Analyzing Auditory Neurons by Learning Distance Functions

Inna Weiner

School of Computer Science and Engineering

The Hebrew University of Jerusalem

Jerusalem 91904, Israel

A thesis submitted for the degree of

Master of Science

Supervised by Prof. Daphna Weinshall

September 2005

Acknowledgements

First of all, I would like to deeply thank my supervisor, prof. Daphna Weinshall for bringing me into the world of research. Thank you for giving me the opportunity to try and fall and finally find my own way.

I would like to thank prof. Israel Nelken, who was an initiator of the project that lead to this thesis. It was a great honor for me to work with such a man of vision, dedicated fully to his work, inspiring everyone in his environment to do the same.

I am specially thankful to my collaborator and another initiator of the project, Tomer Hertz. You are the model for getting things done. Not thought of, not started, not almost finished. Done. I am glad I had the opportunity to work with you.

I would also like to thank my fellow friends in HUJI Computer Vision Lab, especially Chen, Talya and Amit, for your encouragement and support.

I am more than thankful to students from prof. Nelken's lab - Yael Bitterman and Jonathan Rubin, for your ideas, time, patience and coffee when needed.

I am indebted to my family and friends for their support, understanding and love all along the way.

And last but not least, I would like to thank Sasha for his endless support and love.

Abstract

We present a novel approach to the characterization of complex sensory neurons. One of the main goals of characterizing sensory neurons is to characterize dimensions in stimulus space to which the neurons are highly sensitive (causing large gradients in the neural responses) or alternatively dimensions in stimulus space to which the neuronal response are invariant (defining iso-response manifolds). The dominant approach attempts to predict the response from a stimulus by learning a linear filter that imposes it. We propose a new problem definition as that of learning a geometry on stimulus space that is compatible with the neural responses: the distance between stimuli should be large when the responses they evoke are very different, and small when the responses they evoke are similar. Here we show how to successfully train such distance functions using rather limited amount of information.

The data consisted of the responses of neurons in the IC of anesthetized guinea pigs and in primary auditory cortex (A1) of anesthetized cats to 32-40 stimuli derived from natural sounds. A distance function was trained and tested using a cross-validation scheme. The resulting distance functions generalized to predict the distances between the responses of a test stimulus and the trained stimuli. Surprising differences in stimuli predictability were found between IC and A1 neurons: while IC neurons generalization was similar for all stimuli, A1 neurons showed better generalization for wide-band stimuli than for narrow-band stimuli.

Contents

Chapter 1

Introduction

A major challenge in auditory neuroscience is to understand how cortical neurons represent the acoustic environment. Cortical neural responses to complex sounds are idiosyncratic, and small perturbations in the stimuli may give rise to large changes in the responses. Furthermore, different neurons, even with similar frequency response areas, may respond very differently to the same set of stimuli. The dominant approach to the functional characterization of sensory neurons attempts to estimate a linear model, the spectrotemporal receptive field (STRF), using the responses to a set of test stimuli. The STRF is then used to predict neuronal responses to new stimuli. However, STRFs have been recently shown to have low predictive power for cortical neurons (18; 27).

Here we take a novel approach to the characterization of auditory neurons. Our approach attempts to learn the non-linear warping of stimulus space that is induced by the neuronal responses. This approach is motivated by the observation of Chechik et. al. (5) that different neurons impose different partitions of the stimulus space, which are not necessarily simply related to the spectro-temporal structure of the stimuli. More specifically, we characterize a neuron by learning a pairwise distance function over the stimulus domain that will be consistent with the similarities between the responses to different stimuli (see Chapter 4).

We consider a semi-supervised learning scenario, in which data is augmented by some sparse side-information in the form of equivalence constraints. Equivalence constraints are the natural way to define labels over *pairs* of data points: a pair of data points will have a positive constraint between them if they come

from the same class and a negative constraint between them if they come from different classes. Such constraints carry *less* information than explicit labels on the original data points, since equivalence constraints can be obtained from explicit labels but **not** vice versa. Distance learning algorithm will accept as an input a stimuli data set and a subset of equivalence constraints based on the cell's responses and computed off-line. It will compute a hypothesis that complies with the constraints and provides distance (or similarity) measure for any 2 stimuli. Intuitively a good distance function would assign small values to pairs of stimuli that elicit a similar neuronal response, and large values to pairs of stimuli that elicit different neuronal responses.

In recent years there has been a growing interest in employing constraints to learn an informative distance function. Most of the work in this area has focused on the learning of Mahalanobis distance functions of the form $(x-y)^T A(x-y)$ (30; 33). In these papers the parametric Mahalanobis metric was used in combination with some suitable parametric clustering algorithm, such as K-means or EM of a mixture of Gaussians. Several algorithms which incorporate unlabeled data into the boosting process have been suggested in (6; 13). In these algorithms, the incorporation of unlabeled points is achieved by extending the 'margin' concept to the unlabeled points. Several margin extensions were previously suggested, relating the margin of a hypothesis over an unlabeled point to the certainty of the hypothesis regarding the point's classification. The extended margins are then used in a 'MarginBoost' algorithm (19).

In this work we propose to use the *DistBoost* algorithm (14) to learn a cell-specific distance function over the stimuli space. This algorithm learns a non-parametric distance function and has shown its dominance over other algorithms' performance in different distance-learning problems (14; 34). However, the novel problem formulation is the main contribution of this work and clearly any distance-learning algorithm can be incorporated in the proposed scheme.

This approach has a number of potential advantages over the STRF approach. First, unlike most functional characterizations that are limited to linear or weakly non-linear models, distance learning can approximate functions that are highly non-linear. Second, it allows to aggregate information from a number of neurons, in order to learn a good distance function even when the number of stimuli in

the test set is rather small. Finally, given some distance function on stimulus space, it may be possible to determine the stimulus features that most strongly influence the responses of a cortical neuron by examining the properties of such a function.

In this thesis I focus on two questions:

- Can one learn distance functions over the stimulus domain for single cells using information extracted from responses collected during standard electrophysiological experiments?
- What is the predictive power of these cell specific distance functions when presented with novel stimuli?

In order to address these questions we used extracellular recordings from 28 cells in the Inferior Colliculus (IC) of guinea pigs and from 22 cells in the primary auditory cortex (A1) of cats in response to natural bird chirps and modified versions of these chirps (1). To estimate the distance between responses, we used a normalized distance measure between the peri-stimulus time histograms of the responses to the different stimuli. Distances between responses were used for two purposes: (1) obtain equivalence constraints for the distance learning algorithm, and (2) evaluate the performance of the algorithm by computing the correlation between response distances and learnt stimuli distances.

Our results show that it is possible to learn compatible distance functions on the stimulus domain with relatively low training errors. This result is interesting by itself as a possible characterization of cortical auditory neurons, a goal which eluded many previous studies (1; 5). Using cross validation, we measure the test error (or predictive power) of our method, and report generalization power which is significantly higher than previously reported for natural stimuli (18; 27). We then show that performance can be further improved by learning a distance function using information from pairs of related neurons. Finally, for cortical neurons, we show better generalization performance for wide-band stimuli as compared to narrow-band stimuli, while for IC neurons a less profound difference is found. These latter two contributions may have some interesting biological

implications regarding the nature of the computations done by auditory cortical neurons.

The thesis is organized as follows: Chapter 2 presents a general overview of the auditory system and a brief summary of recent work in the area. Chapter 3 provides an overview of computational methods and ideas used in this work. The biological subject of sensory neuron examination as a computational distance learning problem is formalized in Chapter 4. Experimental results are described in Chapter 5. A discussion of methods, results and future goals is presented in Chapter 6.

Chapter 2

Auditory System

2.1 Physiology overview

Processing of auditory information provides us one of the basic sensations of the world. Mammals have the ability to solve hard problems of auditory perception: using sounds for within- and across-species communication, extracting information in noisy background, and sound localization (23).

The accepted view of sensory systems considers neurons as feature detectors arranged in an anatomical and functional hierarchy. For example, in the visual system, information from simple feature detectors in the retina converges at the level of the primary visual cortex (V1) to extract more complex oriented features. The neurons that process these features send them in turn to higher cortical areas to give rise to even more complicate neuronal detectors. The auditory system is only partially consistent with such a view. Whereas visual information flows almost unchanged through the thalamus to the primary visual cortex, auditory information is intensely processed in the brainstem and the midbrain (see fig. 2.1) (21). Starting from the cochlear nucleus, multiple information streams can be identified by their anatomical sources and targets, by the cellular morphology of the participating neurons, or by their physiological properties (31). Even in the cochlear nucleus there are neurons with highly complex response functions (25). This extensive subcortical processing and the complexity of auditory stimuli

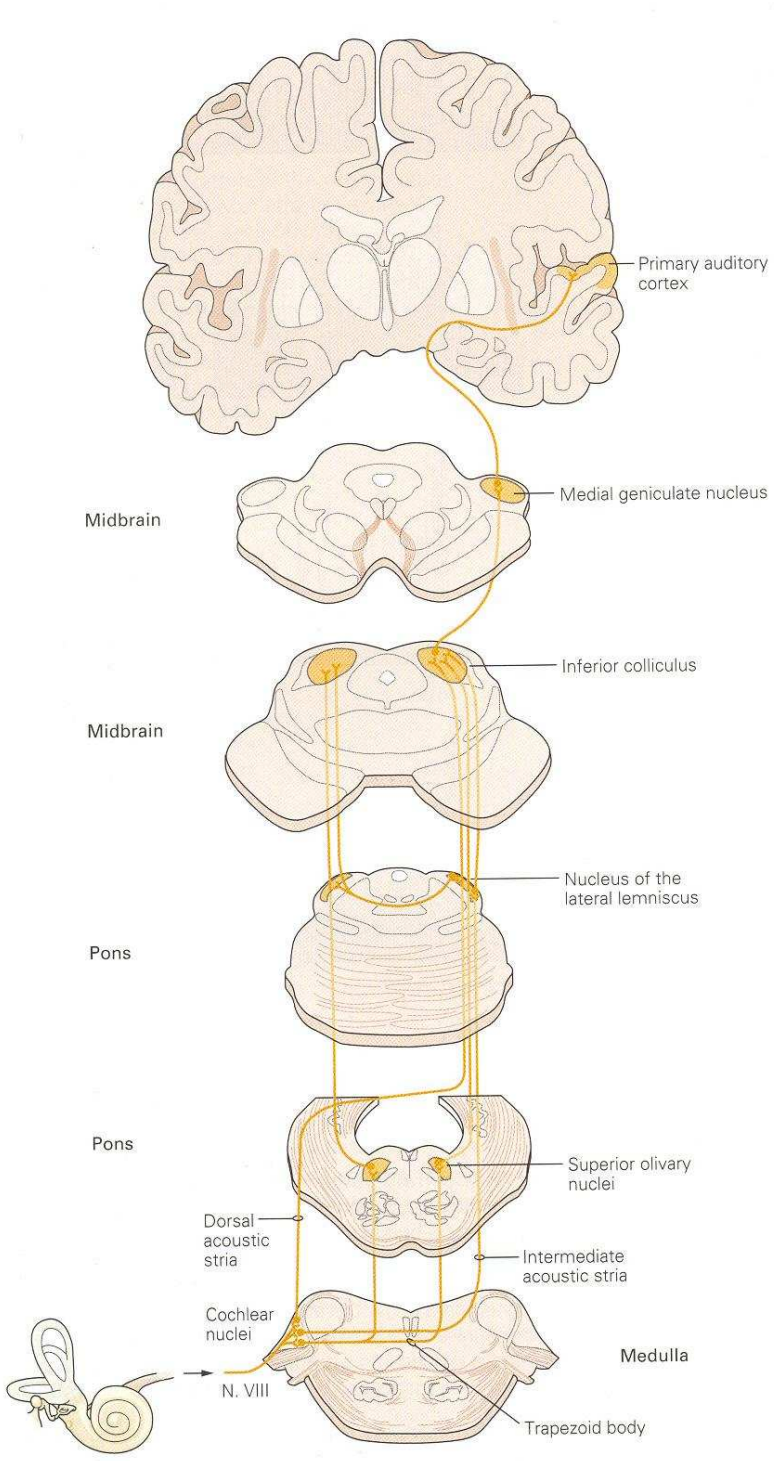

Figure 2.1: The primary auditory pathway from the cochlear nucleus to the auditory cortex. In our research we examined neurons from 2 stations in the auditory pathway: Inferior Colliculus (IC) and Primary Auditory Cortex (A1) (15)

