THE HEBREW UNIVERSITY

A Machine Learning Approach to Analysis of Biologger Data in Movement Ecology

by:
Yehezkel S. Resheff

Thesis for the degree
"Doctor of Philosophy"
in
Brain Sciences: Computation and Information Processing

Submitted to the Senate of the Hebrew University of Jerusalem

December 2017

This work was carried out
under the supervision of
Daphna Weinshall
Ran Nathan

Acknowledgements

First and foremost, I would like to thank my advisors, Prof. Daphna Weinshall, and Prof. Ran Nathan. With any other combination of advisors I probably would never have reached this point. I would like to thank the committee members Yonatan Loewenstein and Uzi Motro for their input and support, the *Edmond and Lily Safra Center for Brain Sciences* for the carefully thought out program, and funding. Most of all I would like to thank my family, friends, and environment that made the eternal student-hood and associated lifestyle so natural.

The Hebrew University

Abstract

Edmond and Lily Safra Center for Brain Sciences

Doctor of Philosophy

A Machine Learning Approach to Analysis of Biologger Data in Movement Ecology

by Yehezkel S. Resheff

Recent technological advancements have dramatically increased the availability of wearable devices fitted with multiple sensors. Two fields have greatly benefited from this revolution: animal tracking, and medical devices. With ever-growing datasets becoming readily available, it has now become necessary to develop methods to turn these data into insights.

In this research I studied and developed machine learning methods for analysis of data from bio-logger wearable devices. The first chapter deals with unsupervised analysis of accelerometer data, framing the problem as topic modeling over short bursts of the accelerometer signal. This method is applied to both focal domains, with results demonstrating the utility both for animal understanding and continuous monitoring systems for hospitalized patients.

The second chapter deals with the most common data type derived from wearable sensors – GPS location data – and provides a method to deal with and process the vast amounts of data that are accumulating, for purposes such as visualization and further analysis.

In the third chapter methods are developed to tackle problems which typically arise in the context of movement and bio-logger data, but that are more general in scope and appropriate for a somewhat wider audience of data-science practitioners. The first such problem is the imputation of missing data, for which a solution in the form of a single optimization problem is proposed. The second problem is learning a multi-class classifier with arbitrary cost assigned to each of the $O(k^2)$ types of error associated with confusing one of the k labels with another.

Additional work on supervised learning methods for classification of behavioral modes from accelerometer data, behavioral anomaly detection from smart-watch sensors, and predicting doctor assessments is presented in the appendix.

Letter of Contribution

Chapter II

Topic modeling of behavioral modes using sensor data

Yehezkel S. Resheff, Shay Rotics, Ran Nathan, and Daphna Weinshall

YR devised the main conceptual ideas, and wrote the initial version of the manuscript together with DW. SR and RN helped discuss the fundamental concepts in Movement Ecology giving inspiration to this method. All authors reviewed and accepted the final version of the manuscript.

Topic Models for Automated Motor Analysis in Schizophrenia Patients

Talia Tron, Yehezkel S. Resheff, Mikhail Bazhmin, Abraham Peled, Alexander Grinshpoon and Daphna Weinshall.

Yehezkel S. Resheff and Talia Tron are equal contributors to this manuscript. They both devised the main conceptual ideas, planned and designed clinical experiment. TT has designed experimental interfaced, directed and supervised data collection, while YR has monitored data storing and preprocessing. When all data was collected, YR derived motor 'code-book' and performed topic model analysis on the words generated. TT has processed the topic representation to obtain 'consistency', 'typicality' and 'richness' measures, analyzed clinical data to derive sub-clinical populations over time and designed a classifier for automatic clinical classification. Both authors, together with DW, contribute to the interpretation of the results and writing of the manuscript. AP AG and MB gave the theoretical clinical knowledge for experimental design and interpretation. MB collected the experimental data and performed the clinical evaluations under the supervision and guidance of AP and TT. All authors discussed the results and contributed to the final manuscript.

Chapter III

Online Trajectory Segmentation and Summary With Applications to Visualization and Retrieval

Yehezkel S. Resheff

YR is the single author of this paper.

Chapter IV

Optimal Linear Imputation with a Convergence Guarantee

Yehezkel S. Resheff, and Daphna Weinshall

YR and DW devised the main conceptual ideas, discussed the methods and implementations. YR wrote the initial version of the paper. All authors reviewed and accepted the final version.

Every Untrue Label is Untrue in its Own Way: Controlling Error Type with the Log Bilinear Loss

Yehezkel S. Resheff, Amit Mandelbaum, and Daphna Weinshall

YR devised the main conceptual ideas. AM was involved in discussions leading to some of the implementation details. YR wrote the initial version of the manuscript. All authors reviewed and approved the final version.

Appendix Papers

Real-time Schizophrenia Monitoring using Wearable Motion Sensitive Devices

Talia Tron, Yehezkel S. Resheff, Mikhail Bazhmin, Abraham Peled and Daphna Weinshall.

TT and YR devised the main conceptual ideas, planned and designed clinical experiment. TT has designed experimental interfaced, directed and supervised data collection, while YR has monitored data storing and preprocessing. MB collected the experimental data, and performed the clinical evaluations under the supervision and guidance of AP and TT. When all data was collected, YR derived the motor features (step count, energy etc.), and TT performed the correlations with clinical data over different time windows. Both authors, together with DW and AP, contribute to the interpretation of the results and writing of the manuscript. AP and MB gave the theoretical clinical knowledge for experimental design and interpretation.

ARIMA-based motor anomaly detection in schizophrenia inpatients

Talia Tron, Yehezkel S. Resheff, Mikhail Bazhmin, Abraham Peled, and Daphna Weinshall.

TT and YR devised the main conceptual ideas, planned and designed clinical experiment. TT has designed experimental interfaced, directed and supervised data collection, while YR has monitored data storing and preprocessing. MB collected the experimental data and performed the clinical evaluations under the supervision and guidance of AP and TT. When all data was collected, YR designed a step count detector using raw motor data. TT used the step count feature to design an ARIMA model for patients' abnormal behavior detection and systematically compared its performance with clinical and medication data. Both authors, together with DW and AP, contribute to the interpretation of the results and writing of the manuscript. AP and MB gave the theoretical clinical knowledge for experimental design and interpretation.

"Beware of bugs in the above code; I have only proved it correct, not tried it."

Donald E. Knuth

Contents

Chapter 1

Introduction

1.1 Wearable Devices

The term *wearable devices* refers to electronics designed to be worn on the body or otherwise attached to it or embedded in it. While much of the promise initially attributed to the concept has not come to fruition, in the specific niches of health monitoring and animal tracking these devices have proven to be invaluable sources of data and drivers of cutting edge scientific research.

Wearable devices with various sensors are becoming increasingly popular, with ongoing research into applications to health monitoring [20] and context detection [15]. Many fields of animal behavior and conservation have also begun to utilize similar devices in order to remotely monitor the whereabouts and behavior of their research subjects [23], and this has especially been the case in the field of Movement Ecology.

The main narrative of this thesis is the improvement and utilization of methods for analysis of sensor data from wearable devices, with specific applications to animal tracking and medical devices. In the following, I survey these fields, and outline the background and motivations for the current work.

1.1.1 Wearable Devices in Movement Ecology – the Biologger

Movement ecology aims to unify organism movement research and aid in the development of a general theory of whole-organism movements [18]. Recent technological advances in tracking tools and especially the appearance of cheap and small GPS devices [11], have driven the field into a period of rapid growth in knowledge and insight [13], and have led to the emergence of various methods of analyzing movement patterns [27].

These advances have motivated the development of integrative conceptual frameworks unifying cognitive, bio-mechanical, random and optimality paradigms to study movements of all kinds by all types of organisms [18]. Nevertheless, movement data, however accurate, are unlikely to suffice for inference on the links between behavioral, ecological, physiological, and evolutionary processes driving the movement of individuals, and can't by itself link these subjects which have traditionally been researched separately in

their respective fields. Thus, promoting movement ecology research and the desirable unification across species and movement phenomena requires the development of additional data sources: sensors and tools providing simultaneous information not only about the movement, but also about energy expenditure and behavior of the focal organisms, together with the environmental conditions they encounter en route [19].

The past few years have seen tremendous growth in the amounts of data being collected with respect to the movement and behavior of wild animals. Data which is the past would have seemed impossible to obtain, such as high resolution life-tracks, and sub-second sensor data, are now routinely collected by many research groups [6], and for an increasingly large number and diverse group of species. Remote sensing and bio-logging affords us an "eye on planet" [14], with respect to our non-human surroundings. With these large and rich data accumulating in the research community, the need for accompanying tools and methods to turn them into valuable insights is pressing [3].

Surprisingly, research into methods of analysis of this data is somewhat lagging behind the technical ability to collect it. Several reasons may contribute to this phenomenon. First, Movement Ecology and animal tracking are traditionally field-centric scientific fields, where intimate knowledge of the focal animal is held in the highest regard. Arguably, methods always come second. Furthermore, with the inevitable transition into a data-oriented discipline, the necessary augmentation from relevant fields such as computer science, data science, and machine learning, may necessitate a re-design of the core education in Movement Ecology. This process is slow by nature, but we are currently seeing the first steps in this direction.

That being said, the bio-logger is currently driving a data-revolution in Movement Ecology. Even when the most rudimentary methods are applied, the existence of this new data is opening up a vast array of research opportunities, and the low hanging fruit are definitely being picked. Going forward, we will need to develop, popularize, and standardize methods to utilize these data sources to the full extent possible.

1.1.2 Wearable Devices in Health

The second field heavily impacted by the advent of miniaturized sensors and wearable devices is tracking for health and medical devices. Here, these devices are used for continuous monitoring of various aspects of the physical, mental, or behavioral state of patients, either for the purpose of advanced analytics, or to facilitate immediate response (as in the case of fall detection for the elderly).

The most popular application in health and wellbeing is for sports and activity tracking. Wearable devices and smart-phones are routinely used to track the extent of movement and exercise during a user's day, features of structured exercise such as running, and even help improve more complicated body weight exercise. Other applications in the scientific research

of sports and movement use devices such as smart shoes, or accelerometer-embedded smart-watches, to measure gait and posture.

For movement disorder patients, such as Parkinson's and Huntington disease, wearable devices enable lifelong tracking driving novel research that was not possible using traditional observational methods. Furthermore, ongoing tracking of symptoms and behavior provide an objective and reliable augmentation to clinical evaluation, providing a longitudinal picture on the progression of a patient's state.

In this thesis I have applied (together with a number of colleagues) these and similar methods for the continuous tracking of patients in a close ward mental hospital. In the pilot study, 27 inpatients from the closed wards at Shaar-Meashe MHC participated. Each participant was fitted with a smart-watch (GeneActiv[1]) with tri-axial accelerometer embedded sensors. Preliminary results indicate that a tracking system for ongoing patient monitoring holds tremendous promise both for detecting of events that necessitate immediate intervention, and for a longitudinal tracking and augmentation of clinical evaluation with objective measures.

1.2 Behavioral Mode Learning

1.2.1 Supervised Learning

As discussed above, the field of Movement Ecology is being rapidly revitalized by tools and methods. More specifically, the accelerometer-biologger (ACC). These sensors allow the determination of the acceleration of the tagged animal's body, and are used as a means of identifying moment-to-moment behavioral modes [32], and estimating energy expenditure [31].

ACC loggers typically record in 1-3 dimensions, either continuously or in short bouts in a constant window [23]. Their output is used to infer behavior most commonly through supervised machine learning techniques, and energy expenditure using the Overall Dynamic Body Acceleration (ODBA) or related metrics [9, 31]. Combined with GPS recordings, acceleration sensors add fine-scale information on the variation in animal's behavior and energy expenditure in space and time (see [4] for a recent review).

ACC-based analysis has also been used to compute many measures of interest, including behavior-specific body posture, movement and activity budgets, measures of foraging effort, attempted food capture events, mortality detection and classifying behavioral modes [4]. These measures have facilitated movement-related research for a wide range of topics in ecology and animal behavior [27, 4, 25, 28] as well as other fields of research such as animal conservation and welfare [25, 7] and biomechanics [12, 26].

The protocol for using ACC data for supervised learning of behavioral modes consists of several steps [23]. First, before deployment, the response of each tag to $\pm 1G$ on each axis is recorded in a controlled environment, in order to calibrate the tag-specific linear transformation from the recorded

[1] https://www.geneactiv.org/

values (mV) to the desired units of acceleration (G). Next, the calibrated tags are given a recording schedule and mounted on the focal animals. The data is later retrieved either by RF methods, or by physically reacquiring the device.

In order to train supervised machine learning models, a labeled dataset is collected through field observations. This time and labor intensive stage requires the researcher to observe the animal, either in its natural habitat or in captivity, and relate the actual behavioral modes to the time-stamp of the ACC recordings. Since some behavioral modes tend to be less common, or be preformed predominantly at specific times, recording a sufficient number of such behavior-measurement samples may be tricky. Furthermore, for nocturnal or cryptic species, observations may not be feasible. In the final stage, models are trained using the labeled data, and the entire dataset is then labeled.

This process is described for animal behavioral mode labeling, but the process is similar in nature when applied to humans for medical devices. The main steps, which also limit the scope and feasibility of the method, are the compilation of the set of relevant behavioral modes, and the collection of the necessary annotated examples to train the models with. In the next section we discuss unsupervised methods where these limitations are mitigated.

1.2.2 Unsupervised Learning

There are several drawbacks to the supervised learning approach in our setting. Observations, even if perfectly accurate, may not adequately represent the behavioral pattern throughout the period of the research (which is desirably the lifetime of the animal or patient). This is true for several reasons: in animal movement, the field work is inherently confined to a specific time and place, and thus only some of the animals are observed. Also, the presence of the observer may impact the behavior of the observed animals.

For patients, in many of the cases discussed the behavior of individuals is expected to be highly variable. Especially when movement disorders or mental illness is involved, the ability to specify the full repertoire of movement is very doubtful. We then would want to rely on a data driven approach instead, one which will enable us to discover the movement range in an automatic and case-by-case fashion.

Furthermore, the need for observations limits the scope of supervised learning behavioral mode annotation projects to observable species, and requires considerable efforts (both in terms of money and manpower). For this reason, even when the supervised approach is technically feasible, it would be beneficial on many occasions to have a simpler unsupervised approach in the field's toolkit.

In this thesis I describe a method for unsupervised analysis of behavioral mode based on (possibly multiple) sensor streams. The method, in two closely related variants, is applied both to animal movement data and to patients in a hospital setting. In both cases, the method is demonstrated to be able to extract meaningful information regarding the movement and behavior of individuals, and connections are drawn to supervised annotations.

1.3 Trajectory Analysis

Another type of data collected by bio-loggers and wearable devices is location GPS information. In the field of Ecology and Animal Tracking, movement data is being collected at a global [24, 10] as well as a regional [2, 30] scale. Car [17, 16] and ship [29] trajectories are recorded for control, optimization, and safety purposes, and pedestrians are tracked for health, safety, and navigation utilities [8, 1]. With this large amount of data rolling in, new and more efficient methods must be developed in order to visualize, analyze, and gain insight and knowledge.

One of the most fundamental computations associated with understanding movement data is segmentation of trajectories. Traditionally, a segmentation proceeds by defining a feature of a single point, then dividing the entire trajectory into sub-trajectories which are uniform (in some sense) with respect to this feature [5]. Essentially, trajectory segmentation is the process of subdividing a trajectory into parts either by grouping points similar with respect to some measure of interest, or by minimizing a global objective function.

In this thesis I present a novel online algorithm for segmentation and summary, based on point density along the trajectory, and based on the nature of the naturally occurring structure of intermittent bouts of locomotive and local activity. I show an application to visualization of trajectory datasets, and discuss the use of the summary as an index allowing efficient queries which are otherwise impossible or computationally expensive, over very large datasets.

1.4 Missing Data

A central theme in this thesis is *missing data*. Several aspects of this are covered in various parts of this thesis. Missing data in the most common sense is treated in the methods presented in Chapter 4 [22]. This work deals with imputation of missing data in the form of random entries in a matrix which are not available. The structure of the data (information shared between columns of the matrix in this case) is used to recover the missing entries.

In the context of supervised vs. unsupervised learning methods of behavioral modes, the term *missing* is used to denote the lack of observations in some cases, and motivate methods that do not require a labeled dataset. This should not be confused with the meaning of the word in the first line of work. In the latter, all labels are "missing" and thus we must rely on external structure in order to "recover" them. In fact, it would be more natural to say that we are going to *describe*, rather than *recover* these labels.

The third context in which missing data is a key issue is in trajectory analysis (Chapter 3). In many data acquisition systems for trajectory data there is an inherent problem of missing data. This is often the case due to data acquisition limitations (coverage, reception, etc.), but may also stem from limit

memory, or power constraints. Either way, methods for analysis of movement data must often be resilient to a large extent to missing parts of a trajectory. The method described in Chapter 3 [21] does not try to fill in missing parts of a trajectory, but is by construction resistant to adverse effects of holy trajectories, and able to adequately and efficiently represent them in the interest of downstream processing.

References

[1] Moustafa Alzantot and Moustafa Youssef. "UPTIME: Ubiquitous pedestrian tracking using mobile phones". In: *2012 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE. 2012, pp. 3204–3209.

[2] Allert I Bijleveld et al. "Understanding spatial distributions: negative density-dependence in prey causes predators to trade-off prey quantity with quality". In: *Proc. R. Soc. B*. Vol. 283. 1828. The Royal Society. 2016, p. 20151557.

[3] Luca Börger. "Stuck in motion? Reconnecting questions and tools in movement ecology". In: *Journal of Animal Ecology* 85.1 (2016), pp. 5–10.

[4] Danielle D Brown et al. "Observing the unwatchable through acceleration logging of animal behavior". In: *Animal Biotelemetry* 1.1 (2013), p. 20.

[5] Maike Buchin et al. "An algorithmic framework for segmenting trajectories based on spatio-temporal criteria". In: *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM. 2010, pp. 202–211.

[6] Francesca Cagnacci et al. *Animal ecology meets GPS-based radiotelemetry: a perfect storm of opportunities and challenges*. 2010.

[7] Steven J Cooke. "Biotelemetry and biologging in endangered species research and animal conservation: relevance to regional, national, and IUCN Red List threat assessments". In: *Endangered species research* 4.1-2 (2008), pp. 165–185.

[8] Eric Foxlin. "Pedestrian tracking with shoe-mounted inertial sensors". In: *IEEE Computer graphics and applications* 25.6 (2005), pp. 38–46.

[9] Adrian C Gleiss, Rory P Wilson, and Emily LC Shepard. "Making overall dynamic body acceleration work: on the theory of acceleration as a proxy for energy expenditure". In: *Methods in Ecology and Evolution* 2.1 (2011), pp. 23–33.

[10] Roi Harel, Nir Horvitz, and Ran Nathan. "Adult vultures outperform juveniles in challenging thermal soaring conditions". In: *Scientific reports* 6 (2016).

[11] Mark Hebblewhite and Daniel T Haydon. "Distinguishing technology from biology: a critical review of the use of GPS telemetry data in ecology". In: *Philosophical Transactions of the Royal Society of London B: Biological Sciences* 365.1550 (2010), pp. 2303–2312.

[12] Allyson G Hindle, David AS Rosen, and Andrew W Trites. "Swimming depth and ocean currents affect transit costs in Steller sea lions Eumetopias jubatus". In: *Aquatic Biology* 10.2 (2010), pp. 139–148.

[13] Marcel Holyoak et al. "Trends and missing parts in the study of movement ecology". In: *Proceedings of the National Academy of Sciences* 105.49 (2008), pp. 19060–19065.

[14] Roland Kays et al. "Terrestrial animal tracking as an eye on life and planet". In: *Science* 348.6240 (2015), aaa2478.

[15] Nicky Kern, Bernt Schiele, and Albrecht Schmidt. "Multi-sensor activity context detection for wearable computing". In: *Ambient Intelligence*. Springer, 2003, pp. 220–232.

[16] Yong-Shik Kim and Keum-Shik Hong. "An IMM algorithm for tracking maneuvering vehicles in an adaptive cruise control environment". In: *International Journal of Control Automation and Systems* 2 (2004), pp. 310–318.

[17] Lars-Henrik Kullingsjö and Sten Karlsson. "The Swedish car movement data project". In: *Proceedings to EEVC Brussels, Belgium, November 19-22, 2012*. 2012.

[18] Ran Nathan et al. "A movement ecology paradigm for unifying organismal movement research". In: *Proceedings of the National Academy of Sciences* 105.49 (2008), pp. 19052–19059.

[19] Ran Nathan et al. "Using tri-axial acceleration data to identify behavioral modes of free-ranging animals: general concepts and tools illustrated for griffon vultures". In: *Journal of Experimental Biology* 215.6 (2012), pp. 986–996.

[20] Alexandros Pantelopoulos and Nikolaos G Bourbakis. "A survey on wearable sensor-based systems for health monitoring and prognosis". In: *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 40.1 (2010), pp. 1–12.

[21] Yehezkel S Resheff. "Online trajectory segmentation and summary with applications to visualization and retrieval". In: *Big Data (Big Data), 2016 IEEE International Conference on*. IEEE. 2016, pp. 1832–1840.

[22] Yehezkel S. Resheff and Daphna Weinshall. "Optimized Linear Imputation". In: *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods, ICPRAM 2017, Porto, Portugal, February 24-26, 2017*. 2017, pp. 17–25. DOI: 10.5220/0006092900170025. URL: https://doi.org/10.5220/0006092900170025.

[23] Yehezkel S Resheff et al. "AcceleRater: a web application for supervised learning of behavioral modes from acceleration measurements". In: *Movement ecology* 2.1 (2014), p. 27.

[24] Shay Rotics et al. "The challenges of the first migration: movement and behaviour of juvenile vs. adult white storks with insights regarding juvenile mortality". In: *Journal of Animal Ecology* 85.4 (2016), pp. 938–947.

[25] Kentaro Q Sakamoto et al. "Can ethograms be automatically generated using body acceleration data from free-ranging birds?" In: *PloS one* 4.4 (2009), e5379.

[26] WI Sellers and RH Crompton. "Automatic monitoring of primate locomotor behaviour using accelerometers". In: *Folia primatologica* 75.4 (2004), pp. 279–293.

[27] Peter E Smouse et al. "Stochastic modelling of animal movement". In: *Philosophical Transactions of the Royal Society of London B: Biological Sciences* 365.1550 (2010), pp. 2201–2211.

[28] Orr Spiegel et al. "Mixed strategies of griffon vultures'(Gyps fulvus) response to food deprivation lead to a hump-shaped movement pattern". In: *Movement ecology* 1.1 (2013), p. 5.

[29] Thomas Statheros, Gareth Howells, and Klaus McDonald Maier. "Autonomous ship collision avoidance navigation concepts, technologies and techniques". In: *Journal of navigation* 61.01 (2008), pp. 129–142.

[30] Adi Weller Weiser et al. "Characterizing the Accuracy of a Self-Synchronized Reverse-GPS Wildlife Localization System". In: *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE. 2016, pp. 1–12.

[31] Rory P Wilson et al. "Moving towards acceleration for estimates of activity-specific metabolic rate in free-living animals: the case of the cormorant". In: *Journal of Animal Ecology* 75.5 (2006), pp. 1081–1090.

[32] K Yoda et al. "Precise monitoring of porpoising behaviour of Adélie penguins determined using acceleration data loggers". In: *Journal of Experimental Biology* 202.22 (1999), pp. 3121–3126.

Chapter 2

Unsupervised Analysis of Behavioral Modes from Sensor Data

2.1 Movement Ecology

This paper [2] was published in the *International Journal of Data Science and Analytics*. A short version [1] was presented at the 2015 IEEE International Conference on Data Science and Advanced Analytics, which took place in Paris, France, during October 19-21, 2015.

References

[1] Yehezkel S Resheff et al. "Matrix factorization approach to behavioral mode analysis from acceleration data". In: *Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on*. IEEE. 2015, pp. 1–6.

[2] Yehezkel S Resheff et al. "Topic modeling of behavioral modes using sensor data". In: *International Journal of Data Science and Analytics* 1.1 (2016), pp. 51–60.

Int J Data Sci Anal 12
DOI 10.1007/s41060-016-0003-4

REGULAR PAPER

Topic modeling of behavioral modes using sensor data

Yehezkel S. Resheff[1] · **Shay Rotics**[2] · **Ran Nathan**[2] · **Daphna Weinshall**[3]

Received: 30 November 2015 / Accepted: 7 January 2016

Abstract The field of movement ecology is experiencing a period of rapid growth in availability of data. As the volume rises, traditional methods are giving way to machine learning and data science, which are playing an increasingly large part in turning these data into science-driving insights. One rich and interesting source is the biologger. These small electronic wearable devices are attached to animals free to roam in their natural habitats and report back readings from multiple sensors, including GPS and accelerometer bursts. A common use of accelerometer data is for supervised learning of behavioral modes. However, we need unsupervised analysis tools as well, in order to overcome the inherent difficulties of obtaining a labeled dataset, which in some cases either is infeasible or does not successfully encompass the full repertoire of behavioral modes of interest. Here, we present a matrix factorization-based topic model method for accelerometer bursts, derived using a linear mixture property of patch features. Our method is validated via comparison with a labeled dataset and is further compared to standard clustering algorithms.

Keywords Behavioral modes · Topic model · Movement ecology · MS-BoP

1 Introduction

Wearable devices with various sensors are becoming increasingly popular, with ongoing research into applications to health monitoring [22] and context detection [13]. Many fields of animal behavior and conservation have also begun to utilize similar devices in order to remotely monitor the whereabouts and behavior of their research subjects [25], and this has especially been the case in the field of movement ecology.

The aim of movement ecology is to unify research of movement of organisms and aid in the development of a general theory of whole-organism movement [18]. Recent technological advances in tracking tools and especially the appearance of cheap and small GPS devices [9] have driven the field into a period of rapid growth in knowledge and insight [11,12] and have led to the emergence of various methods of analyzing movement patterns [29].

Nevertheless, movement data, however accurate, are unlikely to suffice for inference on the links between behavioral, ecological, physiological, and evolutionary processes driving the movement of individuals, and link these subjects which have traditionally been researched separately in their respective fields. Thus, understanding movement phenomena across species requires the development of additional data sources: sensors and tools providing simultaneous information about the movement, energy expenditure and behavior of the focal organisms, together with the environmental conditions they encounter en route [19].

One such tool, which has been introduced into the field of movement ecology, is the accelerometer biologger (ACC).

Invited Extended version of a paper [26] presented at the international conference *Data Science and Advanced Analytics*, Paris, France, 19–21 October 2015.

✉ Yehezkel S. Resheff
yehezkel.resheff@mail.huji.ac.il

1 Edmond and Lily Safra Center for Brain Sciences, The Hebrew University of Jerusalem, 91914 Jerusalem, Israel

2 Movement Ecology Lab, Department of Ecology, Evolution and Behavior, The Hebrew University of Jerusalem, Jerusalem, Israel

3 School of Computer Science and Engineering, The Hebrew University of Jerusalem, Jerusalem, Israel

Published online: 27 January 2016

These sensors allow the determination of the tagged animal's body acceleration and are used as a means of identifying moment-to-moment behavioral modes [35] and estimating energy expenditure [34].

ACC loggers typically record in 1–3 dimensions, either continuously or in short bouts in a constant window [25]. Their output is used to infer behavior, most commonly through supervised machine learning techniques, and energy expenditure using the overall dynamic body acceleration (ODBA) or related metrics [8,34]. When combined with GPS recordings, acceleration sensors add fine scale information on the variation in animal behavior in space and time (see [2] for a recent review).

ACC-based analysis has been used to compute many measures of interest in the field of movement ecology, including behavior-specific body posture, movement and activity budgets, measures of foraging effort, attempted food capture events, mortality detection, classifying behavioral modes [2]. These measures have facilitated research for a wide range of topics in ecology, animal behavior [2,29–31], animal conservation and welfare [3,31], and biomechanics [10,28].

In recent years, there has been considerable interest in the analysis of behavioral modes using ACC data and supervised learning techniques. The protocol for using ACC data for supervised learning of behavioral modes consists of several steps. First, a sensor calibration procedure is preformed in a controlled environment: Before deployment, the response of each tag to $\pm 1G$ acceleration on each axis is recorded, in order to fit the tag-specific linear transformation from the recorded values (mV) to the desired units of acceleration. Next, the calibrated tags are given a recording schedule and mounted on the focal animals, after these are captured. Finally, the data are retrieved using RF (radio) methods, cellular transmission, or physically reacquiring the device.

Before supervised machine learning models can be used, a labeled dataset is collected through field observations. This time- and labor-intensive stage requires the researcher to observe the animal, either in its natural habitat or in captivity and relate the actual behavioral modes to the time stamp of the ACC recordings. Since some behavioral modes tend to be less common, or are performed predominantly at specific times, recording a sufficient number of such behavior measurement samples may be tricky. Furthermore, for aquatic and nocturnal species, observations may not be feasible. In the final stage, models are trained using the labeled data, and the entire dataset is then classified.

Supervised machine learning methods have been applied to ACC data from many species and for a diverse range of behavioral modes. However, there are several drawbacks to the supervised approach. Observations, even if perfectly accurate, may not be adequately representative of the behavioral pattern throughout the period of the research (which is desirably the lifetime of the animal), for several reasons: Field work is inherently confined to a specific time and place; moreover, only some of the animals are observed, and the presence of the observer may in some cases have an impact on the behavior of the observed animals. Furthermore, the need for observations limits the scope of such research projects to observable species and to research laboratories with the necessary resources (in money, manpower, and knowledge) to carry out all the steps listed above.

In this paper, we present a framework for unsupervised analysis of behavioral modes from ACC data. First, we suggest a multi-scale bag of patches (MS-BoP) descriptor of ACC signals reminiscent of "bag of visual words" descriptors in computer vision (see [4,36]). Next, we present a simple topic model for behavioral modes incorporating a linear mixture property of the MS-BoP features and demonstrate how it can be used for unsupervised analysis of behavioral modes.

The rest of the paper is organized as follows: The next section describes related work both in movement ecology and in matrix factorization for clustering and topic modeling. In Sect. 3, we introduce the features and model. Finally, in Sect. 4 we present the results of an analysis on a large real-world dataset and the comparison with other methods.

2 Previous work

Previous work on behavioral mode analysis using ACC data in movement ecology focused predominantly on supervised learning, with an emphasis on constructing useful features and finding the right classifiers for a specific use, such as monitoring dairy cows [6], or determining the flight type of soaring birds [33].

While this line of work proved very successful, both in terms of classifier performance and of scientific discovery that it was able to drive, it still suffers from the inherent limitations of supervised learning, compounded by the very high cost of obtaining labeled data for behavioral observations of wild animals. It remains the case that for some animals (nocturnal or sea species for instance), obtaining a labeled dataset is currently infeasible. Thus, in order to use all available ACC data for behavioral mode analysis in the field of movement ecology, an unsupervised framework is essential.

To the best of our knowledge, there have been two attempts at such an approach. In [27], K-means was applied to a representation of the ACC data, to achieve behavior mode clusters. In [7,16], a Gaussian mixture model (GMM) variant was used to cluster a low-dimensional representation of ACC signals into a small number of useful behavioral modes. Unsupervised techniques have also been used for discovery of behavioral patterns in human subjects for logging and medical purposes [17] and for detection of surprising events

[20]. Our method goes one step further by allowing samples to be a mixture (more precisely, a convex combination) of behavioral modes, accounting for the observation that ACC samples do indeed tend to be mixed this way (Fig. 1).

In wearable devices research on human behavioral analytics, topic models were often used to describe behavior on a multiple-minute timescale, based on the moment-to-moment behavioral annotation provided by sensor measurements through supervised learning [21,24]. These methods enable a long-range behavioral layer (such as going to work) which is fundamental to understanding the context of the user. While extremely relevant to the ecology community, these methods cannot be transported to animal behavior since the density of moment-to-moment behavior annotation (typically 1 per 5–10 minutes [25]) is insufficient for such an analysis.

Nonnegative matrix factorization (NNMF) has extensively been studied in the context of clustering [14,32] and topic modeling [1]. Connections have been shown to various popular clustering algorithms such as K-means and spectral clustering [5]. Our proposed method is essentially topic modeling with NNMF, based on theoretical justification that incorporates the nature of our signals and the features under consideration.

Our approach is novel in the modeling of a short-timescale behavior (4 seconds in our experiments) as a sequential mixture of behavioral modes. The features we suggest allow this problem to be naturally cast into a linear mixture model which is then solved using standard optimization techniques.

3 Methodology

3.1 Feature generation

In the field of natural language processing (NLP), textual documents are commonly described as word-count histograms. These descriptors are generally known as *bag-of-word* representations (BoW), since during their creation all the words in a document are (figuratively speaking) thrown into a bag, losing all proximity information, and then each word in a predefined dictionary is assigned the number of times it repeats in the bag. The final representation of the document is a vector of these counts.

The BoW representation was adopted in recent years into computer vision for the representation of images. Since images are not naturally divided into discrete elements (like words in a document), the first step is to transform the image into a series of word analogues which can then be thrown into a bag. This discretization process is often achieved by clustering *patches* of images, then assigning each patch the index of its cluster. The resulting feature vector for a given image is the histogram of the cluster associations of its patches. The cluster centroids are often referred to as the *codebook*, and the method as bag of visual words (BoVW).

Here, we adapt the BoVW method to be used with the ACC signal. We start by defining the notion of a patch of an ACC signal.

3.1.1 Definition: patch in an ACC signal

Let

$$s = [s_1, \ldots, s_N]$$

be an ACC signal of length N. The patch of length l starting at index i of s is the subvector:

$$[s_i, \ldots, s_{i+l-1}]$$

thus, there are $N - l + 1$ distinct patches in s.

3.1.2 Codebook generation

As in the BoVW case, ACC signals and patches do not consist of discrete elements. In order to count and histogram types of patches, we must first construct a patch codebook. We suggest the following construction: Given a codebook size k and a patch length l, for each ACC signal in the dataset, extract and pool all of the l-length patches. Next, using K-means clusters the patches into k clusters. The resulting k centroids will be called the codebook. The intuition behind using patches to describe an ACC signal is that behavioral modes should be definable by the distribution of short-timescale movements that they are comprised of. Since different behavioral modes occur at various characteristic timescales, we would like to repeat the process for more than one patch length, in order to efficiently capture all ACC patterns of relevance. Thus, we generate a separate codebook for several timescales in the appropriate range, depending on the behavioral modes we are interested in (Algorithm 1).

3.1.3 Feature transformation

Once we have constructed the codebook for all of the scales, we are ready to transform our ACC signals into the final multi-scale bag of patches (MS-BoP) descriptor. For each ACC record in the dataset, and for each scale, we extract all patches of the signal at that scale and assign each one the index of the nearest centroid in the appropriate codebook. For each scale, we then histogram the index values to produce a (typically sparse) vector of the length of the codebook. The final representation is the concatenation of histograms for the various scales (Algorithm 2).

Algorithm 1 Multi Scale Codebook Generation

input:

$\{s_i\}_{i=1}^{p}$ the set of raw acceleration measurements
$l_1, .., l_m$ list of scales to use
$k_1, \ldots, k_m$ list of corresponding sizes per codebook

output:

$CB_1, \ldots, CB_l$ the generated codebooks. $CB_i[j]$ is the $j-th$ word in the $i-th$ codebook ($i = 1, \ldots, l$; $j = 1, \ldots, k_i$)

1: **for** scale := 1,...,l **do**
2: patches := list of all patches of scale l_{scale} in $\{s_i\}_{i=1}^{p}$
3: CB_i := K_means(patches, k_{scale}).centroids
4: **end for**
5: **return** $CB_1, \ldots, CB_l$

Algorithm 2 MS-BoP feature transformation

input:

$CB_1, \ldots, CB_l$ The l codebooks, output of Algorithm 1.
$l_1, .., l_m$ list of the patch scales that were used in Algorithm 1.
s an ACC signal to transform

output:

f The MS-BoP representation of signal s

1: **for** scale := 1,...,l **do**
2: f_{scale} := a zeros vector of the same length as CB_{scale}
3: patches := list of all patches of scale l_{scale} in s
4: **for each** p in patches **do**
5: idx := index of the closest word to p in the codebook CB_{scale}
6: increment $f_{scale}[idx]$ by 1
7: **end for**
8: **end for**
9: f := stack_vectors($f_1, \ldots, f_l$)
10: **return:** f

3.2 Mixture property of patch features

In order to motivate the proposed model (next section), we present the mixture property of patch features. We assume that our signals have the property that a large enough part of a sample from a certain behavioral mode will have distribution of patches that is the same as the distribution in the entire sample. The meaning of this assumption is that each behavioral mode has a distribution of patches that characterizes it at each scale.