are probably responsible for the difficulty of assigning a function to neurons in auditory cortex.

2.2 Transformation of stimulus representations in the ascending auditory system

The question of transformation of stimulus representations in the ascending auditory system is fundamental. While certain basic features, such as frequency sensitivity, are shared by auditory neurons of all stations, several studies suggest that dramatic changes in stimulus representation occur as information flows from IC through Medial Geniculate Nucleus (MGB) to A1. Perhaps the first difference between IC and higher stations appears at the basic characterization of an auditory neuron - its frequency response area (FRA). The FRA maps the neurons response properties on a two dimensional frequency-level space. Two fundamental characteristics of neurons in IC, MGB and A1 are their best frequency (BF; the frequency at which they have their lowest threshold) and their bandwidth. IC neurons exhibit narrow FRAs, while the FRAs characteristically become wider as we advance to the thalamus and the cortex. Another difference is the temporal resolution of the neurons: IC neurons typically follow repetitive stimuli up to about 100 Hz while typical A1 neurons follow this type of stimuli up to about 10 Hz and MGB neurons are intermediate. Such characterization suggest that neurons in A1 (and in MGB) are less sensitive to low-level features of auditory stimuli than neurons in the preceding station, the IC.

To a large extent, neurons in the IC behave roughly as linear filters, i.e. neurons with the BF essentially process the same signal and therefore are highly redundant (5). In contrast, neurons in A1 are highly non-linear; It has been shown that they are extremely sensitive to weak acoustic components, even in the presence of much stronger acoustic components (1; 17). In addition, this sensitivity is idiosyncratic: different neurons with overlapping frequency areas show very different response profiles to the same set of sounds. As a result, A1 neurons are much less redundant than IC neurons (5). MGB seems closer to A1 than to

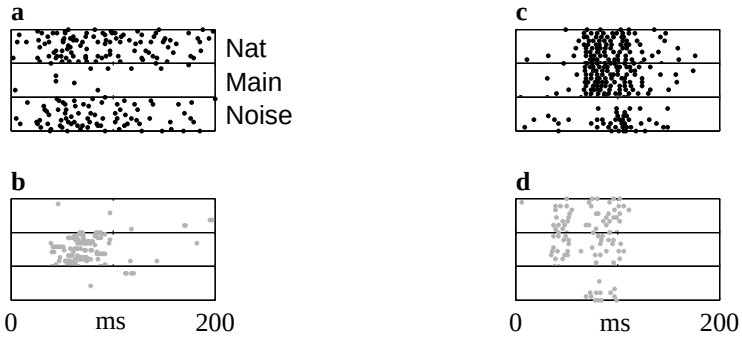

Figure 2.2: (a),(b): PSTHs of two A1 neurons with overlapping FRAs, when presented with Natural stimulus (higher row), Main chirp (middle row) and Noise (lower row). It can be easily seen that both neurons are highly non-linear to the stimuli (response to natural stimulus is more like the response to noise than the response to the main chirp for both neurons). The neurons are non-redundant between themselves: while the neuron in (a) fires extensively when presented with natural stimulus or noise and is almost silent when presented with the main chirp, the neuron in (b) behaves in the opposite way. (c),(d): Two IC neurons with overlapping FRAs. IC neurons' behavior is a linear filter of the stimuli (main and natural invoke almost the same response for both neurons, which is different from the response to noise) and redundant between themselves - firing patterns are similar for both neurons (23).

IC in these respects.

Changes in the neuronal encoding along the auditory pathway are illustrated in Nelken et. al. (24), where neuronal responses to a partial set of stimuli used in this work were collected at the three different stations: IC, MGB and A1. The attention was given to 3 variations of the stimuli: the natural stimulus (bird vocalization), the main dominant tonal component and the background weak noise component. Most of the acoustic energy of the natural stimulus is preserved in the main component, and only the remainder is in the background noise. Therefore the natural and the main variations could be considered more acoustically alike than the natural and the noise. Two pairs of simultaneously-recorded neuronal responses, one from the IC (right) and the second from A1 (left), are shown in fig. 2.2. Although both pairs had largely overlapping frequency response areas, which included the frequency range of the chirps, it is clear that while the two IC neurons had very similar response patterns, the two A1 neurons

responses were almost opposite. Moreover, the responses of the IC neurons appear to be a linear function of the stimuli: similar stimuli (the natural stimulus and the main tonal component) elicited similar responses that differ from the response to the weak noise component. In contrast, A1 neurons responses not only differ one from another, both neurons process a non-linear function of the input stimulus: responses to the natural stimulus resemble responses to the weak component more than responses to the strong tonal component.

2.3 Related work

Recently, considerable attention has been focused on spectrotemporal receptive fields (STRFs) as characterizations of the function of auditory cortical neurons (3; 7; 16; 20; 32). The STRF model is appealing in several respects: it is a conceptually simple model that provides a linear description of the neuron's behavior. It can be interpreted both as providing the neuron's most efficient stimulus (in the time-frequency domain), and also as the spectro-temporal impulse response of the neuron (18; 22). Finally, STRFs can be efficiently estimated using simple algebraic techniques.

However, while there were initial hopes that STRFs would uncover relatively complex response properties of cortical neurons, several recent reports of large sets of STRFs of cortical neurons concluded that most STRFs are relatively simple (9), and that STRFs are typically rather sluggish in time, therefore missing the highly precise synchronization of some cortical neurons (20). Furthermore, STRFs often fail to predict the responses to natural stimuli (11; 18). For example, in Machens et al. only 11% of the response power could be predicted by STRFs on average (18). Similar results were also reported in (27), who found that STRF models account for only $18 - 40\%$ of the stimulus-related power in auditory cortical neural responses to random chord stimuli. Various other studies have shown that there are significant and relevant non-linearities in auditory cortical responses to natural stimuli (1; 17; 18; 26).

The analysis of our dataset of natural sounds started with the work of Bar-Yosef et. al. (1), who have shown that cortical auditory neurons are extremely

sensitive to small perturbations in the (natural) acoustic context. Chechik et. al. (5), who worked on the same stimuli and response set for cortical neurons, have shown that different neurons impose different partitions of the stimulus space, which are not necessarily simply related to the spectro-temporal structure of the stimuli. It appears that these non-linearities cannot be sufficiently explained using linear models such as the STRF.

Chapter 3

Computational Methods

3.1 Statistical Framework For Modeling Data

3.1.1 Mixture Model

The problem of clustering data into meaningful groups is one of the basic problems in machine learning. Traditionally, when the learning is performed on labeled samples the problem is called classification, and when the learning is done on unlabeled data samples - clustering. In order to learn something from unlabelled data one has to make some assumptions about the data.

Suppose we have a data set of size N, i.e. $\mathcal{X} = \{x_1, \cdots, x_N\}$, drawn from an unknown distribution $p(\mathbf{x}|\mathbf{\Theta})$. We will assume here that the functional forms of the probability densities are known, and we need to learn the unknown parameter vector.

To begin with, the following assumptions will be made (10):

1. The samples come from a known number c of classes.
2. The a priori probabilities $P(\omega_j)$ for each class are known, $j = 1, \cdots, c$.
3. The forms for the class-conditional probability densities $p(\mathbf{x}|\omega_j, \Theta_j)$ are known, $j = 1, \cdots, c$.
4. All that is unknown are the values for the c parameter vectors $\mathbf{\Theta_1}, \cdots, \mathbf{\Theta_c}$

Each data sample $\mathbf{x}$ comes from class ω_j with probability $P(\omega_j)$, and its probability is $p(\mathbf{x}|\omega_j, \mathbf{\Theta_j})$. Thus, the probability density function for the sample is given by

$$p(\mathbf{x}|\mathbf{\Theta}) = \sum_{j=1}^{c} p(\mathbf{x}|\omega_j, \mathbf{\Theta_j}) P(\omega_j) \tag{3.1}$$

where $\mathbf{\Theta} = \mathbf{\Theta_1}, \cdots, \mathbf{\Theta_c}$. A density function of this form is called a *mixture density*. The conditional densities $p(\mathbf{x}|\omega_j, \Theta_j)$ are called the *component densities*, and the a priori probabilities $P(\omega_j)$ are called the *mixing parameters*. In the version used in this work, we leave the mixing parameters unknown, and estimate them from the data along with the other unknown parameters.

The basic goal in this setup is to estimate the unknown parameter vector $\mathbf{\Theta}$, using the data samples from this mixture density. Once we know $\mathbf{\Theta}$ (and $\omega_1, \cdots, \omega_c$ is left unknown) we can decompose the mixture into its components and the problem is solved: we have a sufficient decomposition of the input space into c clusters.

3.1.2 Gaussian Mixture Model

A case of special interest is a Gaussian Mixture Model (GMM). In this parametric statistical model, the component densities are drawn from the normal distribution

$$p(\mathbf{x}|\omega_i, \mathbf{\Theta_i}) \sim N(\mu_i \Sigma_i) \tag{3.2}$$

Gaussian Mixture Models are appealing because they provide a generative model for the data examined. A generative model is useful, because we can provide predictions regarding previously unseen points. Each Gaussian component is usually used to represent a different source/type of data.

3.2 Parameter Estimation

3.2.1 Maximum likelihood Estimation Problem

Suppose we have a density function $p(\mathbf{x}|\mathbf{\Theta})$ which is governed by a set of parameters $\mathbf{\Theta}$ and the data set $\mathcal{X} = \{x_1, \cdots, x_N\}$ which is drawn from this distribution.

We assume that these data vectors are independent and i.i.d. with distribution p. Therefore, the resulting density for the data set is

$$p(\mathcal{X}|\mathbf{\Theta}) = \prod_{i=1}^{N} p(x_i|\Theta) = \mathcal{L}(\Theta|\mathcal{X}) \tag{3.3}$$

This function $\mathcal{L}(\Theta|\mathcal{X})$ is called the likelihood of the parameters given the data, or just the *likelihood function*. The likelihood is actually a function of $\mathbf{\Theta}$ while the data set $\mathcal{X}$ is fixed. In the maximum likelihood problem our goal is to find $\mathbf{\Theta}$ that maximizes $\mathcal{L}$ (2). We wish to find $\mathbf{\Theta}^*$ where

$$\mathbf{\Theta}^* = argmax_{\Theta}\ \mathcal{L}(\Theta|\mathcal{X}) \tag{3.4}$$

Often we maximize $log\mathcal{L}(\Theta|\mathcal{X})$ instead because it is analytically easier. For some easy problems it's possible to find a closed solution for finding parameters $\mathbf{\Theta}$, but for many interesting problems it is not feasible to find such analytical expressions and more elaborate techniques have to be used.

3.2.2 Expectation Minimization

The Expectation Minimization (EM) algorithm is one of the popular techniques for finding the maximum likelihood estimate (MLE) in various problems. The EM algorithm (8) is a general method for finding the MLE of the parameters of an underlying distribution from a given data set when the data is incomplete or has missing values.

There are two main applications for the EM algorithm. The first occurs when the data has missing values due to noisy measurements or other limitations of the observation process. The second occurs when optimizing the likelihood function is analytically intractable but when the likelihood function can be simplified by assuming the existence of some *missing* (or *hidden*) parameters. The latter application is the one we are going to relate to in this work.

As before, we assume that data $\mathcal{X}$ is observed and is generated by some distribution. We call $\mathcal{X}$ the *incomplete data*. We assume that a complete data set

exists $\mathcal{Z} = \mathcal{X}, \mathcal{Y}$ and also specify a joint density function:

$$p(z|\Theta) = p(x, y|\Theta) = p(y|x, \Theta)p(x|\Theta) \tag{3.5}$$

One example of such hidden variables are the mixing parameters in a mixture model.

With this new density function we can define a new likelihood function, $\mathcal{L}(\Theta|\mathcal{Z}) = \mathcal{L}(\Theta|\mathcal{X}, \mathcal{Y}) = p(\mathcal{X}, \mathcal{Y}|\Theta|)$, called the *complete data likelihood* (2). This function is in fact a random variable since the missing information $\mathcal{Y}$ is unknown, random and presumably governed by an underlying distribution.

The EM algorithm first finds the expected value of the complete data log-likelihood $log\ p(\mathcal{X}, \mathcal{Y}|\Theta|)$ with respect to the unknown data $\mathcal{Y}$ given the observed data $\mathcal{X}$ and the current parameter estimators. Thus, we can define:

$$Q(\Theta, \Theta^{(i-1)}) = E[log\ p(\mathcal{X}, \mathcal{Y}|\Theta)|\mathcal{X}, \Theta^{(i-1)}] \tag{3.6}$$

where the meaning of the two arguments in the function $Q(\Theta, \Theta^{(i-1)})$ is the following: the first argument Θ corresponds to the parameters that will be optimized in the attempt to maximize the likelihood; the second argument $\Theta^{(i-1)}$ is constant and we use it to evaluate the expectation. The evaluation of this expectation is called the E-step of the algorithm.

The second step (the M-step) of the EM algorithm is to maximize the expectation we computed in the first step. Thus, we find:

$$\Theta^{(i)} = argmax_{\Theta}\ Q(\Theta, \Theta^{(i-1)}) \tag{3.7}$$

These two steps are repeated as necessary. Each iteration is guaranteed to increase the log-likelihood and the algorithm is guaranteed to converge to a local maximum of the likelihood function (8).

3.2.3 Finding Maximum Likelihood GMM Parameters via EM

The mixture density parameter estimation is probably one of the most widely used applications of the EM algorithm. In this case, we assume the probabilistic model given by eq. 3.1. Let's define $\alpha_j = P(\omega_j)$, and we get:

$$p(\mathbf{x}|\mathbf{\Theta}) = \sum_{i=1}^{M} \alpha_i p_i(\mathbf{x}|\mathbf{\Theta_i}) \tag{3.8}$$

where the parameters vector we wish to optimize is $\Theta = (\alpha_1, \cdots, \alpha_M, \Theta_1, \cdots, \Theta_M)$ such that $\sum_{i=1}^{M} \alpha_i = 1$ and each p_i is a density function parametrized by Θ_i. For example, if we assume d-dimensional Gaussian component distributions with mean μ and covariance matrix Σ, i.e. $\Theta_i = (\mu_i, \Sigma_i)$, then

$$p_i(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{(-1)}(x-\mu_i)} \tag{3.9}$$

Explicitly, we assume that we have M Gaussian distributions mixed together with M missing coefficients α_i (probability of each Gaussian model). The incomplete data log likelihood for density defined in Eq. 3.8 from the data $\mathcal{X}$ is given by:

$$log(\mathcal{L}(\Theta|\mathcal{X})) = log \prod_{i=1}^{N} p(x_i|\Theta) = \sum_{i=1}^{N} log(\sum_{j=1}^{M} \alpha_j p_j(x_i|\Theta_j)) \tag{3.10}$$

This expression is hard to estimate because it contains a log of a sum. But if we consider $\mathcal{X}$ incomplete, and assume that each data point x_i has an unobserved data point y_i, whose value informs us to which density component it belongs, the likelihood expression can be significantly simplified. We assume that $y_i \in \{1, \cdots, M\}$, and $y_i = l$ if the i^{th} sample was generated by the l^{th} mixture component. Then the complete data log likelihood is:

$$log(\mathcal{L}(\Theta|\mathcal{X}, \mathcal{Y})) = log(P(\mathcal{X}, \mathcal{Y}|\Theta)) = \sum_{i=1}^{N} log(P(x_i|y_i)P(y_i)) = \sum_{i=1}^{N} log(\alpha_{y_i} p_{y_i}(x_i|\Theta_{y_i})) \tag{3.11}$$

From this equation we can derive update equations for $\alpha_i, \Theta_i | i = 1, \cdots, M$, or in the Gaussian Mixture Model case for $\alpha_i, (\mu_i, \Sigma_i) | i = 1, \cdots, M$. The estimates of the new optimal parameters for maximizing the log-likelihood in terms of old parameters in this case are:

$$\alpha_l^{new} = \frac{1}{N} \sum_{i=1}^{N} p(y_i = l | x_i, \Theta^{old}) \tag{3.12}$$

$$\mu_l^{new} = \frac{\sum_{i=1}^{N} x_i p(y_i = l | x_i, \Theta^{old})}{\sum_{i=1}^{N} p(y_i = l | x_i, \Theta^{old})} \tag{3.13}$$

$$\Sigma_l^{new} = \frac{\sum_{i=1}^{N} p(y_i = l | x_i, \Theta^{old} (x_i - \mu_l^{new})(x_i - \mu_l^{new})^T}{\sum_{i=1}^{N} p(y_i = l | x_i, \Theta^{old})} \tag{3.14}$$

For full details according the derivation of these update equations, see Bilmes (2). The above equations perform both the expectation step and the maximization step simultaneously. The algorithm proceeds by using the newly derived parameters as the guess for the next iteration.

3.2.4 Finding Maximum Likelihood GMM Parameters via EM using Equivalence Constraints

In previous paragraphs Gaussian Mixture Models were used for density estimation in an unsupervised manner. But in many cases additional information for specific data points is available. For example, we may have access to the labels of *part* of the data set. In this case our problem becomes semi-supervised since the estimation relies on both labeled and unlabeled points. Side-information relevant to this work is *equivalence constraints* between pairs of data points. Two points will have a *positive* constraint between them if they were generated by the same source (they belong to the same Gaussian density distribution) and a *negative* constraint if they were generated by different sources.

The additional value of incorporating equivalence constraints is in two levels: First, it will result in faster convergence of the EM algorithm to a solution of higher likelihood, Second, and more importantly, the equivalence constraints should change the likelihood function itself. It can give rise to a different solution,

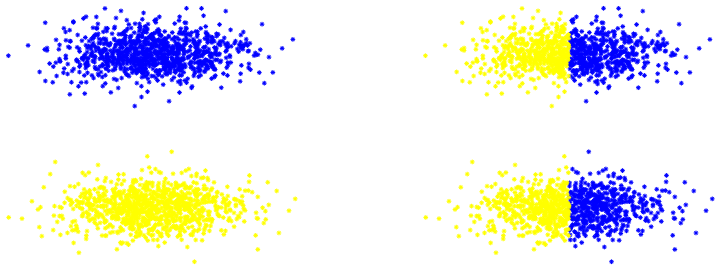

Figure 3.1: Illustrative example to demonstrate the added value of equivalence constraints. The data set consists of two *vertically aligned* classes. Left: given no additional information, the EM algorithm identifies two *horizontally aligned* classes. Right: additional side information in the form of equivalence constraints changes the probability function and the vertical partition arises as the most likely solution (29).

that could be rejected based on unconstrained GMM density model. A simple example demonstraiting this point is shown in Fig. 3.1

Shental et al (29) introduced a method to incorporate positive and negative equivalence constraints into the EM procedure of evaluating a constrained GMM. The main idea of the algorithm is in modifying the Expectation Step in the following way: instead of summing over *all* possible assignments of data points to sources, sum only over the assignments that comply with the given constraints. For example, if points x_i and x_j are positively constrained - only assignments in which both points are assigned to the *same* Gaussian source (in the notation presented before $y_i = y_j = l$, $l \in \{1, \cdots, M\}$) are considered. On the other hand, if these points are negatively constrained - only assignments in which they are assigned to *different* Gaussians are considered. The full details of the algorithm may be found in the article.

3.3 Boosting

Boosting is a general method that attempts to "boost" the accuracy of any given learning algorithm. It is a method of finding a highly accurate hypothesis (classification rule) by combining many "weak" hypotheses, each of which is only moderately accurate. Typically, each weak hypothesis is a simple rule which can

be used to generate a predicted classification for any instance, and the basic idea of boosting is to use such weak learners repeatedly each time on a different sample of the learning examples. The output is a weighted sum of the weak learners' outputs.

A common boosting algorithm is AdaBoost presented by Freund and Shapire (see Fig. 1) (12). The algorithm takes as input a training set $(x_1, y_1), ..., (x_n, y_n)$ where each x_i belongs to some *instance space* X and each *label* y_i is in some label set Y. In the algorithm presented here it is assumed that the label space is -1, +1. The main effect of AdaBoost's update rule (assuming $\alpha_t > 0$) is to decrease the weight of correctly classified train examples and to increase the weight of those classified incorrectly. The final hypothesis H is a weighted majority vote of T weak hypotheses where α_t is the weight assigned to h_t.