Intuitively, if a signal s is constructed by taking the first half of a signal s_a and the second half of an equal length signal s_b, then the distribution of patches in s will be approximately an equal parts mixture of those in s_a and in s_b. The reason for this is that a patch in s either (a) is completely contained in s_a and will then be distributed like patches in s_a, or (b) is completely in s_b and will then be distributed like patches in s_b, or (c) starts in s_a and continues into s_b, in which case we know little about the patch distribution and consider it as noise. The key point is that the number of patches of type (c) is at most twice the length of the patch and thus can be made small in relation to the total number of patches which is in the order of the length of the signal. More formally:

Let s be an ACC signal composed of a concatenation of t_1 consecutive samples during behavioral mode a and t_2 consecutive samples during behavioral mode b (see Fig. 1). Denote $p_{mode}(v)$ the probability of a patch v of length l in behavioral $mode \in \{a, b\}$. Let v be a patch drawn uniformly from s, then:

$$\begin{aligned} p(v) &= Pr(A)p(v|A) + Pr(B)p(v|B) + Pr(C)p(v|C) \\ &\geq Pr(A)p_a(v) + Pr(B)p_b(v) \\ &= \frac{t_1 - l}{t_1 + t_2} p_a(v) + \frac{t_2 - l}{t_1 + t_2} p_b(v) \end{aligned}$$

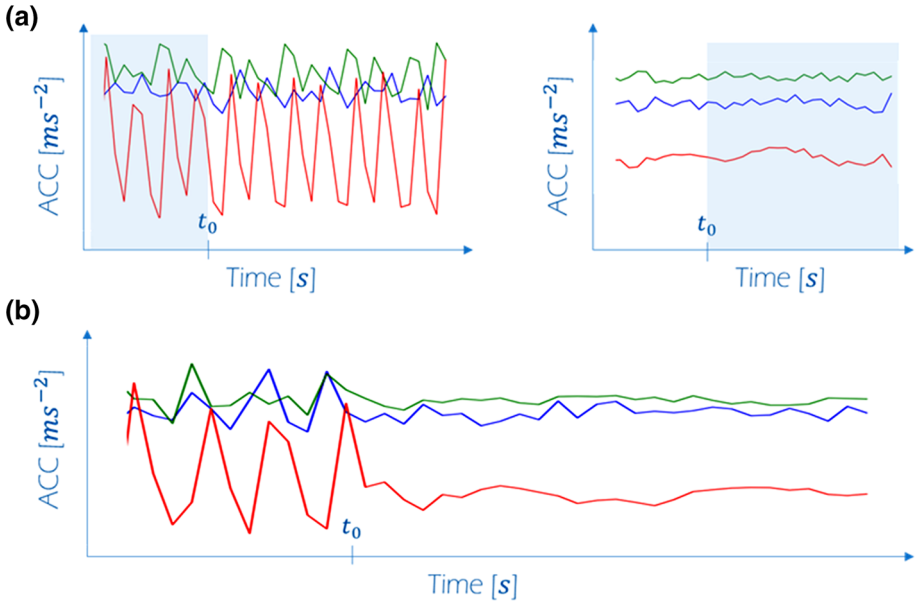


Fig. 1 Pure and mixed triaxial ACC signals. Pure ACC signals (**a**) are measured during a single behavioral mode. However, in most cases a single measurement contains a mixture of more than one behavioral mode (**b**) and may be viewed as a concatenation of the beginning/end of two pure signals. The *colors* represent each of the three acceleration dimensions (color figure online)

$$= \frac{t_1}{t_1 + t_2} p_a(v) + \frac{t_2}{t_1 + t_2} p_b(v) - \epsilon$$

where events A, B, C denote the patch being all in s_1, all in s_2, and starting in $s1$ and ending in s_2, respectively, and:

$$\epsilon = \frac{l}{t_1 + t_2}[p_a(v) + p_b(v)]$$

ϵ can be made arbitrarily small by making $t_1 + t_2$ large and keeping l constant, meaning that for patches small enough in relation to the length of the entire signal, the distribution of patches of the concatenated signal is a mixture (convex combination) of the distributions of the parts, with mixing coefficients proportional to the part lengths. We note that this result can easily be extended to a concatenation of any finite number of signals, as long as each one is sufficiently long in comparison with the patch width.

Since behaviors of real-world animals may start and stop abruptly, and a recorded ACC signal is likely to be a concatenation of signals representing different behavioral modes (typically 1–3), the above property inspires a model that is able to capture such mixtures in an explicit fashion. Furthermore, the resulting mixture coefficients may provide some insight into the nature of the underlying behaviors and the relationships between them— for example, which often appear alongside each other, and which are more temporally separated.

3.3 The proposed model

Let k denote the number of behavioral modes under consideration and p the dimension of the representation of ACC observations. Following the mixture property presented in the previous section, we assume that every sample is a convex combination of the representation of a "pure" signal of the various behavioral modes. Further, we assume the existence of a matrix $F \in R^{pk}$, the *factor matrix*, such that the ith column of F is the representation of a pure signal of the ith behavioral mode, which we will call the factor associated with the ith behavioral mode. Let s be an ACC sample, then:

$$s = F\alpha + \epsilon \tag{1}$$

where $\epsilon \in R^p$ is some random vector. In other words, we say that the sample s is a linear combination of the factors associated with each of the behavioral modes with some remainder term. For the full dataset, we then have:

$$S = FA + \epsilon \tag{2}$$

where F is the same matrix, $A's$ columns are the factor loadings for each of the samples denoted α in (1), and $\epsilon \in R^{pN}$ is a random matrix. Since our features are nonnegative histograms, and we would like the factor loadings to be nonnegative, we constrain the matrices F, A to have nonnegative values. We solve for F, A using a least squares criterion:

$$\begin{aligned} \underset{F,A}{\text{argmin}} \quad & \|FA - S\|_F^2 \\ \text{subject to} \quad & F_{i,j}, A_{i,j} \geq 0 \; \forall i, j \end{aligned} \tag{3}$$

This is by now a standard problem, which can be solved, for instance, using alternating nonnegative least squares [32]. The idea behind the algorithm (Algorithm 3) is that while the complete problem is not convex, and not easily solved, for a set A it becomes a simple convex problem in F, and vice versa. This inspires the simple block coordinate descent algorithm which minimizes alternately w.r.t each of the matrices. Since this procedure generates a (weakly) monotonically decreasing series of values of the objective (3), it is guaranteed to converge to a local minimum[1].

Algorithm 3 Alternating Non-Negative Least Squares

input:

S	the complete matrix $S \in R^{pN}$
k	factorization rank

output:

F, A matrices $F \in R^{pk}$, $A \in R^{kN}$

1: $F :=$ random initialization
2: $A :=$ random initialization
3: **while** not converged **do**
4: $\quad F := \underset{F}{\text{argmin}} \|FA - S\|_F^2 \;\; s.t. \; F_{i,j} \geq 0 \quad \forall i, j$
5: $\quad A := \underset{A}{\text{argmin}} \|FA - S\|_F^2 \;\; s.t. \; A_{i,j} \geq 0 \quad \forall i, j$
6: **end while**
7: **return** F, A

3.4 Speedup via sampling

Since this method may potentially be applied to large datasets (containing at least hundreds of millions of records and many billions of patches), it is worth mentioning that all parameter-learning steps of the algorithm can be processed (identically to the original method) on a sample of the dataset. During codebook generation, records in the dataset and/or patches in each used record could be sampled to reduce the number of resulting patches we have to cluster. Next, fitting F and A on a sample of the records gives us the factor matrix, but not the factor loadings per record of the dataset. However, once

[1] The objective is bounded from below by 0.

we have F the optimization problem (3) turns into:

$$\begin{aligned} &\underset{A}{\text{argmin}} && \|FA - S\|_F^2 \\ &\text{subject to} && A_{i,j} \geq 0 \quad \forall i, j \end{aligned} \tag{4}$$

a simple convex problem in which the factor loadings per record (columns of A) can be minimized independently for each record s in the dataset, as follows:

$$\begin{aligned} &\underset{\alpha}{\text{argmin}} && \|F\alpha - s\|^2 \\ &\text{subject to} && \alpha_i \geq 0 \quad \forall i \end{aligned} \tag{5}$$

3.5 Extension to the multi-sensor case

Thus far, we have constructed a topic model applicable for data derived from a single (albeit possibly multi-dimensional) sensor. The multi-sensor (or sensor integration) case is of particular interest in this case because many devices containing accelerometers also include other sensors such as gyroscopes and magnometers. Since each of these is recording at different frequencies, we cannot simply consider them to be extra dimensions in the same time series produced by the 3D accelerometer. The integrative framework we suggest assumes that the same behavioral modes are manifested in distinct patterns for each of the sensors. Thus, we will have separate factor matrices:

$$F^1, \ldots, F^l$$

for the l sensor types, and a single shared factor loading matrix A. Denoting the features matrices of the MS-BoP features for each of the l sensor types:

$$S^1, \ldots, S^l$$

we now look for matrices:

$$A, F^1, \ldots, F^l$$

such that:

$$\forall i : S^i \approx F^i A$$

which we encode in the following optimization problem:

$$\begin{aligned} &\underset{F^1,\ldots,F^l,A}{\text{argmin}} && \frac{1}{l}\sum_{i=1}^{l} \|F^i A - S^i\|_F^2 \\ &\text{subject to} && F^k_{i,j}, A_{i,j} \geq 0 \quad \forall i, j, k \end{aligned} \tag{6}$$

This problem is solvable using the same type of method. Specifically, we will now show that this new problem can be rewritten in the same form as (3), with both the sample and factor matrices stacked. Denote:

$$F = \begin{bmatrix} [F^1] \\ \vdots \\ [F^l] \end{bmatrix}$$

and:

$$S = \begin{bmatrix} [S^1] \\ \vdots \\ [S^l] \end{bmatrix}$$

and then (6) becomes:

$$\begin{aligned} &\underset{F,A}{\text{argmin}} && \|FA - S\|_F^2 \\ &\text{subject to} && F_{i,j}, A_{i,j} \geq 0 \quad \forall i, j \end{aligned}$$

since the $\frac{1}{l}$ scaling factor makes no difference to the $argmin$. In summary, the multi-sensor case where a separate factor matrix is allocated to each sensor, with a joint factor loading matrix, is identical to the single-sensor case when the MS-BoP features for each sensor are stacked vertically.

3.6 Extension to the supervised and semi-supervised cases

Suppose that observation (or any other mechanism) allowed us to obtain "pure" ACC signals for some (or all) of the behavioral modes. Using the mean MS-BoP representation of the signals in each of these modes for the corresponding column of F, we are left with a convex problem similar to (3), where the optimization is over the remaining elements of F only.

In the extreme case, when we have labeled samples for a pure ACC signal for all the behavioral modes under consideration, and thus all of F is predetermined, the resulting problem is equivalent to (4). Namely, we are left with the task of obtaining the factor loadings for the remaining (unlabeled) data.

3.7 Limitations

Consider a solution, matrices F, A that minimize objective (3), so that:

$$S \approx FA$$

Clearly, for any orthogonal matrix Q (of the appropriate dimensions):

$$FA = FQQ^TA = (FQ)(A^TQ)^T$$

thus, the solution:

$$F' = FQ$$
$$A' = (A^T Q)^T$$

is also a minimizer of objective (3), iff the matrices F', A' obey the constraints:

$$F'_{i,j}, A'_{i,j} \geq 0 \quad \forall i, j \tag{7}$$

While this clearly holds if Q is a permutation matrix, there are (always) orthogonal matrices Q which contain negative elements for which the constraints in (7) hold. From the construction of F' and A', we can interpret them as an entanglement of our factors and loadings (technically, what we find is the span of the correct factors, but not the factors themselves). We note that while this property limits the ability to recover factors that generate the data, in practice the factors themselves are useful for analysis of behavioral topics, as demonstrated in the section below.

We leave to future research the issue of the disentanglement, which should be achieved via regularization with respect to A in the original optimization problem (3).

4 Results

In this section, we present experiments designed to compare our method to alternatives and derive insights about the data. Results are then discussed in the next section.

Data for these experiments consist of $3D$ acceleration measurements from biologgers which were recorded during 2012. Each measurement consists of 4 seconds at $10Hz$ per axis, giving a total of 120 values.

A ground truth partitioning of the data was obtained using standard machine learning techniques (see [19,25] for more details regarding the methodology), based on 3815 field observations each of which was assigned one of 5 distinct behavioral modes (walking, standing, sitting, flapping, gliding). Experiments were conducted using stratified sampling of 100,000 measurements (20,000 per behavioral mode).

Matrix factorization was preformed using the scikit-learn [23] python software library (see [15] for method details). In all experiments, the results were stable across repetitions, leading to essentially zero standard deviation, and therefore the reported results correspond to single repetitions.

The purpose of these experiments is to assess to what extent the soft partitioning via our topic model method relates to the hard, ground truth partitions. Our method is compared to the following:

Random partitioning: each sample is assigned a value drawn uniformly from the set of possible partitions $\{1, 2, .., k\}$
Uniform partition: each sample is assigned the same distribution of $\frac{1}{k}$ per partition, over the k partitions.
K-means: the sample are partitioned using K-means.
Gaussian mixture model (GMM): GMM is used to assign samples k partition coefficients.

where (a) and (b) are used as controls, (c) and (d) are used as representative hard and soft clustering methods, respectively.

The data are then divided randomly into two equal parts designated train and test. Using the training set, we learn the partitioning of the data for each of the methods (random, uniform, K-means, GMM, and NNMF). Next, for each method separately, we assign each of the partitions one of the semantic labels (flapping, gliding, walking, standing, sitting). In order to do this, we group the data in the training set according to the semantic label it received (the supervised annotation) and compute the average loading for each label in the partition. The final assignment for the partition is the label with the highest mean loading in it (see schematic in Fig. 2).

The evaluation stage is performed on the test set only. Resemblance to the ground truth is measured using log-loss (Fig. 3) and $0-1$ loss (Fig. 4), after partition values are converted to soft label assignments using the mapping derived from the training set (see schematic in Fig. 2). For an assignment $l_1, \ldots, l_5$ for the 5 behavior labels, where the ground truth label is i, we use the $0-1$ loss:

$$l_{0-1} = \begin{cases} 0 & i = argmax\{l_1, \ldots, l_5\} \\ 1 & otherwise \end{cases} \tag{8}$$

and the log-loss:

$$l_{log} = -log(l_i) \tag{9}$$

Table 1 shows the average distribution of supervised (ground truth) behavioral modes for partitions assigned each of the labels, in the form of a confusion matrix. Partitions were obtained using nonnegative matrix factorization (NNMF) with $k = 30$, and associations between partitions and labels as described above. Data are presented after row normalization to facilitate between-row comparison.

5 Discussion

As expected, both $0-1$ and log-loss error plots are monotonically decreasing in the number of clusters (we use the term clusters here for cluster/partition/topic depending on the method under consideration). The most striking result is

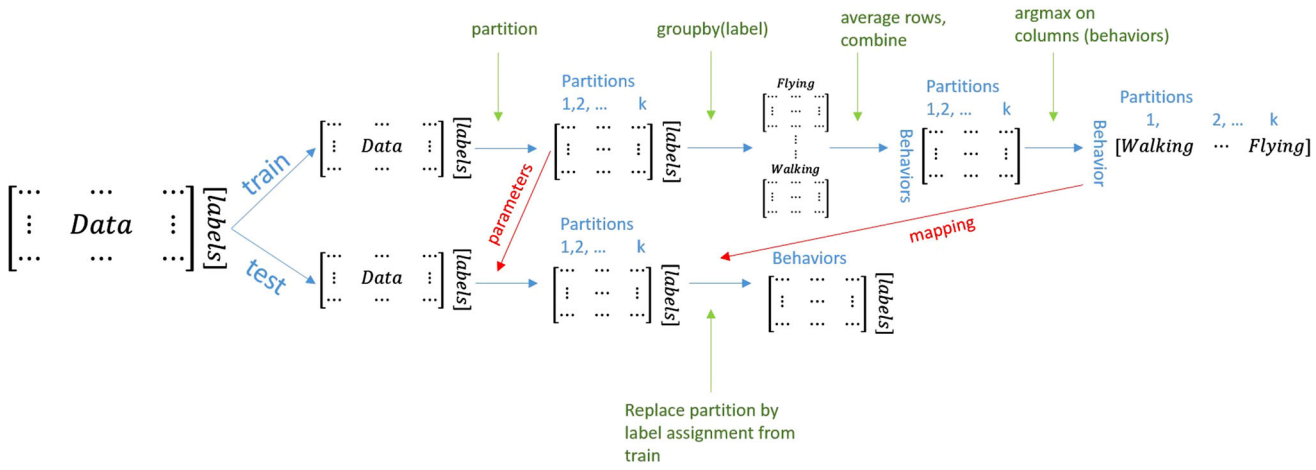


Fig. 2 Schematic flow of partition evaluation

Fig. 3 Log-loss of soft assignment to each of the ground truth classes using each of the methods under consideration. (*NNMF* nonnegative matrix factorization, *GMM* Gaussian mixture model)

that while the matrix factorization topic model method performs well compared to the other methods with respect to the log-loss metric (Fig. 3), it is not quite as good with respect to the 0–1 loss (Fig. 4).

In order to better understand these phenomena, we take a closer look at the data. Consider an observation where the animal takes a single step during the 4-s acceleration measurement window and stands still for the rest of it. In order not to dramatically underestimate the amount of walking, an observer will label this sample as walking (in fact, most samples are probably mixtures).

From the mixture property of the MS-BoP features (see Sect. 3), when using the matrix factorization topic model approach, we would expect to get a walking factor proportional to the time spent doing so in the measurement windows. Thus, for a sample with some walking (say, $<50\,\%$) we get a miss in the 0–1 loss metric, but a better score in the log-loss which is more sensitive to assignment of low probabilities to the correct class.

Table 1 sheds more light on the aforementioned result by showing average assignment of factors for each of the ground truth classes, in the form a confusion matrix. Flapping samples indeed received the highest weight, on average, on flapping factors (51.25 %), but the gliding and walking factors get over 13 % each. This may be due to the fact that Storks indeed glide between wing flaps, and may have walked prior to taking off during the observations which are inherently biased to behavior close to the ground (where the observer is). Conversely, none of the other behavioral modes include a significant amount of flapping factors.

This result may also point to the tendency (or strategy) of field observers to assign the more active behavior to mixed samples (in which case a sample where the bird flaps for a part of the duration of the measurement would be assigned to flapping, in the same sense that a step or two would qualify an otherwise stationary sample as walking).

We note that the sitting factors received factor weights higher than expected in all other behavioral modes. It might

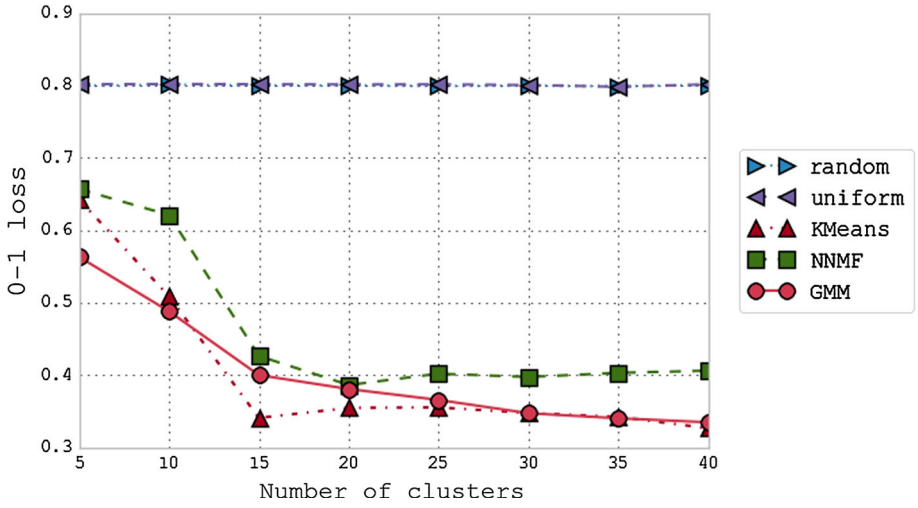


Fig. 4 0–1 loss of hard assignment to each of the ground truth classes using each of the methods under consideration. For the soft assignment partitioning methods, hard assignment is achieved using argmax. (*NNMF* nonnegative matrix factorization, *GMM* Gaussian mixture model)

Table 1 Mean label association per ground truth behavioral mode

Ground truth/assignment	Flapping (%)	Gliding (%)	Walking (%)	Standing (%)	Sitting (%)
Flapping	51.25	13.66	13.37	4.33	17.39
Gliding	0.75	49.98	8.49	3.95	36.84
Walking	2.41	19.71	43.92	20.56	13.41
Standing	0.86	13.30	1.04	74.93	9.88
Sitting	0.01	30.88	0.15	10.46	58.50

NNMF with 30 factors. Normalized rows

be interesting to try and overcome this sort of systematic error using a column normalization. We defer this to future research.

6 Conclusions

In this paper, we describe a matrix factorization-based topic model approach to behavioral mode analysis from accelerometer data and demonstrate its qualities using a large movement ecology dataset. While clustering and topic modeling with matrix factorization is by no means a new idea, the novelty here is in the integration with patch features (MS-BoP) that theoretically motivate the method in the context of time series sensor readings for behavioral mode analysis.

The main contribution of this paper is in presenting a framework that will allow for a widespread use of behavioral mode analysis in movement ecology and related fields where determining movement patterns from remote sensor readings is necessary. Further, we introduce the MS-BoP features, which may be applicable for many continuous sensor readings, and show that a linear mixture model is justified when using such features.

Acknowledgments This work was supported in part by a grant from the Israel Science Foundation (ISF) to Prof. Daphna Weinshall.

References

1. Arora, S., Ge, R., Moitra, A.: Learning topic models-going beyond svd. In: Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on, pp. 1–10. IEEE (2012)
2. Brown, D.D., Kays, R., Wikelski, M., Wilson, R., Klimley, A.: Observing the unwatchable through acceleration logging of animal behavior. Anim. Biotelem. **1**(1), 20 (2013)
3. Cooke, S.: Biotelemetry and biologging in endangered species research and animal conservation: relevance to regional, national, and IUCN Red List threat assessments. Endanger. Species Res. **4**(January), 165–185 (2008)
4. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. Workshop Stat. Learn. Comput. Vis. ECCV **1**(1–22), 1–2 (2004)
5. Ding, C.H.Q., He, X., Simon, H.D.: On the equivalence of nonnegative matrix factorization and spectral clustering. In: SDM, vol. 5, pp. 606–610. SIAM (2005)
6. Diosdado, J.A.V., Barker, Z.E., Hodges, H.R., Amory, J.R., Croft, D.P., Bell, N.J., Codling, E.A.: Classification of behaviour in housed dairy cows using an accelerometer-based activity monitoring system. Anim. Biotelem. **3**(1), 15 (2015)
7. Garriga, J., Palmer, J.R., Oltra, A., Bartumeus, F.: Expectation-maximization binary clustering for behavioural annotation. arXiv preprint arXiv:1503.04059 (2015)

8. Gleiss, A.C., Wilson, R.P., Shepard, E.L.C.: Making overall dynamic body acceleration work: on the theory of acceleration as a proxy for energy expenditure. Methods Ecol. Evolut. **2**(1), 23–33 (2011)
9. Hebblewhite, M., Haydon, D.T.: Distinguishing technology from biology: a critical review of the use of GPS telemetry data in ecology. Philos. Trans. R. Soc. Lond. Ser. B Biol. Sci. **365**(1550), 2303–2312 (2010)
10. Hindle, A., Rosen, D., Trites, A.: Swimming depth and ocean currents affect transit costs in Steller sea lions Eumetopias jubatus. Aquat. Biol. **10**(2), 139–148 (2010)
11. Holyoak, M., Casagrandi, R., Nathan, R., Revilla, E., Spiegel, O.: Trends and missing parts in the study of movement ecology. Proc. Natl. Acad. Sci USA **105**(49), 19060–19065 (2008)
12. Kays, R., Crofoot, M.C., Jetz, W., Wikelski, M.: Terrestrial animal tracking as an eye on life and planet. Science **348**(6240), aaa2478 (2015)
13. Kern, N., Schiele, B., Schmidt, A.: Multi-sensor activity context detection for wearable computing. In: Ambient Intelligence, pp. 220–232. Springer (2003)
14. Li, T., Ding, C.: The relationships among various nonnegative matrix factorization methods for clustering. In: Data Mining, 2006. ICDM'06. Sixth International Conference on, pp. 362–371. IEEE (2006)
15. Lin, C.-B.: Projected gradient methods for nonnegative matrix factorization. Neural Comput. **19**(10), 2756–2779 (2007)
16. Louzao, M., Weigand, T., Bartumeus, F., Weimerskirch, H.: Coupling instantaneous energy-budget models and behavioural mode analysis to estimate optimal foraging strategy: an example with wandering albatrosses. Mov. Ecol. **2**(8), 8 (2014). doi:10.1186/2051-3933-2-8
17. Minnen, D., Starner, T., Ward, J.A., Lukowicz, P., Tröster, G.: Recognizing and discovering human actions from on-body sensor data. In: Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on, pp. 1545–1548. IEEE (2005)
18. Nathan, R., Getz, W.: A movement ecology paradigm for unifying organismal movement research. Proc. Natl. Acad. Sci. USA **105**(49), 19052–19059 (2008)
19. Nathan, R., Spiegel, O., Fortmann-Roe, S., Harel, R., Wikelski, M., Getz, W.M.: Using tri-axial acceleration data to identify behavioral modes of free-ranging animals: general concepts and tools illustrated for griffon vultures. J. Exp. Biol. **215**(6), 986–996 (2012)
20. Nguyen, A., Moore, D., McCowan, I.: Unsupervised clustering of free-living human activities using ambulatory accelerometry. In: Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE, pp. 4895–4898. IEEE (2007)
21. Nguyen, T., Phung, D., Gupta, S., Venkatesh, S.: Extraction of latent patterns and contexts from social honest signals using hierarchical dirichlet processes. In: Pervasive Computing and Communications (PerCom), 2013 IEEE International Conference on, pp. 47–55. IEEE (2013)
22. Pantelopoulos, A., Bourbakis, N.G.: A survey on wearable sensor-based systems for health monitoring and prognosis. Syst. Man Cybern. Part C Appl. Rev. IEEE Trans. **40**(1), 1–12 (2010)
23. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
24. Rai, A., Yan, Z., Chakraborty, D., Wijaya, T.K., Aberer, K.: Mining complex activities in the wild via a single smartphone accelerometer. In: Proceedings of the Sixth International Workshop on Knowledge Discovery from Sensor Data, pp. 43–51. ACM (2012)
25. Resheff, Y.S., Rotics, S., Harel, R., Spiegel, O., Nathan, R.: AcceleRater: a web application for supervised learning of behavioral modes from acceleration measurements. Mov. Ecol. **2**(1), 25 (2014)
26. Resheff, Y.S., Rotics, S., Nathan, R., Weinshall, D.: Matrix factorization approach to behavioral mode analysis from acceleration data. In: Data Science and Advanced Analytics (DSAA), 2015 International Conference on IEEE (2015)
27. Sakamoto, K.Q., Sato, K., Ishizuka, M., Watanuki, Y., Takahashi, A., Daunt, F., Wanless, S.: Can ethograms be automatically generated using body acceleration data from free-ranging birds? PloS one **4**(4), e5379 (2009)
28. Sellers, W.I., Crompton, R.H.: Automatic monitoring of primate locomotor behaviour using accelerometers. Folia Primatol. Int. J. Primatol. **75**(4), 279–293 (2004)
29. Smouse, P.E., Focardi, S., Moorcroft, P.R., Kie, J.G., Forester, J.D., Morales, J.M.: Stochastic modelling of animal movement. Philos. Trans. R. Soc. Lond. Ser. B Biol. Sci. **365**(1550), 2201–2211 (2010)
30. Spiegel, O., Harel, R., Getz, W.M., Nathan, R.: Mixed strategies of griffon vultures (Gyps fulvus) response to food deprivation lead to a hump-shaped movement pattern. Mov. Ecol. **1**(1), 5 (2013)
31. Takahashi, M., Tobey, J.R., Pisacane, C.B., Andrus, C.H.: Evaluating the utility of an accelerometer and urinary hormone analysis as indicators of estrus in a Zoo-housed koala (Phascolarctos cinereus). Zoo Biol. **28**(1), 59–68 (2009)
32. Wang, Y.-X., Zhang, Y.-J.: Nonnegative matrix factorization: a comprehensive review. Knowl. Data Eng. IEEE Trans. **25**(6), 1336–1353 (2013)
33. Williams, H., Shepard, E., Duriez, O., Lambertucci, S.: Can accelerometry be used to distinguish between flight types in soaring birds? Anim. Biotelem. **3**(1), 1–11 (2015)
34. Wilson, R.P., White, C.R., Quintana, F., Halsey, L.G., Liebsch, N., Martin, G.R., Butler, P.J.: Moving towards acceleration for estimates of activity-specific metabolic rate in free-living animals: the case of the cormorant. J. Anim. Ecol. **75**(5), 1081–1090 (2006)
35. Yoda, K., Sato, K., Niizuma, Y., Kurita, M., Bost, C., Le Maho, Y., Naito, Y.: Precise monitoring of porpoising behaviour of Adélie penguins determined using acceleration data loggers. J. Exp. Biol. **202**(22), 3121–3126 (1999)
36. Zagoris, K., Pratikakis, I., Antonacopoulos, A., Gatos, B., Papamarkos, N.: Distinction between handwritten and machine-printed text based on the bag of visual words model. Pattern Recognit. **47**(3), 1051–1062 (2014)

2.2 Mental Health

This paper was accepted for publication in the upcoming IEEE Conference on Body Sensor Networks (BSN), to be held 4-7 March, 2018, in Las Vegas, Nevada, USA.

Topic Models for Automated Motor Analysis in Schizophrenia Patients

Talia Tron*[1,4] Yehezkel S. Resheff*[1,4] Mikhail Bazhmin[2] Abraham Peled[2,3] Alexander Grinsphoon[2,3] Daphna Weinshall[4]

***Abstract*— Wearable devices fitted with various sensors are increasingly being used for the automatic and continuous tracking and monitoring of patients. Only first steps have been taken in the field of psychiatric care, where long term tracking of patient behavior holds the promise to help practitioners to better understand both individual patients, and the disorders in general. In this paper we use topic models for unsupervised analysis of movement activity of schizophrenia patients in a closed ward setting. Results demonstrate that features computed on the basis of this analysis differentially characterize interesting sub-populations of schizophrenia patients. Positive-signs schizophrenia sub-population was found to have high *motor richness* and low *typicallity*, while negative-signs patients had low *motor richness* and lower *typicality*. In addition we design a classifier which correctly classified up to 80% of the clinical sub-population (f-score=0.774) based on motor features.**

I. INTRODUCTION

Motor peculiarities are an integral part of the schizophrenia disorder, both as aspects of the more general symptom repertoire, and in response to medications. To date, these symptoms are typically evaluated in a descriptive manner based on psychiatric rating scales such as the Positive and Negative Syndrome Scale (PANSS) [1], or targeted specifically using subjective clinical scales such as the Unified Dyskinesia Rating Scale (UDysRS) [2] and the Unified Parkinson's Disease Rating Scale (MDS-UPDRS) [3]. The lack of objective, quantitative methods for measuring these symptoms, and the insufficient conceptual clarity around them, may cause multiple interpretations of phenomenology, leading to low reliability and validity of diagnosis. In addition, the symptom evaluation process requires expert staff and availability of resources, and is therefore not done frequently enough to capture more subtle changes in spontaneous and drug-induced conditions. Clearly there is an urgent need for automatic monitoring and assessment tools.

The last decade has seen a steep rise in the use of wearable devices for medical applications in a range of fields, from human physiology [4] to movement disorders [5], [6] and mental health [7]. Accelerometers and gyroscopes, which are commonly embedded in smart-watches and other wearable devices, are now used to assess mobility and recognize activity. In a clinical setting, these sensors may be used in order to detect changes in high-level movement parameters, track their dynamics and correlate them with mental state.

Measures of activity such as step counts and overall activity, as well as changes thereof, have already been shown to effectively provide insights into the state of patients in a closed ward mental hospital setting [8]. Unsupervised behavioral mode analysis of sensor data, such as topic models, have previously been used in other domains to provide a high level description of behavior [9]. Here we combine these ideas and use topic models for unsupervised analysis of patient activity. These models allow a richer, qualitative description of behavior than the aforementioned measures. We demonstrate that features computed on the basis of topic model analysis differentiate sub-populations of patients.

II. MATERIALS AND METHODS

A. Study Design

27 inpatients from the closed wards at Shaar-Meashe MHC participated in the study. Most participants (21/27) were diagnosed with schizophrenia according to the DSM-5, 3 with paranoid schizophrenia, 2 with schizoaffective disorder, and one with psychotic state cannabinoids. Participants' age varied from 21 to 58 (mean 37.5), with course of illness varying from 0 (first hospitalization) up to 37 years (mean 16.9 years). Two of the patients dropped out of the study after less than a day due to lack of cooperation. The remaining 25 patients were followed for a period of three weeks on average (6-52 days).

The study was conducted in natural settings, where patients were *not* required to change any personal or medical procedure. On top of the normal care, every patient underwent an additional evaluation by a trained psychiatrist twice a week. The procedure included clinical evaluation of symptom severity using PANSS; Neurological Evaluation Scale (NES [10]) assessment was conducted as a control. In addition, continuous medication monitoring (type, dosage and frequency) by the clinical staff was observed.

All procedures performed in the study were in accordance with the ethical standards of the institutional research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards.

B. Data Acquisition

Each participant was fitted with a smart-watch (GeneActiv[1]) with tri-axial accelerometer embedded sensors, the high

* T. Tron and Y.S. Resheff contributed equally to this work.
[1]The Edmond and Lily Safra center (ELSC) for Brain Science, Hebrew University, Jerusalem 91904, Israel talia.tron@mail.huji.ac.il
[2]Rappaport Faculty of Medicine, Technion Institute of Technology, Haifa 3200003, Israel
[3]Sha'ar Menashe Mental Health Center, Sha'ar Menashe 38706, Israel
[4]The Rachel and Selim Benin School of Computer Science and Engineering, Hebrew University, Jerusalem 91904, Israel

[1]https://www.geneactiv.org/

frequency output ($50Hz$) of which was stored on memory cards. Data was collected continuously throughout the experiment for a total of 489 days, from 25 patients. In order to reduce noise introduced by the variability in patient activity, the analysis focused on fixed time windows corresponding to regular departmental daily activity: Occupational therapy time slots (10am-11am), lunch (12pm-1pm), and indoor free time (4pm-5pm). In addition, we calculated full day features (6am-10pm) and used night time features (10pm-6am) to evaluate sleep quality. Weekends were excluded from the analysis.

III. ANALYSIS

A. Revising clinical assessment

The 30-item Positive and Negative Syndrome Scale (PANSS) was reduced to a five-factor description (Positive, Negative, Disorganized/Concrete, Excited and Depressed), according to the consensus model suggested by Wallwork et al. [11], based on 25 previously published models and refined with confirmatory factor analysis (CFA). Only the positive and negative factors were used for further analysis.

Clinical observations show that changes in a patient's symptoms occur continuously on a daily basis [12]. We therefore interpolated the bi-weekly PANSS factor scores, to achieve smooth daily scoring of symptoms. This was done using the PCHIP 1-d monotonic cubic interpolation, resulting in 494 data points (originally 118).

Interpolated data points were used to classify clinical sub-populations of patients on a daily basis. Sub-populations included patients with "High positive" symptoms, "High negative", "High negative and positive", and "Low" level symptoms. The remaining intermediate data points were discarded from the classification. This sub-typing allowed us to explore how different motor features are expressed in different clinical manifestations. Clustering was done based on the percentile of the positive and negative factors, each axis separately (Fig. 1).

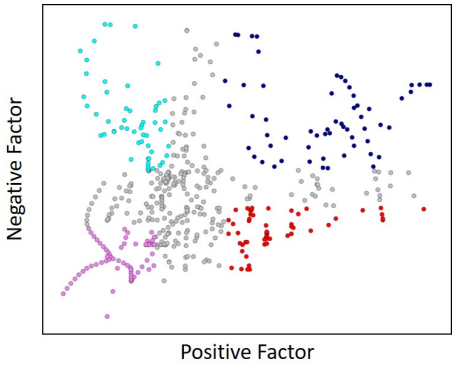


Fig. 1. Clinical sub-populations. Each data point represents the severity of the positive and negative factors for a specific patient in a specific day (N=494) based on the interpolated PANSS factors data. In the "Low" sub-population (magenta, N=65) both negative and positive symptoms lie in the bottom quartile, while in the "High negative and positive" (blue, N=59) both lie in the top quartile. The "High negative" sub-population (cyan, N=53) lies in the top vertical quartile with positive symptom values lower than median, while the "High positive" (red, N=57) lies in the top horizontal quartile with lower than median negative symptom values. The remaining data points (N=260) were classified as "Intermediate" (green).