Algorithm 1 The boosting algorithm AdaBoost

Given $(x_1, y_1), ..., (x_n, y_n)$; $x_i \in X$, $y_i \in \{-1, +1\}$

Initialize $D_1(i) = 1/n \quad i = 1, .., n$

For $t = 1, .., T$

1. Train weak learner using distribution D_t
2. Get weak hypothesis $h_t : X \to [-1, +1]$ with error
   $\epsilon_t = Pr_{i* \ *D_t}[h_t(x_i) \neq y_i]$
3. Choose $\alpha_t = \frac{1}{2} \ln(\frac{1-\epsilon_t}{\epsilon_t})$
4. Update:
$$D_{t+1}(i) = \begin{cases} D_t(i) \exp(-\alpha_t) & h_t(x_i) = y_i \\ D_t(i) \exp(\alpha_t) & h_t(x_i) \neq y_i \end{cases}$$
5. Normalize: $D_{t+1}(i) = D_{t+1}(i)/Z_{t+1}$
   where $Z_{t+1} = \sum_{i=1}^{n} D_{t+1}(i)$
6. Output the final hypothesis $H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$

The ability of AdaBoost to reduce the training error was proved theoretically by Freund and Shapire (12). Let us write the error ϵ_t of weak hypothesis h_t as $\frac{1}{2} - \gamma_t$. Intuitively, since any random classifier guesses the right binary classification at a rate of $\frac{1}{2}$, γ_t measures how much better current weak classifier h_t is as compared to random classifiers. Freund and Shapire (12) proved that the training error (the fraction of mistakes on the training set) of the final hypothesis H is at most $exp(-2\sum_t \gamma_t^2)$. Thus, if each weak hypothesis is slightly better than random (such that $\epsilon_t < \frac{1}{2}$), then the training error drops exponentially fast.

3.4 DistBoost

The DistBoost algorithm (14) is a distance learning algorithm. It learns a distance function which is based on boosting binary classifiers with a confidence level in a *product* space, using weak learners that are trained in the *original* feature space. The algorithm proposes a boosting scheme that incorporates unlabeled data points (points that don't participate in any of the equivalence constraints). These unlabeled points provide a density prior and their weights rapidly decay during the boosting process. The weak learner is based on a constrained EM algorithm which computes a Gaussian mixture model (see 3.2.4). The GMM provides in each boosting step a partition of the original space, from which a weak product space hypothesis is made. The algorithm is shortly described in Alg. 2.

Algorithm 2 The *DistBoost* Algorithm

Input:

Data points: $(x_1, \cdots, x_n), \quad x_k \in \mathcal{X}$

A set of equivalence constraints: (x_{i_1}, x_{i_2}, y_i), where $y_i \in \{-1, 1\}$

Unlabeled pairs of points: $(x_{i_1}, x_{i_2}, y_i = *)$, implicitly defined by all unconstrained pairs of points

- Initialize $W^1_{i_1 i_2} = 1/(n^2) \quad i_1, i_2 = 1, \ldots, n$ (weights over pairs of points)
  $w_k = 1/n \qquad\qquad k = 1, \ldots, n$ (weights over data points)
- For $t = 1, .., T$

  1. Fit a constrained GMM (weak learner) on weighted data points in $\mathcal{X}$ using the equivalence constraints.
  2. Generate a weak hypothesis $\tilde{h}_t : \mathcal{X} \times \mathcal{X} \to [-1, 1]$ and define a weak distance function as $h_t(x_i, x_j) = \frac{1}{2}\left(1 - \tilde{h}_t(x_i, x_j)\right) \in [0, 1]$
  3. Compute $r_t = \sum_{(x_{i_1}, x_{i_2}, y_i = \pm 1)} W^t_{i_1 i_2} y_i \tilde{h}_t(x_{i_1}, x_{i_2})$, only over **labeled** pairs. Accept the current hypothesis only if $r_t > 0$.
  4. Choose the hypothesis weight $\alpha_t = \frac{1}{2}\ln(\frac{1+r_t}{1-r_t})$
  5. Update the weights of **all** points in $\mathcal{X} \times \mathcal{X}$ as follows:

$$W^{t+1}_{i_1 i_2} = \begin{cases} W^t_{i_1 i_2} \exp(-\alpha_t y_i \tilde{h}_t(x_{i_1}, x_{i_2})) & y_i \in \{-1, 1\} \\ W^t_{i_1 i_2} \exp(-\alpha_t) & y_i = * \end{cases}$$

  6. Normalize: $W^{t+1}_{i_1 i_2} = \frac{W^{t+1}_{i_1 i_2}}{\sum_{i_1, i_2 = 1}^{n} W^{t+1}_{i_1 i_2}}$
  7. Translate the weights from $\mathcal{X} \times \mathcal{X}$ to $\mathcal{X}$: $w^{t+1}_k = \sum_j W^{t+1}_{kj}$

Output: A final distance function $\mathcal{D}(x_i, x_j) = \sum_{t=1}^{T} \alpha_t h_t(x_i, x_j)$

Chapter 4

Formalizing the problem as a distance learning problem

4.1 Experimental setup

4.1.1 Acoustic stimuli

Four stimuli, each of length 60 - 100 ms, consisted of a main tonal component with frequency and amplitude modulation and of a background noise consisting of echoes and unrelated components. Each of these stimuli was further modified by separating the main tonal component from the noise, and by further separating the noise into echoes and background. All possible combinations of these components were used here, in addition to a stylized artificial version that lacked the amplitude modulation of the natural sound. In total, 8 versions of each stimulus were used in A1 experiments and another 2 versions were used in IC experiments. Several stimuli with different variations are shown in fig. 4.1. Therefore each neuron in A1 had a dataset consisting of 32 datapoints and each neuron in IC had 40 points dataset[1]. The sounds were taken from the Cornell Laboratory of Ornithology.

[1] For more detailed methods for obtaining the different versions of the stimuli see Bar-Yosef et al. (1).

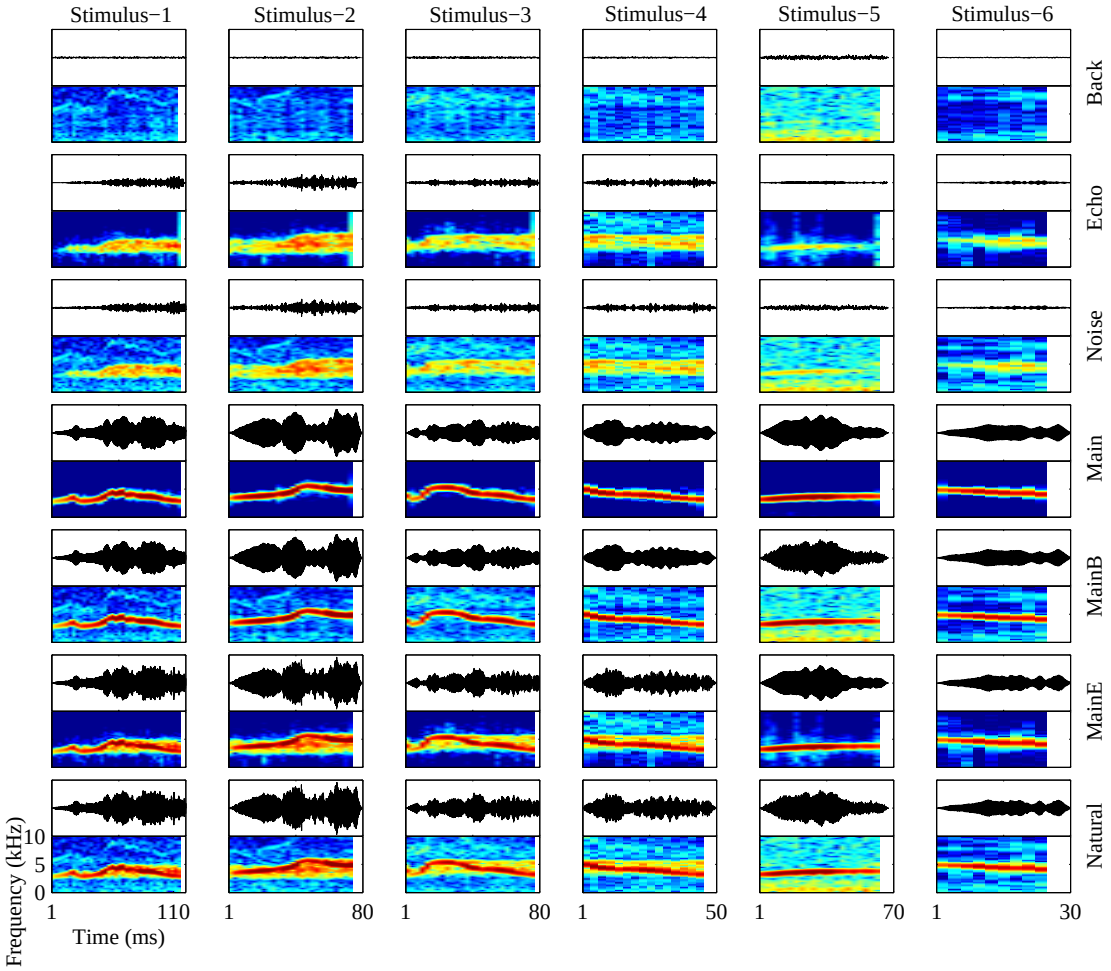

Figure 4.1: Each column refers to a stimulus. The bottom line presents the natural sound for each stimulus, row named "Main" shows only the extracted main chirp for each stimulus, also shown echo, background, noise (echo+background), MainE (main chirp + echo) and MainB (main chirp + background). (1)(unpublished fig.)

4.1.2 IC setup

Recordings were made in the right IC of fifteen pigmented guinea pigs weighing $335 - 507$ gm. Animals were anesthetized with urethane (1.3 gm/kg, i.p., in 20% solution in 0.9% saline) and Hypnorm (Janssen, High Wycombe, UK) (0.2 ml, i.m., comprising fentanyl citrate 0.315 mg/ml and fluanisone 10 mg/ml). To prevent bronchial secretions, atropine sulfate (0.06 mg/kg, s.c.) was administered at the start of the experiment. Anesthesia was supplemented with further doses of Hypnorm (0.2 ml, i.m.), on indication of pedal withdrawal reflex. A tracheotomy was performed, and core temperature was maintained at $38°C$ via a heating blanket and rectal probe. The animals were placed inside a sound attenuating room in a stereotaxic frame in which hollow plastic speculas replaced the ear bars to allow sound presentation and direct visualization of the tympanic membrane. A craniotomy was performed over the position of the IC. The dura was reflected, and the surface of the brain was covered by a solution of 1.5% agar in 0.9% saline. Respiratory rate was monitored by means of a fine polythene tube inserted into the tracheal cannula connected to a low-pressure transducer; heart rate was monitored using a pair of electrodes inserted into the skin to either side of the animal's thorax. Recordings were made with glass-insulated tungsten electrodes (4) advanced into the IC (optional charge) through the intact cortex, in a vertical penetration, by a piezoelectric motor (Burleigh Inchworm IW-700/710). Extracellular action potentials were amplified (Axoprobe 1A, Axon Instruments, Foster City, CA), discriminated using a level-crossing detector (SD1, Tucker-Davies Technologies), and their time of occurrence was recorded with a resolution of 1 μsec. For further details see Shackleton et. al. (28).

4.1.3 A1 setup

Extracellular recordings were made in primary auditory cortex of nine halothane-anesthetized cats. Anesthesia was induced by ketamine and xylazine and maintained with halothane (0.25-1.5%) in 70% N_2O using standard protocols authorized by the committee for animal care and ethics of the Hebrew University - Haddasah Medical School. Single neurons were recorded using metal microelectrodes and an online spike sorter (MSD, alpha-omega). All neurons were well sep-

arated. Penetrations were performed over the whole dorso-ventral extent of the appropriate frequency slab (between about 2 and 8 kHz). Stimuli were presented 20 times using sealed, calibrated earphones at 60-80 dB SPL, at the preferred aurality of the neurons as determined using broad-band noise bursts. For further details see Bar-Yosef et. al. (1).

4.2 Computational problem formulation

Our approach is based on the idea of learning a cell-specific distance function over the space of all possible stimuli, relying on partial information extracted from the neuronal responses of the cell. The initial data consists of stimuli and the resulting neural responses. In a typical auditory experiment each neuron is presented with several repetitions of a set of input stimuli. Usually, these pairs of stimuli and responses are used to directly learn the neuron's input-output function. In our approach we use these pairs to train a distance learning algorithm, defined over all stimuli. We use the neuronal responses to identify pairs of stimuli to which the neuron responded similarly and pairs to which the neuron responded very differently. These pairs can be formally described by equivalence constraints. Equivalence constraints are relations between pairs of datapoints, which indicate whether the points in the pair belong to the same category or not. We term a constraint *positive* when the points are known to originate from the same class, and *negative* if they belong to different classes. In this setting the goal of the algorithm is to learn a distance function that attempts to comply with the equivalence constraints. In order to measure the similarity between neuronal responses, we used the normalized χ^2 distance measure (see Section 4.4 for details).

We can therefore formally define the computational task as follows:

Input: a set of input stimuli which were presented to a set of neurons, and their recorded responses.

1. Represent the input stimuli using some standard representation (such as its first Cepstral coefficients).

2. Use the responses to extract positive and negative equivalence constraints.

3. Learn a distance function using the input stimuli and the equivalence constraints thus gathered.

4. Test the predictive power of the learned distance function using cross validation.

This formalism allows to combine information from a number of cells to improve the resulting characterization. Specifically, we combine equivalence constraints gathered from pairs of cells which have similar responses by taking the intersection of two cells' constraints, and train a single distance function for both cells. Our results demonstrate that this approach improves prediction results of the "weaker" cell, and almost always improves the result of the "stronger" cell in each pair. Another interesting result of this formalism is the ability to classify stimuli based on the responses of the total recorded cell ensemble. For some stimuli, the predictive performance based on the learned inter-stimuli distance was very good, whereas for other stimuli it was rather poor. For cortical neurons these differences were correlated with the acoustic structure of the stimuli, partitioning them into narrowband and wideband stimuli.

4.3 Data representation

We used the first 60 ms of each stimulus. Each stimulus was represented using the first d real Cepstral coefficients. The real Cepstrum of a signal x was calculated by taking the natural logarithm of magnitude of the Fourier transform of x and then computing the inverse Fourier transform of the resulting sequence. In our experiments we used the first 21-36 coefficients. Neuronal responses were represented by creating Peri-Stimulus Time Histograms (PSTHs) using 20 repetitions recorded for each stimuli. Response duration was 100 ms.

4.4 Obtaining equivalence constraints over stimuli pairs

The distances between responses were measured using a normalized χ^2 distance measure. All responses to both stimuli (40 responses in total) were superimposed to generate a single high-resolution PSTH. Then, this PSTH was non-uniformly binned so that each bin contained at least 10 spikes. The same bins were then used to generate the PSTHs of the responses to the two stimuli separately. For similar responses, we would expect that on average each bin in these histograms would contain 5 spikes. More formally, let N denote the number of bins in each histogram, and let r_1^i,r_2^i denote the number of spikes in the i'th bin in each of the two histograms respectively. The distance between pairs of histograms is given by:

$$D_{chi} = \frac{\sum_{i=1}^{N} \frac{(r_1^i - r_2^i)^2}{(r_1^i + r_2^i)/2}}{N-1} \tag{4.1}$$

In order to identify pairs (or small groups) of similar responses, we computed the normalized χ^2 distance matrix over all pairs of responses, and used the complete-linkage algorithm (10) to cluster the responses into $8-12$ clusters. All of the points in each cluster were marked as similar to one another, thus providing positive equivalence constraints. In order to obtain negative equivalence constraints, for each cluster c_i we used the $2-3$ furthest clusters from it to define negative constraints. All pairs, composed of a point from cluster c_i and another point from these distant clusters, were used as negative constraints.

4.5 Evaluation of the distance learning method

In order to evaluate the quality of the learned distance function, we measured the correlation between the distances computed by our distance learning algorithm to those induced by the χ^2 distance over the responses. For each stimulus that was tested in leave-one out manner, we measured the distances to all other stimuli using the learnt distance function. We then computed the rank-order Spearman

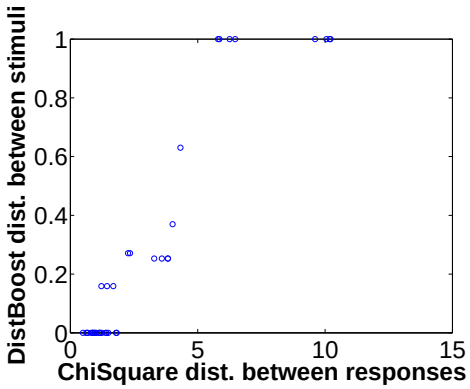

Figure 4.2: IC: DistBoost induced distances between stimuli versus χ^2 distances between appropriate neuronal responses for one left-out data point.

correlation coefficient and Pearson correlation coefficient between these learnt distances in the stimulus domain and the χ^2 distances between the appropriate responses. A typical result for one left out stimulus in IC cell (IC26) is presented in Fig 4.2. The X axis of the figure is the χ^2 distances between left-out stimulus *response* and 39 remaining neural responses. The Y axis is DistBoost induced distances between the left-out stimulus and 39 train stimuli. The correlation is measured between these two vectors (of 39 distances each). Since these vectors contain less than 40 values both for IC and A1 cells, in some cases the correlation results are not highly significant. This procedure produced a single correlation coefficient for each left-out stimulus, and the average correlation coefficient across all stimuli was used as the overall performance measure for each cell.

In order to compare our new approach to the STRF approach we performed the following procedure, using STRFPAK-2.0.1 package by Theunissen et. al. (32), for all cells:

- For each left-out stimulus compute STRF based on the train stimuli (the remaining $N-1$ stimuli) with 4 different tolerance values.
- Choose one tolerance value per cell by choosing the one that gave the highest average correlation between the *predicted* firing rate for the train stimuli based on the STRFs and the actual recorded firing rate.

- For the chosen tolerance value, calculate distances between the predicted firing rates for each left-out stimulus by simply calculating the Euclidean distance between the left-out stimulus prediction and all the other predictions. This results in $N-1$ distances per left-out stimulus.
- Compute the rank-order Spearman correlation coefficient and Pearson correlation coefficient between these distances in the stimulus domain and the χ^2 distances between the appropriate responses, same as in our distance learning paradigm. Produce a single correlation coefficient for each of the left-out stimuli, and the average correlation coefficient across all stimuli is used as the overall performance measure for each cell.

4.6 Parameter selection

The following parameters of the *DistBoost* algorithm can be fine-tuned:

1. The input dimensionality d = 21-36.
2. The number of Gaussian models in each weak learner M = 2-4.
3. The number of clusters used to extract equivalence constraints C = 8-12.
4. The number of distant clusters used to define negative constraints $numAnti$ = 2-3.

Optimal parameters were determined separately for each of the 22 A1 cells and the 28 IC cells, based *solely* on the training data. Specifically, in the cross-validation testing we used the following validation paradigm: Given N stimuli, in the leave-one out manner the training is done on $N-1$ stimuli. Before performing the training, we removed an additional datapoint and trained our algorithm on the remaining $N-2$ points. We then validated the algorithm's performance using the left out datapoint. The optimal cell-specific parameters for the final LOU training (with $N-1$ stimuli) were determined using this approach.

Chapter 5

Results

5.1 Fitting power of cell-specific distance function

The fitting power of a method is measured by analyzing the fraction of the information that the model can capture on train examples. In the present context, this is a measure of how well our method captures the relevant structure of the auditory stimulus space as induced by a specific cell, while trained with all the stimulus set. We begin our analysis with an evaluation of the fitting power of our method, by training A1 neurons with the entire set of 32 stimuli (see Fig. 5.1). For each cell, rank order correlation coefficient between learnt distances in the stimulus domain and the χ^2 distances between the appropriate responses was calculated (see Section 4.5). In general almost all of the correlation values are positive and quite high. The average correlation over all cells is 0.62 with $STandardError(ste) = 0.0096$.

5.2 Generalization power of the method

In order to evaluate the generalization potential of our approach, we used a leave one out (LOU) cross-validation paradigm. In each run, we removed a single stimulus from the dataset of N stimuli (40 for IC and 32 for A1), trained our algorithm on the remaining $N-1$ stimuli, and then tested its performance on the

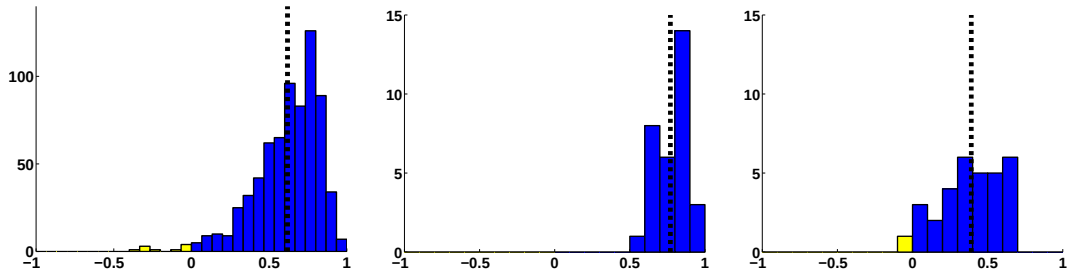

Figure 5.1: Left: Histogram of train rank-order correlations on the entire ensemble of A1 cells. The rank-order correlations were computed between the learnt distances and the distances between the recorded responses for each single stimulus ($N = 22*32$). Center: train correlations for a "strong" cell. Right: train correlations for a "weak" cell. The dotted lines represent the average value of each distribution.