B. Online computation of "patch features"

Topic model analysis requires the discretization of the continuous accelerometer signal both in time and in intensity, to produce word analogues – motor words. This mapping involves the creation of a code-book. The patch feature topic model procedure described in [9] contains a codebook generation stage where clustering (k-means) is applied to segments (a.k.a. patches) from the entire dataset. Given the larger dataset at hand, we designed an online greedy approximation to this procedure.

Specifically, the idea behind an online generation of code-book is to follow the way a dictionary would be created for a natural language corpus. The process proceeds with a single pass over the data. Each word is considered sequentially, and added to the dictionary on first encounter.

Since the words we are using describe the continuous accelerometer signal, we must also define what we mean by a word. Ideally, the dictionary should not be affected by small random changes in the signal. Additionally, since many behavioral modes are periodic to some extent, we would like the representation to allow wrap-around of patches. This would imply that the sequences of patches ABC and BCA, for example, have similar representation in the dictionary.

We achieve both these goals by using a discretized version of the signal and wrap-around equivalence classes. We use a SAX-like method [13] to encode each patch into a string. The process is as follows: Each interval on the time-axis is replaced by the mean value in the interval. Next, these point-values are replaced by a letter (discretized) according to their value. The output of this process is a string of length $\frac{patch-size}{interval-size}$ over a pre-determined alphabet.

Algorithm 1 Online codebook creation

```
codebook ← empty list
for each patch in the dataset do
    patch_word ← SAX_representation(patch)
    if patch_word (or equivalent) not in codebook then
        append patch_word to the codebook
    end if
end for
return codebook
```

The procedure resulted in 150K distinct words which described the entire dataset, distributed much as would be expected from a text corpus (see top panel in Fig. 2).

C. Topic Modeling over Motor Words

Latent Dirichlet Allocation (LDA) is a widely used topic model, with origins in natural language processing, and applications in many domains ranging from music modeling to motion of cars. On top of their traditional purpose of finding hidden semantic structures in data, these models have been shown to be useful for detecting surprising (or novel) events [14], [15].

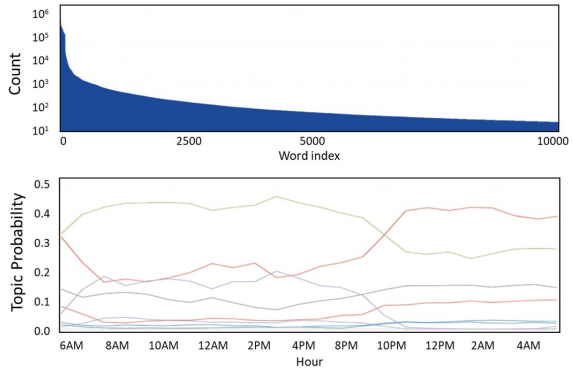


Fig. 2. Top - the motor-word frequency histogram for the entire dataset, truncated after the most common 10K words. These pseudo-words demonstrate the long-tail scale-free property characteristic of a natural language. Bottom- the daily topic distribution vector averaged over all patients.

Data was divided into blocks of 15 minutes of continuous signal; these serve as documents for the topic model, each represented as a histogram over the motor words as described above. The LDA process provides as output both a distribution over topics for each of the documents, and a distribution over words for each of the topics. Subsequently, a specific time window of a specific patient is characterized by a probability vector over the topics. The bottom panel in Fig. 2 demonstrates the distribution of the 10 topics used here over all patients and days. We can see that topic 6 (green) and topic 10 (purple) are typically prominent during the day, while other topics are more likely to occur during the night or throughout.

D. Topic Features

The advantage of using a data-driven unsupervised representation is that its features, unlike the supervised energy and step-count measures [8], are not directly connected with the intensity of the motor signal. Instead, this representation captures the *quality* of motor behavior in a given period of time. For example, a very repetitive behavior can be expressed by a low number of unique 'motor-words' in a specific time window. This allows us to compare patients behavior to themselves and to others in different activity windows, and thus be able to measure 'typicality' of the behavior for instance. Three Features were calculated based on topic models, separately for each data point (namely for each patient on each day, and each of the predefined activity windows described in section II-B):

1) Motor Richness: The normalized distinct word count per activity window.

$$\text{Motor Richness} = \frac{\text{distinct word count in window}}{\text{window length}}$$

This measure represents the range of motor activity repertoire. A low score implies that the patient repeatedly performed similar movements, while a high score corresponds to the use of many different movement patterns.

2) Consistency:

$$\text{Consistency} = 1 - D_{KL}(v \parallel \bar{v})$$

where v denotes the topic distribution over the time window, and $\bar{v}$ the mean topic distribution for the patient in the same window over all measured days. D_{KL} denotes the Kullback-Leibler divergence. This score measures how regular the patient's motor behavior is in the given time window.

3) Typicality: the entropy of the topic distribution vector.

$$\text{Typicality} = H[v] = -\sum_{i=1}^{10} v_i \log(v_i)$$

Low entropy implies that a small number of topics can capture the activity. High entropy implies that the observed activity is a mixture of many topics. We name this measure *typicality* since typical activity should be captured by one (or a few) topics, thus producing low-entropy topic distributions [15].

The manifestation of each feature in clinical sub-populations was tested using one-way ANOVA separately for each of the activity windows. In addition, a learning algorithm for automated sub-population classification was designed and evaluated.

E. Classification Algorithm

Classification was carried out using a two-step algorithm based on linear support vector machines (SVM) and decision trees classifiers, in order to distinguish between different sub-populations (described in III-A) based on motor features (Algorithm 2). The algorithm was trained to discriminate sub-populations, and specifically classify "High positive" vs. "High negative" and "Low" vs. "High positive and negative".

In the first step, individual classifiers were trained for each activity window separately (lunch, occupational therapy, free-time, day, night, and all). In the second step, the probabilistic output of the 6 time-specific first-stage classifiers was used to train a second, daily-model, which determined the clinical category.

Feature selection was done based on the ANOVA f-values of each individual feature on train data. These were calculated separately for each activity window, and the same features were used also for testing.

Algorithm 2 Two stage algorithm for patient sub-type classification based on activity in time-windows.

1: **for all** time-windows w_i **do**
2: train base classifier c_i on w_i and target y
3: **end for**
4: **for all** time-windows w_i **do**
5: $\hat{y}_i \leftarrow$ prediction of c_i on w_i
6: **end for**
7: train final classifier c on the set of first-stage predictions $\hat{y}_i$ and target y

The algorithm was evaluated in a leave-one-out framework, where in each iteration a different observation (specific

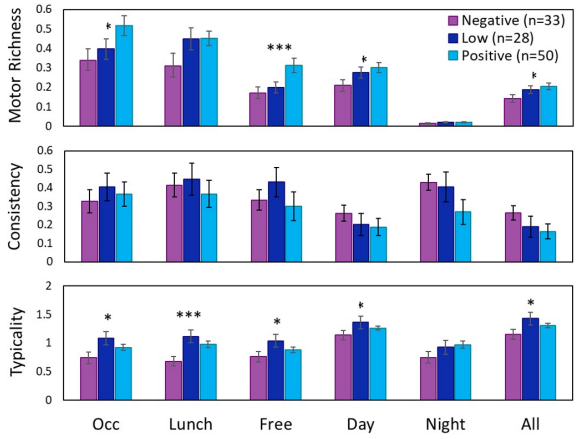


Fig. 3. ANOVA results for topic features in 3 clinical sub-populations (see Fig. 1): "High positive" (denoted *Positive*), "High negative" (denoted *Negative*), and "Low". The analysis was repeated separately for each time window (X-axis). *Motor richness* was highest in the *Positive* sub-population (cyan) and lowest in the *Negative* sub-population (magenta). This was most significant during free time, but was also true for all other activity windows (p-values between 0.05-0.07 are marked by half an asterisk). *Typicality* was generally highest in the *Low* sub-population, and lowest in the *Negative* sub-population, with the most significant difference during lunch time. No significant group different was found for *Consistency* although it was lowest in the *Positive* sub-population in all activity windows.

patient in a specific day) was left out and the model was trained on the remaining data and tested on the left out sample. To avoid possible contamination of test data (leakage) due to observation interpolation, when using an interpolated point as the test, all actual observations it was based upon were excluded from the train data.

IV. RESULTS

A. Motor Activity in different Clinical Sub-populations

Fig. 3 summarizes the results of subjecting all features to ANOVA analysis. *Motor richness* is consistently highest for the "High positive" sub-population, and lowest for the "High negative" sub-population, with the "Low" sub-population somewhere in the middle. This indicates that patients with active positive symptoms tend to have a higher variety of motor activities, while negative symptoms are expressed in poorer movement repertoire. The trend was evident in all activity windows but was only found significant during free time ($F = 5.09$, $p = 0.0077$).

As expected, *typicality* is highest for the "Low" sub-population, consistently over all activity windows. The lowest *typicality* is observed in the "High negative" sub-population, indicating that the motor activity of these patients is less similar to the common motor behavior. The biggest group difference was found over lunch time ($F = 7.48$, $p = 0.00090$) but it was also significant during occupational therapy ($F = 4.39$, $p = 0.015$), free time ($F = 3.38$, $p = 0.037$) and throughout the day ($F = 3.78$, $p = 0.026$). No group difference was found for *consistency*, although it was lower in the "High positive" sub-population in all activity windows.

B. Classification Results

For "High positive" vs. "High negative" classification, the best results were achieved using linear SVM for the first stage (window-based model, see Algorithm 2) with top-5 selected features, and decision tree for the second (daily) stage. The algorithm correctly classified 78% of the"High negative" observations, and 58% of the "High positive" observations. All together the mean precision was 0.651 and mean recall was 0.654 on test data (f-score=0.652).

Slightly better results were achieved for the "Low" vs. "High positive and negative" classification, using linear SVM for both stages and top-5 selected features. Here the algorithm correctly classified 81% of the "Low" observations and 70% of the "High positive" observations. All together the mean precision was 0.757 and mean recall was 0.748 on test data (f-score=0.774).

REFERENCES

[1] S. R. Kay, A. Flszbein, and L. A. Opfer, "The positive and negative syndrome scale (panss) for schizophrenia.," *Schizo. bulletin*, vol. 13, no. 2, p. 261, 1987.

[2] C. G. Goetz, J. G. Nutt, and G. T. Stebbins, "The unified dyskinesia rating scale: presentation and clinimetric profile," *Mov. Dis.*, vol. 23, no. 16, pp. 2398–2403, 2008.

[3] C. G. Goetz, B. C. Tilley, S. R. Shaftman, G. Stebbins, *et al.*, "Movement disorder society-sponsored revision of the unified parkinson's disease rating scale (mds-updrs): Scale presentation and clinimetric testing results," *Mov. dis.*, vol. 23, no. 15, pp. 2129–2170, 2008.

[4] J. Staudenmayer, S. He, A. Hickey, J. Sasaki, and P. Freedson, "Methods to estimate aspects of physical activity and sedentary behavior from high-frequency wrist accelerometer measurements," *Journal of Applied Physiology*, vol. 119, no. 4, pp. 396–403, 2015.

[5] R. LeMoyne, T. Mastroianni, M. Cozza, C. Coroian, and W. Grundfest, "Implementation of an iphone for characterizing parkinson's disease tremor through a wireless accelerometer application," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pp. 4954–4958, IEEE, 2010.

[6] A. Wagner, N. Fixler, and Y. S. Resheff, "A wavelet-based approach to moniotring parkinson's disease symptoms," *International Conference on Acoustics, Speech, and Signal Processing*, 2017.

[7] R. Wang, F. Chen, Z. Chen, T. Li, G. Harari, S. Tignor, X. Zhou, D. Ben-Zeev, and A. T. Campbell, "Studentlife: assessing mental health, academic performance and behavioral trends of college students using smartphones," in *Proceedings of the 2014 ACM Int Joint Conf on Pervasive and Ubiquitous Computing*, pp. 3–14, ACM, 2014.

[8] T. Tron, Y. S. Resheff, M. Bazhmin, A. Peled, and D. Weinshall, "Real-time schizophrenia monitoring using wearable motion sensitive devices," in *MobiHealth*, Springer, 2016.

[9] Y. S. Resheff, S. Rotics, R. Nathan, and D. Weinshall, "Topic modeling of behavioral modes using sensor data," *International Journal of Data Science and Analytics*, vol. 1, no. 1, pp. 51–60, 2016.

[10] R. W. Buchanan and D. W. Heinrichs, "The neurological evaluation scale (nes): a structured instrument for the assessment of neurological signs in schizophrenia," *Psychiatry research*, vol. 27, no. 3, pp. 335–350, 1989.

[11] R. Wallwork, R. Fortgang, R. Hashimoto, D. Weinberger, and D. Dickinson, "Searching for a consensus five-factor model of the positive and negative syndrome scale for schizophrenia," *Schizophrenia research*, vol. 137, no. 1, pp. 246–250, 2012.

[12] S. Arieti, *Interpretation of schizophrenia*. Basic Books (AZ), 1974.

[13] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pp. 2–11, ACM, 2003.

[14] U. Shalit, D. Weinshall, and G. Chechik, "Modeling musical influence with topic models," in *International Conference on Machine Learning*, pp. 244–252, 2013.

[15] A. Hendel, D. Weinshall, and S. Peleg, "Identifying surprising events in videos using bayesian topic models," in *Asian Conference on Computer Vision*, pp. 448–459, Springer, 2010.

Chapter 3

Trajectory Segmentation

This paper [1] was presented at the IEEE International Conference on Big Data (Big Data 2016), held December 5-8, 2016, in Washington D.C., USA.

References

[1] Yehezkel S Resheff. “Online trajectory segmentation and summary with applications to visualization and retrieval”. In: *Big Data (Big Data), 2016 IEEE International Conference on*. IEEE. 2016, pp. 1832–1840.

Online Trajectory Segmentation and Summary With Applications to Visualization and Retrieval

Yehezkel S. Resheff
Edmond and Lily Safra Center for Brain Sciences
The Hebrew University of Jerusalem
yehezkel.resheff@mail.huji.ac.il

***Abstract*—Trajectory segmentation is the process of subdividing a trajectory into parts either by grouping points similar with respect to some measure of interest, or by minimizing a global objective function. Here we present a novel online algorithm for segmentation and summary, based on point density along the trajectory, and based on the nature of the naturally occurring structure of intermittent bouts of locomotive and local activity. We show an application to visualization of trajectory datasets, and discuss the use of the summary as an index allowing efficient queries which are otherwise impossible or computationally expensive, over very large datasets.**

***Index Terms*—trajectory analysis; visualization; retrieval; data-management**

I. Introduction

The steadily dropping cost of data collection and storage is impacting both industry and science. This so-called "data-era" has not skipped movement data, which is now being acquired at unprecedented rates. In the field of Ecology and Animal Tracking, movement data is being collected at a global [25], [13] as well as a regional [7], [29] scale. Car [16], [15] and ship [26] trajectories are recorded for control, optimization, and safety purposes, and pedestrians are tracked for health, safety, and navigation utilities [12], [2]. With this large amount of data rolling in, new and more efficient methods must be developed in order to visualize, analyze, and gain insight and knowledge.

One of the most fundamental computations associated with understanding movement data is segmentation of trajectories. Traditionally, a segmentation proceeds by defining a feature of a single point, then dividing the entire trajectory into sub-trajectories which are uniform (in some sense) with respect to this feature [9].

Features used for this purpose include truly point-wise metrics, such as speed and heading [9], as well as those designed to capture the behavior in the vicinity of a point, such as first passage time (FPT) [11] and residence time [5].

A slightly different approach to trajectory segmentation is based on directly optimizing a cost function related to the segments themselves. Warped K-Means [18] is a trajectory oriented adaptation of the well-known K-Means algorithm, where the regular (mean square distance from centroid) objective is locally optimized, under the additional constraint that a cluster may only contain consecutive points.

The method we propose here addresses two main limitations of the warped K-Means algorithm. First, for very large datasets it would be necessary to have an online algorithm able to deal with an unknown number of segments. Second, the optimization criterion used should depend on the local trajectory density, allowing long-distance locomotion to be segmented differently from dense local regions. These ideas are formalized in section III.

We further present two main applications of the proposed method, addressing major tasks related to large-scale trajectory data. The first is the task of visualization of a large number of possibly dense and overlapping trajectories. The challenge arising in this context is that amounts of data are often prohibitively large and can't be rendered simply on a map. Even if all points are plotted, the trajectories will be obscured by the sheer number of points. We show that our method is appropriate for descriptive visualization of a large number of trajectories, and allows single paths to be easily resolved and compared.

The second application which we briefly discuss is fast querying and retrieval over a trajectory database. Here, the proposed method serves as an index into the full trajectory data, allowing various otherwise intractable queries. Combined with the visualization component, we suggest a real-time visual search engine for exploration of large scale trajectory data.

Our main contributions are the online segmentation algorithm, which is a fundamental pre-processing step and may be used for a wide range of downstream processing, and the visualization method based on it, allowing concurrent visualization of many trajectories. The query engine initially described here will be further developed in future research.

The rest of the paper is structured as follows: section II briefly reviews related work in trajectory segmentation. In section III, the algorithm is described and discussed. Finally, section IV contains the proposed applications of the segmentation method for visualization and data retrieval.

II. Related Work

In the field of Ecology, Trajectory segmentation is widely used in order to process the path of an animal into functionally homogeneous units. The main approach used is to compute a single point-wise feature along the trajectory and then group similar points, with respect to the feature. For example, a method of change-point analysis [17] has been used in conjunction with Residence Time (a metric of the total amount

of time spent in the vicinity of a point [5]). Another approach is to segment a trajectory with respect to the momentary behavior of an animal along a path [23].

A more general framework was also suggested [9], based on finding the segmentation with the minimum number of segments, such that a given metric will not differ within any segment by more than a pre-defined factor. For a wide range of metrics (such as speed, velocity, heading, curvature, etc.), this can be achieved in $O(nlogn)$.

Warped k-means [18] adopts a completely different view to the segmentation problem. Since this is essentially a problem of clustering similar points, the method attempts (as in the well known K-means algorithm [20]) to find centroids and assignments, in order to minimize the mean square distance of each point to the centroid it is assigned to, under the additional constraint that if two points are in the same cluster, so are all the points between them along the trajectory.

As the volume of information increases, traditional methods which were very useful for small to medium datasets are no longer applicable, and fast (preferably linear time, online methods) must be developed. In the next section we start by introducing definitions and notation, then describe the proposed algorithm. Then in section IV we continues with two applications of the method, which are fundamental to trajectory analysis. The first application we demonstrate is concurrent visualization of many trajectories. Next, we discuss the application of the method to the problem of data retrieval over large datasets.

III. Trajectory Segmentation

Definition 1. *A trajectory T is a continuous mapping from a time interval to positions:*

$$T : [a, b] \rightarrow R^k$$

where $T(t)$ is the position at time $t \in [a, b]$

in most cases, since we are dealing with trajectories in the physical world, we will have $k \in \{2, 3\}$. While all data presented in this paper is $2D$ trajectory data, the concepts and methods discussed are valid for trajectory in arbitrary dimensional space.

Definition 2. *A sampled trajectory is a set of time and position tuples:*

$$\{(t_i, p_i)\}_{i=1}^{n}$$

where for some trajectory T, and some set of timestamps $t \in [a, b]$, $p = T(t)$.

In practice, almost all trajectory data is obtained in the form of a sampled trajectory. We note that this notation allows for either equally or unequally spaced samples. We will always assume that the samples are sorted by time, ascending.

Definition 3. *A segmentation of a sampled trajectory $\{(t_i, p_i)\}_{i=1}^{p}$ is a list of cutoff indexes:*

$$1 = c_0 < c_1 < ... < c_k = p$$

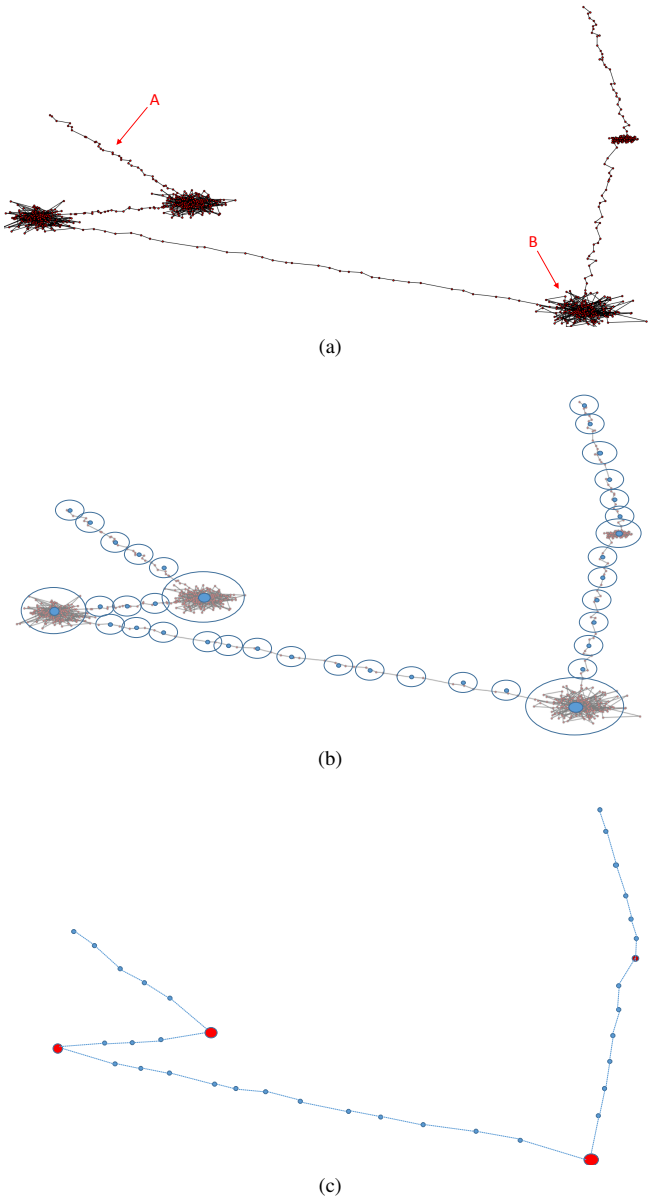


Figure 1: An outline of the segmentation method. (a) A raw trajectory consisting of locomotive parts (a-A) and local parts (a-B). (b) A segmentation overlayed over the raw trajectory. During locomotive periods, a new point is added to the segmentation only when a pre-defined distance is crossed from the previous point. Local periods are represented by a single point (c) the centroids of the segments form a condensed representation of the trajectory, which is in itself a trajectory. Red points correspond to local parts of the original trajectory.

The $i-th$ segment is then a subset of the sampled trajectory:

$$\{(t_i, p_i)\}_{i=c_{i-1}}^{c_i}$$

Each segment, is itself a sampled trajectory.

Algorithm 1 Trajectory Segmentation

input:

- T - sampled trajectory
- min_r- minimal radius for a segment
- $min\_density$ - minimal density for a segment or radius larger than min_r

output:

- $cutoffs$ - the cutoff values for the segmentation of T
- $centroids$ - the list of centroids of each segment. By adding a time-stamp this becomes a sampled trajectory. Omitted for simplicity.

$cutoffs \leftarrow$ empty list
append 0 to $cutoffs$
$centroids \leftarrow$ empty list
$current\_centroid \leftarrow T[0]$
$n\_points \leftarrow 1$
$radius \leftarrow 0$
for i = 1 to T.length **do**
  $n\_points \leftarrow n\_points + 1$
  $radius \leftarrow max(radius, dist(current\_centroid, T[i]))$
  if $radius > min\_r$ **then**
    $density \leftarrow n\_points/(pi * radius^2)$
    if density $< min\_density$ **then**
      append i to $cutoffs$
      append $current\_centroid$ to $centroids$
      $current\_centroid \leftarrow T[i]$
      $n\_points \leftarrow 1$
      $radius \leftarrow 0$
      continue
    end if
  end if
  $current\_centroid \leftarrow ((n-1) * current\_centroid + T[i])/n$
end for
append $T.length$ to $cutoffs$
append $current\_centroid$ to $centroids$
return $cutoffs, centroids$

A. Algorithm outline

The proposed method is based on the observation that for many types of trajectories, there exist intermittent periods of dense and sparse positions (Figure 1a). Consider for instance the trajectory of a person throughout a day. Dense local segments correspond to time spent at and around the home, workplace, shopping mall etc. (sometimes called a stay-point [33], [19], [32]), whereas between these segments we might expect long and sparsely distributed locomotive activity. The same pattern is seen for animals; foraging behavior is characterized by dense usage of relatively small areas, whereas long locomotive segments connect between such areas.

The spread of points during a period of local activity has more than one source. When measuring the position of a completely static entity, measurement noise will cause the locations to be presented as a cloud around the actual location. While this in itself can be overcome by averaging (neglecting adverse boundary effects depending on the length of the filter), for a locally moving entity, the measured spread of points will be a sum of the actual movement and the noise. Furthermore, when operating on a stream, it is not clear how to average local periods while leaving locomotive periods intact.

The segmentation approach allows differential consideration of local (or static) versus locomotion parts of a trajectory. The proposed segmentation algorithm will average a full local segment, reducing it to a single centroid, while maintaining much of the structure of the other parts of the trajectory (up to a pre-defined radius).

When constructing the algorithm, we require the following:

1) The segmentation must work on a trajectory stream (i.e. $O(n)$ and finite memory)
2) For a given trajectory, the segmentation should be invariant to sampling density. For instance, for a trajectory derived from a drive from home to work, whether it is a fast drive, or a constant traffic jam (corresponding to dense or sparse sampling in space, assuming a constant sampling rate in time), the size of the resulting segmentation should be the same.
3) Dense regions corresponding to local activity (shopping area for people, foraging for animals), should be placed in a single segment irrespective of time or area.

In order to satisfy the first requirement, the algorithm (Algorithm 1) proceeds in one pass over each trajectory. The number of points, and the radius and density of the current segment are constantly maintained. The second requirement is satisfied by starting a new segment only after a pre-defined radius is passed (i.e. there is a minimum radius for a segment). As a result, denser sampling will have no effect on the segments produced.

The third requirement is satisfied by letting a segment grow past the pre-defined radius as long as the density in the segment is large enough. See Algorithm 1 for the full account of the method, and Figure 1b-c for a visualization.

The output of Algorithm 1 is both the segmentation and the centroids of each of the segments. Recall a segmentation (definition 3) is a list of the cutoff indexes, and thus does not serve in itself as a (standalone) condensed representation of the trajectory. The segment centroids however do serve as such a representation, and the applications we suggest for this method (section IV below), are all based on considering them instead of the full data. We note that by adding a time-stamp to the centroids, they become themselves a sampled trajectory. This time-stamp, will in practice be either the mean time of the points in the segment, or the time of the beginning of the segment.

B. Comments on the calculation of density of points in a segment

A central component of Algorithm 1 is maintaining of the measure of density of points in the current segment of the trajectory. This is done by maintaining the centroid and radius of the segment – effectively, the smallest circle (or sphere if the trajectory is in higher dimension), in which the segment is contained. The density is then the number of points divided by the area (Algorithm 1; line 11).

The main limitation of this approach is that when the segment is not a in the shape of a circle (as is often the case with trajectory data), the circle may highly over-estimate the volume in which the points are actually contained. In this section we develop the idea of replacing the circle with other means of calculation, to better estimate the density, while maintaining the linear runtime and bounded memory characteristics of the original algorithm.

Algorithm 2 Online Rectangular Approximation of Point Density

input:

- stream - sampled trajectory in the form of a stream

1: $maxx, maxy \leftarrow -\infty$
2: $minx, miny \leftarrow +\infty$
3: $n \leftarrow 0$
4: **while** stream.has_next() **do**
5: $\quad x, y \leftarrow$ stream.next()
6: $\quad n \leftarrow n + 1$
7: $\quad maxx \leftarrow max(maxx, x)$
8: $\quad maxy \leftarrow max(maxy, y)$
9: $\quad minx \leftarrow min(minx, x)$
10: $\quad miny \leftarrow min(miny, y)$
11: $\quad density \leftarrow \frac{n}{(maxx-minx)(maxy-miny)}$
12: **end while**

We now formalize the above discussion and first point out that both circles and rectangles are worst-case unbound under-approximators of density. The choice of approximation is dependent therefor on the nature of points expected. We further show that the rectangular approximation can be computed on a stream in linear time and bounded memory, as we require in this setting. Finally, we discuss other more general approaches.

Definition 4. *The density of a set of points is the number of points divided by the volume of the convex hull of the points*

Statement 1. *The estimation of point density using the number of points divided by the volume of the smallest (a) sphere or (b) axis-aligned rectangle around the points is an unbound under-estimation.*

Proof. (part a) Consider the points $(0,0),(a,0),(0,\frac{4}{a}),(a,\frac{4}{a})$ for some $a \in R$. These 4 points form a rectangle with an area of 4 and therefor have a density of 1. The smallest circle around the points has a circumference of at least a (since both the points $(0,0)$ and $(0,a)$ are inside the circle), and therefor an area of at least $\pi\frac{a^2}{4}$. The density estimate using the circle is now at most:

$$\frac{4}{\pi\frac{a^2}{4}} = \frac{1}{\pi a^2} \xrightarrow[a\to\infty]{} 0$$

meaning the approximation is arbitrarily bad.

(part b) Consider the points $(\epsilon,0),(0,\epsilon),(1,1-\epsilon),(1-\epsilon,1)$ for some $\epsilon \in R$. These 4 points are enclosed in an axis-aligned square of area 1, while forming themselves a rectangle of area:

$$1-(1-\epsilon)^2-\epsilon^2 \xrightarrow[\epsilon\to 0]{} 0$$

□

Statement 2. *The rectangular estimation of density can be computed on a stream with $O(n)$ runtime and $O(1)$ memory.*

Proof. The computation proceeds by considering each point in the stream at a time, and updating the maximal and minimal coordinate encountered on each axis, the total count of points, and the density (See Algorithm 2). □

A more general approach is to directly compute the convex hull of the points in the segment, and thus derive the density as the ratio of the number of points and the area of the convex hull. Exact convex hull computation is achieved in $O(nlogn)$ (even in the online setting [22]), although linear time approximations exist [14].

The downside of this approach is that these methods use $O(n)$ memory. For segments that are expected not to be too long, this may still be feasible. In such cases we are able to compute the density in a more precises manner, by adapting algorithm 1 to use the convex hull computation rather than the circle approximation.

An alternative approach would be to define the density with respect to a cover of the points with a given number of balls, rather than the full convex hull. This area can be approximated using streaming k-means [1].

C. The online algorithm

The method as described in Algorithm 1 is applied to the entire trajectory (the algorithm is assumed to receive the entire trajectory as input). However, since the algorithm iterates a single time over the points, and keeps only a constant-size summary of the current segment at each point, the adaptation necessary for the application to a stream from many concurrent trajectories is straightforward.

In the streaming version of the algorithm there needn't be an initialization phase, other than constructing an empty list of known trajectories. The points arriving are considered each at a time, together with a unique ID of the trajectory object they belong to. If the ID is not in the list of known trajectories then it is added, and a new centroid, point counter, and radius are initialized for the new trajectory (as in lines 4-6 of algorithm 1). Then, the update step (rows 8-17) is conducted using the three stored values belonging to the current trajectory ID. The exceptions are rows 13-14 where the cutoff and centroid are stored to a disk rather then maintained by the process.

As a result, the amount of memory used by the process is constant (and extremely small) per trajectory, and linear in the number of trajectories being processed concurrently. This property makes it feasible to construct a high throughput mechanism when a very large number of trajectories are processed either sequentially or in parallel with shared memory, and then saved both in the raw and segmented formats to separate storage components. This idea is further developed in section IV-B below, in the context of data retrieval.

D. Selecting Hyper-parameters

The proposed algorithm uses two hyper-parameters (namely the minimal radius of a segment, and minimal density of the large local segments). The choice of the radius parameter effectively controls the granularity of the locomotive parts of trajectory which will be kept in the segmented representation (see Fig. 1). The selection of this parameter is a trade-off between the size and granularity of the result (the larger the radius, the fewer points we keep, but the less precise the information retained).

In some cases, it may be beneficial to apply the algorithm in parallel with more than one radius parameter, thus constructing more than one segmented representation at different levels of granularity. Alternatively, and more interestingly, it is possible to iteratively apply the method on the output of itself, thus effectively constructing a hierarchy of segmentations. This is especially applicable when movement has similar structure regardless of scale [24].

The density parameter is chosen according the the minimal density expected in local segments, and requires some trial and error. When the density parameter is too high, dense local regions will be split into more than one segment, thus increasing the overall size of the output.

In order to compute the total size of the segmentation, it is enough to consider a consecutive locomotive and local region (since they appear intermittently). Consider a length l locomotive part of a trajectory consisting of n_1 points, followed by a local region consisting of n_2 points. Using a radius r segmentation we will keep at most $\frac{l}{r}$ point out of the locomotive part, and a single point for the local part. In total the ratio between the size of the segmentation and the raw data is $\frac{\frac{d}{r}+1}{n_1+n_2}$, which can be made small by choosing a large r, as discussed above.

IV. Applications

A. Visualization

Simultaneous visualization of multiple trajectories is especially challenging because of the large amount of data to be displayed all at once. Even when the actual points don't cover the entire display, visual clatter renders individual trajectories indiscernible.

Several approaches exist for overcoming this problem. In essence, all methods approach the inherent difficulty by presenting a representation of the information [4], and differ from each other in the aspect of the data that is to be maintained.

Trajectory stacking approaches [28], [27], [3] are used when all (or most) trajectories in a set are expected to approximately follow a common path. This is a natural state of affairs when dealing with objects moving along a pre-defined path. Such a path could be a road, shipping line, or convenient path for pedestrians. Using the stacking approach, planar trajectories are presented at varying heights on a virtual axis perpendicular to the plane, thus forming a wall-like shape that enables to see the general shape of the common trajectory as well as deviations from it.

There are two main limitations of the stacking approach. First, directly comparing two trajectories presented at different heights is not visually easy (in order to do this one would have to "imagine" the projection of both onto the plane simultaneously). In fact, to determine the exact position of a trajectory (corresponding to a single projection) is not always easy.

The second limitation is that while appropriate for visualization of trajectories mostly following a common route, the stacking approach is completely useless for a set of general trajectories in a plane.

Another common visualization approach is the space-use density map (see for example figure 3). This approach assumes that the pertinent information in a large set of trajectories is the density in space of points pooled over all the set. The result is a visualization as a heatmap presenting the density in a grid.

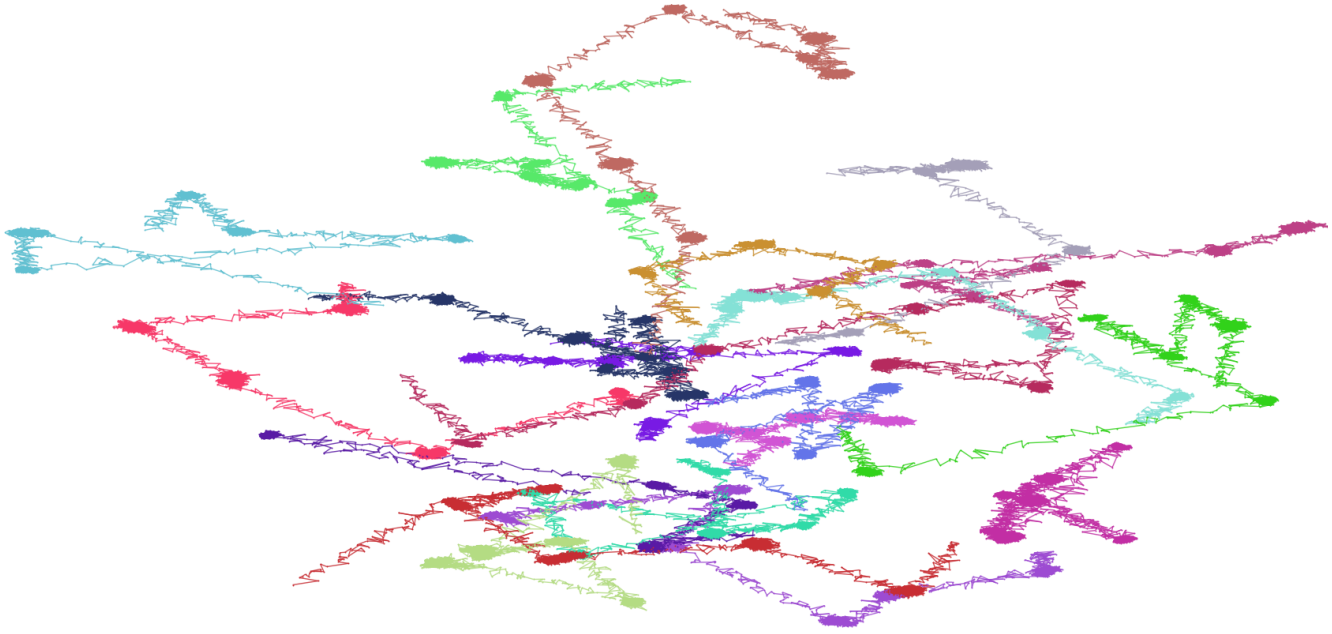
The density heatmap approach is mostly useful when single trajectories are of no interest. Clearly, the heatmap does not preserve any information regarding particular trajectories, and thus comparing trajectories is also not possible.

The assumption made by the visualization method presented here is regarding the nature of the trajectories under consideration, rather than the information needed by the viewer. Namely, we assume that trajectories have the property and consistency of intermittent locomotive and local periods. The locomotive periods are characterized by a directional movement forming a path, while the local periods present a dense use of space, often for extended periods of time. Measurement noise, and small-scale local movement or the order of the noise level turn these periods in to a perceived "cloud" when presenting the full trajectory.

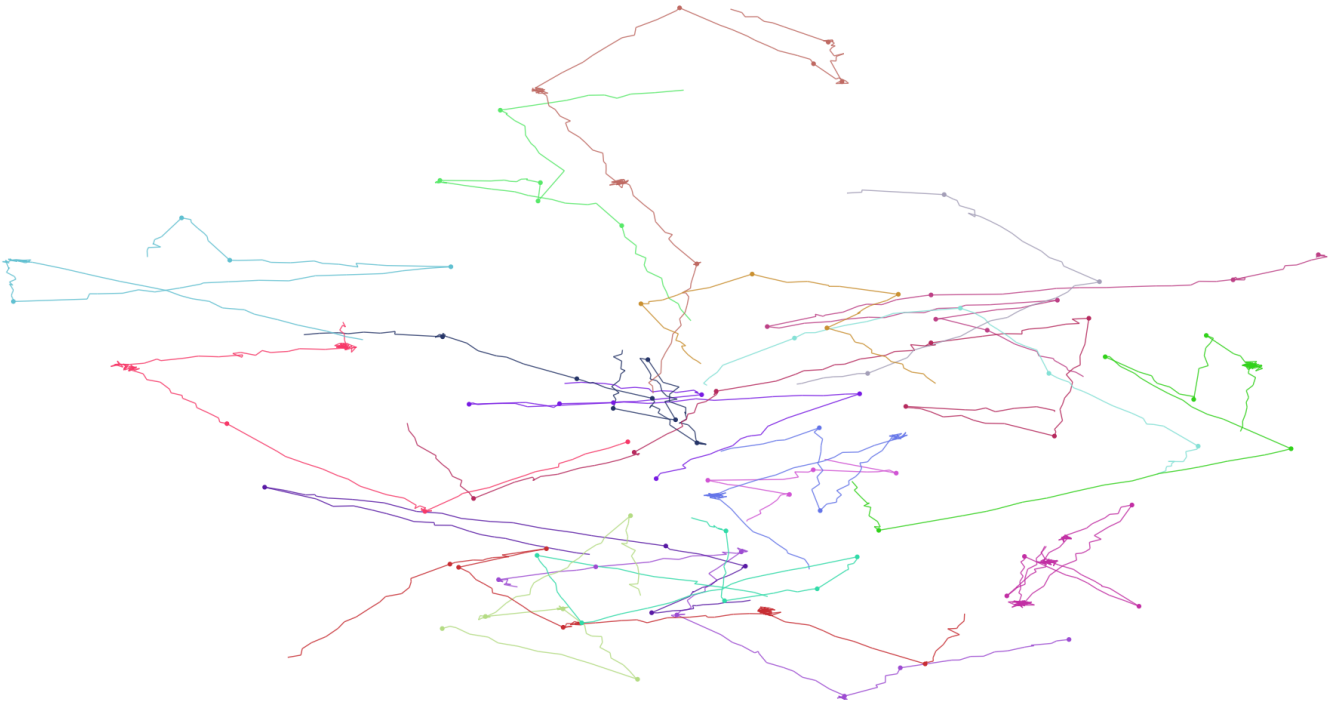
The proposed method then represents such a trajectory by a decimation in space of the locomotive periods, and a single point for local periods (see algorithm 1). This representation preserves the entire pertinent information in the trajectory, since locomotive paths are maintained (in the chosen spatial resolution), and local periods are naturally replaced by a single point representing them, without loosing much information.

The extreme reduction in volume of points to be presented allows simultaneous visualization of many trajectories, while maintaining the ability to resolve single trajectories and compare nearby ones.

Figure 2 presents 20 synthetic trajectories, constructed to have a varying number of locomotive and local segments, and some overlap between them. When presenting the entire



(a) Full data; 20 trajectories



(b) Segmented data; 20 trajectories

Figure 2: A set of 20 trajectories with a varying number of locomotive and local periods, presented as raw data (a) or the segmented representation (b).

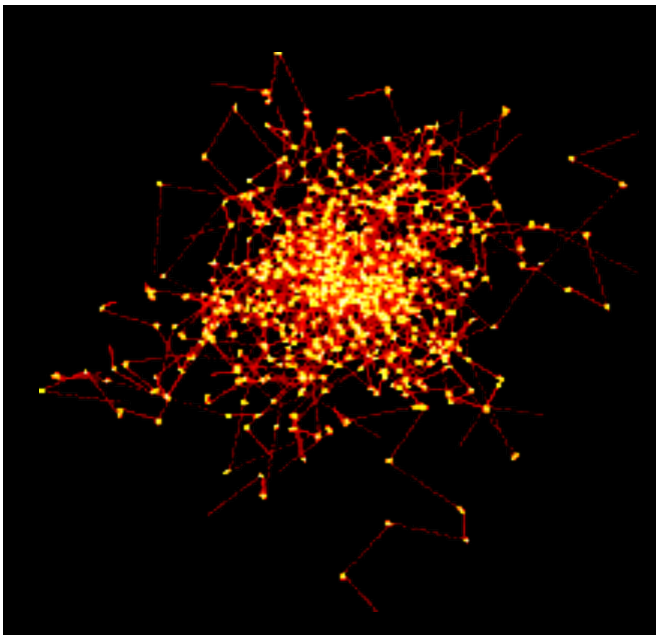


Figure 3: A heatmap representation of 100 trajectories. Areas with intense local activity show up while other parts of the trajectory remain invisible. Individual trajectories are not resolvable.

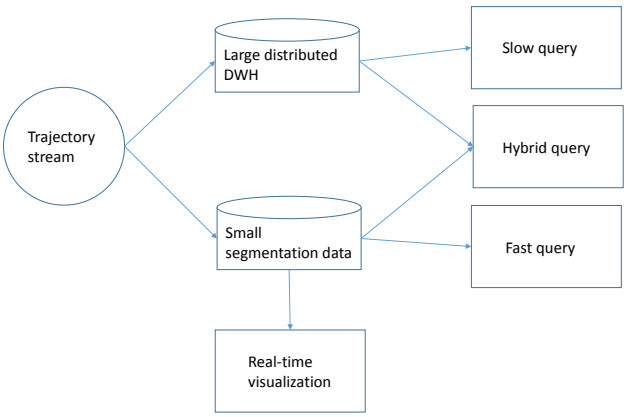


Figure 4: The proposed architecture. The trajectory data management system feeds the input stream into a large distributed Data Warehouse (DWH) and also into the small data segmentation representation. Three types of queries are then defined on the bases of the data source they use. Real-time visualization is fed from the small segmentation data.

raw data (top panel), single trajectories are not easy to trace and compare to each other. Much of the pertinent information is lost by trajectories occluding each-other, and the general clutter. The segmented representation of the same trajectories (bottom panel) does however expose all the aforementioned information in an intuitive and effortless way.

In a dynamic setting where an analyst is able to zoom into interesting parts of a trajectory dataset, the segmented representation allows a very large number of trajectories to be examined and compared, in a way not possible with other types of visualization. In the next section we describe a more general storage and analytics mechanism, in which such a visualization capability will serve as one component.

B. Fast Trajectory Query

With the growing volume of trajectory data collected and stored, the need for new methods of exploring and analyzing such data becomes eminent [8]. Indeed, there has been some recent interest in querying abilities over trajectory data, stemming from multiple domains, as part of the emerging field of trajectory data mining [31], [34], [21].

In the field of video analysis, several approaches have been presented for the task of querying using semantics of object trajectories in the video [10], [6], [30]. Trajectories embedded in Geographic Information Systems (GIS) have also been addressed, with query methods that take into account a trajectory and its background geographical information [8].

So far, none of the proposed methods are able to deal in real-time with truly massive amounts of trajectory data, necessitating large parallel systems. The reason for this is that the size of the index is approximately the size of the raw data. Searching for similarities and patterns over big trajectory data is particularly challenging, since the relevant semantics often stem from large pieces of each single trajectory, which may often be naturally distributed over multiple machines. For this reason, there must be a global index to direct the search.

The method we propose here is based on the use of compact representation of the segmented trajectories (algorithm 1) as an index into the full trajectory data, allowing many useful and interesting queries over this small representation. Thus, using segmentation as an index, we are able to convert a hard big-data problem into a tractable small-medium data computation.

Furthermore, this method allows to preform many trajectory oriented queries without considering actual locations. Using only the segmentation index, without keeping the actual data, may be useful in cases where some computations are necessary, but privacy considerations do not allow storage of full trajectory, or exact location information. (For instance, one could find entities with similar movement patters, or frequent encounters, with only the aggregated segmentations, without any actual exact locations. Another privacy maintaining application would be to determine the context of a user, without storing any locations whatsoever. To this end it would be necessary to assert whether the user is at home, work etc. The details of the method are outside the scope of the current paper.)

The sort of queries for which we can directly use the index comprise most of the typical trajectory based queries:

1) Range queries: this sort of query retrieves the identity of entities which were in a certain spatial area, possibly at

a certain time. Withing the granularity of the segmentation, this can be preformed without considering the full data.

2) KNN queries: this common sort of query retrieves the top K trajectories similar in a given metric to an example trajectory. There are many commonly used options for trajectory similarity metrics [31], most of which can be used on the segmented representation.
3) pairwise queries: many queries over trajectories consider two (or a small set of) trajectories, rather than the entire dataset. Such queries include finding meeting or intersecting points; the point in time where two trajectories come closest to each-other; detecting periods where a small set of trajectories are moving together. All such queries may be computed directly using the segmented representation (where the spatial scale under consideration is no finer than the segmentation radius, otherwise a two-stage method is needed, as is described below).

Another sort of query are the hybrid queries. These queries can be preformed by first narrowing down the search using the segmentation representation, and then diving into the full data only where necessary. For instance, in order to find all trajectories that meet (or intersect) exactly (within a tolerance such smaller than the granularity of the segmentation) with a given trajectory, it is possible to first find all trajectories that pass near the given trajectory, using the course segmentation data. Next, the full data is used only in times and places where such proximity was detected, in order to check whether or nor the trajectories actually meet. When access to a specific trajectory at a specific time in the full data is fast (as is typically the case), this combined method will allow this real-time query in a manner that otherwise may not possible.

A data management system supporting such queries consists of several building blocks (see Figure 4 for a schematic of a possible system). First, the input data is received via a stream of locations along a trajectory, for many (or a huge number of) trajectories simultaneously. These locations are processed in parallel into both a main large Data Warehouse (DWH) and a much smaller database of segmentation data, which is derived from the stream (algorithm 2).

A real-time component then allows visualization and fast querying on the small segmentation data. An additional retrieval layer preforms queries either on the full data (slow queries) or hybrid queries where the segmentation data is able to drastically narrow the search in the full data. We defer the full characterization and testing of the system to future research.

V. Conclusion

In this paper we describe an online algorithm for segmentation and summary of trajectory data. We then demonstrate the application of the method to visualization of parts of large trajectory datasets, and discuss the potential use as an index into big trajectory data allowing fast queries.

Visualization of large trajectory datasets is particularly challenging because of the need to represent simultaneously a tremendous number of data-points, in a way the allows single trajectories to be resolved on the one hand, and the big picture of all movement to be discernible of the other hand. Previous methods used either stacked representations which are suitable for the description of many trajectories on the same approximate route (such as a shipping line), or space-use-density based visualizations which are appropriate for describing aggregate behavior over many trajectories.

The novelty in the proposed approach is in the generation of trajectory summaries, allowing visualization of all the pertinent information across many trajectories at once. The resulting visualization keeps the identity of single trajectories clear and comparable, while at the same time presenting the entire movement in the dataset in a way that is accessible at first glance.

The second application we discuss for the segmentation algorithm is in the field of data retrieval, for indexing of large trajectory data for fast queries. Thus far, the literature contains several trajectory specific indexing and querying frameworks with varying degrees of complexity. However, the size of the index produced is similar to the size of the raw data. The method we propose here utilizes the segmented representation of the trajectory data as a small index allowing fast queries that are otherwise prohibitively slow over large datasets.

Together with the proposed visualization, we then describe an end-to-end system for ingestion, retention, and real-time visualization and analysis over big trajectory data. Such a system will allow an analyst to utilize both real-time and historical data. Future research will concentrate on building such a system with direct applications to research in Movement Ecology, traffic, and pedestrian movement.

We note that in addition to the applications of the segmentation method for visualization and data retrieval, the method may be of interest for the sake of the segmentation itself. In many trajectory oriented computations, this is a first step which serves to reduce noise and data volume. The proposed online method is well suited for this purpose, especially when the volume of the data prohibits retention of large parts at the same time, and thus a method must be applied on the stream.

For example, downstream processing may require differential treatment of locomotive and local segments of trajectories. This is desirably the case for smoothing, where when used on the entire raw data, local segments will tend to get smudged along the locomotive segments leading to and from them. This adverse effect if overcome when preforming the smoothing only on the locomotive segments.

Another target for future research is the combination of data-structure based trajectory indexing with the current approach. Since the segmentation is in itself a (albeit much smaller) trajectory, methods for efficient representation of complete trajectories which are not suitable for the large-scale data at hand, can be applied to the small segmentation trajectory in order to further improve the approach. In this sense, using the segmentation as an index may serve as a a

general framework which can be used in synergy with most existing methods.

VI. Acknowledgments

We would like to thank Michal Moskovitz for the helpful discussions and remarks regarding computational geometry, and Tom Hope and e-Thai Lieder for their illuminating feedback and suggestions regarding the manuscript and methods described in it.

References

[1] Nir Ailon, Ragesh Jaiswal, and Claire Monteleoni. Streaming k-means approximation. In *Advances in Neural Information Processing Systems*, pages 10–18, 2009.

[2] Moustafa Alzantot and Moustafa Youssef. Uptime: Ubiquitous pedestrian tracking using mobile phones. In *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 3204–3209. IEEE, 2012.

[3] Natalia Andrienko and Gennady Andrienko. Visual analytics of movement: An overview of methods, tools and procedures. *Information Visualization*, page 1473871612457601, 2012.

[4] Benjamin Bach, Pierre Dragicevic, Daniel Archambault, Christophe Hurter, and Sheelagh Carpendale. A review of temporal data visualizations based on space-time cube operations. In *Eurographics conference on visualization*, 2014.

[5] Frédéric Barraquand and Simon Benhamou. Animal movements in heterogeneous landscapes: identifying profitable places and homogeneous movement bouts. *Ecology*, 89(12):3336–3348, 2008.

[6] Faisal I Bashir, Ashfaq A Khokhar, and Dan Schonfeld. Segmented trajectory based indexing and retrieval of video data. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 2, pages II–623. IEEE, 2003.

[7] Allert I Bijleveld, Robert B MacCurdy, Ying-Chi Chan, Emma Penning, Rich M Gabrielson, John Cluderay, Eric L Spaulding, Anne Dekinga, Sander Holthuijsen, Job ten Horn, et al. Understanding spatial distributions: negative density-dependence in prey causes predators to trade-off prey quantity with quality. In *Proc. R. Soc. B*, volume 283, page 20151557. The Royal Society, 2016.

[8] Vania Bogorny, Bart Kuijpers, and Luis Otavio Alvares. St-dmql: a semantic trajectory data mining query language. *International Journal of Geographical Information Science*, 23(10):1245–1276, 2009.

[9] Maike Buchin, Anne Driemel, Marc van Kreveld, and Vera Sacristán. An algorithmic framework for segmenting trajectories based on spatio-temporal criteria. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 202–211. ACM, 2010.

[10] T Catarci, ME Donderler, E Saykol, O Ulusoy, and U Gudukbay. Bilvideo: A video database management system. *IEEE MultiMedia*, 10(1):66–70, 2003.

[11] Per Fauchald and Torkild Tveraa. Using first-passage time in the analysis of area-restricted search and habitat selection. *Ecology*, 84(2):282–288, 2003.

[12] Eric Foxlin. Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Computer graphics and applications*, 25(6):38–46, 2005.

[13] Roi Harel, Nir Horvitz, and Ran Nathan. Adult vultures outperform juveniles in challenging thermal soaring conditions. *Scientific reports*, 6, 2016.

[14] M Zahid Hossain and M Ashraful Amin. On constructing approximate convex hull. *American Journal of Computational Mathematics*, 3(01):11, 2013.

[15] Yong-Shik Kim and Keum-Shik Hong. An imm algorithm for tracking maneuvering vehicles in an adaptive cruise control environment. *International Journal of Control Automation and Systems*, 2:310–318, 2004.

[16] Lars-Henrik Kullingsjö and Sten Karlsson. The swedish car movement data project. In *Proceedings to EEVC Brussels, Belgium, November 19-22, 2012*, 2012.

[17] Marc Lavielle. Using penalized contrasts for the change-point problem. *Signal processing*, 85(8):1501–1510, 2005.

[18] Luis A Leiva and Enrique Vidal. Warped k-means: An algorithm to cluster sequentially-distributed data. *Information Sciences*, 237:196–210, 2013.

[19] Quannan Li, Yu Zheng, Xing Xie, Yukun Chen, Wenyu Liu, and Wei-Ying Ma. Mining user similarity based on location history. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, page 34. ACM, 2008.

[20] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[21] Dieter Pfoser, Christian S Jensen, Yannis Theodoridis, et al. Novel approaches to the indexing of moving object trajectories. In *Proceedings of VLDB*, pages 395–406, 2000.

[22] Franco P Preparata and Michael Ian Shamos. Convex hulls: Basic algorithms. In *Computational geometry*, pages 95–149. Springer, 1985.

[23] Yehezkel S Resheff, Shay Rotics, Roi Harel, Orr Spiegel, and Ran Nathan. Accelerater: a web application for supervised learning of behavioral modes from acceleration measurements. *Movement ecology*, 2(1):1, 2014.

[24] Injong Rhee, Minsu Shin, Seongik Hong, Kyunghan Lee, Seong Joon Kim, and Song Chong. On the levy-walk nature of human mobility. *IEEE/ACM transactions on networking (TON)*, 19(3):630–643, 2011.

[25] Shay Rotics, Michael Kaatz, Yehezkel S Resheff, Sondra Feldman Turjeman, Damaris Zurell, Nir Sapir, Ute Eggers, Andrea Flack, Wolfgang Fiedler, Florian Jeltsch, et al. The challenges of the first migration: movement and behaviour of juvenile vs. adult white storks with insights regarding juvenile mortality. *Journal of Animal Ecology*, 85(4):938–947, 2016.

[26] Thomas Statheros, Gareth Howells, and Klaus McDonald Maier. Autonomous ship collision avoidance navigation concepts, technologies and techniques. *Journal of navigation*, 61(01):129–142, 2008.

[27] Guo-Dao Sun, Ying-Cai Wu, Rong-Hua Liang, and Shi-Xia Liu. A survey of visual analytics techniques and applications: State-of-the-art research and future challenges. *Journal of Computer Science and Technology*, 28(5):852–867, 2013.

[28] Christian Tominski, Heidrun Schumann, Gennady Andrienko, and Natalia Andrienko. Stacking-based visualization of trajectory attribute data. *IEEE Transactions on visualization and Computer Graphics*, 18(12):2565–2574, 2012.

[29] Adi Weller Weiser, Yotam Orchan, Ran Nathan, Motti Charter, Anthony J Weiss, and Sivan Toledo. Characterizing the accuracy of a self-synchronized reverse-gps wildlife localization system. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 1–12. IEEE, 2016.

[30] Yutaka Yanagisawa, Jun-ichi Akahani, and Tetsuji Satoh. Shape-based similarity query for trajectory of mobile objects. In *International Conference on Mobile Data Management*, pages 63–77. Springer, 2003.

[31] Yu Zheng. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(3):29, 2015.

[32] Yu Zheng, Xing Xie, and Wei-Ying Ma. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull.*, 33(2):32–39, 2010.

[33] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th international conference on World wide web*, pages 791–800. ACM, 2009.

[34] Xiaofang Zhou, Kai Zheng, Hoyoung Jueng, Jiajie Xu, and Shazia Sadiq. Making sense of spatial trajectories. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 671–672. ACM, 2015.

Chapter 4

General Methods

4.1 Optimized Linear Imputation

This paper [1] was presented at the International Conference on Pattern Recognition Applications and Methods, which was held during February 24-26, in Porto, Portugal. The **extended version** presented here was accepted to the Springer series *Lecture Notes on Computer Science (LNCS)*.

References

[1] Yehezkel S. Resheff and Daphna Weinshall. "Optimized Linear Imputation". In: *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods, ICPRAM 2017, Porto, Portugal, February 24-26, 2017*. 2017, pp. 17–25. DOI: 10.5220/0006092900170025. URL: https://doi.org/10.5220/0006092900170025.

Optimal Linear Imputation with a Convergence Guarantee*

Yehezkel S. Resheff[1,2] and Daphna Weinshall[1]

[1] School of Computer Science and Engineering, The Hebrew University of Jerusalem
{heziresheff, daphna}@cs.huji.ac.il
[2] Edmond and Lily Safra Center for Brain Sciences, The Hebrew University of Jerusalem

Abstract. It is a common occurrence in the field of data science that real-world datasets, especially when they are high dimensional, contain missing entries. Since most machine learning, data analysis, and statistical methods are not able to handle missing values gracefully, these must be filled in prior to the application of these methods. It is no surprise therefore that there has been a long standing interest in methods for imputation of missing values. One recent, popular, and effective approach, the IRMI stepwise regression imputation method, models each feature as a linear combination of all other features. A linear regression model is then computed for each real-valued feature on the basis of all other features in the dataset, and subsequent predictions are used as imputation values. However, the proposed iterative formulation lacks a convergence guarantee. Here we propose a closely related method, stated as a single optimization problem, and a block coordinate-descent solution which is guaranteed to converge to a local minimum. Experiment results on both synthetic and benchmark datasets are comparable to the results of the IRMI method whenever it converges. However, while in the set of experiments described here IRMI often diverges, the performance of our methods is shown to be markedly superior in comparison to other methods.

1 Introduction

The typical modus operandi in the field of data science evolves a wrangling stage where either the raw data or features computed on the basis of the raw data are organized in the form of a table. Indeed, the vast majority of data analysis, machine learning, and statistical methods rely on complete data [9], mostly structured in a tabular or relational form.

Since in real-world datasets more often than not some of the entries are missing, imputation is an important part of data preprocessing and cleansing [13,24]. Naturally, this topic has has been of long-standing interest in many fields associated with data analysis.

As is often the case, simple and elegant linear methods with interpretable results have gained a special place in the heart of the field, and are used in practice whenever applicable. More advanced methods (see Section 3 for a brief review) are typically reserved for special cases.

* Invited extension of [29] – presented at the 6th International Conference on Pattern Recognition Applications and Methods (ICPRAM2017).

The trivial option for many application domains is to discard complete records in which there are any missing values. Clearly this method is sub-optimal, for several reasons: first and foremost, when missing values are not missing at random [20,11], discarding these records may bias the resulting analysis [21] (consider for instance a classification task where many of the examples from one of the classes have some missing features. Discarding these entire examples will lead to a very unbalanced problem, with a potentially detrimental effect on the final results).

Other limitations include the needless loss of information when discarding entire records which may actually include valuable information for the down-stream task. Furthermore, when dealing with datasets with either a small number of records or a large number of features, omitting complete records when any feature value is missing may result in discarding a large proportion of the records (or in the extreme case – all of them), and insufficient data for the required analysis.

Early methods for data imputation include procedures for replacing each missing value by the mean or median of the feature value across records [7,5]. While per-feature summary statistics may indeed provide a "good guess" when there is no other information present, this is often not the case we are dealing with. Namely, for each missing feature value there are other non-missing values in the same record. It is likely therefore (or in fact we assume) that other features contain information regarding the value of the missing feature, and imputation should therefore take into account known feature values in the same record. This is done by all subsequent methods.

Multiple imputation (see [31] for a detailed review) imputes several sets of missing values, drawn from the posterior distribution of the missing values under a given model, given the data. Subsequent processing is then to be performed on each version of the imputed data, and the resulting multiple sets of model parameters are finally combined to produce a single result.

While extremely useful in traditional statistical analysis and heavily utilized in analysis of public survey data, this may not be feasible in a machine learning and modern statistical setting. First, the run-time cost of performing the analysis on several copies of the full-data may be prohibitive. Second, being a model-based approach it depends heavily on the type and nature of the data, and can't be used as an out-of-the-box preprocessing step. More importantly though, while traditional model parameters may (for the most part) be combined between versions of the imputed data (regression coefficients for instance), many modern machine learning methods do not produce a representation that is straightforward to combine (consider the parameters of an Artificial Neural Network or a Random Forest for example [3]).

In [25], a method for imputation on the basis of a sequence of regression models is introduced. The method, popularized under the acronym MICE [1,36], uses a non-empty set of complete features (i.e. features with values which are known in all the records) as its base, and iteratively imputes one feature at a time on the basis of the completed features up to that point.

[3] In this case it would be perhaps more natural to train the model using data pooled over the various copies of the completed data rather than train separate models and average the resulting parameters and structure. This is indeed done artificially in methods such as denoinsing neural nets [37], and has been known to be useful for data imputation [6].

Since each step of this method produces a single complete feature, the number of iterations needed to impute the entire table is exactly the number of features that have a missing value in at least one record. The drawbacks of this method are twofold. First, there must be at least one complete feature to be used as the base (however if there is no complete feature then the feature with fewest missing values may be imputed using the a feature-wise summary statistic). More importantly though, the values imputed at the $i-th$ step can only use a regression model that includes the features which were originally full or those imputed in the first $i-1$ steps. Ideally, the regression model for each feature should be able to use all other feature values, thus not discarding any available information.

The IRMI method [34] goes one step further by again building a sequence of regression models for each feature, this time utilizing the values in all other features. This iterative method initially uses a simple imputation method such as median imputation, to produce temporary imputation values. In each subsequent iteration it computes for each feature the linear regression model based on all other feature values, and then re-imputes the missing values based on these regression models. The process is terminated upon convergence or after a per-determined number of iterations (Algorithm 1). The authors state that although they do not have a proof of convergence, experiments show fast convergence in most cases (However, in our experiments the method often failed to converge. See Section 4.2).

In this paper we present a method similar in spirit to IRMI, formulated as a single optimization problem, and provide an optimization procedure with a guarantee of convergence. This method of Optimized Linear Imputation (OLI) is related in spirit to IRMI in that it performs a linear regression imputation for the missing values of each feature, on the basis of all other features. Our method is defined by a single optimization objective which we then solve using a block coordinate-descent method. Thus our method is guaranteed to converge, which is its most important advantage over IRMI. The OLI method is then compared to IRMI as well as other methods using both synthetic, benchmark, and real world datasets.

The rest of the paper is organized as follows: In Section 2 we present the novel method of Optimized Linear Imputation (OLI), and a method of optimization which guarantees convergence. We discuss and analyze the relationship to previous methods, and further show that our algorithm may be easily extended to use any form of regularized linear regression.

In Section 4 we compare the OLI method to the IRMI, MICE and Median Imputation (MI) methods. Using the same simulation studies as appear in the original IRMI paper, we show that the results of OLI are rather similar to the results of IRMI. With benchmark and real-world datasets we show that our method usually outperforms the alternatives MI and MICE in accuracy, while providing comparable results to IRMI. However, IRMI did not converge in many of these experiments, while our method always provided good results.

Algorithm 1 the IRMI method for imputation of real-valued features (see [34] for more details)

input:

- X - data matrix of size $N \times (d+1)$ containing N samples and d features. Zeros in locations of missing values.
- m - missing data mask
- max_iter - maximal number of iterations

output:

- Imputation values

1: $\tilde{X} := median\_impute(X)$ {assigns each missing value the median of its column}
2: **while** not converged and under max_iter iterations **do**
3: **for** i := 1...d **do**
4: regression = linear_regression($\tilde{X}_{-i}[!m_i], \tilde{X}_i[!m_i]$)
5: $\tilde{X}_i[m_i]$ = regression.predict($\tilde{X}_{-i}[m_i]$)
6: **end for**
7: **end while**
8: **return** $\tilde{X} - X$

2 The Optimized Linear Imputation method

2.1 Notation

We start by listing the notation used throughout the paper.

N	Number of samples
d	Number of features
$x_{i,j}$	The value of the $j-th$ feature in the $i-th$ sample
$m_{i,j}$	Missing value indicators: $m_{i,j} = \begin{cases} 1 & x_{i,j}\ is\ missing \\ 0 & otherwise \end{cases}$
m_i	Indicator vector of missing values for for the $i-th$ feature

The following notation is used in the algorithms' pseudo-code:

$A[m]$	The rows of a matrix (or column vector) A where the boolean mask vector m is $True$
$A[!m]$	The rows of a matrix (or column vector) A where the boolean mask vector m is $False$
linear_regression(X, y)	A linear regression from the columns of the matrix X to the target vector y, having the following fields: .parameters: parameters of the fitted model. .predict(X): the target column y as predicted by the fitted model.

2.2 Optimization problem

We formulate the linear imputation objective as a single optimization problem. We start by constructing a design matrix:

$$X = \begin{bmatrix} & & 1 \\ & [x_{i,j}(1-m_{i,j})] & \vdots \\ & & 1 \end{bmatrix} \tag{1}$$

where the constant-1 rightmost column is a convenience used for the intercept terms in the subsequent regression models. Multiplying the data values $x_{i,j}$ by $(1-m_{i,j})$ simply sets all missing values to zero, keeping non-missing values as they are.

Our approach essentially aims to find consistent missing value imputations and regression coefficients at once, as a single optimization problem. By consistent we mean that (a) the imputation values are the values obtained by the regression formulas, and (b) the regression coefficients are the values that would be computed after the imputations if the another iteration of the algorithm was to be applied (i.e. a stationary point of the algorithm). We propose the following optimization formulation:

$$\begin{cases} \min\limits_{A,M} & ||(X+M)A-(X+M)||_F^2 \\ s.t. & m_{i,j}=0 \Rightarrow M_{i,j}=0 \\ & M_{i,d+1}=0 \; \forall i \\ & A_{i,i}=0 \quad i=1...d \\ & A_{i,d+1}=\delta_{i,d+1} \quad \forall i \end{cases} \tag{2}$$

where $||.||_F$ denotes the Frobenius norm.

Intuitively, the objective that we minimize measures the square error of reconstruction of the imputed data $(X+M)$, where each feature (column) is approximated by a linear combination of all other features plus a constant (that is, linear regression of the remaining imputed data). The imputation process by which M is defined is guaranteed to leave the non-missing values in X intact, by the first and second constraints which make sure that only missing entries in X have a corresponding non-zero value in M. Therefore:

$$(X+M) = \begin{cases} M & for\ missing\ values \\ X & for\ non\ missing\ values \end{cases}$$

The regression for each feature is further constrained to use only **other** features, by setting the diagonal values of A to zero (the third constraint). The forth constraint makes sure that the constant-1 rightmost column of the design matrix is copied as-is and therefore does not impact the objective.

We note that all the constraints set variables to constant values, and therefore this can be seen as an unconstrained optimization problem on the remaining set of variables. This set includes the non-diagonal elements of A and the elements of M corresponding to missing values in X. We further note that this is not a convex problem in A, M since it contains the MA factor. In the next section we show a solution to this problem that

is guaranteed to converge to a local minimum. This convergence guarantee is the major advantage of the proposed formulation over the prior IRMI method.

2.3 Block coordinate descent solution

We now develop a coordinate descent solution for the proposed optimization problem. Coordinate descent (and more specifically alternating least squares; see for example [33,18,2,17]) algorithms are extremely common in machine learning and statistics, and while don't guarantee convergence to a global optimum (but only to a local optimum), they often preform well in practice.

As stated above, our problem is an unconstrained optimization problem over the following set of variables:

$$\{A_{i,j}|i,j=1,..,d;i\neq j\}\cup\{M_{i,j}|m_{i,j}=1\}$$

Algorithm 2 Optimized Linear Imputation (OLI)

input:

- X_0 - data matrix of size $N \times d$ containing N samples and d features
- m- missing data mask

output:

- Imputation values

$X := median\_impute(X_0)$
$M := zeros(N, d)$
$A := zeros(d, d)$
while not converged **do**
  for $i := 1...d$ **do**
    $\beta := linear\_regression(X_{-i}, X_i).parameters$
    $A_i := [\beta_1, ..., \beta_{i-1}, 0, \beta_i, ..., \beta_d]^T$
  end for‘
  while not converged **do**
    $M := M - \alpha[(X + M)A - (X + M)](A - I)^T$
    $M[!m] := 0$
  end while
  $X := X + M$
end while
return M

Keeping this in mind, we re-write the objective function in a form that will facilitate subsequent derivation:

$$L(A, M) = ||(X + M)A - (X + M)||_F^2 \tag{3}$$

$$= \sum_{i=1}^{d} ||(X + M)_{-i}\beta_i - (X + M)_i||_F^2 \tag{4}$$

where C_{-i} denotes the matrix C without its $i - th$ column, C_i the $i - th$ column, and β_i the $i - th$ column of A without the $i - th$ element (recall that the $i - th$ element of the $i - th$ column of A is always zero). The term $(X + M)_{-i}\beta_i$ is therefore a linear combination of all but the $i - th$ column of the matrix $(X + M)$. The sum in (4) is over the first d columns only, since the term added by the rightmost column is zero (see fourth constraint in (2) which enforces the exact copy of the rightmost column).

We now suggest the following coordinate descent algorithm for the minimization of the objective (3) (the method is summarized in Algorithm 2):

1. Fill in missing values using median/mean (or any other) imputation
2. Repeat until convergence:
   (a) Minimize the objective (3) w.r.t. A (compute the columns of the matrix A)
   (b) Minimize the objective (3) w.r.t. M (compute the missing values entries in matrix M)
3. Return M [4]

As we will show shortly, step (a) in the iterative part of the proposed algorithm reduces to calculating the linear regression for each feature on the basis of all other features, essentially the same as the first step in the IRMI algorithm [34] Algorithm 1.

Step (b) can be solved either as a system of linear equations or in itself as an iterative procedure, by gradient descent on (3) w.r.t M using (5).

We now begin by briefly showing that step (a) indeed reduces to linear regression. Taking the derivatives of (4) w.r.t the non-diagonal elements of column i of the matrix A we have:

$$\frac{\partial L}{\partial \beta_i} = 2(X + M)_{-i}^T[(X + M)_{-i}\beta_i - (X + M)_i]$$

Setting the partial derivatives to zero gives:

$$\begin{aligned}(X + M)_{-i}^T[(X + M)_{-i}\beta_i - (X + M)_i]&=0\\ \Rightarrow \beta_i = ((X + M)_{-i}^T(X + M)_{-i})^{-1}(X + M)_{-i}^T(X + M)_i&\end{aligned}$$

which is exactly the linear regression coefficients for the $i - th$ feature from all other (imputed) features, as claimed.

[4] Alternatively, in order to stay close in spirit to the linear IRMI method, we may prefer to use $(X + M)A$ as the imputed data, meaning the imputed values are in fact derived from the all other features using a linear model. Clearly, at the point of convergence of the algorithm the two are identical.

Next, we obtain the derivatives of the objective function w.r.t M:

$$\nabla_M = \frac{\partial L}{\partial M} = 2[(X+M)A - (X+M)](A-I)^T \tag{5}$$

leading to the following gradient descent algorithm for step (b), the minimization of the objective w.r.t M:
Repeat until convergence:

(i) $M := M - \alpha \nabla_M L(A, M)$
(ii) $\forall_{i,j} : \; M_{i,j} = M_{i,j} m_{i,j}$

where α is a predefined step size and the gradient is given by (5). Step (ii) above makes sure that only missing values are assigned imputation values [5].