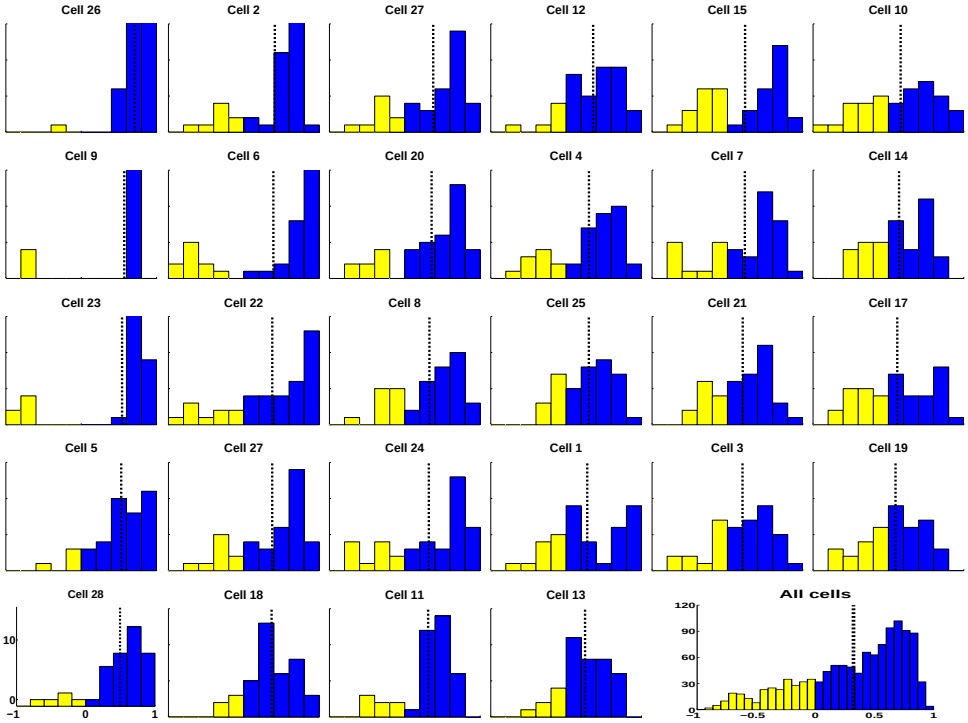

Figure 5.2: IC: Histograms of cell specific test rank-order correlations for the 28 cells in the dataset. The rank-order correlations compare the predicted distances to the distances between the recorded responses, measured on a single stimulus which was left out during the training stage. For visualization purposes, cells are ordered (columns) by their average test correlation per stimulus in descending order. Negative correlations are in yellow, positive in blue.

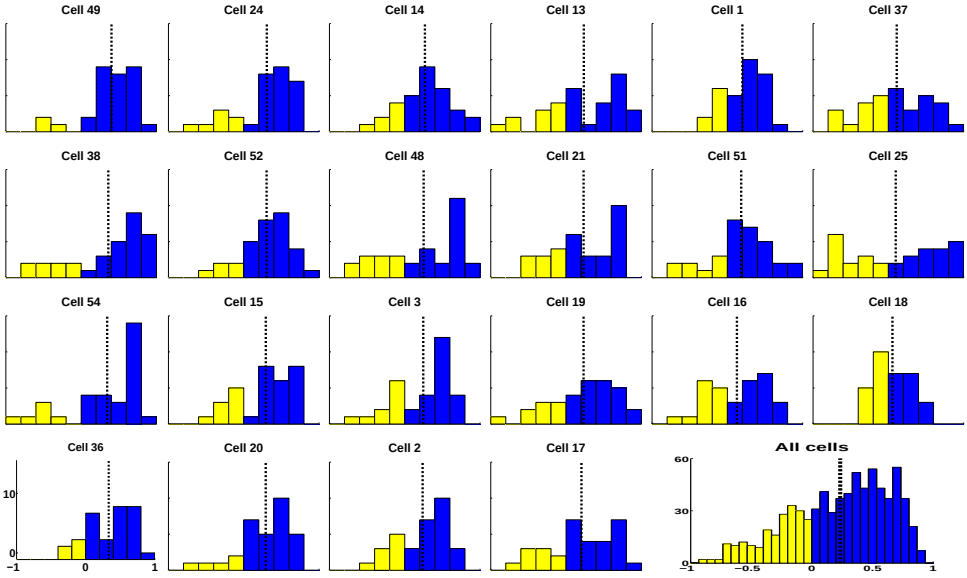

Figure 5.3: A1: Histograms of cell specific test rank-order correlations for the 22 cells in the dataset. The rank-order correlations compare the predicted distances to the distances between the recorded responses, measured on a single stimulus which was left out during the training stage. For visualization purposes, cells are ordered (columns) by their average test correlation per stimulus in descending order. Negative correlations are in yellow, positive in blue.

datapoint that was left out (see Fig. 5.3 and Fig. 5.2). The train result in this paradigm is the mean correlation for output distances of the $N-1$ stimuli the algorithm trained on and the neuronal distances for the responses to these stimuli. The test result is the correlation of the left-out stimulus output distances and its neuronal response distances. In each histogram we plot the test correlations of a single cell, obtained using the LOU paradigm over all of the N stimuli (N boxes in each cell histogram).

As can be seen, results for test performance are varied, even when cells from the same sub-system (A1 or IC) are considered. While for some IC neurons our algorithm obtains average rank-order correlations that are as high as 0.71, for a minority of the cells the average correlation drops to below 0.2. This variability in the predictive power of the method is also apparent in the results obtained for the cortical cells. The highest test correlation achieved on A1 cell is 0.41, while for some cells the average test correlation is less then 0.1. On average, A1 test correlations are lower than IC test correlations. The average rank-order correlation over all IC cells is 0.34 with $ste = 0.034$ and the average rank-order correlation over all A1 cells is 0.24 with $ste = 0.0019$.

Not surprisingly, both for IC and for A1 the train results in LOU procedure are better than the test results (see Fig. 5.4). Interestingly, however, we found that there was a significant correlation between the training performance and the test performance both for IC ($C = 0.61,\ p < 0.001$) and for A1 ($C = 0.41,\ p < 0.05$).

5.3 Boosting the performance of weak cells

In order to boost the performance of cells with low average correlations, we constructed the following experiment: We clustered the responses of each cell, using the complete-linkage algorithm over the χ^2 distances with 4 clusters. We then used the $F_{\frac{1}{2}}$ score that evaluates how well two clustering partitions are in agreement with one another ($F_{\frac{1}{2}} = \frac{2*P*R}{P+R}$, where P denotes precision and R denotes recall.). This measure was used to identify pairs of cells whose partition of the stimuli was most similar to each other. In our experiment we took the four cells with the lowest performance (right column of Fig 5.3), and for each of them used the $F_{\frac{1}{2}}$ score to retrieve the most similar cell. For each of these pairs, we

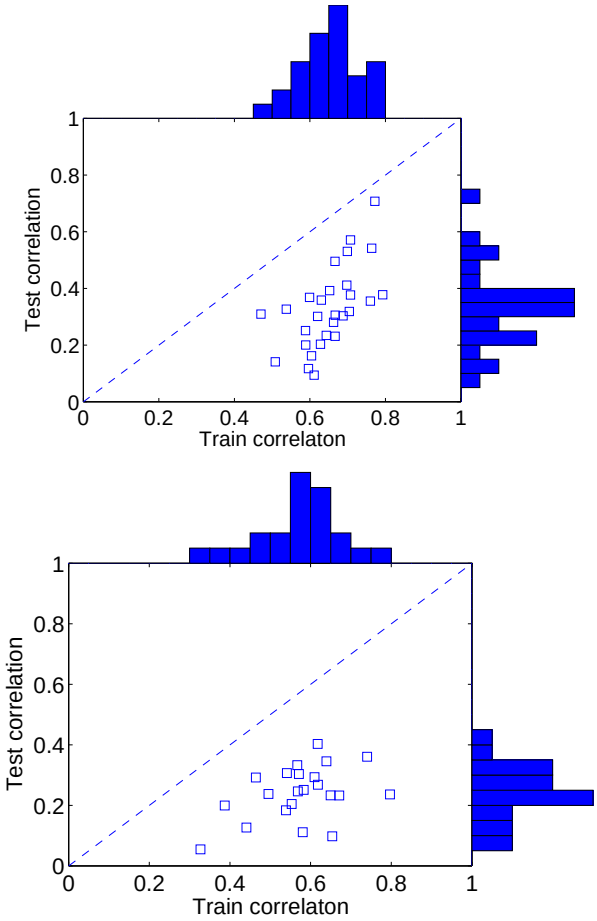

Figure 5.4: Train vs. test cell specific correlations. Each point marks the average correlation of a single cell. The distribution of train and test correlations is displayed as histograms on the top and on the right respectively. Upper: IC data set. The correlation between train and test is 0.61 with $p = 0.0006$. Lower: A1 data set. The correlation between train and test is 0.4 with $p = 0.05$.

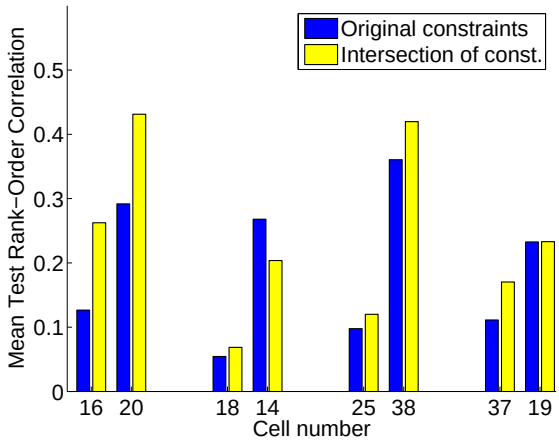

Figure 5.5: Test rank-order correlations when training using constraints extracted from each cell separately, and when using the intersection of the constraints extracted from a pair of cells. This procedure always improves the performance of the weaker cell, and usually also improves the performance of the stronger cell.

trained our algorithm once more, using the constraints obtained by intersecting the constraints derived from the two cells in the pair, in the LOU paradigm. The results can be seen in Fig 5.5. On all of the four original low-performing cells, this procedure improved LOU test results. Interestingly and counter-intuitively, when training the better performing cell in each pair using the intersection of its constraints with those from the poorly performing cell, results deteriorated only for one of the four better performing cells. In general, for 11 of 22 A1 cells, one of the four most similar cells caused an average improvement of 0.09 in test rank order correlation, while for 5 of the remaining cells no other cell improved the generalization performance.