Our proposed algorithm uses a gradient descent procedure for the minimization of the objective (3) w.r.t M. Alternatively, one could use a closed form solution by directly setting the partial derivative to zero. More specifically, let

$$\frac{\partial L}{\partial M} = 0 \tag{6}$$

Substituting (5) into (6), we get

$$M(A-I)(A-I)^T = -X(A-I)(A-I)^T$$

which we rewrite as:

$$MP = Q \tag{7}$$

with the appropriate matrices P, Q. Now, since only elements of M corresponding to missing values of X are optimization variables, only these elements must be set to zero in the derivative (6), and hence only these elements must obey the equality (7). Thus, we have:

$$(MP)_{i,j} = Q_{i,j} \;\; \forall i,j | m_{i,j} = 1$$

which is a system of $\sum_{i,j} m_{i,j}$ linear equations in $\sum_{i,j} m_{i,j}$ variables.

2.4 Discussion

In order to better understand the difference between the IRMI and OLI methods, we rewrite the IRMI iterative method [34] using the same notation as used for our OLI method. We start by defining an error matrix:

$$E = (X+M)A - (X+M)$$

[5] Note that this is not a projection step. Recall that the optimization problem is only over elements M_{ij} where x_{ij} is a missing value, encoded by $m_{ij} = 1$. The element-wise multiplication of M by m guarantees that all other elements of M are assigned 0. Effectively, the gradient descent procedure does not treat them as independent variables, as required.

In the following, E is the error matrix of the linear regression models on the basis of the imputed data. Unlike our method, however, IRMI considers the error only in the non-missing values of the data, leading to the following objective function:

$$L(M, A) = \sum_{i,j|m_{i,j}=0} E_{i,j}^2$$

In order to minimize this loss function, at each step the IRMI method (Algorithm 1) optimizes over a single column of A (which in effect reduces to fitting a single linear regression model), and then assigns as the missing values in the corresponding column of M the values predicted for it by the regression model.

While this heuristic for choosing M is quite effective, it is **not** a gradient descent step and consequently leads to a process with unknown convergence properties. The main motivation for proposing our method was to fix this undesired property within the same general conceptual framework of linear imputation; namely, propose a method that is similar in spirit, with a convergence guarantee.

Another advantage of the proposed OLI formulation is the ability to easily extend it to any regularized linear regression. This can be done by re-writing the itemized form of the objective (4) as follows:

$$L(A, M) = \sum_i [||(X + M)_{-i}\beta_i - (X + M)_i||_F^2 + \Omega(\beta_i)]$$

where $\Omega(\beta_i)$ is the regularization term.

Now, assuming that the resulting regression problem can be solved (that is, minimizing each of the summands in the new objective with a constant M), and since step (b) of our method remains exactly the same (the derivative w.r.t M does not change as the extra term does not depend on M), we can use the same method to solve this problem as well.

Another possible extension is to use kernelized linear regression. In this case the imputation is preformed in an implicit feature space:

$$L(A, M) = ||\phi((X + M)A) - \phi(X + M)||_F^2$$

This may be useful in cases when the dependencies between the features are not linear (see a further discussion of this case in Section 3).

The method of initialization is another issue deserving further investigation. Since our procedure converges to a local minimum of the objective, it may be advantageous to start the procedure from several random initial points, and choose the best final result. However, since the direct target (missing values) are obviously unknown, we would need an alternative measure of the "goodness" of a result. The missing values are usually assumed to be missing at random, so it would make sense to use the distance between the distributions of known and imputed values (per feature) as a measure of appropriateness of an imputation.

3 Non-linear Imputation Methods

Linear imputation methods are the family of methods which model a missing value as a linear combination of other values in the same record (either only non-missing or both missing and non-missing). Using these methods makes sense when a notion of a *record* exists in the data, in the sense that a set of measurements refers to the same entity in some way. This is often the case in tabular (or relational) data, where each row represents information about a specific entity. In this case, redundancy in the structure of the record often allow linear data imputation.

However, the structure of the data is often such that non-linear relationships exist between columns, and thus non-linear methods are required in order to model and impute missing data.

One of the most utilized non-linear imputation methods uses the method of Expectation Maximization in order to obtain maximum likelihood estimates for the missing data [4]. A major advantage of this approach is the convergence guarantee, and a vast literature regarding statistical properties in various settings and under different assumptions regarding the underlying model.

Deep learning techniques have become overwhelmingly popular in recent years for many machine learning tasks. Indeed, this shift has not skipped the very important task of data imputation.

Stacked Denoising Autoencoders [37] (SDA) is a training method for deep learning models where noise is added to the training examples and fed into the network. The objective is then to recover the original version of the data (prior to having the noise added to it). The main use of the SDA method is for learning representations (see for instance [16,23,40,22]), however in [6] this method is proposed as a means for imputation of traffic data.

In order to use SDA for data imputation, during the training stage a "missing data" mask is randomly selected for each sample, and the corresponding values are zeroed-out or replaced with noise values. The objective function the network is trained with respect to, as in the case of denoising, is the reconstruction error of the output versus the original data. When the trained model is used for imputation, the actual missing data is treated like the mask during training, and the output of the network is used as the imputed data.

The most straightforward version of this method would require a substantial amount of data without any missing elements, since these are used for the training process described above. However, one might use training data which does contain missing elements, and use a loss which takes this into account (essentially by requiring the reconstruction error to be low only for true data).

In image processing, the task of image denoising is to clean up noise in a digital image, which appears either due to noise in the acquisition process (dust, rain, etc.), or as the result of some intentional post-processing (such as overlayed text). Although often treated differently, image denoising is essentially an imputation problem. Here too, denoising autoencoders have been employed successfully [39] to achieve state of the art results.

In the recently proposed learning setting of *Ballpark Learning* [12], an entire column is imputed (or estimated) based on rough group comparisons. In this setting, rather

than having column-wise partial information, upper and lower bounds on the proportions of labels in so-called "bags" are used together with constraints on bag differences to obtain an optimization problem yielding the desired imputations of the target column.

Non-linear imputation methods are potentially superior to linear methods, when the linear structure assumption the latter are based on is good description of the data. However, when applicable, linear methods have the very desirable advantages of simplicity and interpretability, which arguably is what makes them so popular in practice.

In the next section we present an in-depth evaluation of the OLI method proposed in this paper, and compare it to other linear imputation methods. We start of using synthetic data, them move on to some benchmark datasets.

4 Evaluation

In order to evaluate our method, we compared its performance to other imputation methods using various types of data. We used complete datasets (real or synthetic), and randomly eliminated entries in order to simulate the missing data case. To evaluate the success of each imputation method, we used the mean square error (MSE) of the imputed values as a measure of error. MSE is computed as the mean square distance between stored values (the correct values for the simulated missing values) and the imputed ones.

In Section 4.1 we repeat the experimental evaluation from [34] using synthetic data, in order to compare the results of our method to the results of IRMI. In Section 4.2 we compare our method to 3 other methods - IRMI, MI (median imputation), and MICE - using standard benchmark datasets from the UCI repository [19] . In Section 4.3 we augment the comparisons with an addition new real-life dataset of behavioral modes of migrating storks [30].

For some real datasets in the experiments described below we report that the IRMI method did not converge (and therefore did not return any result). This decision was reached when the MSE of the IRMI method rose at least 6 orders of magnitude throughout the allocated 50 iterations, or (when tested with unlimited iterations) when it rose above the maximum valid number in the system of approximately $1e + 308$.

4.1 Synthetic data

The following simulation studies follow [34] and compare OLI to IRMI. All simulations are repeated 20 times with $10,000$ samples. 5% of all values across records are selected at random and marked as missing. Values are stored for comparison with imputed values. Simulation data is multivariate normal with mean of 1 in all dimensions. Unless stated otherwise, the covariance matrix has 1 in its diagonal entries and 0.7 in the off-diagonal entries.

The aim of the first experiment is to test the relationship between the actual values imputed by the IRMI and OLI methods. The simulation is based on multivariate normal data with 5 dimensions. Results show that the values imputed by the two methods are very near (Fig. 1), with the vast majority of values imputed by the two methods with an absolute difference of up to .02 (compared to the standard deviation of 1.0 in the

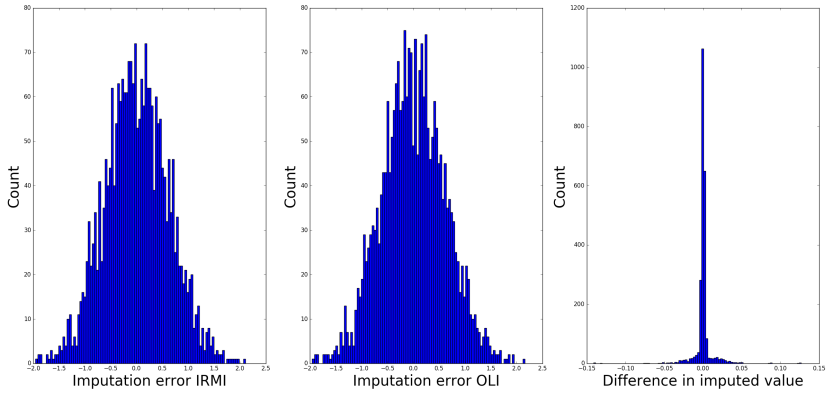


Fig. 1. Distribution of imputation error (imputed − actual) for the IRMI method lLeft). Distribution of imputation error for the OLI method (center). Distribution of the difference in the imputed value between the IRMI and OLI methods (right). Data is $10,000$ samples from a 5-dimensional multivariate normal distribution. All columns have a standard deviation of 1.0 and all pairs of columns have a correlation of 0.7, 5.0% of the data was randomly selected and designated as missing.

data. Furthermore, the distribution of imputation error derived from the two methods is identical. Together, these findings point to the similarity in the results these two methods produce.

In the next simulation we test the performance of the two methods as we vary the number of features. The simulation is based on multivariate normal data with $3 - 20$ dimensions. The results (Fig. 2b) show almost identical behavior of the IRMI and OLI algorithms, which also coincides with the results presented for IRMI in [34]. Median imputation (MI) is also shown for comparison as baseline. Fig 3 shows a zoom into a small segment of figure 2.

As expected, imputing the median (which is also the mean) of each feature for all missing values results in an MSE equal to the standard deviation of the features (i.e., 1). While very close, the IRMI and the OLI methods do not return the exact same imputation values and errors, with an average absolute deviation of 0.053

Next we test the performance of the two methods as we vary the covariance between the features. The simulation is based on multivariate normal data with 5 dimensions. Non-diagonal elements of the covariance matrix are set to values in the range $0.1 - 0.9$. The results (Fig. 2a) show again almost identical behavior of the IRMI and OLI algorithms. As expected, when the dependency between the feature columns is increased, which is measure by the covariance between the columns (X-axis in Fig. 2a), the performance of the regression-based methods IRMI and OLI is monotonically improving, while the performance of the MI method remain unaltered.

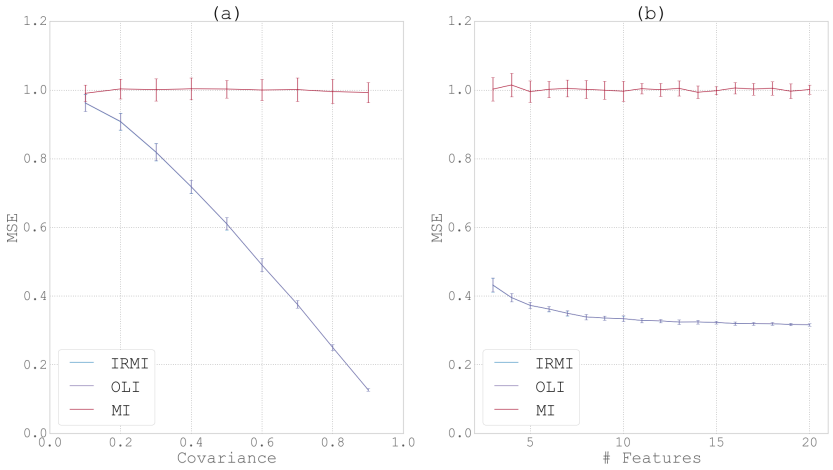


Fig. 2. (a) MSE of the IRMI, OLI and MI methods as a function of the covariance. Data is 5 dimensional multivariate normal. (b) MSE of the IRMI, OLI and MI methods as a function of the dimensionality, with a constant covariance of 0.7 between pairs of features. In both cases error bars represent standard deviation over 20 repetitions.

4.2 UCI datasets

The UCI machine learning repository [19] contains several popular benchmark datasets, some of which have been previously used to compare methods of data imputation [32]. In the current experiment we used the following datasets: *iris* [8], *wine* (white) [3], *Ecoli* [14], *Boston housing* [10], and *power* [35]. Each feature of each dataset was normalized to have mean 0 and standard deviation of 1, in order to make error values comparable between datasets. Categorical features were dropped. For each dataset, 5% of the values were chosen at random and replaced with a missing value indicator. The procedure was repeated 10 times. For these datasets we also consider the MICE method [1] using the *winMice* [15] software.

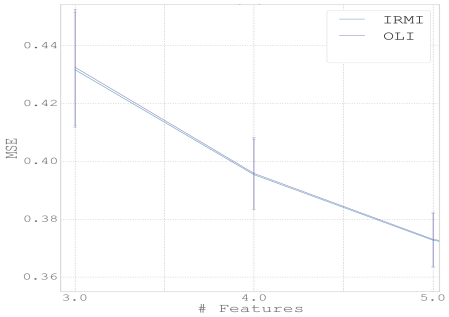


Fig. 3. Zoom into a small part of figure 2

Table 1. Comparison of the imputation results of the IRMI, OLI, MICE and MI methods with 5% missing data. The *converged* column indicates the number of runs in which the IRMI method converged during testing; the MSE of IRMI was calculated for converged repetitions only.

Dataset	# Features	correlation	IRMI		OLI	MI	MICE
			converged	MSE			
Iris	4	0.59	9/10	**0.20**	**0.20**	1.00	0.33
Ecoli	7	0.18	9/10	8.26	5.75	1.72	**1.20**
Wine	11	0.18	**0/10**	-	**0.87**	1.05	1.10
Housing	11	0.45	10/10	**0.28**	0.30	1.14	0.56
Power	4	0.45	**3/10**	0.44	**0.47**	1.02	0.88
Storks	20	0.24	**0/10**	-	**0.31**	1.07	0.42

Overall, the results are quite good, demonstrating the superior ability of the linear methods to impute missing data in these datasets (Table 1, rows 1-5). In the Iris dataset our OLI method achieved an average error identical to IRMI, which successfully converged only 9 out of the 10 runs. Both outperformed the MI and MICE standard methods. In the Ecoli dataset both the IRMI and OLI methods performed worse than the alternative methods, with MICE achieving the lowest MSE. In the Wine dataset the IRMI failed to converge in all 10 repetitions, while the OLI method outperformed the MI and MICE methods. The IRMI method outperformed all other methods in the Housing dataset, but failed to converge 7 out of 10 times for the Power dataset.

In summary, in cases where the linear methods were appropriate, with sufficient correlation between the different features (shown in the second column of Table 1), the proposed OLI method was comparable to the IRMI method with regard to mean square error of the imputed values when the latter converged, and superior in that it always converges and therefore always returns a result.

While the IRMI method achieved slightly better results than OLI in some cases, its failure to converge in others gives the OLI method the edge. Overall, better results were achieved for datasets with high mean correlation between features, as expected when using methods utilizing the linear relationships between features.

4.3 Storks behavioral modes dataset

In the field of Movement Ecology, readings from accelerometers placed on migrating birds are used for both supervised [26] and unsupervised [27,28] learning of behavioral modes. In the following experiment we used a dataset of features extracted from 3815 such measurements. As with the UCI datasets, 10 repetitions were performed, each with 5% of the values randomly selected and marked as missing. Results (Table 1, final row) of this experiment highlight the relative advantage of the OLI method. While the IRMI method failed to converge in all 10 repetitions, OLI achieved an average MSE considerably lower than the MI baseline, and also outperformed the MICE method.

5 Conclusion

Since the problem of missing values often haunts real-word datasets, while most data analysis methods are not designed to deal with this problem, imputation is a necessary pre-processing step whenever discarding entire records is not a viable option. Here we proposed an optimization-based linear imputation method that augments the IRMI [34] method with the property of guaranteed convergence, while staying close in spirit to the original method. Since our method converges to a local optimum of a different objective function, the two methods should not be expected to converge to the same value exactly. However, simulation results show that the results of the proposed method are generally similar (nearly identical) to IRMI when the latter does indeed converge.

The contribution of our paper is two-fold. First, we suggest an optimization problem based method for linear imputation and an algorithm that is guaranteed to converge. Second, we show how this method can be extended to use any number of methods of regularized linear regression. Unlike matrix completion methods [38], we do not have a low rank assumption. Thus, OLI should be preferred when data is expected to have some linear relationships between features and when IRMI fails to converge, or alternatively, when a guarantee of convergence is important (for instance in automated processes).

References

1. Buuren, S., Groothuis-Oudshoorn, K.: mice: Multivariate imputation by chained equations in r. Journal of statistical software 45(3) (2011)
2. Comon, P., Luciani, X., De Almeida, A.L.: Tensor decompositions, alternating least squares and other tales. Journal of chemometrics 23(7-8), 393–405 (2009)
3. Cortez, P., Cerdeira, A., Almeida, F., Matos, T., Reis, J.: Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems 47(4), 547–553 (2009)
4. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the royal statistical society. Series B (methodological) pp. 1–38 (1977)
5. Donders, A.R.T., van der Heijden, G.J., Stijnen, T., Moons, K.G.: Review: a gentle introduction to imputation of missing values. Journal of clinical epidemiology 59(10), 1087–1091 (2006)
6. Duan, Y., Yisheng, L., Kang, W., Zhao, Y.: A deep learning based approach for traffic data imputation. In: Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on. pp. 912–917. IEEE (2014)
7. Engels, J.M., Diehr, P.: Imputation of missing longitudinal data: a comparison of methods. Journal of clinical epidemiology 56(10), 968–976 (2003)
8. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annals of eugenics 7(2), 179–188 (1936)
9. García-Laencina, P.J., Sancho-Gómez, J.L., Figueiras-Vidal, A.R.: Pattern classification with missing data: a review. Neural Computing and Applications 19(2), 263–282 (2010)
10. Harrison, D., Rubinfeld, D.L.: Hedonic housing prices and the demand for clean air. Journal of environmental economics and management 5(1), 81–102 (1978)
11. Heitjan, D.F., Basu, S.: Distinguishing missing at random and missing completely at random. The American Statistician 50(3), 207–213 (1996)

12. Hope, T., Shahaf, D.: Ballpark learning: Estimating labels from rough group comparisons. Joint European Conference on Machine Learning and Knowledge Discovery in Databases pp. 299–314 (2016)
13. Horton, N.J., Kleinman, K.P.: Much ado about nothing. The American Statistician 61(1) (2007)
14. Horton, P., Nakai, K.: A probabilistic classification system for predicting the cellular localization sites of proteins. In: Ismb. vol. 4, pp. 109–115 (1996)
15. Jacobusse, G.: Winmice users manual. TNO Quality of Life, Leiden. URL http://www. multiple-imputation. com (2005)
16. Kandaswamy, C., Silva, L.M., Alexandre, L.A., Sousa, R., Santos, J.M., de Sá, J.M.: Improving transfer learning accuracy by reusing stacked denoising autoencoders. In: Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on. pp. 1380–1387. IEEE (2014)
17. Kim, H., Park, H.: Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. SIAM journal on matrix analysis and applications 30(2), 713–730 (2008)
18. Kroonenberg, P.M., De Leeuw, J.: Principal component analysis of three-mode data by means of alternating least squares algorithms. Psychometrika 45(1), 69–97 (1980)
19. Lichman, M.: UCI machine learning repository (2013), `http://archive.ics.uci.edu/ml`
20. Little, R.J.: A test of missing completely at random for multivariate data with missing values. Journal of the American Statistical Association 83(404), 1198–1202 (1988)
21. Little, R.J., Rubin, D.B.: Statistical analysis with missing data. John Wiley & Sons (2014)
22. Lu, X., Tsao, Y., Matsuda, S., Hori, C.: Speech enhancement based on deep denoising autoencoder. In: Interspeech. pp. 436–440 (2013)
23. Masci, J., Meier, U., Cireşan, D., Schmidhuber, J.: Stacked convolutional auto-encoders for hierarchical feature extraction. Artificial Neural Networks and Machine Learning–ICANN 2011 pp. 52–59 (2011)
24. Pigott, T.D.: A review of methods for missing data. Educational research and evaluation 7(4), 353–383 (2001)
25. Raghunathan, T.E., Lepkowski, J.M., Van Hoewyk, J., Solenberger, P.: A multivariate technique for multiply imputing missing values using a sequence of regression models. Survey methodology 27(1), 85–96 (2001)
26. Resheff, Y.S., Rotics, S., Harel, R., Spiegel, O., Nathan, R.: Accelerater: a web application for supervised learning of behavioral modes from acceleration measurements. Movement ecology 2(1), 25 (2014)
27. Resheff, Y.S., Rotics, S., Nathan, R., Weinshall, D.: Matrix factorization approach to behavioral mode analysis from acceleration data. In: Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on. pp. 1–6. IEEE (2015)
28. Resheff, Y.S., Rotics, S., Nathan, R., Weinshall, D.: Topic modeling of behavioral modes using sensor data. International Journal of Data Science and Analytics 1(1), 51–60 (2016)
29. Resheff, Y.S., Weinshal, D.: Optimized linear imputation. In: Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods - Volume 1: ICPRAM,. pp. 17–25 (2017)
30. Rotics, S., Kaatz, M., Resheff, Y.S., Turjeman, S.F., Zurell, D., Sapir, N., Eggers, U., Flack, A., Fiedler, W., Jeltsch, F., et al.: The challenges of the first migration: movement and behaviour of juvenile vs. adult white storks with insights regarding juvenile mortality. Journal of Animal Ecology 85(4), 938–947 (2016)
31. Rubin, D.B.: Multiple imputation after 18+ years. Journal of the American statistical Association 91(434), 473–489 (1996)

32. Schmitt, P., Mandel, J., Guedj, M.: A comparison of six methods for missing data imputation. Journal of Biometrics & Biostatistics 2015 (2015)
33. Takane, Y., Young, F.W., De Leeuw, J.: Nonmetric individual differences multidimensional scaling: An alternating least squares method with optimal scaling features. Psychometrika 42(1), 7–67 (1977)
34. Templ, M., Kowarik, A., Filzmoser, P.: Iterative stepwise regression imputation using standard and robust methods. Computational Statistics & Data Analysis 55(10), 2793–2806 (2011)
35. Tüfekci, P.: Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. International Journal of Electrical Power & Energy Systems 60, 126–140 (2014)
36. Van Buuren, S., Oudshoorn, K.: Flexible multivariate imputation by mice. Leiden, The Netherlands: TNO Prevention Center (1999)
37. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. The Journal of Machine Learning Research 11, 3371–3408 (2010)
38. Wagner, A., Zuk, O.: Low-rank matrix recovery from row-and-column affine measurements. arXiv preprint arXiv:1505.06292 (2015)
39. Xie, J., Xu, L., Chen, E.: Image denoising and inpainting with deep neural networks. In: Advances in Neural Information Processing Systems. pp. 341–349 (2012)
40. Zhou, G., Sohn, K., Lee, H.: Online incremental feature learning with denoising autoencoders. Ann Arbor 1001, 48109 (2012)

4.2 Controlling Imbalanced Error in Deep Learning

This paper is an extended version of [1] which was presented at *the First International Workshop on Learning with Imbalanced Domains: Theory and Applications*, held at the *European Conference on Machine Learning*, 18-22 September 2017, Skopje, Macedonia.

References

[1] Yehezkel S Resheff, Amit Mandelbom, and Daphna Weinshall. "Controlling Imbalanced Error in Deep Learning with the Log Bilinear Loss". In: *First International Workshop on Learning with Imbalanced Domains: Theory and Applications*. 2017, pp. 141–151.

Every Untrue Label is Untrue in its Own Way: Controlling Error Type with the Log Bilinear Loss

Yehezkel S. Resheff[1,2], Amit Mandelbaum[1], and Daphna Weinshall[1]

[1] School of Computer Science and Engineering, The Hebrew University of Jerusalem
{heziresheff, amit.mandelbaum, daphna}@cs.huji.ac.il
[2] Edmond and Lily Safra Center for Brain Sciences, The Hebrew University of Jerusalem

Abstract. Deep learning has become the method of choice in many application domains of machine learning in recent years, especially for multi-class classification tasks. The most common loss function used in this context is the cross-entropy loss, which reduces to the log loss in the typical case when there is a single correct response label. While this loss is insensitive to the identity of the assigned class in the case of misclassification, in practice it is often the case that some errors may be more detrimental than others. Here we present the bilinear-loss (and related log-bilinear-loss) which differentially penalizes the different wrong assignments of the model. We thoroughly test this method using standard models and benchmark image datasets. As one application, we show the ability of this method to better contain error within the correct super-class, in the hierarchically labeled CIFAR100 dataset, without affecting the overall performance of the classifier.

1 Introduction

Multi-class classification may well be the most studied machine learning problem, from both the theoretical and the practical aspects. The problem is often phrased as an optimization problem, where we seek a classifier (or algorithm) which minimizes some loss function [9]. To begin with we may phrase the problem using the 0-1 loss, which effectively counts the number of misclassified points. Most methods then replace this function by a surrogate loss due to various computational reasons (often just to achieve a tractable problem), or by the design of a more specific goal-oriented or application motivated loss function.

In this paper we go back to basics, and question the use of the 0-1 loss function in multi-class classification. Already in the binary scenario, we see the need for a more subtle loss function, which distinguishes between the 2 different types of possible errors: false negative - when a point of class 1 is wrongly classified as class 2, and false positive - when a point of class 2 is wrongly classified as class 1. When class 1 symbolizes the diagnosis of some serious illness while class 2 refers to the lack of findings, differential treatment of the two errors may be

a matter of life and death. In most applications this distinction doesn't make it into the definition of the loss function (but see review of related work below), and it is usually approached post-hoc by such means as the ROC curve, which essentially describes the behavior of the classifier as a trade-off between the two types of error.

These days the field of applied machine learning is swept by deep learning [12], and we customary solve very large multi-class classification problems with a growing number of classes. Now the question becomes more acute, and the 0-1 loss function does not seem to provide a good model for the real world anymore. In most application domains, different misclassification errors are likely to have different detrimental implications, and should be penalized accordingly to achieve an appropriate classifier. For example, the consequence of diagnosing a certain illness wrongly by coming up with a related illness may be considered less harmful to the patient than missing it altogether.

Unlike binary classification, with multi-class problems we don't have only two types of errors to worry about, but rather $O(k^2)$ errors if there are k labels to choose from. We can no longer postpone the resolution of the problem to some post-hoc stage, and the choice of a single threshold. Furthermore, when considering deep learning, encoding the desired trade-off between types of errors will hopefully lead to a representation more suitable to make the necessary distinctions.

Yet in practice, the basic loss function one uses in the training of neural network classifiers has changed little, revealing its deep roots in the world of binary classification and the 0-1 loss function. Specifically, the categorical cross-entropy (which when using a single correct response reduces simply to log-loss), has the property of invariance to the division of the output weight among the erroneous labels, and therefore it is also invariant to the identity of the class an example is assigned to, in case of misclassification.

This is the problem addressed in this paper. We propose two loss functions, which provide the means for distinguishing between different misclassification errors by penalizing each error differently, while maintaining good performance in terms of overall accuracy. Specifically, in Section 2 we develop the bilinear loss (and related log-bilinear loss) which directly penalize a deep learning model differentially for weights assigned in the output layer, depending on both the correct label and the identity of the wrong labels which have weight assigned to them.

In Section 3 we describe the empirical evaluation of training deep networks with these loss functions, using standard deep learning models and benchmark multi-class datasets. Thus we empirically show the ability of these methods to maintain good overall performance while redistributing the error differently between the possible wrong assignments.

In Section 4 we show an application of the proposed method for controlling error within the correct super-class in the hierarchically labeled CIFAR100 dataset, without significant reduction in the total accuracy of the final classifier (in fact, slight improvement is observed in our empirical evaluation).

The contribution of this paper is twofold. First, we introduce the bilinear/log-bilinear loss functions and show their utility for controlling error location in deep learning models. Second, we suggest the task of confining error to the correct super-class in a hierarchical settings, and show how these loss functions can achieve this goal.

Related Work Asymmetric loss functions have been studied extensively in the context of binary classification, both theoretically and algorithmically, see e.g. [2,15,18,17]. Thus asymmetric loss functions have been shown to improve classification performance with asymmetrical error costs or imbalanced data. In a related approach, boosting methods (e.g. [5]) are based on the asymmetric treatment of training points, and can naturally be modified to include cost sensitive loss functions [20]. The optimal selection (with respect to a differential cost of the two types of error) of the final classifier's threshold has been discussed in [8,10], for example. This question also received specific attention in the context of the design of cascades of detectors [23,22], since as part of the construction of the cascade one must give the different errors - miss the event or detect it when it is absent - different weights in the different levels of the cascade.

In the context of multiclass classification, proper rescaling methods are discussed in [25] in order to address the issue of imbalanced datasets; rescaling methods, however, typically penalize error based on the identify of the wrong label only. [4] describes a wrapper method to transform every multi-class classifier to a cost-sensitive one, while [16] shows how multi-class decision trees can be trained to approximate a general loss matrix. A truly asymmetric cost-sensitive loss function is used in [14], employing a *generalized cost matrix* somewhat similar to matrix A defined in Section 2, while [24] offers a formulation which is based on the Bayes-decision theory and the k-nearest neighbor classifier.

In the context of deep learning, the purpose of a cost sensitive objective is not only to change the output of the model, but first and foremost to learn a representation in intermediate layers of the network that is able to better capture the aspects of the data that are important according to the relative cost of the different types of error. This is demonstrated in the hierarchical experiments below (Section 3).

2 The [Log] Bilinear-loss

We assume a model with k non-negative output units, such that the output for the $i-th$ example is:

$$\hat{y}^{(i)} = \hat{y}_1^{(i)}, \ldots, \hat{y}_k^{(i)}$$

and further assume a per-example normalized output, i.e.: $\forall i : \sum_{j=1}^{k} \hat{y}_j^{(i)} = 1$

In most cases in practice, the training set is labeled with a single correct response per example. Thus, the common cross-entropy loss reduces to $-log(y_{l_i}^{(i)})$, where l_i is the correct label for the i'th example. This loss essentially represents

an implicit policy of rewarding for weight placed on the correct answer, while being indifferent to the identity of the wrong labels which have weight assigned to them.

Arguably, this common practice is often in direct opposition to the goal of the process of learning; in many real world scenarios, some mistakes are extremely costly while others are of little consequence. In this section we develop alternative loss functions which address this issue directly.

We start by modifying the loss function used for training the classifier. We augment the usual loss function with a term where a non-negative cost is assigned to any wrong classification, and where the cost is based on *both* the correct label and the identity of the misclassified label in an asymmetrical manner. Specifically, let $a_{i,j}$ denote the relative cost associated with assigning the label j to an example whose correct label is i, and let $A = \{a_{i,j}\} \in \mathcal{R}^{k \times k}$ denote the penalty matrix.

Using A we define two related loss functions: The Bilinear loss is defined as:

$$L_B = y^T A \hat{y} \tag{1}$$

where y denotes the correct output (y is a probability vector). Similarly, the log-Bilinear loss is defined as:

$$L_{LB} = -y^T A log(1 - \hat{y}) \tag{2}$$

where $log(\cdot)$ operates element-wise.

Finally, we combine the regular cross-entropy loss with (1) and (2) to achieve a loss function with the known benefits of cross-entropy, and which also provides the deferential treatment of errors:

$$L_{CE+B} = (1 - \alpha) L_{CE} + \alpha y^T A \hat{y} \tag{3}$$

$$L_{CE+LB} = (1 - \alpha) L_{CE} - \alpha y^T A log(1 - \hat{y}) \tag{4}$$

where $L_{CE} = -\sum y_i log(\hat{y}_i)$ is the regular cross-entropy loss.

The fundamental difference between the bilinear and log-bilinear loss formulations is the implied view on what in the output of the model is to be penalized (and thus controlled). The bilinear formulation adds a constant cost $a_{ij} \Delta p_j$ for each Δp_j increase in the j'th normalized output unit, for a training example from the i'th class. The log-bilinear formulation on the other hand is insensitive to this sort of increase as long as the overall value in the j'th output is small, but picks up sharply as p_j approaches 1.

Furthermore, for two equally penalized mistakes j, k for an example from class i (meaning $a_{ij} = a_{ik}$), the bilinear loss is insensitive to the division of error between the two classes, since $\alpha_1 a_{ij} + \alpha_2 a_{ik} = (\alpha_1 + \alpha_2) a_{ij}$, so only the sum of the assignments to these two classes matters. The log-bilinear loss however amounts to $\alpha_1 log(1 - a_{ij}) + \alpha_2 log(1 - a_{ik})$, which for a constant sum is maximized when all the weight is placed on one of the errors.

The meaning of this property is that the bilinear loss is penalizing for the total weight placed on erroneous classes weighted by the relative importance (as measured by the penalty matrix A). The log-bilinear loss is penalizing peakiness of the wrong assignment, again weighted by importance. Thus, the bilinear loss is a more likely candidate when we would like to control the locations where erroneous weights concentrate, and the log-bilinear loss when we would like to control which errors the model is confident about.

We note that while we naturally apply the method to deep learning models in our experiments (section 3), these loss functions are applicable to any gradient based model with outputs as described above.

3 Empirical evaluation of the loss functions

In this section we evaluate and compare the efficacy of the two loss functions defined in (3) and (4), in terms of two measures and the trade-off between them: (i) the total un-weighted error in the original classification task; (ii) the error distribution as measured by the different elements of the confusion matrix at train and test time.

Specifically, the purpose of the following experiments is: (a) to evaluate and compare the ability of the proposed formulation to control the location of error, using standard deep learning methods and benchmark datasets; (b) test the influence of the trade-off parameter α in the above formulations (3) and (4); and (c) evaluate the trade-off between control of error location, and overall accuracy of the model.

3.1 Methods

In order to test the ability to control the location of error, we randomly select a varying number n of specific errors to be avoided. Each such error is defined by the identity of both the correct label and the wrong label (such as for example: "don't mistake a 2 for an 8"). In practice, this is done by sampling a random mask (Boolean matrix) of size k^2, with exactly n positive off-diagonal elements, where k is the number of classes. These n locations are henceforth called *the masked zone*.

The datasets we use for empirical evaluation in this section are MNIST and CIFAR10, where $k = 10$. For each combination of n in $\{10, 20, 30, 40, 50\}$ and the trade-off parameter α in $\{0, .1, .5, .9, .95, .99\}$, we trained 50 and 10 models (MNIST and CIFAR10 respectively). Thus we trained a total of 1500 MNIST models, and 300 CIFAR10 models. This design was repeated for the Bilinear and Log-Bilinear loss functions, leading to a total of 3600 models.

All models were trained using the open source Keras and TensorFLow [1] software packages [3]. In order to facilitate the training of a large number of models, MNIST models were each trained for only 10 epochs, and CIFAR10 models for 300 epochs with an early stopping criterion.

[3] Materials available at: https://github.com/Hezi-Resheff/paper-log-bilinear-loss

3.2 MNIST dataset: results

The MNIST dataset [13] is an old benchmark dataset of small images (28×28 pixels) of hand-written digits. The data is divided into $55,000$ images in the training set, $5,000$ for validation, and $10,000$ images in the test set. In recent years, deep learning methods have been used successfully to reach almost perfect classification when using the MNIST dataset (see for example [3,7]).

Table 1. The model used in all MNIST experiments

	Layer	#params
1	convolution (20; 5X5)	520
2	max-pooling(2X2)	
3	dropout(20%)	
4	convolution(50; 5X5)	25,050
5	max-pooling(2X2)	
6	dropout(20%)	
7	fully-connected(500)	1,225,500
8	fully-connected(10)	5,010
	total:	1,256,080

The model we use in our experiments (detailed in Table 1) is a typical small model with two convolutional layers, followed by a single fully-connected layer. Max-pooling and dropout [19] are used following each convolutional layer. This model was selected for the current experiments primarily because of the small training time required to achieve satisfactory results (approx. 99.2% correct after 10 epochs), which allowed us to generate a very large number of models, as discussed in the methods section above.

Bilinear Loss. Results are shown in Fig. 1. Clearly the number of test errors in the selected mask spots (the 'masked zone') is dramatically reduced when using our proposed loss function (3) with $\alpha > 0$, as compared to the baseline models ($\alpha = 0$). As expected, the total number of errors increases linearly with the size of the mask. At the same time the reduction in error count in the masked zone is attenuated by the value of the trade-off parameter α, with higher values of α leading to fewer errors in this zone. Noticeably, as the number n of selected spots in the mask is increased, a larger value of α is necessary in order to retain the 82 reduction in the number of errors relative to the baseline.

The overall model accuracy (Fig. 2) changes very little for all but the largest value of α. However, for $\alpha = .99$ we see a substantial increase in overall error. This value of α was also responsible for the most dramatic decrease in errors in the masked zone. Thus in this parameter value regime the network achieves an inferior overall solution, but succeeds in pushing most of the errors outside the masked zone.

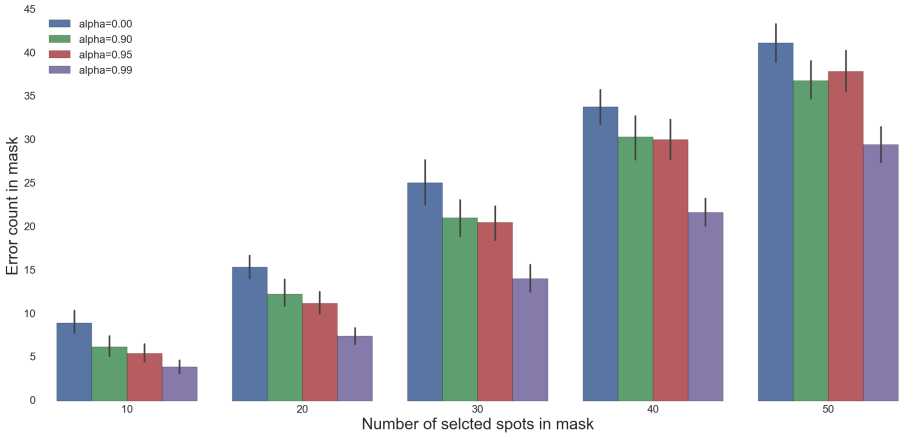


Fig. 1. MNIST, Bilinear loss. We plot the number of erroneous test examples in the masked zone (y-axis) as a function of the zone's size (x-axis), mean over the 50 repetitions per configuration with error bars showing the 95% confidence interval.

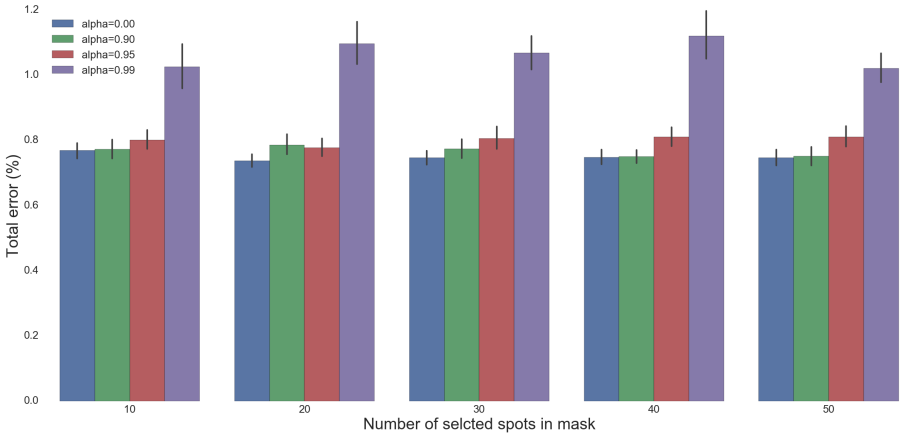


Fig. 2. MNIST, Bilinear loss. We plot the percent of overall model error, mean over the 50 repetitions per configuration with error bars showing the 95% confidence interval.

Overall, for intermediate values of the trade-off parameter α, the number of errors in the masked zone is reduced dramatically, even for relatively large masks with as many as $40-50$ points, without an appreciable increase in overall error. This result demonstrates the feasibility of the proposed bilinear loss (when added to the regular cross-entropy loss), as a means of controlling the location of error without harming overall accuracy in deep learning models.

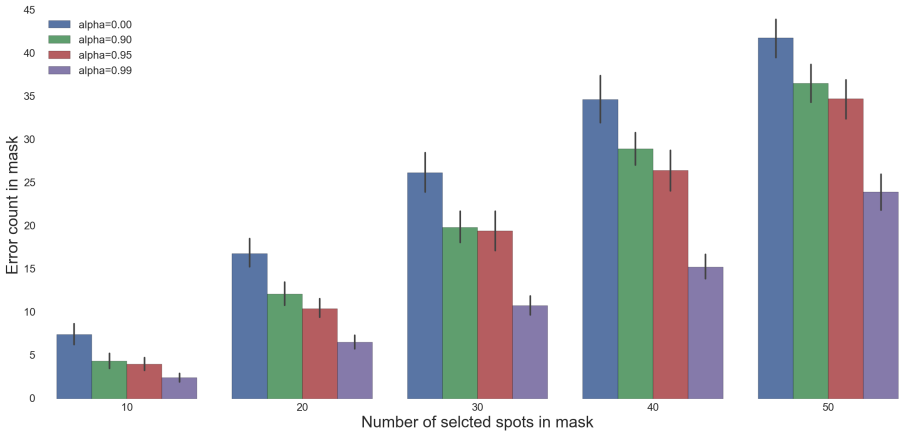


Fig. 3. MNIST, Log Bilinear loss; see caption of Fig. 1.

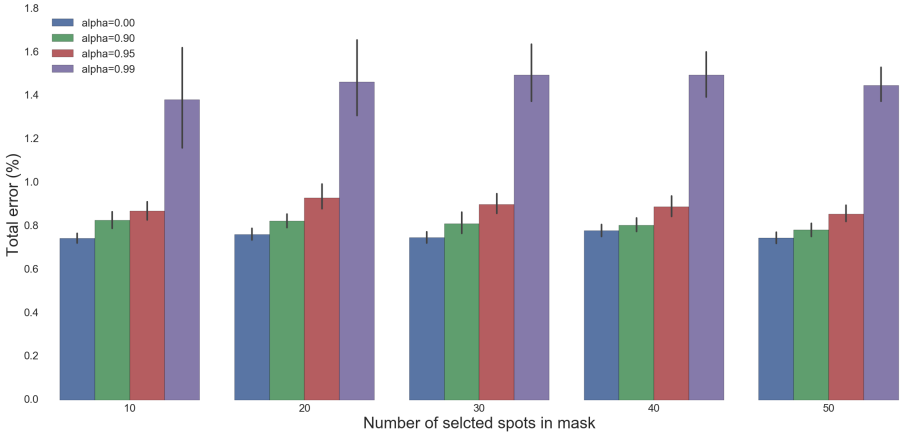


Fig. 4. MNIST, Log Bilinear loss; see caption of Fig. 2.

Log-Bilinear Loss. The results for the log-bilinear loss are qualitatively similar to the bilinear loss presented above. The reduction in the number of errors in the masked zone (Fig. 3) relative to the baseline is larger than with the bilinear loss for small mask sizes ($n = 10 - 20$), and comparable for the larger masks. The overall model accuracy (Fig. 4) is, however, significantly worse than in the linear case, with small adverse effects showing already for small α values.

Overall, the log-bilinear loss, much like the linear version, is able to reduce the number of errors in an MNIST model in a selected zone. However, the reduction in error in the masked zone when using the log-bilinear loss is more substantial, at a price of a worse model overall; this harmful effect is slight for small values of α.

3.3 CIFAR-10: results

The CIFAR-10 dataset [11] is made up of 60,000 small images (32×32 color pixels), each belonging to one of 10 classes (airplane, car, bird, cat, deer, dog, frog, horse, ship, truck). The data is divided into a training set of 50,000 images, and a test set of the remaining 10,000 images.

As in the MNIST case, the model we use (Table 2) is selected on the basis of typicality, accuracy, and relatively short training time to allow many models to be computed. Here, the model consists of three blocks of convolutional layers (each containing two convolutional layers, followed by max-pooling and dropout layers), and two fully-connected layers. Overall accuracy is approximately 92% after 300 epochs at most. (300 was set to be the maximal number of epochs; early stopping was employed when convergence was achieved earlier, which was often the case.)

Table 2. The model used in all CIFAR10/100 experiments

	Layer	#params
1	convolution (64; 3X3)	1,792
2	convolution (64; 3X3)	36,928
3	max-pooling(2X2)	
4	dropout	
5	convolution(128; 3X3)	73,856
6	convolution(128; 3X3)	147,584
7	max-pooling(2X2)	
8	dropout	
9	convolution(256; 3X3)	295,168
10	convolution(256; 3X3)	590,080
11	max-pooling(2X2)	
12	dropout	
13	fully-connected(1000)	257,000
14	dropout (CIFAR100 only)	
15	fully-connected(1000)	1,001,000
16	dropout (CIFAR100 only)	
17	fully-connected(10/100)	10,010/100,100
	total:	2,413,418 (CFIAR-10) 2,503,508(CIFAR-100)

Bilinear Loss. Results are shown in Fig. 5. Like before, the number of test errors in the masked zone is dramatically reduced when using our proposed loss function (3) with $\alpha > 0$, as compared to the baseline models ($\alpha = 0$).

Compared to the MNIST results, for a small mask of 10 locations, the reduction in error in the masked zone is appreciably larger for CIFAR-10. We attribute this to *compulsory errors* – errors which stem from true class overlap, and thus

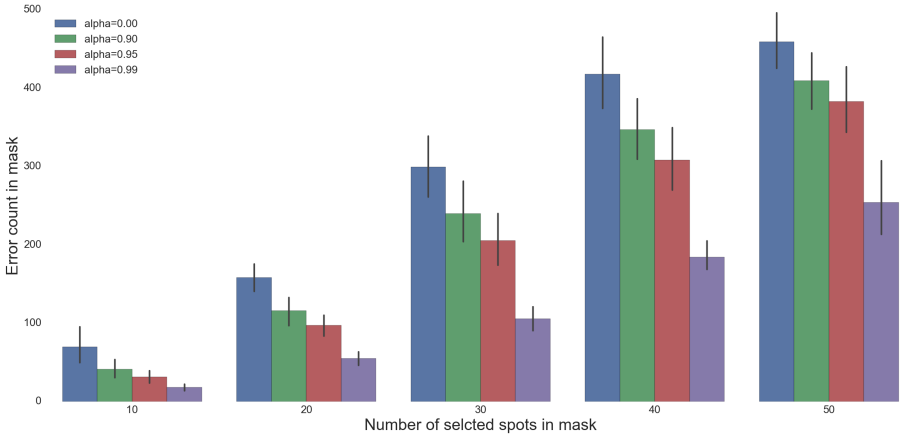


Fig. 5. CIFAR-10, Bilinear loss. We plot the number of erroneous test examples in the masked zone (y-axis) as a function of the zone's size (x-axis), mean over the 10 repetitions per configuration with error bars showing the 95% confidence interval.

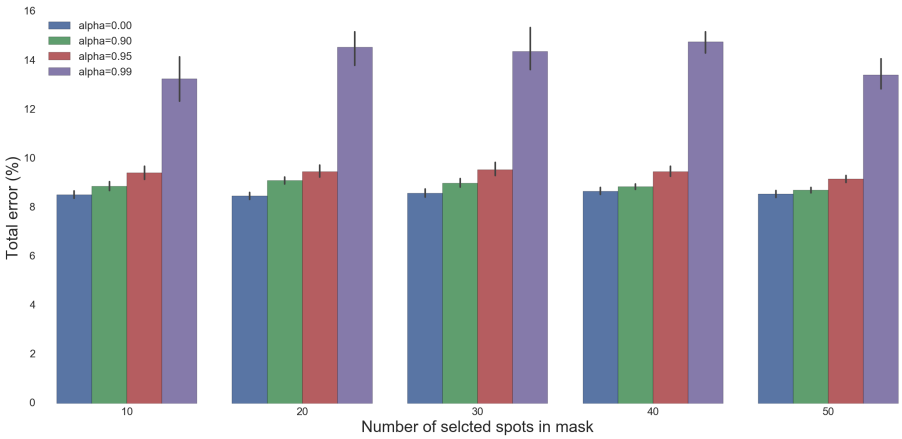


Fig. 6. CIFAR-10, Bilinear loss. We plot the percent of overall model error, mean over the 10 repetitions per configuration with error bars showing the 95% confidence interval.

can't be overcome by changing the objective. Arguably, the MNIST dataset has a higher volume of these, explaining the better ability to control the error away from the mask when using the CIFAR-10 dataset.

The overall model accuracy (Fig. 6) changes very little for all but the largest value of α. Unlike the MNIST case, however, we see a real (albeit small) increase in overall error even for lower values of α. For $\alpha = .99$ we again see a substantial increase in overall error. This value of α was also responsible for the most dramatic decrease in errors in the masked zone. Thus in this regime of parameter

values the network achieves an inferior overall solution, but succeeds in pushing most of the errors outside the masked zone.

Overall, for intermediate values of the trade-off parameter α, the number of errors in the masked zone is reduced dramatically, even for relatively large masks with as many as $40 - 50$ points, without drastically harming the overall performance of the model.

Log-Bilinear Loss. The log-bilinear loss results show a large reduction in the error in the masked zone (Fig. 7), but at the same time an increase in overall model error (Fig. 8) already for small masks and smaller values of α.

We show data in this case (Fig. 7 and 8) for masks of size up to 30, and α values of up to 0.95. The log-bilinear loss is less effective than the bilinear loss already in this regime, and is not of practical use for this dataset with larger masks. It would seem that for this dataset the bilinear loss produces better results overall in terms of control of error on the one hand, while maintaining reasonable overall model accuracy on the other.

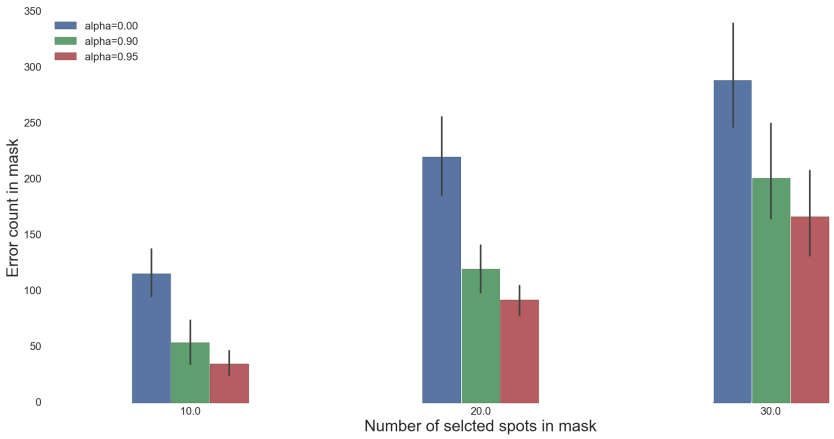


Fig. 7. CIFAR-10, Log Bilinear loss; see caption of Fig. 5.

4 Hierarchical classification: empirical evaluation

When the data is organized hierarchically in a tree, a natural order in imposed on the errors that the classifier can make. In some domains it may be beneficial to penalize for errors based on the distance between the node of the true label and the node corresponding to the erroneous label, where the distance reflects the lowest node in the tree that is ancestral to both nodes, since the detrimental effect of errors may be proportional to this distance.

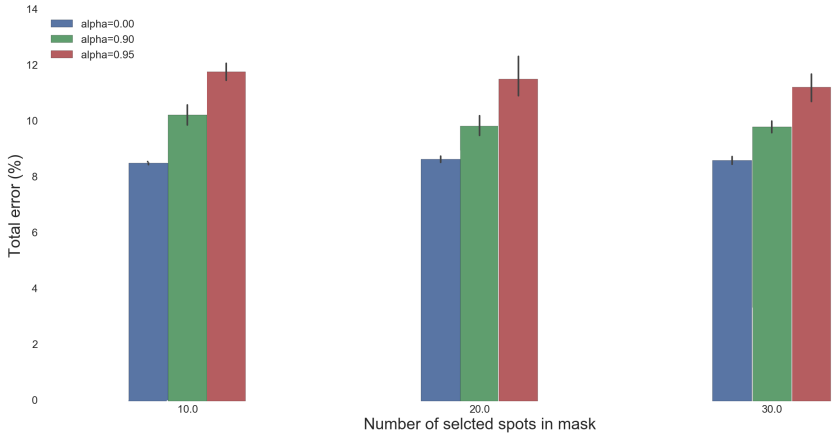


Fig. 8. CIFAR-10, Log Bilinear loss; see caption of Fig. 6.

In this section we investigate this scenario using the hierarchical CIFAR100 dataset [11]; this dataset contains 100 classes, which are grouped into 20 coarse-grained super-classes (for example, the *reptile* super-class contains: crocodile, dinosaur, lizard, snake, and turtle; the *insects* super-class contains: bee, beetle, butterfly, caterpillar, and cockroach).

Hierarchical structure is often utilized by classifiers in order to improve the regular notion of classification [21,6]. The objective here is to generate a classifier that, when wrong, will be more likely to err by choosing another class from the same super-class containing the true label.

We achieve this goal by using the bilinear and log-bilinear loss functions defined in (3) and (4), where the elements of the penalty matrix $A = \{a_{i,j}\}$ are defined as follows:

$$a_{i,j} = \begin{cases} 0 & i = j \\ 1 & \text{i and j are in the same super-class} \\ 5 & \text{o.w.} \end{cases}$$

This penalty matrix reflects the notion that we would much rather err within the correct super-class.

As in the above experiments, we test multiple values of the trade-off parameter α for each of the bilinear and log-bilinear models, and 10 repetitions of each combination. The results show that when using either the bilinear (Table 3) or the log-bilinear (Table 4) loss functions, we are able to reduce the error for coarse-grained classification (into super-classes) without hindering the overall model accuracy.

Specifically, the coarse-grained overall error is reduced from 25.45% to 24.01% when using the bilinear loss with $\alpha = 0.5$, and to 24.52% when using the log-bilinear loss with the same values of α. Interestingly, in the latter case the overall model accuracy is slightly (but significantly) reduced as well.

Table 3. CIFAR-100, Bilinear loss. α: the trade-off parameter (see (3)). Total error: the mean overall test set error. Total error coarse: the mean overall test set error when using the coarse classification into the 20 super-classes. Relative error with correct super-class label: the mean percent of errors which are assigned a label in the correct super-class.

α	total error	total error coarse	relative error with correct super-class label
0	37.36	25.45	30.88
0.05	37.41	25.33	31.22
0.10	37.33	25.27	31.63
0.25	37.13	24.62	32.92
0.50	36.93	24.01	34.61
0.75	38.33	24.52	35.76
0.90	41.41	26.45	35.98
0.95	44.57	28.82	35.27

Table 4. CIFAR-100, Log-Bilinear loss. See caption in Table 3.

α	total error	total error coarse	relative error with correct super-class label
0	37.30	25.43	30.88
0.05	37.23	25.12	31.55
0.10	37.25	24.92	32.20
0.25	37.61	24.71	33.60
0.50	38.52	24.52	36.58
0.75	41.53	25.73	38.32
0.90	46.18	28.26	38.87
0.95	48.98	30.31	38.21

In the baseline model ($\alpha = 0$), 30.88% of all mislabeled examples are mislabeled into the correct super-class (i.e. assigned one of the other 4 classes that form the same super-class as the correct label). [4] When using the bilinear loss, the percent of erroneous classification labels that fall within the correct super-class in increased to 34.61% with $\alpha = 0.5$. Likewise, the value increases to 36.58% with the log-bilinear loss. However, in the latter case the total error both in the regular (100 classes) and coarse classification (20 super-classes) is somewhat higher.

It seems that for this task the bilinear loss is more effective than the log-bilinear loss. The reason for this may be that the many misclassification errors are evaluated with low confidence, which gives a relatively flat response for the

[4] This value, rather than the expected $\frac{4}{99}$ reflects the hierarchical structure that exists in this dataset, as captured in the super-class labels.

100 classes. Thus, while the bilinear loss accumulates these errors, the log-bilinear loss adds up values which are very small (the sum of $log(1-\epsilon)$).

We note that while in the MNIST/CIFAR10 experiments we generate masks to direct the error away from as many as 50 out of the 90 possible error types, here only 400 out of the possible $9,900$ possible errors are within super-class errors (each super-class is of size 5). Remarkably, even in this extreme case we see a clear effect of the error being controlled as defined in the bilinear loss.

Small sample In the final experiment presented here, we test the effect of the bilinear loss in the small-sample case. From each of the 20 super-classes of the CIFAR-100 dataset, we select a class based on the highest super-class-typicality, which we define as the percent of the errors for each class that are assigned to another class within the same super-class, in the baseline model.

Each of the 20 selected classes is down-sampled to $0, 10$, or 50 training examples. In the 10 and 50 cases, the small sample is then duplicated back to the original sample size. Models are then trained using the bilinear loss, using the matrix described above (penalizing differentially errors in the correct and wrong super-class), with values of the trade-off parameter α in $\{0, 0.25, 0.5\}$.

The results (Table 5) show a small improvement in model accuracy - the accuracy of the small-sample classes, and small improvement in the assignment of the small-sample classes to the correct super-class for samples of size $N = 10, 50$. Interestingly, even when the selected classes are not seen at at all during training ($N = 0$), and even for the baseline model ($\alpha = 0$), almost half (49.42%) of the test examples from the selected classes are assigned to the correct super-class. While somewhat expected, since the most typical classes are used, this again mostly reflects the visually informative super-class structure in the CIFAR-100 dataset.

5 Conclusions

Often in real-world classification tasks, the cost of an error depends on the identity of the correct label as well as that of the misclassification target. The cross-entropy objective most commonly used in classification with deep learning models does not accommodate this in a natural way.

In this paper we present the bilinear and log-bilinear loss functions, which directly add to the regular cross-entropy loss a component that depends on the true and wrongly assigned labels. We evaluate these formulations extensively using standard models and benchmark datasets.

First, we show that with the MNIST/CIFAR-10 datasets we are able to direct error away from a randomly chosen mask of up to 50 out of the 90 possible errors (the non-diagonal elements of the 10 by 10 confusion matrix), while preserving the overall model accuracy.

Next, we show that in the hierarchically annotated CIFAR-100 dataset, these methods are able to help contain error to within the correct super-class, thus producing models with the same overall accuracy as the baseline model, but

Table 5. CIFAR-100 with selected small-sample classes. N: the number of samples in each of the 20 small-sample classes. alpha: the value of the trade-off parameter α. model accuracy: the overall accuracy of the model. small sample accuracy: the mean accuracy for the 20 small sample classes. small sample super-class accuracy: the percent of test examples from the small-sample classes that are assigned a class in the correct super-class

N	α	model accuracy	small sample accuracy	small sample super-class accuracy
0	0	54.92	0.00	49.42
	0.25	54.94	0.00	49.24
	0.50	54.91	0.00	50.54
10	0	51.76	4.23	47.82
	0.25	52.33	4.45	49.20
	0.50	52.72	6.02	50.14
50	0	56.88	21.98	56.43
	0.25	56.87	22.43	56.85
	0.50	57.11	23.47	58.52

with a higher inclination to choose a label from sibling classes (as defined by the super-class) when mistakes occur.

Future work will focus on extending this "hierarchical learning" by coercing the layers of a deep model to gradually focus on higher and higher levels of a fuzzy hierarchy.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), `http://tensorflow.org/`, software available from tensorflow.org
2. Bach, F.R., Heckerman, D., Horvitz, E.: Considering cost asymmetry in learning classifiers. Journal of Machine Learning Research 7(Aug), 1713–1741 (2006)
3. Ciregan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. pp. 3642–3649. IEEE (2012)
4. Domingos, P.: Metacost: A general method for making classifiers cost-sensitive. In: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 155–164. ACM (1999)
5. Fan, W., Stolfo, S.J., Zhang, J., Chan, P.K.: Adacost: misclassification cost-sensitive boosting. In: Icml. pp. 97–105 (1999)
6. Gopal, S., Yang, Y.: Recursive regularization for large-scale classification with hierarchical and graphical dependencies. In: Proceedings of the 19th ACM SIGKDD

international conference on Knowledge discovery and data mining. pp. 257–265. ACM (2013)
7. Jarrett, K., Kavukcuoglu, K., LeCun, Y., et al.: What is the best multi-stage architecture for object recognition? In: Computer Vision, 2009 IEEE 12th International Conference on. pp. 2146–2153. IEEE (2009)
8. Karakoulas, G., Shawe-Taylor, J.: Optimizing classifiers for imbalanced training sets. In: Proceedings of the 11th International Conference on Neural Information Processing Systems. pp. 253–259. MIT Press (1998)
9. Kotsiantis, S.B., Zaharakis, I., Pintelas, P.: Supervised machine learning: A review of classification techniques (2007)
10. Koyejo, O.O., Natarajan, N., Ravikumar, P.K., Dhillon, I.S.: Consistent binary classification with generalized performance metrics. In: Advances in Neural Information Processing Systems. pp. 2744–2752 (2014)
11. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images (2009)
12. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature 521(7553), 436–444 (2015)
13. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)
14. Lee, Y., Lin, Y., Wahba, G.: Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. Journal of the American Statistical Association 99(465), 67–81 (2004)
15. Lin, Y., Lee, Y., Wahba, G.: Support vector machines for classification in nonstandard situations. Machine learning 46(1), 191–202 (2002)
16. Margineantu, D.D., Dietterich, T.G., et al.: Learning decision trees for loss minimization in multi-class problems. Tech. rep., Corvallis, OR: Oregon State University, Dept. of Computer Science (1999)
17. Masnadi-Shirazi, H., Vasconcelos, N.: Risk minimization, probability elicitation, and cost-sensitive svms. In: ICML. pp. 759–766 (2010)
18. Scott, C., et al.: Calibrated asymmetric surrogate losses. Electronic Journal of Statistics 6, 958–992 (2012)
19. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15(1), 1929–1958 (2014)
20. Sun, Y., Kamel, M.S., Wong, A.K., Wang, Y.: Cost-sensitive boosting for classification of imbalanced data. Pattern Recognition 40(12), 3358–3378 (2007)
21. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. Journal of machine learning research 6(Sep), 1453–1484 (2005)
22. Wu, J., Brubaker, S.C., Mullin, M.D., Rehg, J.M.: Fast asymmetric learning for cascade face detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(3), 369–382 (2008)
23. Wu, J., Mullin, M.D., Rehg, J.M.: Linear asymmetric classifier for cascade detectors. In: Proceedings of the 22nd international conference on Machine learning. pp. 988–995. ACM (2005)
24. Zhang, Y., Zhou, Z.H.: Cost-sensitive face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(10), 1758–1769 (2010)
25. Zhou, Z.H., Liu, X.Y.: On multi-class cost-sensitive learning. Computational Intelligence 26(3), 232–257 (2010)

Chapter 5

Discussion

The aim of this study was to investigate and develop machine learning techniques for analysis of wearable device derived data, in the fields of Movement Ecology and Medical Devices. Results indicate the usefulness of Inertial Measurement Unit (IMU) sensor data for supervised as well as unsupervised scenarios for both applications (Chapter 2, and Appendix A+B). In this thesis we go on to show that smart aggregations allow us to analyze huge amounts of trajectory data, while keeping pertinent information (Chapter 3). Finally, some more general machine learning methods are developed, with insights and motivations from the problems at hand, and wider applicability to various data-science domains (Chapter 4).

Supervised learning has long been used for automatic annotation of behavioral modes from IMU stream data [6, 1]. In Movement Ecology, behavioral annotation from accelerometer data, overlayed on animal trajectories, has led to many otherwise insurmountable discoveries. In the field of medical devices, smart-watches and other wearable devices have been used to track symptoms of Parkinson's patients [10]. Activity recognition based on supervised methods has also been used extensively for tracking and monitoring of general wellbeing using smart-watches and smart-phone devices.

The application of supervised learning is however inherently limited to research projects where labeled data is obtainable (Chapter 2). The availability of such labeled data is subject to a number of resources, both financial and technical. It is infeasible when collecting a sufficiently large labeled dataset is too expensive to be practical, or when the knowledge and technical ability is in question (this is the case for instance for nocturnal animals, which are often extremely hard to track and observe in their natural habitats, as well as for mental health where between-patient variation renders annotation of specific behaviors all but useless). As a corollary, it is evident that for successful application of this technique one must have a full anthology of all pertinent behavioral modes. This is often not the case.

For these reasons, unsupervised machine learning techniques are an indispensable part of the wearable devices analytics toolkit. Surprisingly, very little work has previously been done on adapting and building such tools for this intriguing field. Existing methods were limited to simple clustering on snippets from sensor streams. Although these rudimentary methods were able to produce some results, it is evident that such general tools luck a unique perspective and insights into the particularities of the wearable device data.

The method proposed in this thesis (Chapter 2) is built from the ground up, in a principled manner, taking into account the nature and relevant assumptions regarding the data. After an initial representation stage, the problem is framed as a topic modeling over motor words, allowing a plethora of existing methods to be used in conjunction with the idiosyncratic representation phase. This is exploited in the second paper in Chapter 2, where the Latent Dirichlet Allocation (LDA) topic modeling method is used following an online codebook generation.

The wearable device sensor unsupervised analysis with topic models developed in Chapter 2 is used both for analysis of data from a large migrating bird (White Stork, *Ciconia ciconia*), and to help gain insights in a pilot study examining the behavior and mental state of patients in a closed ward mental institution. Together, this demonstrates the general applicability of the method when considering sensor data of this sort.

In the first study [7, 8], up to several years of data was collected from migrating White Storks (*Ciconia ciconia*), using mounted bio-loggers fitted with GPS and accelerometers. In addition, field observations allowed to annotate a small portion (several thousand) of the accelerometer readings with behavioral modes (for example: Active and Passive Flight, Walking, Standing, Sitting). Results show that using the topic-modeling unsupervised approach we are able to recover to a large extent the labels that are obtained from a supervised learning based labeling. Thus, this method could be applied when field observations are not possible, as discussed above.

We point out that utilizing this method would allow for a wide spread use of behavioral mode analysis in the field of Movement Ecology. The importance of this issue is ever-growing, with huge amounts of rich sensor tracking data becoming increasingly easy to collect, whereas annotation and observation remain illusive for the most part. Standardization, consolidations, and enrichment of the general purpose toolbox in this field are thus paramount to the successful utilization of these data to their full potential.

In the second study in Chapter 2, 27 inpatients from the closed wards at Shaar-Meashe MHC fitted with a smart-watch (GeneActiv[1]) with tri-axial accelerometer embedded sensors, the high frequency output ($50Hz$) of which was stored on memory cards. Data was collected continuously throughout the experiment for a total of 489 days. Subpopulations of patients were defined on the basis of clinical assessment scores, and general behavioral features were calculated on the basis of an LDA topic model, computed over motor-words.

Results indicate that clinically interesting sub-populations can be distinguished on the basis of the motor-behavioral features. This result holds much promise for the use of smart watches or other wearable devices in a clinical setting for monitoring of patients. In a follow-up study (Appendix B) we further demonstrate that anomaly detection over the time-series of such motor-behavioral features may shed light on abrupt changes in the mental state of patients, as shown by tracking change in medication regime and ongoing clinical assessments.

[1] https://www.geneactiv.org/

Ideally, such a monitoring system would transmit real-time continuous sensor data to be analyzed in the cloud. The psychiatrist would then have a report available, summarizing the behavior of the patient over time, as a decision support system. Additionally, specific alerts would be pushed when an abrupt change in behavior is detected, which is believed to correlate to a change in mental state of an adverse reaction to medication (for instance, a sharp increase or decrease in mobility following change in medication regime).

In the next chapter (Chapter 3) we turn our gaze toward the location data obtained routinely by bio-loggers mounted on animals in the field of Movement Ecology. Unlike tracking of humans which often relies heavily on the the known underlying structure of the physical world (i.e. roads, buildings, coffee-shops etc.), when tracking wild animals one must rely to a much larger extent on the structure in the GPS (or other movement data) itself. Indeed, it is an interesting problem to infer real-world place semantics from raw movement data (see for instance [11]).

Trajectory segmentation is the process of subdividing a trajectory into parts either by grouping points similar with respect to some measure of interest, or by minimizing a global objective function. The aim of the trajectory analysis method developed in Chapter 3 is to provide a smart segmentation and summary, based on point density along the trajectory, and based on the nature of the naturally occurring structure of intermittent bouts of locomotive and local activity. We show an application to visualization of trajectory datasets, and discuss the use of the summary as an index allowing efficient queries which are otherwise impossible or computationally expensive, over very large datasets [2].

In Chapter 4 more general methods are developed. While these draw insight from, and are originally motivated by the problems revolving around analysis of sensor data from wearable devices, they are more general in scope and applicable in a wide range of applications and tasks (and as such should be regarded as basic research in the field of statistics, machine learning, or data science).

The first problem addressed is that of data imputation [5]. More often than not, and especially in high dimensional real-world data, some feature values are missing. Since most data analysis and statistical methods are not graceful in handling missing values, the first step in the analysis requires imputation. Indeed, many methods have been developed for the imputation of missing values as a pre-processing step, with research into the implications for downstream analysis.

One recent and effective approach, the IRMI stepwise regression imputation method [9], uses a linear regression model for each real-valued feature on the basis of all other features in the dataset. However, the proposed iterative formulation lacks convergence guarantee. In this paper we propose a closely related method, stated as a single optimization problem and a block coordinate-descent solution which is guaranteed to converge to a local minimum.

Experiments show results on both synthetic and benchmark datasets, which

are comparable to the results of the IRMI method whenever it converges. However, while in the set of experiments described here IRMI often does not converge, the performance of our methods is shown to be markedly superior in comparison with other methods. We conclude that especially for automatic analytics engines, where guaranteed convergence is paramount, our method should be preferred.

The second problem we address is a fundamental problem in multi-class classification with specific implication for supervised behavioral mode classification in the field of Movement Ecology. Suppose we are annotating a dataset with behavioral modes such as Active and Passive Flight, Standing and Sitting etc., and derive a confusion matrix summarizing the performance of our system (as in [6]). It is highly probable, depending on the research questions, that confusing some behavior (say Standing) with some specific other behavior (say Passive Flight) is far more detrimental than with another (say Sitting). This asymmetry in the actual loss is not reflected in the natural formulation of general purpose classifiers.

In this work [4, 3] we concentrate on deep learning based classifiers. Deep learning has become the method of choice in many application domains of machine learning in recent years, especially for multi-class classification tasks. The most common loss function used in this context is the categorical cross entropy loss, which reduces to the log loss in the typical case when there is a single correct response label.

We present the bilinear-loss (and related log-bilinear-loss) which differentially penalizes the different wrong assignments of the model. We thoroughly test this method using standard models and benchmark image datasets. As one application, we show the ability of this method to better contain error within the correct super-class, in the hierarchically labeled CIFAR100 dataset, without affecting the overall performance of the classifier.

In summary, this plug-in addition to the regular loss function is able to encode the differential sensitivity toward wrongly labeling examples, based on the true class and the wrong assignment. As such, this method has a wide range of applications, way beyond Movement Ecology and classification of behavioral modes. Indeed, in almost any real-word system this fine-grained determination of the loss, many be of help to practitioners, in order to better tailor the general suit of deep learning methods to the problem at hand.

Several directions for future research arise from this study, in the respective sub-fields which are touched upon in each of the three chapters. In the context of unsupervised analysis of behavioral modes, there is much work still to be done, for instance in order to extend the current methodology to include external information such as location type (field, water, residential etc.), as well as weather and other local information which give behavior along a trajectory extra meaning, and could help derive deeper insights into the drivers of behavior.

In the field of medical devices, much work is needed in order to bring the insights we already know are accessible in the data to physicians in order to promote a better understanding of mental and physical state of patients, and the development thereof following change of medication, and as time

passes. Our pilot study already shows there is a lot of available information that could potentially be used both on an ongoing basis, and to troubleshoot specific issues as they arise.

In conclusion, this study promoted our understanding of the correspondence between movement on the sub-second level as measured by IMUs, and the underlying higher-level behavioral modes, and all the way up to cognitive state. In addition, general methods were developed to solve specific problems with a more general applicability throughout the data fields.

References

[1] Ran Nathan et al. "Using tri-axial acceleration data to identify behavioral modes of free-ranging animals: general concepts and tools illustrated for griffon vultures". In: *Journal of Experimental Biology* 215.6 (2012), pp. 986–996.

[2] Yehezkel S Resheff. "Online trajectory segmentation and summary with applications to visualization and retrieval". In: *Big Data (Big Data), 2016 IEEE International Conference on*. IEEE. 2016, pp. 1832–1840.

[3] Yehezkel S Resheff, Amit Mandelbaum, and Daphna Weinshall. "Every Untrue Label is Untrue in its Own Way: Controlling Error Type with the Log Bilinear Loss". In: *arXiv preprint arXiv:1704.06062* (2017).