5.4 Stimulus classification

After computing the test performance per cell we measured the predictability of each stimulus by averaging the LOU test results obtained for the stimulus across all cells separately in IC and in A1. This analysis recovers differences in performance that can be linked to a basic attribute of auditory stimulus - its bandwidth. In our experiment, we used both narrow-band and wide-band

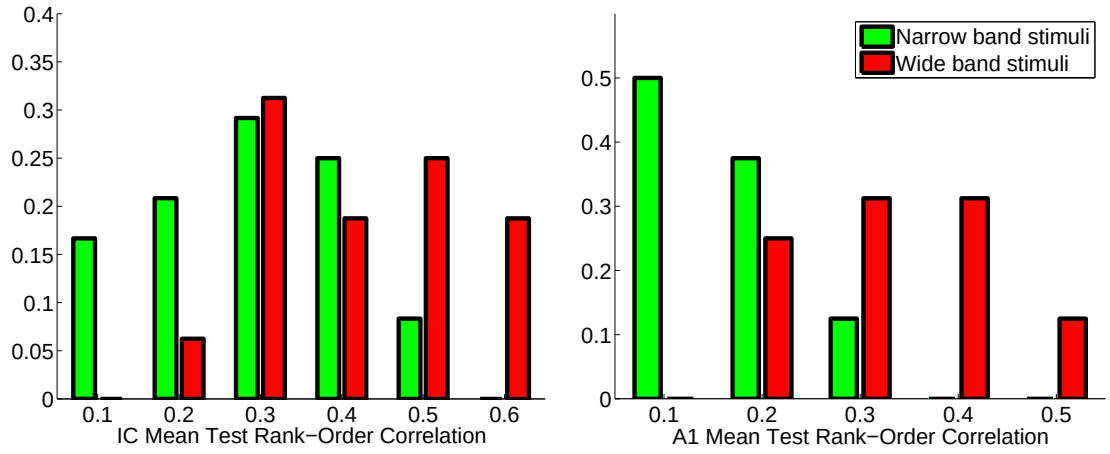

Figure 5.6: Left: Stimuli specific correlation values averaged over the entire ensemble of IC cells. Mean test correlation for narrow-band stimuli: $0.292, std = 0.12$; mean correlation for wide-band: $0.416, std = 0.12$. Narrow-band stimuli are slightly less predictable, but the distributions are largely overlapping. Right: Stimuli specific correlation values averaged over the entire ensemble of A1 cells. Mean test correlation for narrow-band stimuli: $0.151, std = 0.08$; mean correlation for wide-band: $0.333, std = 0.09$. The predictability of wideband stimuli in A1 case is clearly better than that of the narrowband stimuli.

variations of stimuli (for example, Main component is a narrow-band stimulus and Noise component is a wide-band stimulus, see Fig 4.1). As can be seen in Fig. 5.6, the cross-validation results in A1 induced a partition of the stimulus space into narrow-band and wide-band stimuli. In IC the mean predictability of wide-band stimuli is higher than the mean predictability of narrow-band stimuli, but the two correlation distributions are largely overlapping.

Our analysis shows that for cortical auditory neurons wide-band stimuli are more predictable than narrow-band stimuli, despite the fact that the neuronal responses to these two groups are not different as a whole. Whereas the non-linearity in the interactions between narrow-band and wide-band stimuli has already been noted before (17; 23; 24), here we further refine this observation by demonstrating a significant difference between the behavior of narrow and wide-band stimuli with respect to the predictability of the similarity between their responses. IC neurons show less difference in predictability, a fact that complies with IC cells' description as linear filters depending solely on the Frequency Response Area of the neuron.

5.5 Comparison to STRF

As another evaluation of our novel approach, STRF was calculated for all IC and A1 cells using leave-one out manner (see 4.5). Rank-order Spearman correlation coefficient was computed between predicted responses' distances and true responses distances. For 28 IC cells in the data set mean rank-order test correlation coefficient achieved by STRF linear approach is 0.37, $std = 0.09$. For A1 neurons data set mean rank-order test correlation coefficient is 0.21, $std = 0.07$. Three typical results for A1 cells are presented in Fig. 5.7. Although the average results in both approaches are similar for IC and for A1, the correlations are distributed uniformly and they remain small for all stimuli in STRF results, and are very different (very high and very low correlations per stimulus in the same cell) in DistBoost results.

After computing STRF test performance per cell we measured the predictability of each stimulus by averaging the LOU test results obtained for the stimulus across all cells separately in IC and in A1. Using STRF approach we did not find

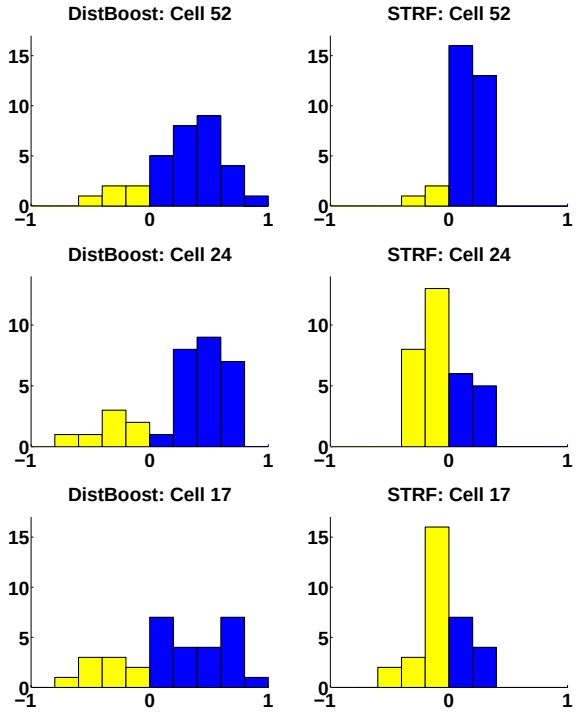

Figure 5.7: Comparison of DistBoost and STRF for A1 set. Typical examples of test rank order correlations using DistBoost (left) and STRF (right). While mean test correlation per-cell is similar for both approaches, DistBoost predicts the distances very well for part of the stimuli and poorly for another part and STRF generally returns small correlation values for all stimuli. Negative correlations are in yellow, positive in blue.

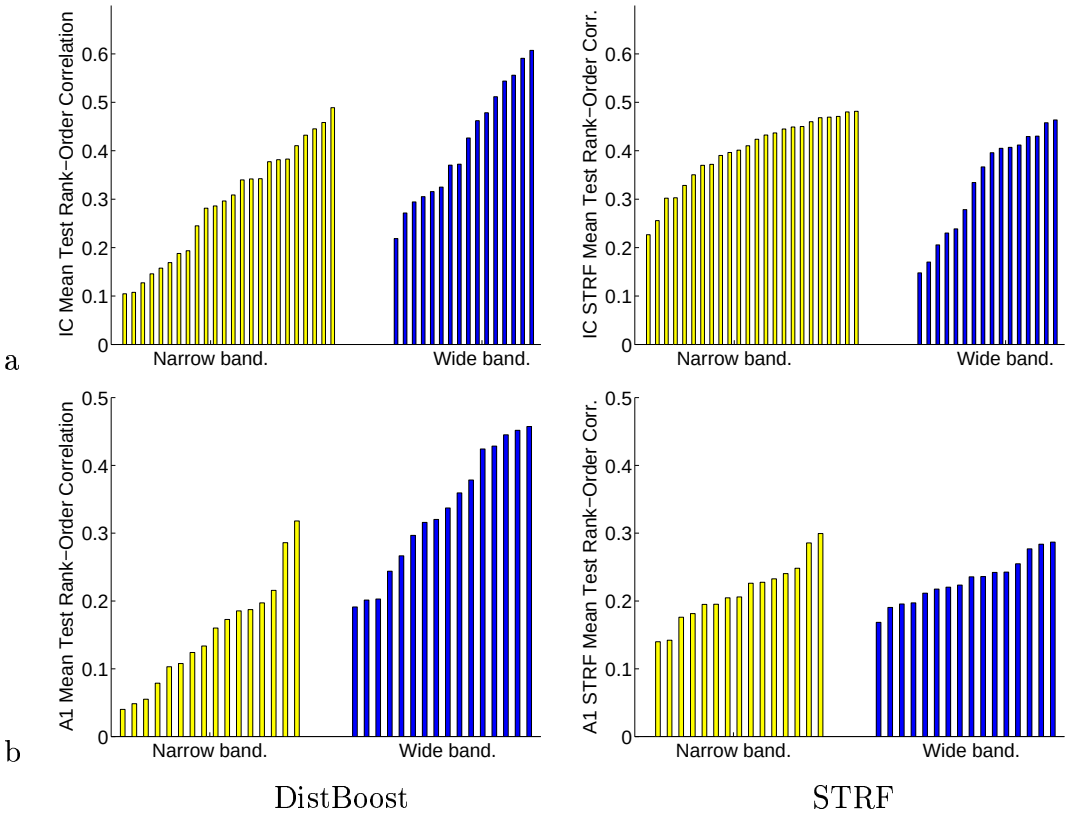

Figure 5.8: Histograms of mean test rank-order correlations for narrow-band and wide-band stimuli separately. (a) IC results. Left: DistBoost correlations. Right: STRF correlations. (b) A1 results. Left: DistBoost correlations. Right: STRF correlations.

any difference in predictability for narrow-band and wide-band stimuli in a sense of distance learning in the stimuli space in IC nor in A1 (see Fig. 5.8).

Chapter 6

Discussion

In the standard approach to auditory modeling, a linear or weakly non-linear model is fitted to the data, and neuronal properties are read from the resulting model. Extensive research have been done in order to define the basic features that cortical neurons are sensitive to, in analogy to oriented moving lines for V1 neurons. Thus the bulk of the experiments use simple and well-defined stimuli, such as pure tones, amplitude-modulated tones, frequency-modulated tones, random chords, etc. Unfortunately, due to the high sensitivity of A1 neurons to small perturbations in the stimulus space, powerful predictions could be made only on stimuli identical in nature to those in the train set. As a rule, models based on ensembles of simple and tailored stimuli failed to capture the responses of auditory cortical neurons to natural stimuli. Recent experiments use complex stimuli in general, and natural stimuli in particular, to create models for the neuronal responses to complex sounds. However, these studies suggest that the predictability of these models is weak and the usefulness of linear modeling is limited due to the highly non-linear behavior of auditory cortical neurons in response to natural stimuli.

Motivated by the limitations of linear modeling, we propose a new formulation of the problem. We frame the problem as a distance learning problem over the auditory stimulus space. Neuronal data is used as a guide for training a highly non-linear distance function on stimulus space, compatible with the neural responses. After training the model, we can predict the similarity between a cell's response to a test stimulus and the cell's responses to the training stimuli. The

main result of this thesis is to demonstrate the feasibility of this approach. As another validation of the novel method, we present here results not only for A1 neurons, but for pre-cortical IC neurons as well.