[4] Yehezkel S Resheff, Amit Mandelbom, and Daphna Weinshall. "Controlling Imbalanced Error in Deep Learning with the Log Bilinear Loss". In: *First International Workshop on Learning with Imbalanced Domains: Theory and Applications*. 2017, pp. 141–151.

[5] Yehezkel S. Resheff and Daphna Weinshall. "Optimized Linear Imputation". In: *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods, ICPRAM 2017, Porto, Portugal, February 24-26, 2017*. 2017, pp. 17–25. DOI: 10.5220/0006092900170025. URL: https://doi.org/10.5220/0006092900170025.

[6] Yehezkel S Resheff et al. "AcceleRater: a web application for supervised learning of behavioral modes from acceleration measurements". In: *Movement ecology* 2.1 (2014), p. 27.

[7] Yehezkel S Resheff et al. "Matrix factorization approach to behavioral mode analysis from acceleration data". In: *Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on*. IEEE. 2015, pp. 1–6.

[8] Yehezkel S Resheff et al. "Topic modeling of behavioral modes using sensor data". In: *International Journal of Data Science and Analytics* 1.1 (2016), pp. 51–60.

[9] Matthias Templ, Alexander Kowarik, and Peter Filzmoser. "Iterative stepwise regression imputation using standard and robust methods". In: *Computational Statistics & Data Analysis* 55.10 (2011), pp. 2793–2806.

[10] Avishai Wagner, Naama Fixler, and Yehezkel S Resheff. "A wavelet-based approach to monitoring Parkinson's disease symptoms". In: *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE. 2017, pp. 5980–5984.

[11] Yu Zheng et al. "Mining interesting locations and travel sequences from GPS trajectories". In: *Proceedings of the 18th international conference on World wide web*. ACM. 2009, pp. 791–800.

Appendix A

AcceleRater: a web-service for Supervised Learning in Movement Ecology

This paper [1] was published in the journal *Movement Ecology*, December 2014.

References

[1] Yehezkel S Resheff et al. "AcceleRater: a web application for supervised learning of behavioral modes from acceleration measurements". In: *Movement ecology* 2.1 (2014), p. 27.

Resheff et al. Movement Ecology (2014) 2:27
DOI 10.1186/s40462-014-0027-0

MOVEMENT ECOLOGY

SOFTWARE ARTICLE Open Access

AcceleRater: a web application for supervised learning of behavioral modes from acceleration measurements

Yehezkel S Resheff[1,2*], Shay Rotics[1], Roi Harel[1], Orr Spiegel[1,3] and Ran Nathan[1]

Abstract

Background: The study of animal movement is experiencing rapid progress in recent years, forcefully driven by technological advancement. Biologgers with Acceleration (ACC) recordings are becoming increasingly popular in the fields of animal behavior and movement ecology, for estimating energy expenditure and identifying behavior, with prospects for other potential uses as well. Supervised learning of behavioral modes from acceleration data has shown promising results in many species, and for a diverse range of behaviors. However, broad implementation of this technique in movement ecology research has been limited due to technical difficulties and complicated analysis, deterring many practitioners from applying this approach. This highlights the need to develop a broadly applicable tool for classifying behavior from acceleration data.

Description: Here we present a free-access python-based web application called AcceleRater, for rapidly training, visualizing and using models for supervised learning of behavioral modes from ACC measurements. We introduce AcceleRater, and illustrate its successful application for classifying vulture behavioral modes from acceleration data obtained from free-ranging vultures. The seven models offered in the AcceleRater application achieved overall accuracy of between 77.68% (Decision Tree) and 84.84% (Artificial Neural Network), with a mean overall accuracy of 81.51% and standard deviation of 3.95%. Notably, variation in performance was larger between behavioral modes than between models.

Conclusions: AcceleRater provides the means to identify animal behavior, offering a user-friendly tool for ACC-based behavioral annotation, which will be dynamically upgraded and maintained.

Keywords: AcceleRater, Animal behavior, Biologging, Classification, Ethology, Movement ecology, Supervised learning, Tri-axial acceleration, Web application

Background

Movement ecology aims to unify organismal movement research and to aid in the development of a general theory of whole-organism movements [1]. The field has recently experienced a period of rapid growth in knowledge and insights [2], triggered by the advent of movement tracking tools and GPS devices in particular [3], as well as various methods of analyzing movement patterns [4]. These advances have motivated the development of integrative conceptual frameworks unifying cognitive, biomechanical, random and optimality paradigms to study movements of all kinds by all types of organisms [1]. Nevertheless, movement data, however accurate, are unlikely to suffice for inference on the links between behavioral, ecological, physiological, and evolutionary processes driving the movement of individuals, which have traditionally been studied in isolation in each of the movement research paradigms. Thus, promoting movement ecology research and the desirable unification across species and movement phenomena requires developing additional sensors and

* Correspondence: Yehezkel.resheff@mail.huji.ac.il
[1]Movement Ecology Laboratory, Department of Ecology, Evolution and Behavior, Alexander Silberman Institute of Life Sciences, The Hebrew University of Jerusalem, Jerusalem, Israel
[2]Edmond and Lily Safra Center for Brain Sciences, The Hebrew University, Jerusalem 91904, Israel
Full list of author information is available at the end of the article

tools providing simultaneous information about the movement, energy expenditure and behavior of the focal organisms, and the environmental conditions they encounter *en route* [5].

To help bridge this gap, accelerometers were introduced as a means of identifying moment-to-moment behavioral modes [6] and estimating energy expenditure [7] of tagged animals. These sensors record body acceleration either in short bouts or continuously, along one, two or three orthogonal axes. Their output is used to infer behavior, most commonly through supervised machine learning techniques, and energy expenditure using the Overall Dynamic Body Acceleration (ODBA) or related metrics [7,8]. Combined with GPS recordings, acceleration sensors add fine scale information on the variation in animal's behavior and energy expenditure in space and time (see [9] for a recent review). ACC-based analysis allows us to compute many measures of interest, including behavior-specific body posture, movement and activity budgets, measures of foraging effort, attempted food capture events, mortality detection, classifying behavioral modes and more [9]. These measures have facilitated movement-related research for a wide range of topics in ecology and animal behavior [5,9-11] as well as other fields of research such as animal conservation and welfare [10,12] and biomechanics [13,14].

An ACC dataset typically consists of anywhere between tens of thousands to millions of records, together with a small subset of hundreds or thousands of records corresponding to field observations which have known behavioral modes attached to them. A variety of machine learning algorithms have recently been applied for ACC-based supervised learning of behavioral modes [5,15-20]. These methods require a calibration set for ground-truthing, which associates behavioral classes to ACC measurements, by time-matching behavioral observations of tagged individuals to the recorded ACC. This calibration set is generally collected through field observations of free-ranging animals [5,9], but can also be obtained by observing animals in captivity [9,21]. In principle, the calibration dataset can also be generated from a biomechanical model, by generating the acceleration patterns expected in each behavioral mode using a model of an animal, though we are not aware of a published example of this alternative option. The entire calibration set, or its sub-set (called training set, see *cross validation* below), is used to learn how to classify ACC measurements to behavioral classes. An underlying assumption here is that during each measurement, the animal is engaged in a single behavioral mode. To assess classification performance, measures like accuracy, precision and recall are calculated, as illustrated in the *Results* section below. Typically, the calibration set constitutes only a (very) small sample of the recorded dataset; hence, in the final stage of ACC-based behavioral analysis, the classifier is used to assign behavioral modes to the whole dataset which may span the lifetime of many animals.

ACC-based behavioral data can inform "what" the study animal is doing in addition to the more conventional data on "where" the animal is located, acquired by the GPS units. However, in spite of this and the above-mentioned advantages of ACC data, many ecologists do not utilize this option even when they have acceleration sensors in their tracking devices. In part, this is due to the fact that some elusive species are very difficult to observe in order to obtain the above mentioned calibration set. However, in many other cases we believe that the computational procedures, and the technical challenges involved, deter researchers from using ACC-based behavioral data.

AcceleRater was developed to provide a user-friendly free-access tool for choosing, validating and using models for supervised learning of behavioral modes from ACC data. We hope that this tool will encourage the use of ACC-behavioral data with the promising insights it can provide.

Implementation

AcceleRater is a python-based web application, using the sci-kit learn library [22] for fitting models and for most pre-processing operations. AcceleRater aims to facilitate broad use of ACC-based behavioral classification by including detailed explanations, a variety of models, model reconstruction options, alternative tests, and informative outputs, and by allowing the user to control many aspects of the processing, while setting typical values as default options.

Input data format

AcceleRater requires the user to prepare the input data file in advance. Although the package can be designed to obtain data directly from default output formats of some commercially-available ACC loggers, supervised methods require coupling ACC records with observed behaviors, necessitating some processing of the default ACC file in any case. In addition, accelerometers provide hardware-unit-specific measurements which require calibration for each tag, thereby typically requiring another pre-processing stage. Furthermore, the raw ACC data can be measured along one, two or three axes, and some devices provide some summary statistics rather than the raw data (see Additional file 1: Table S3 in supplementary material). To accommodate these needs and varieties, the user first indicates some basic attributes of the input dataset, including contents (summary statistics or raw data), and, for raw data files, the number

of axes (1, 2 or 3) for which ACC data was measured. For any selection, the user is offered several input file structures, all should be formatted as comma separated values (csv) files, with ACC measurements in rows, and behavior labels in the last column. Example data files can be found on the *demo* page of the application website.

The computing and feature selection protocol

1. Selecting and calculating summary statistics: For input files with raw ACC data, the user needs to select summary statistics to be calculated from the raw data. The list of summary statistics currently implemented in the program is given in Additional file 2: Table S1 (supplementary material). Additional statistics will be added upon user requests.
2. Processing summary statistics: The program calculates and then normalizes (to zero mean and unit standard deviation) all summary statistics selected in step (1).
3. Selecting the cross validation method: Cross-validation methods [23] separate the calibration dataset to training and validation subsets, the former is used to build the model, and the latter enables the user to quantify how well the calibrated model matches independent observations. We offer three options for performing validation: (a) *k*-fold cross-validation, the dataset is randomly split into *k* equal-size parts, *k*-1 parts are used for training and 1 for validation. The procedure is repeated *k* times until all parts have been used for validation; (b) a special case of (a), with $k = 2$, known as train-split method. This is the fastest and most commonly used option, taken here as the default; (c) another special case of (a), known as Leave-One-Out method, with $k = n$ where *n* is the number of labeled samples available. For large *n*, this option is computationally expensive, as well as unnecessary; hence the use of this option should be limited to rather small datasets (currently hundreds of samples).
4. Selecting and computing the models, and presentation of the results: the user selects one or more classifiers, listed in Table 1 and briefly outlined in (Additional file 3: Table S2. Once the selection is completed, the normalized statistics are fed into the chosen classifiers. Then, the cross-validation and some other results are displayed in the form of summary tables, confusion matrices, and accuracy, recall and precision tables (see examples in *Results* section below).
5. Using the calibrated model to label new data, see *"Labeling new data"* below.

Using the application

The minimal requirement is to upload the labeled (ground-truthed) ACC data file and run the program with default selection of its various options. Alternatively, the user can select the summary statistics, the cross validation method and the models.

Table 1 A list of classification models currently implemented in AcceleRater, with representative published applications for classifying animal behavior

Model	Sources
Artificial Neural Network (ANN)	[5]
Decision tree	[5]
Linear support vector machine (L-SVM)	[5]
Linear/Quadratic Discriminant Analysis (LDA/QDA)	[5,16,17]
Nearest neighbors	[19]
Radial basis function kernel for support vector machine (RBF-SVM)	This paper
Random forest	[5,16,20]

Main features

Manual - the manual contains an extensive documentation of the application, and should be referred to for further information.

Upload form - The "gateway" to the application. See *Input data format* above.

Models view - Here the models are summarized. This view contains:

- A page for each model with a confusion matrix in both graphical and tabular form, as well as overall accuracy and recall/precision/accuracy tables.
- A graph comparing the overall accuracy for each of the models
- A precision-recall graph comparing the models.
- A table containing the specific accuracy/recall/precision for every behavior in each model. This may be important when some of the behaviors are of more significance for the purpose at hand, and it is therefore desirable to select a model that does best on these behaviors.

Labeling new data – Beyond its use for assessing the feasibility and reliability of ACC-based behavioral classification for a given dataset, arguably the main purpose of using AcceleRater is to annotate (label) a large set of ACC recordings for which behavioral information is not available. The user should upload a file for annotation in an acceptable format (see *Input data format* above). The output csv file is the same as the input file, with an added last column providing the assigned behavioral labels.

Annotating a trajectory on a map – To visualize a trajectory of an animal on a map, annotated with the ACC-based behavioral labels, the program allows the user to upload a raw data file with both location (e.g. from GPS) and ACC data. The trajectory is then shown

on a Google Map with different colors indicating different behaviors. Currently, the program supports raw data file format of only one manufacturer (E-Obs GmbH; Munich, Germany), but other formats will be implemented upon users' requests.

Results

To test AcceleRater, we used ACC data collected by E-Obs transmitters on Griffon Vultures (*Gyps fulvus*). Acceleration was measured at 10Hz per axis and segments corresponding to single behavioral modes were obtained by field observations. For more details on this dataset see Refs. [5] and [11]. We used a dataset comprising of 488 samples and 6 behavior classes: Lying down (3.5%), Standing (43.6%), Walking (13.7%), Eating (22.3%), Soaring (6.6%), Flapping (10.2%). Typical acceleration signatures of the different behaviors are shown in Figure 1.

The main variation in the overall accuracy (Table 2), and in specific accuracy, precision and recall of assignment in the cross validation tests was attributed to different behaviors rather than different models (Additional file 4: Table S4, Figure 2). The specific accuracy of assignment to a particular behavior – the probability of a sample in the test-set to be assigned correctly to the specific behavior (True Positive; TP) or to another behavior (True Negative; TN) – was on average 91-94% for each model and 90-97% for each behavior across models (Additional file 4: Table S4b). The precision of assignment – the probability that an assigned behavior in the test-set is indeed this particular behavior – was medium to high (78-85%) for the different models, very high (92%) for Standing, high (80-86%) for both flying types and lower (59-75%) for the other three behaviors (Additional file 4: Table S4c). The recall – the probability that a sample with a particular behavior in the test-set will be correctly classified as this behavior – was relatively high (77-85%) for the different models, extremely high (95%) on average for Standing (the most common behavior in the training set), medium (80%) for Soaring and for Eating and lower (51-66%) for Walking, Flapping and Lying down (Additional file 4: Table S4d). These results are effectively summarized by the Precision-Recall plot (Figure 2). Note that overall accuracy, recall and precision of the ANN model were slightly better compared to other models (Table 2 & Additional file 4: Table S4), but in general all models preformed reasonably well (Table 2).

Discussion

The use of accelerometers in movement ecology has become popular in recent years, partly due to improvements in the underlying technologies and the advent of analysis tools [5]. Nevertheless, the non-trivial process of supervised learning of behavioral modes from acceleration data has hindered much more widespread use of this technique. Towards this end, we developed AcceleRater as a specialized web application for rapidly training, visualizing and using models for supervised learning of behavior modes from ACC measurements.

AccleRater was tested with 488 ACC segments collected by GPS-ACC transmitters (E-Obs GmbH; Munich,

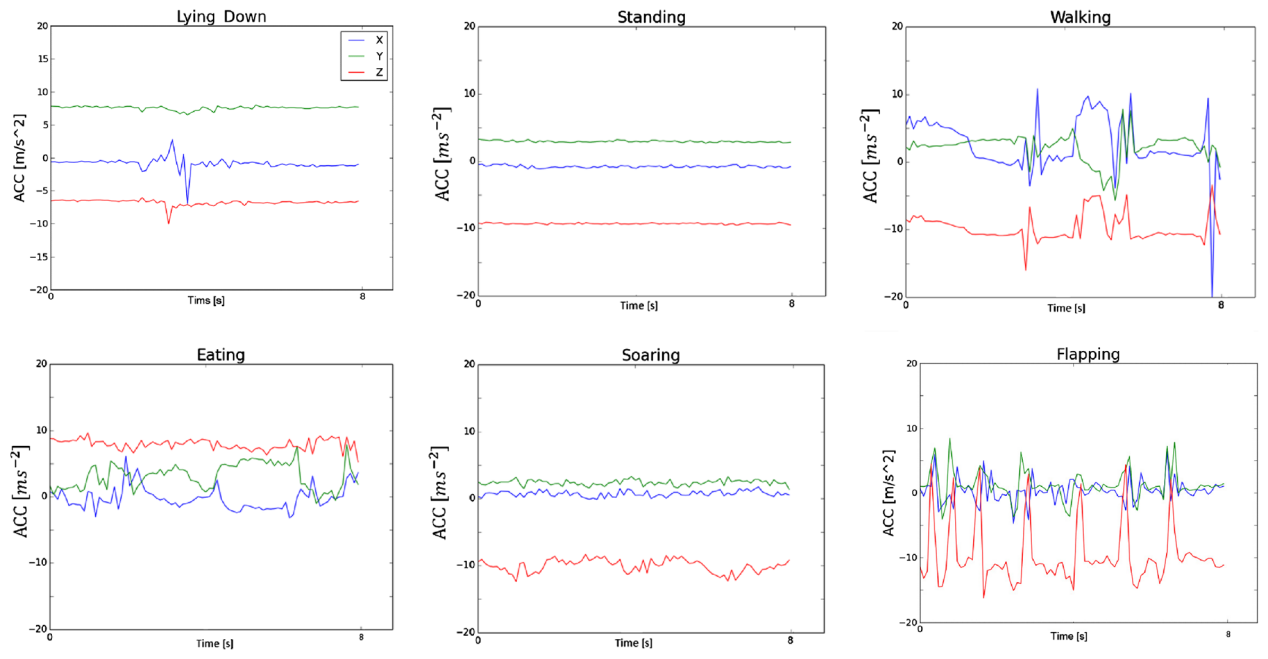


Figure 1 Representative acceleration plots for the six different behavioral modes obtained by AcceleRater from the vulture dataset. Each plot represents a single behavioral segment. Acceleration was sampled at 10Hz per axis.

Table 2 Model accuracy

Model name	% correct	Std
ANN	84.84	2.76
Decision tree	77.68	5.76
LDA	80.75	4.89
Linear SVM	80.13	4.18
Nearest neighbors	80.54	3.18
Random forest	84.02	2.98
RBF SVM	82.58	3.91
Mean	**81.51**	**3.95**

Standard deviation computed using a 10-fold cross validation procedure.

Germany) on Griffon Vultures (*Gyps fulvus*). We ran stratified random selection on a roughly twofold larger dataset [5] to reduce over-dominance of commonly observed behaviors. For this dataset, we found that model selection is a less critical consideration, compared to highly variable results for different behaviors. This might complicate analyses requiring reliable classification of many behaviors, whereas studies focusing a single or few behaviors could choose the best fitted model for their study system. AcceleRater yielded comparable results to those we previously reported for this dataset [5], extending our previous analysis by including additional models (RBF-SVM) and more informative output (e.g., precision and recall, rather than only accuracy). Most importantly, whereas previous contributions from our group as well as others [5,11,9,15,20] have provided guidelines for such analyses, AcceleRater practically implements and extends these guidelines, making this technique available for a broad range of users. It allows a thorough analysis that can be carried out quickly and effectively, yielding informative results within minutes.

Usage considerations

The online nature of the application requires transfer of data files over the internet. This inherently limits the size of the data files to be labeled. When labeling a large dataset with this application, the data should be broken down into manageable size parts, with ≤100,000 rows each.

Future work

The supervised learning framework is based upon observations being sampled from the distribution of the process in question. This sample, however, might not adequately reflect the true distribution of these behaviors throughout the time frame relevant to the research question, due to practical constraints of field observations, for example. Consequently, behavioral modes that are rare in the observation sample, and as such discarded or have weak classifiers, may in fact be more common and/or more influential for the study system. This concern motivates refinement of field observations on the one hand, and development of data-driven methods for unsupervised learning of behavior modes from ACC data on the other hand.

The segmentation of movement tracks has been identified as one of the greatest methodological challenges in movement ecology research [1]. By providing behavioral information highly relevant for distinguishing different movement phases, ACC-based behavioral classification can facilitate addressing this challenge [20]. AcceleRater

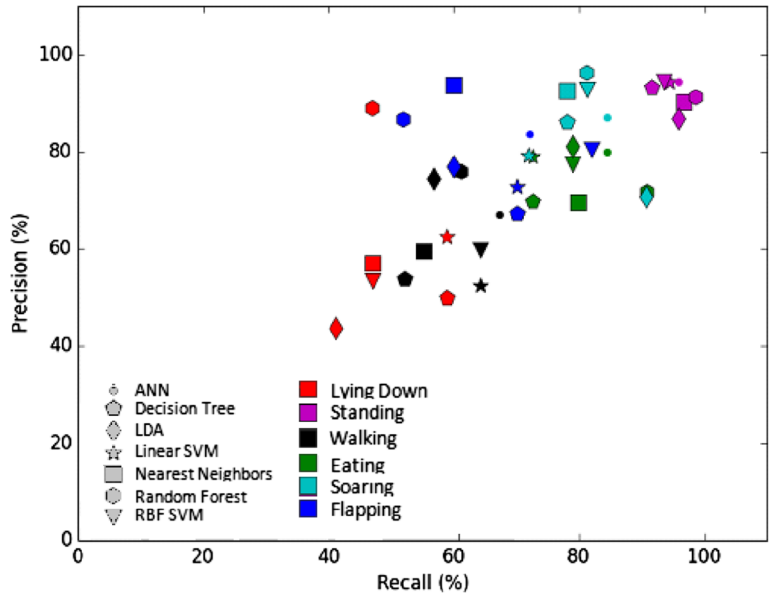


Figure 2 Precision-recall plot generated by accelerater for the vulture dataset (see Additional file 4: Table S4).

can therefore be extended to suggest segmentation pattern for movement tracks based on behavioral classification.

A key limitation of AcceleRater, like other web applications, is the need to upload and download large data files for labeling after a model is trained and chosen. This limitation might prohibit the use of the application on large datasets, with many millions of data points. We plan to address this limitation in future versions by allowing the user to select a model using the web application, and then download a stand-alone program configured to classify new data using the selected model offline, on the user's computer.

Conclusions

We present here a new tool, AcceleRater, allowing fast and intuitive tool for ACC-based behavioral classification, designed to be both flexible and general, with user-friendly interface and informative results displayed in tables and graphs. We demonstrate high performance of this tool in classifying behaviors of free-ranging birds. We encourage broad use and foresee further developments of AcceleRater for advancing more informative analysis of the ecology and behavior of animals in the wild.

Availability and requirements

Project name: AcceleRater.
Project home page: http://accapp.move-ecol-minerva.huji.ac.il/.
Operating system(s): Platform independent.
Programming language: Python, JavaScript.
License: The program was developed by YR and owned by the Minerva Center for Movement Ecology. We encourage its free use, no permission or license is required. The current paper should be cited in resulting publications.
Any restrictions to use by non-academics: none.

Additional files

Additional file 1: Table S3. Examples of ACC tag manufacturers and the type of supported output.

Additional file 2: Table S1. The statistics computed by the application.

Additional file 3: Table S2. The classification models currently (April 2014) implemented in AcceleRater. For updates, please visit the web site of the Minerva Center for Movement Ecology (http://accapp.move-ecol-minerva.huji.ac.il/).

Additional file 4: Table S4. Recall, Precision and Accuracy for each of the models and behaviors.

Abbreviations

ACC: Acceleration; ANN: Artificial neural network; ODBA: Overall dynamic body acceleration; RBF-SVM: Radial basis function SVM; SVM: Support vector machine.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

RN initiated the project. RN, SR, RH and OS contributed from their knowledge and experience in movement ecology. RH and OR collected the vulture data. YR designed and implemented the system, and wrote the first draft of this paper. All authors read and approved the final manuscript.

Acknowledgements

The software was developed by funds available from the Minerva Foundation and the Hebrew University of Jerusalem to the Minerva Center for Movement Ecology. The vulture dataset analyzed here was available from a project funded by the U.S.–Israel Bi-national Science Foundation (BSF 255/2008) and the special BSF Multiplier Grant Award from the Rosalinde and Arthur Gilbert Foundation to RN and Wayne Getz. RN also acknowledge funding from the Adelina and Massimo Della Pergola Chair of Life Sciences. Finally, we are grateful to the three anonymous reviewers for their helpful suggestions, and to Roland Kays and Gil Bohrer for inviting this contribution to the Animal Movement and the Environment (AMoveE) Symposium held in North Carolina on May 2014.

Author details

[1]Movement Ecology Laboratory, Department of Ecology, Evolution and Behavior, Alexander Silberman Institute of Life Sciences, The Hebrew University of Jerusalem, Jerusalem, Israel. [2]Edmond and Lily Safra Center for Brain Sciences, The Hebrew University, Jerusalem 91904, Israel. [3]Present address: Department of Environmental Science & Policy, University of California at Davis, Davis, CA 95616, USA.

Received: 8 September 2014 Accepted: 15 December 2014
Published online: 25 December 2014

References

1. Nathan R, Getz W: **A movement ecology paradigm for unifying organismal movement research.** *Proc Natl Acad Sci U S A* 2008, **105:**19052–9.
2. Holyoak M, Casagrandi R, Nathan R, Revilla E, Spiegel O: **Trends and missing parts in the study of movement ecology.** *Proc Natl Acad Sci U S A* 2008, **105:**19060–5.
3. Hebblewhite M, Haydon DT: **Distinguishing technology from biology: a critical review of the use of GPS telemetry data in ecology.** *Philos Trans R Soc Lond B Biol Sci* 2010, **365:**2303–12.
4. Smouse PE, Focardi S, Moorcroft PR, Kie JG, Forester JD, Morales JM: **Stochastic modelling of animal movement.** *Philos Trans R Soc Lond B Biol Sci* 2010, **365:**2201–11.
5. Nathan R, Spiegel O, Fortmann-Roe S, Harel R, Wikelski M, Getz WM: **Using tri-axial acceleration data to identify behavioral modes of free-ranging animals: general concepts and tools illustrated for griffon vultures.** *J Exp Biol* 2012, **215:**986–96.
6. Yoda K, Sato K, Niizuma Y, Kurita M, Naito Y: **Precise monitoring of porpoising behavior of Adelie Penguins.** *J Exp Biol* 1999, **3126**:3121–3126
7. Wilson RP, White CR, Quintana F, Halsey LG, Liebsch N, Martin GR, Butler PJ: **Moving towards acceleration for estimates of activity-specific metabolic rate in free-living animals: the case of the cormorant.** *J Anim Ecol* 2006, **75:**1081–90.
8. Gleiss AC, Wilson RP, Shepard ELC: **Making overall dynamic body acceleration work: on the theory of acceleration as a proxy for energy expenditure.** *Meth Ecol Evol* 2011, **2:**23–33.
9. Brown DD, Kays R, Wikelski M, Wilson R, Klimley AP: **Observing the unwatchable through acceleration logging of animal behavior.** *Animal Biotelemetry* 2013, **1:**20.
10. Takahashi M, Tobey JR, Pisacane CB, Andrus CH: **Evaluating the utility of an accelerometer and urinary hormone analysis as indicators of estrus in a Zoo-housed koala (*Phascolarctos cinereus*).** *Zoo Biol* 2009, **28:**59–68.
11. Spiegel O, Harel R, Getz WM, Nathan R: **Mixed strategies of griffon vultures' (*Gyps fulvus*) response to food deprivation lead to a hump-shaped movement pattern.** *Mov Ecol* 2013, **1:**5.
12. Cooke S: **Biotelemetry and biologging in endangered species research and animal conservation: relevance to regional, national, and IUCN Red List threat assessments.** *Endangered Species Res* 2008, **4:**165–85.
13. Hindle A, Rosen D, Trites A: **Swimming depth and ocean currents affect transit costs in Steller sea lions *Eumetopias jubatus*.** *Aquat Biol* 2010, **10:**139–48.

14. Sellers WI, Crompton RH: **Automatic monitoring of primate locomotor behaviour using accelerometers.** *Folia Primatol; Int J Primatol* 2004, **75:**279–93.
15. Wilson R, Shepard E, Liebsch N: **Prying into the intimate details of animal lives: use of a daily diary on animals.** *Endangered Species Res* 2008, **4:**123–37.
16. Watanabe S, Izawa M, Kato A, Ropert-Coudert Y, Naito Y: **A new technique for monitoring the detailed behaviour of terrestrial animals: a case study with the domestic cat.** *Appl Animal Behav Sci* 2005, **94:**117–31.
17. Soltis J, Wilson R, Douglas-Hamilton I, Vollrath F, King L, Savage A: **Accelerometers in collars identify behavioral states in captive African elephants *Loxodonta africana*.** *Endangered Species Res* 2012, **18:**255–63.
18. Signer C, Ruf T, Schober F, Fluch G, Paumann T, Arnold W: **A versatile telemetry system for continuous measurement of heart rate, body temperature and locomotor activity in free-ranging ruminants.** *Meth Eco Evol Br Ecol Soc* 2010, **1:**75–85.
19. Bidder OR, Campbell HA, Gómez-Laich A, Urgé P, Walker J, Cai Y, Gao L, Quintana F, Wilson RP: **Love thy neighbour: automatic animal behavioural classification of acceleration data using the K-nearest neighbour algorithm.** *PLoS One* 2014, **9:**e88609.
20. Bom RA, Bouten W, Piersma T, Oosterbeek K, Van Gils JA: **Optimizing acceleration-based ethograms: the use of variable-time versus fixed-time segmentation.** *Mov Ecol* 2014, **2:**6.
21. Rothwell ES, Bercovitch FB, Andrews JRM, Anderson MJ: **Estimating daily walking distance of captive African elephants using an accelerometer.** *Zoo Biol* 2011, **30:**579–91.
22. Pedregosa F, Varoquaux G: **Scikit-learn: Machine learning in Python.** *J Mach Learn Res* 2011, **12:**2825–30.
23. Arlot S, Celisse A: **A survey of cross-validation procedures for model selection.** *Stat Surv* 2010, **4:**40–79.

Appendix B

Studies in Wearable Devices for Mental Health

The following papers are part of a collaboration with Talia Tron, Mikkail Bazmin, Abraham Peld, as well as my advisor Daphna Weinshall. While not a core part of the thesis, these publications contain research using some of the tools developed in the previous papers, and demonstrate the success of the Shaar Menashe project that was driven to a large extent by these tools.

Real-time Schizophrenia Monitoring using Wearable Motion Sensitive Devices

Talia Tron[1], Yehezkel S. Resheff[1], Mikhail Bazhmin[2], Abraham Peled[2,3], and Daphna Weinshall[4]

[1] The Edmond and Lily Safra center (ELSC) for Brain Science, Hebrew University of Jerusalem (HUJI), Israel
[2] Sha'ar Menashe Mental Health Center, Hadera Israel
[3] Rappaport Faculty of Medicine, Technion Institute of Technology, Haifa Israel
[4] The Rachel and Selim Benin School of Computer Science and Engineering, HUJI

Summary. Motor peculiarity is an integral part of the schizophrenia disorder, having various manifestations both throughout the phases of the disease, and as a response to treatment. The current subjective non-quantitative evaluation of these traits leads to multiple interpretations of phenomenology, which impairs the reliability and validity of psychiatric diagnosis. Our long-term objective is to quantitatively measure motor behavior in schizophrenia patients, and develop automatic tools and methods for patient monitoring and treatment adjustment. In the present study, wearable devices were distributed among 25 inpatients in the closed wards of a Mental Health Center. Motor activity was measured using embedded accelerometers, as well as light and temperature sensors. The devices were worn continuously by participants throughout the duration of the experiment, approximately one month. During this period participants were also clinically evaluated twice weekly, including patients' mental, motor, and neurological symptom severity. Medication regimes and outstanding events were also recorded by hospital staff. Below we discuss the general framework for monitoring psychiatric patients with wearable devices. We then present results showing significant correlations between features of activity in various daily time-windows, and measures derived from the psychiatrist's clinical assessment or abnormal events in the patients' routine.

1 Introduction

The relevant clinical literature describes a wide range of motor pattern alternations, manifested in different phases of the schizophrenia disorder. Positive-signs schizophrenia patients are typically psychotic and disorganized, characterized mainly by positive symptoms (e.g. auditory hallucinations, delusions and paranoid thoughts). In clinical settings, these patients show involuntary movements, dyskinesia and catatonic symptoms [9]. In negative-signs schizophrenia, there is usually an observed motor retardation, psycho-motor

poverty, decreased spontaneous movements, psycho-motor slowing and flattened affect [8, 14]. Some patients demonstrate both types simultaneously or during different phases of the illness.

Neurological Soft Symptoms (NSS) can manifest early and during the progression of the disorder, and include deficits in coordination, sensory integration, and sequential motor behaviors [1]. Medical treatment was found to improve some of the motor symptoms, including NSS, involuntary movement and dyskinesia [9]. These medications, however, may also introduce in chronic patients drug-induced movement disorders such as tremor dystonia, Parkinsonism (rigidity and bradykinesia), akathisia and tardive dyskinesia [5].

The diversity and specificity of motor symptoms throughout different phases of the disorder and as a response to drugs, makes them good candidates for patient monitoring and treatment outcome evaluation. Nonetheless, to date, these symptoms are evaluated in a descriptive non etiological manner based on subjective clinical scales such as the Unified Dyskinesia Rating Scale (UDysRS) [3] and the Unified Parkinson's Disease Rating Scale (MDS-UPDRS) [4]. The lack of objective, quantitative methods to measure these symptoms, and the insufficient conceptual clarity around it, causes multiple interpretations of phenomenology, often entailing low reliability and validity of the diagnosis. In addition, symptom evaluation process requires expert staff and availability of resources, and it is not done frequently enough to capture delicate changes in patients' spontaneous and drug-induced conditions.

The last decade has seen a steep rise in the use of wearable devices in medical fields ranging from human physiology [10] to movement disorders [7, 11] and mental health [13]. Accelerometers and gyroscopes, which are commonly embedded in smart-watches and other wearable devices, are now used to assess mobility, recognize activity, and context. In a clinical setting, these sensors may be used in order to detect change in high-level movement parameters, track their dynamics and correlate them with mental state.

The objective of the current study is to develop and evaluate a framework, where wearable devices are used to facilitate continuous motor deficits monitoring in schizophrenia patients in a natural setting. This is an important step towards a detailed automatic evaluation system of symptom severity in schizophrenia. Such a system has a great potential to help understand this illusive disease. An additional goal would be to help with the overwhelming need for detection and characterization of sub-types of the disease towards a better understanding of underlying causes, and the development of better and more personalized treatment.

2 Methods

2.1 Participants and clinical evaluation

Twenty seven inpatients from the closed wards at Shaar-Meashe MHC participated in the study after signing appropriate Helsinki legal consents. Most

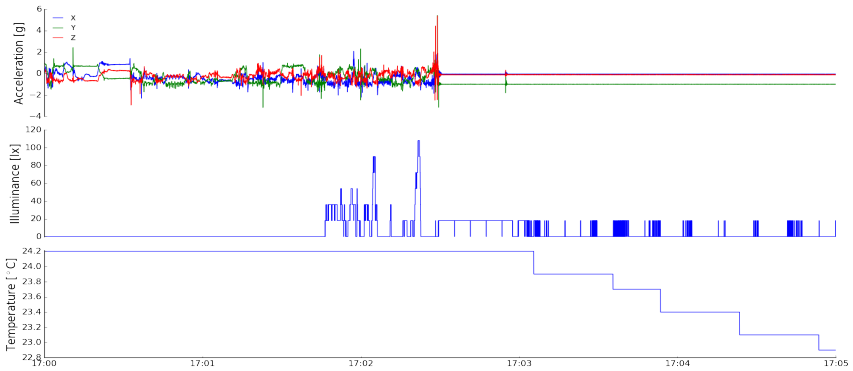


Fig. 1. Raw data as recorded by the smart-watches, including tri-axial accelerometer (top panel), light sensor (middle), and temperature (bottom). This plot shows data from a single patient, recorded on 28 Jan, 2017 at 5:00-5:05pm.

participants (21/27) were diagnosed with schizophrenia according to the DSM-5, 3 with paranoid schizophrenia, 2 with schizoaffective disorder, and one with psychotic state cannabinoids. Participants' age varied from 21 to 58 (mean of 37.48), with course of illness varying from 0 (first hospitalization) up to 37 years (mean of 16.9 years). Two of the patients dropped out of the study after less than a day due to lack of cooperation. The rest (25 patients) were followed for a period of three weeks on average (6-52 days).