First, we learned distance functions over the stimulus domain for single cells using information extracted from responses collected during standard electrophysiological experiments. The evaluation of the fitting power of the model was done on A1 neurons data set and the results were satisfying: thus, a desired distance function can be learnt. Moreover, the predictive power of these cell specific distance functions was checked when presented with novel stimuli. Cross-validation scheme was performed, and a single stimulus was removed from the data set both for IC and for A1 experimental data. While train performance remained high, test results were lower for both subsystems, as expected when handling a small data set. Not surprisingly, in general, both test results and the correlation between train and test results were higher in IC than in A1. Further research is needed to find more compatible evaluation methods for the new approach, possibly based on the identification of the nearest neighbor or a group of nearest neighbors.

Two further results underscore the usefulness of the new formulation. First, we demonstrated that we can improve the test performance of a distance function by using constraints on the similarity or dissimilarity between stimuli derived from the responses of multiple neurons. Whereas we expected this manipulation to improve the test performance of the algorithm on the responses of neurons that were initially poorly predicted, we found that it actually improved the performance of the algorithm also on neurons that were rather well predicted, although we paired them with neurons that were poorly predicted. Thus, it is possible that intersecting constraints derived from multiple neurons uncover regularities that are hard to extract from individual neurons. The description of the best choice of pairs in order to improve both (or one) cell's performance is one of our future research directions.

Second, it turned out that some stimuli consistently behaved better than others across the neuronal population of the cortical neurons. This difference was correlated with the acoustic structure of the stimuli: those stimuli that contained the weak background component (wide-band stimuli) were generally predicted

better. This result is surprising both because the background component is substantially weaker than the other acoustic components in the stimuli (by as much as 35-40 dB). It may mean that the relationship between physical structure (as characterized by the Cepstral coefficients) and the neuronal responses becomes simpler in the presence of the background component, but is much more idiosyncratic when this component is absent. This result underscores the importance of interactions between narrow and wide-band stimuli for understanding the complexity of cortical processing. In contrast, IC neurons are thought to be linear (or almost linear) filters of the stimulus. Considering this, it is not surprising that the generalization performance of IC neurons for both types of stimuli was similar and no such difference was found.

In order to compare our distance learning algorithm to STRF approach we designed a scenario in which from explicit response predictions we construct distances between stimuli, as imposed by STRFs. We use this scenario to compare STRF results to DistBoost results. The average results both for IC and for A1 are similar to those achieved by using DistBoost. But if we take a closer look into the per-stimulus correlations for a typical cell - we find interesting differences between the approaches. STRF method tends to give predictions with uniformly small correlations to responses' distances (in a specific cell all the correlations are similar). DistBoost "learns well" some of the stimuli, and predicts its distances from all the other stimuli with high correlation to responses' distances, while it predicts poorly some of the other stimuli true position in the learnt space. Thus it can be highly useful to identify well-predicted stimuli before running DistBoost and computing their predictability explicitly. However, this still remains one of our future goals. Furthermore, the difference in stimuli predictability that was found using DistBoost for cortical neurons was not present using STRF approach.

A major experimental problem in using any learnt function based on stimuli presented to a neuron is the "life time" of the neuron: only a limited number of stimuli can be presented to a neuron during one recording session. Here we had 32 short stimuli ($60-100$ ms long) presented to A1 neurons and 40 stimuli presented to IC neurons. These data sets are very small for the considered stimulus space (of spectrograms or Cepstral coefficients). Our algorithm is fast enough to be used in near real-time and can therefore be used to guide real experiments: the distance

functions trained here can direct the process of choosing the best set of stimuli for characterizing the responses of a neuron. For example, learnt distance functions can be used to find surprising stimuli: either stimuli that are very different in terms of physical structure but that would result in responses that are similar to those already measured, or stimuli that are very similar to already tested stimuli but that are predicted to give rise to very different responses.

Bibliography

[1] O. Bar-Yosef, Y. Rotman, and I. Nelken. Responses of Neurons in Cat Primary Auditory Cortex to Bird Chirps: Effects of Temporal and Spectral Context. *J. Neurosci.*, 22(19):8619–8632, 2002. 1, 2.2, 2.3, 1, 4.1, 4.1.3

[2] J. Bilmes. A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models, 1997. 3.2.1, 3.2.2, 3.2.3

[3] D. T. Blake and M. M. Merzenich. Changes of AI Receptive Fields With Sound Density. *J Neurophysiol*, 88(6):3409–3420, 2002. 2.3

[4] D.C. Bullock, A.R. Palmer, and Rees A. Compact and easy-to-use tungsten-in-glass microelectrode manufacturing workstation. *Med. Biol. Eng. Comput.*, 26(6):669–672, 1988. 4.1.2

[5] G. Chechik, A. Globerson, M.J. Anderson, E.D. Young, I. Nelken, and N. Tishby. Group redundancy measures reveal redundancy reduction in the auditory pathway. In *NIPS*, 2002. 1, 2.2, 2.3

[6] F. d'Alche Buc, Y. Grandvalet, and C. Ambroise. Semi-supervised marginboost, 2002. 1

[7] R. C. deCharms, D. T. Blake, and M. M. Merzenich. Optimizing Sound Features for Cortical Neurons. *Science*, 280(5368):1439–1444, 1998. 2.3

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum-likelihood from incomplete data via the em algorithm. *J. Royal Statist. Soc. Ser. B.*, 39, 1977. 3.2.2, 3.2.2

[9] D. A. Depireux, J. Z. Simon, D. J. Klein, and S. A. Shamma. Spectro-Temporal Response Field Characterization With Dynamic Ripples in Ferret Primary Auditory Cortex. *J Neurophysiol*, 85(3):1220–1234, 2001. 2.3

[10] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons Inc., 2001. 3.1.1, 4.4

[11] J. J. Eggermont, P. M. Johannesma, and A. M. Aertsen. Reverse-correlation methods in auditory research. *Q Rev Biophys.*, 16(3):341–414, 1983. 2.3

[12] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting, 1997. 3.3, 3.3

[13] Y. Grandvalet, F. d'Alche Buc, and C. Ambroise. Boosting mixture models for semi supervised learning, 2001. 1

[14] T. Hertz, A. Bar-Hillel, and D. Weinshall. Boosting margin based distance functions for clustering. In *ICML*, 2004. 1, 3.4

[15] E. R. Kandel, J. H. Schwartz, and T. M. Jessell. *Principles of Neural Science*. McGraw-Hill, 2000. 2.1

[16] N. Kowalski, D. A. Depireux, and S. A. Shamma. Analysis of dynamic spectra in ferret primary auditory cortex. I. Characteristics of single-unit responses to moving ripple spectra. *J Neurophysiol*, 76(5):3503–3523, 1996. 2.3

[17] L. Las, E. A. Stern, and I. Nelken. Representation of Tone in Fluctuating Maskers in the Ascending Auditory System. *J. Neurosci.*, 25(6):1503–1513, 2005. 2.2, 2.3, 5.4

[18] C. K. Machens, M. S. Wehr, and A. M. Zador. Linearity of Cortical Receptive Fields Measured with Natural Sounds. *J. Neurosci.*, 24(5):1089–1100, 2004. 1, 2.3

[19] L. Mason, J. Baxter, P. Bartlett, and M. Frean. Boosting algorithms as gradient descent in function space, 1999. 1

[20] L. M. Miller, M. A. Escabi, H. L. Read, and C. E. Schreiner. Spectrotemporal Receptive Fields in the Lemniscal Auditory Thalamus and Cortex. *J Neurophysiol*, 87(1):516–527, 2002. 2.3

[21] I. Nelken. Feature detection in the auditory cortex. In Popper A Oertel D and Fay RR, editors, *Integrative functions of the auditory system*, New York, 2002. Springer. 2.1

[22] I. Nelken. Processing of complex stimuli and natural scenes in the auditory cortex. *Current Opinion in Neurobiology*, 14(4):474–480, 2004. 2.3

[23] I. Nelken, L. Las, N. Ulanovsky, and D. Farkas. Levels of auditory processing: The subcortical auditory system, primary auditory cortex, and the hard problems of auditory perception. In R. Konig, P. Heil, E. Budinger, and H. Scheich, editors, *The Auditory Cortex: A Synthesis of Human and Animal Research*, pages 331–346, Mahwah, New Jersey, 2005. Lawrence Erlbaum Assosiates. 2.1, 2.2, 5.4

[24] I. Nelken, N. Ulanovsky, L. Las, O. Bar-Yosef, M. Anderson, G. Chechik, N. Tishby, and E.D. Young. Transformation of stimulus representations in the ascending auditory system. In McAdams S Pressnitzer D, de Cheveigne A and Collet L., editors, *Auditory signal processing: physiology, psychoacoustics, and models*, New York, 2005. Springer. 2.2, 5.4

[25] I. Nelken and E.D. Young. Two separate mechanisms shape the responses of type IV dorsal cochlear nucleus units to narrow-band and wide-band stimuli. *J. Neurophysiol.*, 71:2446–2462, 1994. 2.1

[26] Y. Rotman, O. Bar-Yosef, and I. Nelken. Relating cluster and population responses to natural sounds and tonal stimuli in cat primary auditory cortex. *Hearing Research*, 152(1-2):110–127, 2001. 2.3

[27] M. Sahani and J. F. Linden. How linear are auditory cortical responses? In *NIPS*, 2003. 1, 2.3

[28] Trevor M. Shackleton, Bernt C. Skottun, Robert H. Arnott, and Alan R. Palmer. Interaural Time Difference Discrimination Thresholds for Single Neurons in the Inferior Colliculus of Guinea Pigs. *J. Neurosci.*, 23(2):716–724, 2003. 4.1.2

[29] N. Shental, A. Bar-Hilel, T. Hertz, and D. Weinshall. Computing Gaussian mixture models with EM using equivalence constraints. In *NIPS*, 2003. 3.1, 3.2.4

[30] N. Shental, T. Hertz, D. Weinshall, and M. Pavel. Adjustment learning and relevant component analysis. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Computer Vision - ECCV*, volume 4, 2002. 1

[31] P. H. Smith and G. A. Spirou. From the cochlea to the cortex and back. In D. Oertel, R. R. Fay, and Popper A. N., editors, *Integrative Functions in the Mammalian Auditory Pathway*, pages 6–71, New York, 2002. Springer. 2.1

[32] F. E. Theunissen, K. Sen, and A. J. Doupe. Spectral-Temporal Receptive Fields of Nonlinear Auditory Neurons Obtained Using Natural Sounds. *J. Neurosci.*, 20(6):2315–2331, 2000. 2.3, 4.5

[33] E.P Xing, A.Y. Ng, M.I. Jordan, and S. Russell. Distance metric learnign with application to clustering with side-information. In *Advances in Neural Information Processing Systems*, volume 15. The MIT Press, 2002. 1

[34] C. Yanover and T. Hertz. Predicting protein-peptide binding affinity by learning peptide-peptide distance functions. In *RECOMB*, 2005. 1