The study was conducted in natural settings, where patients were *not* required to change any personal or medical procedure. In addition to routine reports by nurses and physicians, every patient underwent an additional evaluation by a trained psychiatrist twice a week. The procedure included medication monitoring (type, dosage and frequency), as well as clinical evaluation of positive and negative symptom severity (PANSS [6]) and neurological symptoms severity (NES [2]).

All procedures performed in the study were in accordance with the ethical standards of the institutional research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards.

2.2 Data Acquisition

At study onset, each participant was given a smart-watch (GeneActiv[5]). The devices included tri-axial accelerometers, light, and temperature sensors, the high frequency output ($50Hz$) of which was stored on memory cards embedded in the device (see Fig. 1). Data was collected by the aforementioned smart watches worn continuously by patients throughout the experiment (for a total of 489 days of data from 25 patients). The devices were placed and removed

[5] https://www.geneactiv.org/

by the medical staff, and the content of the memory card was uploaded to a central storage location upon termination of the experiment for further analysis.

In order to reduce noise introduced by the variability in patients activity which is due to external circumstances rather than mental state, weekends were excluded from the study and our analysis focused on fixed time windows with regular departmental daily activity. These included occupational therapy time slots (10am-11am), lunch (12pm-1pm), and indoor free time (4pm-5pm). In addition, we calculated full day features (6am-10pm) and used night time features (10pm-6am) to evaluate sleep quality.

2.3 Features

Features were computed on the basis of the accelerometer readings, analyzed in 1 minute windows (see Table 1 and Fig. 2). The point-wise sum of values and sum of square values of the tri-axial accelerometer measurements (Energy Square and Energy Sum respectively) were averaged over 1 minute intervals. The variance of the sum of squares (Energy Variance) was also computed over the same window. Stepping behavior (Step Detector) was detected as large maxima of the smoothed square norm of the point-wise acceleration. Overall Dynamic Body Acceleration (ODBA), a measure of energy expenditure, was computed as the mean norm of the signal after application of a high-pass filter.

Table 1. List of features calculated on the basis of the tri-axial Accelerometers. Average and variance was calculated on a 1 minute time window.

Feature	Description
Step Detector	Simple count of the number of steps per minute.
Energy Square	Averaged sum of point-wise square acceleration.
Energy Sum	Averaged sum of point-wise acceleration.
Energy Variance	Variance of point-wise square acceleration
ODBA	mean norm of a high-passed version of the signal.

2.4 Clinical Assessments

The 30-item scale for positive and negative symptom assessment (PANSS) was reduced to the follwoing 5 literature-based factors: Positive, Negative, Disorganized/Concrete, Excited and Depressed. The dimensionality reduction was done according to the consensus model suggested by Wallwork et al. [12], based on 25 previously published models and refined with confirmatory factor analysis (CFA).

The negative and positive factors had low between-factor correlation ($R = 0.399$), indicating good separation of the symptomatology space. As expected,

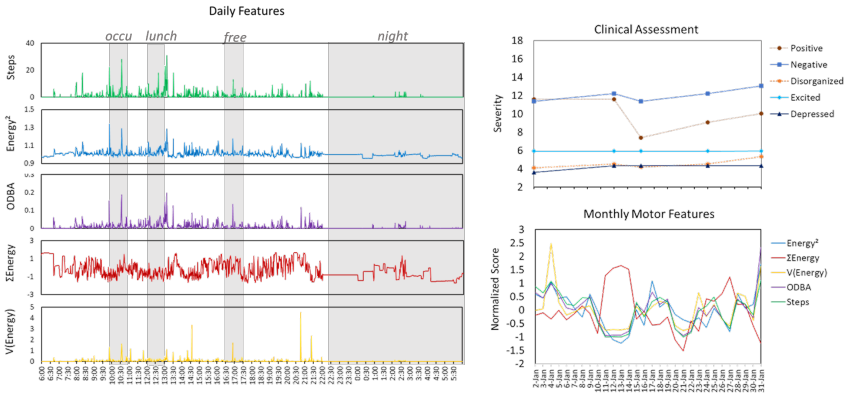


Fig. 2. The daily features of a single subject (left): gray areas indicate the time windows used for aggregated feature calculation. Monthly follow-up of a single patient (right): top panel shows the clinical five-factor PANSS score given by a trained psychiatrist on a bi-weekly basis; bottom panel shows the aggregated features calculated based on the different time windows.

the positive factor was in high correlation with the mean of all positive PANSS items ($R = .944$), and likewise the negative factor was in high correlation with the mean of all negative PANSS items ($R = .972$).

3 Results

We investigated two distinct ways by which wearable devices can be used for patient monitoring, in order to assist physicians in understanding the state of a patient. The first aspect of monitoring relates to the automatic assessment of a patient's condition, in order to provide automated, continuous, and objective measures of mental state. To this end we investigated the correlation between the computed measures and assessments by physicians, as described in Section 3.1. The second aspect of monitoring relates to the detection of change (or anomalous behavior patterns) which warrants additional attention from the medical staff, as described in Section 3.2.

3.1 Movement patterns and mental state

In order to investigate the correspondence between patterns of movement and mental state, *multiple correlation analysis* was computed between activity related features (described in Section 2.2) and PANSS factors. Results (Table 2) indicate the predictive benefit of the computed activity-related features with respect to the PANSS factors. When separately considering features computed

Table 2. Percent Explained Variance based on *Multiple Correlation* between computed features in each of the 5 time-windows and each of the 5 PANSS factors. (See Section 2.2 for time-window specifications.)

	free	lunch	occu	day	night	all
Positive Factor	16.30%	11.14%	12.31%	19.80%	5.21%	53.77%
Negative Factor	19.74%	3.15%	2.06%	18.36%	9.77%	55.50%
Disorganized_Concrete Factor	22.73%	0.50%	15.13%	13.42%	5.82%	64.81%
Excited Factor	23.79%	8.75%	15.08%	10.35%	12.70%	57.10%
Depressed Factor	31.01%	9.23%	8.94%	5.78%	6.39%	58.33%

in each of the time-windows, it is evident that different time windows provide varying predictive value for the 5 different PANSS factors.

Specifically, the Depressed Factor is described relatively well using features from the *free time* window, with 31.01% explained variance, while all other time-windows are below 10%. Both Positive and Negative factors are described well using features from the *free time* as well as *all day* time-windows. The remaining factors are again best described using *free time*. Overall, the *free time* window is the single most effective window, presumably since it imposes less structure on the movement of the subjects, allowing for the manifestation of the underlying mental state. In all cases, combining all time windows (rightmost column in Table 2) leads to substantially higher explained variance, compared to any of the individual windows.

Interestingly, looking at individual variable correlations we see that step count during *free time* was positively correlated with positive, disorganized and exited factor ($R = 0.37, 0.37$ and 0.31 respectively), but not with the negative and depressed factors. In addition, patients who had higher scores in disorganized and exited factors tended to have lower Energy scores during occupational time ($R = -0.28$ for Energy Sum and -0.22 for Energy Variance). This may indicate some motor retardation which is manifested only in non-walking time.

3.2 Continuous Monitoring

Our measures can be used to track changes in the patient's condition as compared to some established normal baseline, and may identify external events which are correlated with the departure from normality. Fig. 3 demonstrates such a case: daily *step counts* of a patient dramatically increased 5-fold, at the same time as a significant change in medication dosage was introduced. Whether the change in medication *caused* the rise in movement propensity or they were both triggered by a change in mental state, this observation points to the relevance of monitoring macro movement patterns as part of routine patient monitoring.

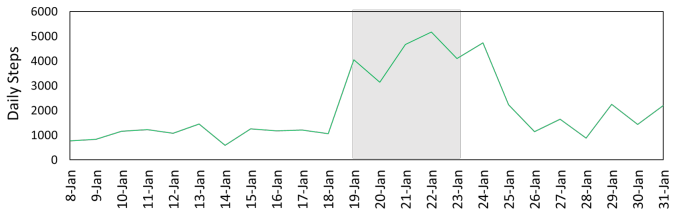


Fig. 3. Mean *daily steps* of a single subject. The gray area corresponds to a short-lasting change in medication regime.

4 Conclusions

We describe a study designed to evaluate the utility of wearable devices fitted with accelerometer, light, and temperature sensors, for the monitoring of schizophrenia patients in a closed ward mental health institution. Initial results show correlations between features of activity in various daily time-windows, and factors derived from the PANSS assessment.

Results indicate that movement features during free time are the most indicative of mental state. This finding is somewhat counter-intuitive, since the more structured activity during occupational therapy or lunch was expected to highlight differences in the state of patients. However, our results clearly show that the behavior of individuals when left to their own devices is better correlated with the PANSS factors.

These findings points to the possibility of automatically and continuously tracking Schizophrenia related symptoms and patient state, in a natural setting hospital environment.The benefits of such a tracking system are twofold; first, the continuous tracking will assist physicians in understanding the state of a patient on an on-going basis, as opposed to specific points in time, when assessed by the doctor. Second, long term monitoring of a large number of patients will produce data which will allow us to develop more objective measures of the motor aspects of the illness, and facilitate a more personalized, objective, and data driven approach which is much needed in the field of mental health.

Future work will focus on measuring the utility of this approach as an augmentation tool from a physicians perspective on the one hand, and the ability to predict physician assessments for automation of diagnosis on the other.

References

1. Bombin, I., Arango, C., Buchanan, R.W.: Significance and meaning of neurological signs in schizophrenia: two decades later. Schizophrenia Bulletin 31(4), 962–977 (2005)

2. Buchanan, R.W., Heinrichs, D.W.: The neurological evaluation scale (nes): a structured instrument for the assessment of neurological signs in schizophrenia. Psychiatry research 27(3), 335–350 (1989)
3. Goetz, C.G., Nutt, J.G., Stebbins, G.T.: The unified dyskinesia rating scale: presentation and clinimetric profile. Movement Disorders 23(16), 2398–2403 (2008)
4. Goetz, C.G., Tilley, B.C., Shaftman, S.R., Stebbins, G.T., Fahn, S., Martinez-Martin, P., Poewe, W., Sampaio, C., Stern, M.B., Dodel, R., et al.: Movement disorder society-sponsored revision of the unified parkinson's disease rating scale (mds-updrs): Scale presentation and clinimetric testing results. Movement disorders 23(15), 2129–2170 (2008)
5. Janno, S., Holi, M., Tuisku, K., Wahlbeck, K.: Prevalence of neuroleptic-induced movement disorders in chronic schizophrenia inpatients. American Journal of Psychiatry 161(1), 160–163 (2004)
6. Kay, S.R., Flszbein, A., Opfer, L.A.: The positive and negative syndrome scale (panss) for schizophrenia. Schizophrenia bulletin 13(2), 261 (1987)
7. LeMoyne, R., Mastroianni, T., Cozza, M., Coroian, C., Grundfest, W.: Implementation of an iphone for characterizing parkinson's disease tremor through a wireless accelerometer application. In: Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE. pp. 4954–4958. IEEE (2010)
8. Morrens, M., Hulstijn, W., Sabbe, B.: Psychomotor slowing in schizophrenia. Schizophrenia bulletin 33(4), 1038–1053 (2007)
9. Peralta, V., Cuesta, M.J.: The effect of antipsychotic medication on neuromotor abnormalities in neuroleptic-naive nonaffective psychotic patients: a naturalistic study with haloperidol, risperidone, or olanzapine. Prim Care Companion J Clin Psychiatry 12(2), e1–11 (2010)
10. Staudenmayer, J., He, S., Hickey, A., Sasaki, J., Freedson, P.: Methods to estimate aspects of physical activity and sedentary behavior from high-frequency wrist accelerometer measurements. Journal of Applied Physiology 119(4), 396–403 (2015)
11. Wagner, A., Fixler, N., Resheff, Y.S.: A wavelet-based approach to moniotring parkinson's disease symptoms. International Conference on Acoustics, Speech, and Signal Processing (2017)
12. Wallwork, R., Fortgang, R., Hashimoto, R., Weinberger, D., Dickinson, D.: Searching for a consensus five-factor model of the positive and negative syndrome scale for schizophrenia. Schizophrenia research 137(1), 246–250 (2012)
13. Wang, R., Chen, F., Chen, Z., Li, T., Harari, G., Tignor, S., Zhou, X., Ben-Zeev, D., Campbell, A.T.: Studentlife: assessing mental health, academic performance and behavioral trends of college students using smartphones. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing. pp. 3–14. ACM (2014)
14. Wichniak, A., Skowerska, A., Chojnacka-Wójtowicz, J., Tafliński, T., Wierzbicka, A., Jernajczyk, W., Jarema, M.: Actigraphic monitoring of activity and rest in schizophrenic patients treated with olanzapine or risperidone. Journal of psychiatric research 45(10), 1381–1386 (2011)

ARIMA-based Motor Anomaly Detection in Schizophrenia Inpatients

Talia Tron[1,4] Yehezkel S. Resheff[1,4] Mikhail Bazhmin[2] Daphna Weinshall[4] Abraham Peled[2,3]

Abstract— **Motor alteration is an important aspect of the elusive schizophrenia disorder, manifested both throughout the various phases of the disease and as a response to treatment. Tracking of patients' movement, and especially in a closed ward hospital setting, can therefore shed light on the dynamics of the disease, and help alert staff to possible deterioration and adverse effects of medication. In this paper we describe the use of ARIMA-based anomaly detection for monitoring of patient motor activity in a closed ward hospital setting. We demonstrate the utility of the approach in several intriguing case studies.**

I. INTRODUCTION

Monitoring of motor behavior is part of the regular assessment of schizophrenia patients and is vital to diagnosis, progress assessment and to the monitoring of medication response. Various alterations of motor behavior are evident throughout the phases of the disease, and as a response to treatment. The psychotic acute phase of schizophrenia is typically accompanied by restlessness, including occasional bizarre movements and gestures, while post psychotic deficiency negative symptoms are related to reduced activity, slowness and even freezing. Antypsychotic medications may cause Parkinsonism, i.e., tremor, rigidity, and slowness, which usually pass after the first week of treatment.

Despite its clinical and diagnostic value, to date, motor monitoring is done in a descriptive non etiological manner based on subjective clinical scales, which may result in biased, inaccurate and typically non quantifiable assessments. This kind of assessment requires expert staff and the availability of resources, and may not be frequent enough to capture significant changes in spontaneous and drug-induced conditions. These issues can be alleviated by carrying out objective, continuous quantifiable monitoring [1], the investigation of which is the goal of this study. Accelerometers and gyroscopes, commonly embedded in smart-watches and other wearable devices, have been extensively used over the last decades in medical applications ranging from human physiology [2] to movement disorders [3] and mental healthcare [4]. These cheap and widely available sensors may be used for continuous qualitative patient monitoring in natural clinical settings. Accelerometer data have already been shown to effectively provide insights into patients clinical state, and motor features were successfully used for clinical sub-typing in a closed ward mental hospital setting [5], [6]. Here we focus on detecting acute abnormal which are either the result or the cause of drug modifications or changes in patients' clinical conditions. Our approach employs forecasting models widely used in statistics and econometrics, applied to step-count data. We demonstrate the utility of this approach with 4 schizophrenia case studies, in which we evaluate monitoring performance based on medical and clinical records.

[1]The Edmond and Lily Safra center for Brain Science, Hebrew University, Jerusalem 91904, Israel talia.tron@mail.huji.ac.il

[2]Rappaport Faculty of Medicine, Technion Institute of Technology, Haifa 3200003, Israel

[3]Sha'ar Menashe Mental Health Center, Sha'ar Menashe 38706, Israel

[4]The Rachel and Selim Benin School of Computer Science and Engineering, Hebrew University, Jerusalem 91904, Israel

II. MATERIALS AND METHODS

A. Study Design

Four inpatients from the closed ward at Shaar-Menashe mental health center, diagnosed with schizophrenia according to the DSM-5, participated in the study. One patient (patient B) was diagnosed with paranoid schizophrenia. Participants' age varied from 24 to 54 (average 36.9), with course of illness varying from 7 to 35 years (average of 13.5 years). After signing the appropriate Helsinki legal consents, participants were tracked for a period of approximately one month (27-31 days) in natural settings. During this period, patients were monitored for medication use (type, dosage, and frequency) by the nurses and the physicians. In addition, every patient underwent a clinical evaluation of Positive and Negative Syndrome Scale (PANSS [7]) and Neurological Evaluation Scale (NES [8]) by a trained psychiatrist twice a week. The neurological evaluation was only utilized to confirm that no psycho-motor deficits were evident in any of the participants during the experiment.

All procedures performed in the study were in accordance with the ethical standards of the institutional research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards.

B. Data Acquisition

At study onset, participants were given smart-watches with embedded accelerometers (GeneActiv[1]). These watches were worn on the wrist throughout the experiment. The output ($50Hz$) of the sensors was stored on internal memory cards. The study was conducted in natural settings, where patients were *not* required to change any personal or medical procedure. None of the patients expressed any discomfort or disturbance from wearing the device.

III. DATA ANALYSIS

A. Building personal ARIMA Models

Analysis focused on the walking pattern of patients, aiming to detect significant quantitative changes. Stepping

[1]https://www.activinsights.com/products/geneactiv/

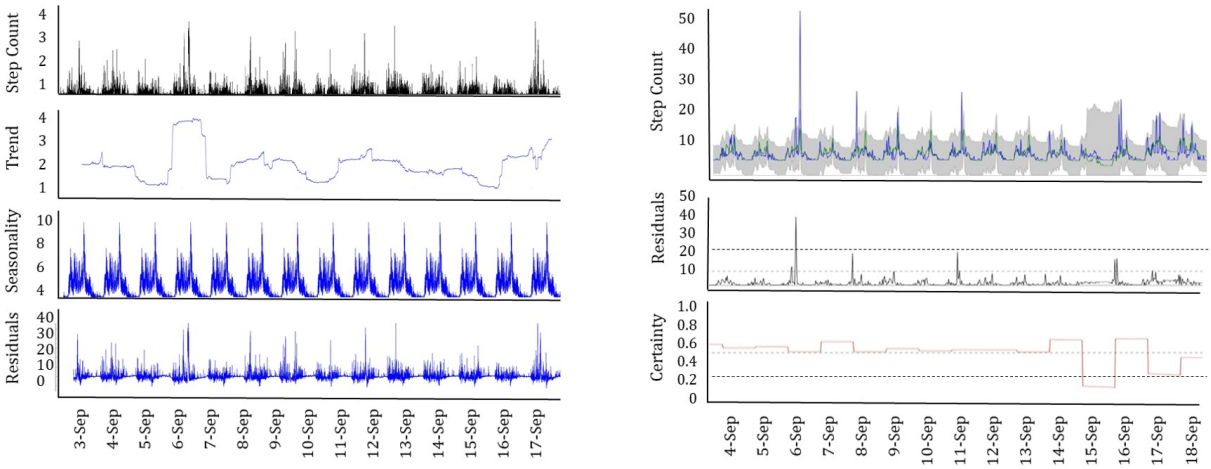


Fig. 1. **Left**- Decomposition of daily steps (top) of a single patient to trend (smoothed series calculated using centered moving average), seasonality (regularly repeating data patterns calculated as the average of the smoothed series for each period) and noise. **Right**- Demonstration of the ARIMA model for patient A. The model returns the predicted mean and a 95% confidence interval (CI) around it. Abnormal behavior is detected when (a) the observed step count value lies outside the CI predicted by the model, (b) the residuals are higher than threshold (e.g. September 6), or (c) when certainty is lower than threshold (e.g. September 15).

behavior was detected as large maxima of the smoothed square norm of the 3-axial $50Hz$ point-wise acceleration, and the number of steps (step count) was averaged over 1 minute intervals (see [5] for further details).

We used AutoRegressive Integrated Moving Average (ARIMA) models to detect abnormal walking patterns. One week of data was used to predict the step count for the following day, together with the associated confidence interval. Repeating this in a rolling window design produced the predicted step count for the entire duration of available data, around 3 weeks for each participant excluding the first week. Predicted values were then compared to those observed in practice for the purpose of anomaly detection [9].

We began by decomposing the step-count data into trend, seasonality and noise components, as shown in the left side of Fig. 1. As expected, strong daily seasonality was seen in the data. It is interesting to note that the trend component, to the extent that it exists, may potentially be used for direct real-time monitoring of patients symptom severity over time.

Next, we aggregated each patient's step-count data in windows of 10-30 minutes (this was done to smooth the data on the one hand, and reduce computation on the other). Both regular and daily seasonal differentiation were computed to obtain a stationary signal. We applied 4 different ARIMA models to all patients, and evaluated them using AIC criteria with mean and absolute errors. The emerging preferred model was ARIMAX(1,1,1) seasonal (1,1,2), which had a consistent lower error and lower AIC over all patients.

B. Abnormal behavior detection

For each patient separately, we ran an ARIMAX(1,1,1) seasonal (1,1,2) model, which was based on 7 days of data in order to predict the following day. The model provided the predicted mean and a 95% confidence interval (CI) around it. Model residuals were calculated as the squared difference between the model predicted values and the observed values during the test period.

A measure of prediction certainty was calculated based on the normalized CI size ($|CI_z|$) as follow:

$$Certainty = 0.95 \times 2 \times \frac{std(data)}{|CI_z|} \tag{1}$$

This is a measure of model confidence, with low values indicating that the model hasn't been able to accurately predict future values based on the patient's history. The multiplier of 0.95 sets the maximum certainty value to 0.95 (model confidence level). Although certainty is somewhat correlated with residuals size, this is an important independent measure. Specifically, it covers cases where the observed value is lower than the predicted value, which is not always expressed in CI range or high residuals.

Abnormal behavior is defined as one the following (see right side of Fig. 1): (a) The predicted value is not in the model CI; (b) the residuals between model prediction and observed values are higher than threshold (set to be 3 times the mean residuals on train data); (c) the $certainty$ of the model is lower than threshold (0.3). In order to avoid trailing errors and secure robustness, when abnormal behavior is detected, the observed values of the training period are replaced with predicted values. On repeated detections (more than twice) the model is adjusted back to observed values.

C. Evaluating model performance

In order to evaluate our model we systematically studied the patients clinical records and drug charts, and compared them with model anomaly detections. No clear abnormal event, such as an outburst of violence or riot, was recorded during the experiment period. We therefore used the PANSS clinical records in order to identify *abnormal* events, which are time stamps corresponding with a steep increase or

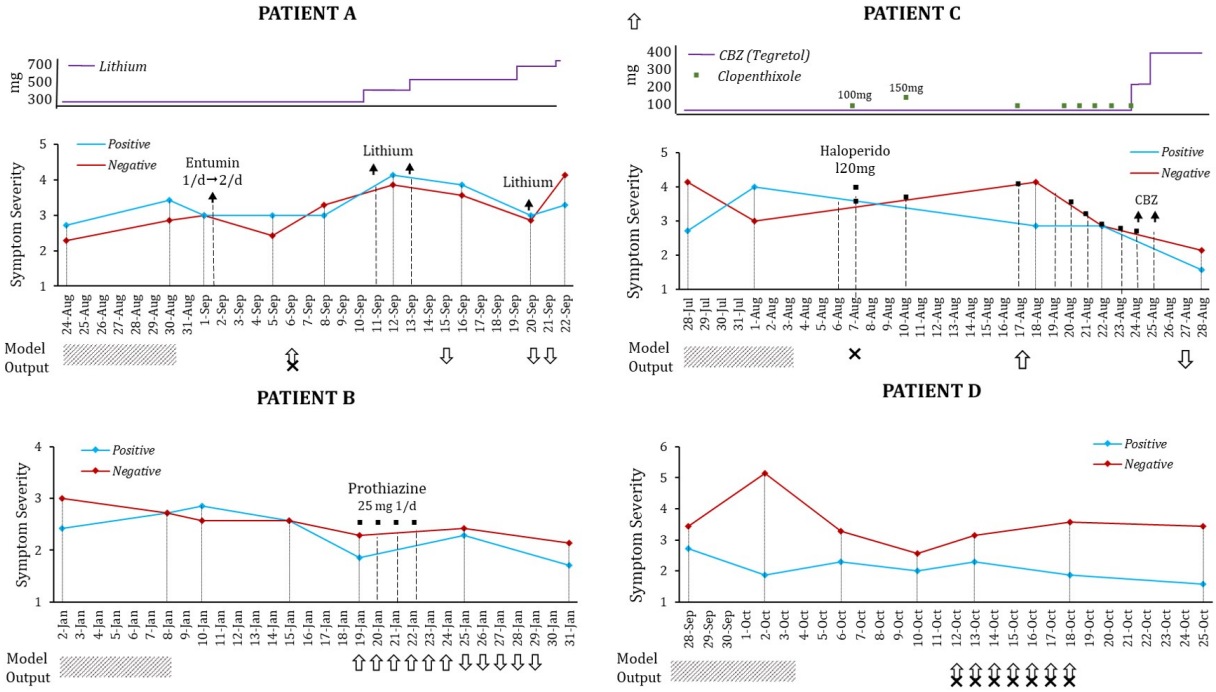


Fig. 2. Description of model prediction vs. clinical and medication records monitoring for all four patients. The direction of the white arrows in the bottom part of each graph indicates whether increased activity (up) or decreased activity (down) has been detected. A cross under the arrow indicates unexplained detection, while a cross without an arrow indicates an event that wasn't detected by the model. The dashed rectangle marks the training period of the model. In the line chart above, the mean severity of positive (red) and negative (blue) symptoms is shown. The black symbols indicate a change in drug dosage (arrow) or a single administration (square). In case of dosage change, the top graph (in patients A and C) indicates its amount (in mg).

decrease in symptom severity (more than 2 degrees on the PANSS scale) between two clinical sessions. Results are summarized in Fig. 2.

In an effort to capture some larger scale dynamics, we took note of the general positive and negative symptoms trend. Every change in drug dosage was also considered an abnormal event, since these changes are rare and usually indicate a change in a patient's clinical condition. It should be noted that increased drug dosage may be either a response to abnormal activity (when the detected event took place prior to drug adjustment) or its trigger (when the detected event followed a drug adjustment). Decreased dosage, on the other hand, is usually followed by continuous improvement in symptom severity, but may still cause side effects. Therefore, in order to obtain a coherent picture, both timing and the direction of the dosage change were taken into account.

For each abnormal event detected by our model, we looked for an explanation (as defined above) in the clinical records (drug dosage and PANSS scores); an event which did not have a satisfactory explanation, was labeled as 'unexplained'. Likewise, a drug change event or a steep change in the clinical evaluation data which was not detected by our model was labeled as 'undetected'. The number of unexplained and undetected events was used to roughly estimate the accuracy and sensitivity of our model. Events in consecutive days were counted as one continuous event.

1) Patient A: Abnormal increased walking behavior was detected on September 6th. On the same day, the dosage of *entumin* (a.k.a *clotiapine*), an atypical anti-psychotic drug, was increased from 40mg 1/day to 40mg 2/day.

On September 15th, and then again during September 20-22, our model detected lower than expected activity. In the clinical records, we see a significant increase in both positive and negative symptoms during September 5-12, with a steep rise in active social avoidance, hostility and social withdrawal. Possibly this behavioral change has resulted from the increased entumin dosage, although we cannot rule out other possible triggers.

Following this deterioration in the patient's condition, on September 11th the dosage of *lithium* was increased, and again on the 13th. Both positive and negative symptoms were reduced in subsequent days, with active social avoidance and hostility returning to normal values. We also see the emergence of increased negative symptoms, including blunted affect and passive apathetic social withdrawal.

Lithium is known to take effect within 1-3 weeks, so the lower activity found by our model during September 20-22 may be the result of the September 11th dosage increase. The September 15th detection remains unexplained by drug records but is congruent with clinical data.

In summary, 2/3 detected events for this patient had a co-found explanation in the clinical and medication records. One event had only a weak co-found in the clinical data. No clinical trend or drug changes remained undetected.

2) Patient B: The model detected a period of extreme increased activity during January 19-24, followed by decreased

activity during January 25-31. On January 19th, this patient was given *prothiazine*, a neuroleptic medication used as a sedative and weak anti-psychotic, for a period of 4 days. We found no significant change in symptom severity for this patient prior to the sedative drug administration, with only a small decrease in overall negative symptoms at that time. This is probably because clinical evaluation was not frequent enough to capture the change. The fact that our model detected this event while the clinical data did not, can be used as evidence for the potential benefit of continuous automated monitoring.

On January 25th, two days after the patient has stopped receiving the medication, we see a small improvement in his clinical condition with normal level of motor activity. In the model this is expressed by a detected 'lower than expected' activity, based on the increased activity in the previous days.

In summary, for this patient all detected events (2) had a co-found explanation in the medication records but no co-found (or a minor one) in the clinical records. No clinical trend or medication alteration remained undetected.

3) Patient C: Increased activity level was detected by the model on August 17th. Clinical data together with medical records clearly suggest that around this period there was an aggravation in the patient's condition. On August 17th, he was injected with 100mg of *clopenthixole acetate* (anti-psychotic and acute sedative medication), and once again in the following days (August 20-25). The drug's effect seems to have been dimmed unsatisfactory, since during August 24-25 the patient was also prescribed 200mg and then 400mg of *carbamazepine* (CBZ), an off label medication used in combination with anti-psychotics when the treatment with anti-psychotics alone has failed [10]. In the clinical data we see a decrease in both negative and positive symptoms severity around August 18-22, with a steep decrease in hallucinations, poor attention, and motor retardation. This improvement is most probably the result of the massive drug treatment. On August 27th, after the patients symptoms were reduced and drug treatment was stabilized, the model detected a significant reduction in patient's activity.

In August 7 the patient received two types of typical anti-psychotic medications (clopenthixole and *haloperidol*), and then again in August 10 (only *clopenthixole*). Since these drugs act on a short term basis, it is not probable that the the worsening in the patient's condition in subsequent days was triggered by this medication change. The most probable explanation is that there was some acute event at that time, which was not detected by our model.

In summary, all detected events (2) had a co-found explanation in the clinical and medication records, while one likely clinical event remained undetected.

4) Patient D: The model reported a period of decreased activity during October 12-18, with low certainty. No medication change was registered in this time period, and no substantial evidence was found in the clinical data (only a steep increase in stereotyped thinking). The overall trend of symptoms' change around that period leaned towards increased negative symptoms and reduced positive symptoms. This happened following approximately a week of steep decrease in negative symptoms.

In summary, the event detected by our model had no co-found explanation in the medication records. No clinical trend or medication alteration remained undetected.

TABLE I
SUMMARY OF ANOMALY DETECTION RESULTS AND PATIENTS' DATA.

	Days	Sessions	Explained	Missed
Patient A	31	10	2/3	0
Patient B	29	7	2/2	0
Patient C	31	11	2/2	1
Patient D	27	7	0/2	0

As summarized in Table I, when aggregating data from all patients, 6/8 anomaly events detected by our model had a co-found explanation in the medication and clinical records (precision of 75%). 6/7 events were detected by our model, with one certain mis-detection in patient C (recall of 85%).Other detected events may have alternative explanation not available to our experimental design.

IV. CONCLUSIONS

Our study demonstrates the benefits of using forecasting models in conjunction with accelerometer data for the continuous monitoring of schizophrenia patients. In three out of four case studies, we found a direct link between detected behavioral events and changes in the patient's clinical condition or drug regime.

REFERENCES

[1] Christopher G Goetz, John G Nutt, and Glenn T Stebbins. The unified dyskinesia rating scale: presentation and clinimetric profile. *Mov. Dis.*, 23(16):2398–2403, 2008.

[2] John Staudenmayer, Shai He, Amanda Hickey, Jeffer Sasaki, and Patty Freedson. Methods to estimate aspects of physical activity and sedentary behavior from high-frequency wrist accelerometer measurements. *Journal of Applied Physiology*, 119(4):396–403, 2015.

[3] Avishai Wagner, Naama Fixler, and Yehezkel S Resheff. A wavelet-based approach to moniotring parkinson's disease symptoms. *International Conference on Acoustics, Speech, and Signal Processing*, 2017.

[4] Rui Wang, Fanglin Chen, et al. Studentlife: assessing mental health, academic performance and behavioral trends of college students using smartphones. In *Proceedings of the 2014 ACM Int Joint Conf on Pervasive and Ubiquitous Computing*, pages 3–14. ACM, 2014.

[5] Talia Tron, Yehezkel S Resheff, Mikhail Bazhmin, Abraham Peled, and Daphna Weinshall. Real-time schizophrenia monitoring using wearable motion sensitive devices. 2016.

[6] Talia Tron, Yehezkel S Resheff, Mikhail Bazhmin, Alexander Grinshpoon, Abraham Peled, and Daphna Weinshall. Topic models for automated motor analysis in schizophrenia patients*. In *Body sensor networks (BSN), 2013 IEEE international conference on*. IEEE, 2017.

[7] Stanley R Kay, Abraham Flszbein, and Lewis A Opfer. The positive and negative syndrome scale (panss) for schizophrenia. *Schizo. bulletin*, 13(2):261, 1987.

[8] Robert W Buchanan and Douglas W Heinrichs. The neurological evaluation scale (nes): a structured instrument for the assessment of neurological signs in schizophrenia. *Psychiatry research*, 27(3):335–350, 1989.

[9] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[10] Daniela Ceron-Litvoc, Bernardo Garcia Soares, Geddes, et al. Comparison of carbamazepine and lithium in treatment of bipolar disorder: a systematic review of randomized controlled trials. *Human Psychopharmacology: Clinical and Experimental*, 24(1):19–28, 2009.

Appendix C

List of Publications in this Thesis

This thesis is submitted in the form of a compilation of published peer-reviewed papers. Table C.1 provides an overview of the papers appearing in this thesis. **The full bibliographic reference is given at the top of the chapter** before each of the papers in the respective chapters.

TABLE C.1: List of Publications in this Thesis

	Title	Year	Published In
1	AcceleRater: a web application for supervised learning of behavioral modes from acceleration measurements	2014	Movement Ecology
2	Topic modeling of behavioral modes using sensor data	2015	J. DSAA
3	Online Trajectory Segmentation and Summary With Applications to Visualization and Retrieval	2016	IEEE BigData
4	Optimized Linear Imputation	2017	ICPRAM†
5	Real-time Schizophrenia Monitoring using Wearable Motion Sensitive Devices	2017	MOBIHEALTH
6	Every Untrue Label is Untrue in its Own Way: Controlling Error Type with the Log Bilinear Loss	2017	ECML (LIDTA)
7	Topic Models for Automated Motor Analysis in Schizophrenia Patients	2018	BSN
8	Automated Schizophrenia Monitoring using Accelerometers	2018	BHI

† Extended version accepted (in production); LNCS journal

תקציר

גישת למידה חישובית לאנליזה של דאטה ביו-לוגרים באקולוגיה של תנועה

A Machine Learning Approach to Analysis of Biologger Data in Movement Ecology

מנחים:
דפנה ווינשל
רן נתן

פריצות דרך טכנולוגיות של העת האחרונה שיפרו במידה ניכרת את הזמינות של מחשוב לביש הכולל סנסורים רבים. שני תחומים נהנו במיוחד מפריחה זו: מעקב אחרי בעלי חיים ומכשור רפואי. עם כמויות דאטה הולכות וגדלות, יש צורך כעת בפיתוח שיטות להפיכת מידע לתובנות.
בעבודה זו חקרתי ופיתחתי שיטות למידה חישובית לאנליזה של דאטה ביו-לוגרים. הפרק הראשון דן בשיטות לא מונחות לניתוח דאטה אקסלרומטרים, כאשר הבעיה מנוסחת כטופיק מודלינג מעל ברסטים של סיגנל אקסלרומטר. השיטה מיושמת בשני התחומים הנדונים, עם תוצאות שממחישות את התועלת שבשימוש במכשור לביש הן להבנת בעלי חיים והן למעקב אחרי חולים.
הפרק השני דן בסוג הדאטה הנפוץ ביותר שמגיע ממכשור לביש – מידע מיקום – ובו מפותחת שיטה לניתוח של כמויות מידע גדולות לצרכים של ויזואיזציה ואנליזה.
הפרק השלישי דן בשיטות כלליות שאמנם עולות בהקשר של אקולוגיה של תנועה, אבל בעלות השלכות רחבות למגוון תחומים של אנליזה של נתונים. הבעייה הראשונה הנדונה היא מילוי מידע חסר והשנייה בניית מסווגים עם פונקציות הפסד שלוקחות בחשבון הן את המחלקה הנכונה והן את המנובאת.

עבודה זו נעשתה בהדרכתם של
פרופסור דפנה ויינשל
פרופסור רן נתן

האוניברסיטה העברית

גישת למידה חישובית לאנליזה של דאטה ביו-לוגרים באקולוגיה של תנועה

חיבור לשם קבלת תואר דוקטור לפילוסופיה

מאת:

יחזקאל ש. רשף

הוגש לסנט האוניברסיטה העברית בירושלים

דצמבר 2018 טבת תשעח