Feature Selection for Unsupervised and Supervised Inference: the Emergence of Sparsity in a Weighted-based Approach [*]

Lior Wolf [†] LIORWOLF@MIT.EDU
Center for Biological and Computational Learning,
The McGovern Institute for Brain Research,
Massachusetts Institute of Technology
Cambridge, MA, 02139 USA

Amnon Shashua SHASHUA@CS.HUJI.AC.IL
School of Computer Science and Engineering,
The Hebrew University,
Jerusalem 91904 Israel

Editor: Donald Geman

Abstract

The problem of selecting a subset of relevant features in a potentially overwhelming quantity of data is classic and found in many branches of science. Examples in computer vision, text processing and more recently bio-informatics are abundant. In text classification tasks, for example, it is not uncommon to have 10^4 to 10^7 features of the size of the vocabulary containing word frequency counts, with the expectation that only a small fraction of them are relevant. Typical examples include the automatic sorting of URLs into a web directory and the detection of spam email.

In this work we present a definition of "relevancy" based on spectral properties of the Laplacian of the features' measurement matrix. The feature selection process is then based on a continuous ranking of the features defined by a least-squares optimization process. A remarkable property of the feature relevance function is that sparse solutions for the ranking values naturally emerge as a result of a "biased non-negativity" of a key matrix in the process. As a result, a simple least-squares optimization process converges onto a sparse solution, i.e., a selection of a subset of features which form a local maxima over the relevance function. The feature selection algorithm can be embedded in both unsupervised and supervised inference problems and empirical evidence show that the feature selections typically achieve high accuracy even when only a small fraction of the features are relevant.

1. Introduction

As visual recognition, text classification, speech recognition and more recently bio-informatics aim to address larger and more complex tasks the problem of focusing on the most relevant information in a potentially overwhelming quantity of data has become increasingly important. Examples from

[*] A short version of this paper was presented at the ICCV, Nice France, Oct. 2003.

[†] The main body of this work was done while LW was at the Hebrew University.

©200? Lior Wolf and Amnon Shashua.

computer vision, text processing and Genomics are abundant. For instance, in visual recognition the pixel values themselves often form a highly redundant set of features; methods using an "over-complete" basis of features for recognition are gaining popularity (Olshausen and Field, 1996), and recently methods relying on abundance of simple efficiently computable features of which only a fraction of are relevant were proposed for face detection (Viola and Jones, 2001) — and these are only few examples from the visual recognition literature. In text classification tasks it is not uncommon to have 10^4 to 10^7 features of the size of the vocabulary containing word frequency counts, with the expectation that only a small fraction of them are relevant (Lewis, 1992). Typical examples include the automatic sorting of URLs into a web directory and the detection of spam email. In Genomics, a typical example is gene selection from micro-array data where the features are gene expression coefficients corresponding to the abundance of cellular mRNA taken from sample tissues. Typical applications include separating tumor from normal cells or discovery of new subclasses of Cancer cells based on the gene expression profile. Typically the number of samples (expression patterns) is less than 100 and the number of features (genes) in the raw data ranges from 5000 to 50000. Among the overwhelming number of genes only a small fraction is relevant for the classification of tissues whereas the expression level of many other genes may be irrelevant to the distinction between tissue classes — therefore, identifying highly relevant genes from the data is a basic problem in the analysis of expression data.

From a practical perspective, large amounts of irrelevant features affects learning algorithms at three levels. First, most learning problems do not scale well with the growth of irrelevant features — in many cases the number of training examples grows exponentially with the number of irrelevant features (Langley and Iba, 1993). Second, is a substantial degradation of classification accuracy for a given training set size (Almuallim and Dietterich, 1991; Kira and Rendell, 1992). The accuracy drop affects also advanced learning algorithms that generally scale well with the dimension of the feature space such as the Support Vector Machines (SVM) as recently observed in (Weston et al., 2001). The third aspect has to do with the run time of the learning algorithm on test instances. In most learning problems the classification process is based on inner-products between the features of the test instance and stored features from the training set, thus when the number of features is overwhelmingly large the run-time of the learning algorithm becomes prohibitively large for real time applications, for example. Another practical consideration is the problem of determining how many relevant features to select. This is a difficult problem which is hardly ever addressed in the literature and consequently it is left to the user to choose manually the number of features. Finally, there is an issue of whether one is looking for the *minimal* set of (relevant) features, or simply a possibly redundant but relevant set of features.

The potential benefits of feature selection include, first and foremost, better accuracy of the inference engine and improved scalability (defying the curse of dimensionality). Secondary benefits include better data visualization and understanding, reduce measurement and storage requirements, and reduce training and inference time. Blum and Langley (1997) in a survey article distinguish between three types of methods: *Embedded*, *Filter* and *Wrapper* approaches. The filter methods apply a preprocess which is independent of the inference engine (a.k.a the predictor or the classification/inference engine) and select features by ranking them with correlation coefficients or make use of mutual information measures. The Embedded and Wrapper approaches construct and select feature subsets that are useful to build a good predictor. The issue being the notion of *relevancy*, i.e., what constitutes a good set of features. The modern approaches, therefore, focus on building feature selection algorithms in the context of a *specific* inference engine. For example, (Weston et al., 2001;

Bradley and Mangasarian, 1998) use the Support Vector Machine (SVM) as a subroutine (wrapper) in the feature selection process with the purpose of optimizing the SVM accuracy on the resulting subset of features. These wrapper and embedded methods in general are typically computationally expensive and often criticized as being "brute force". Further details on relevancy versus usefulness of features and references to historical and modern literature on feature selection can be found in the survey papers (Blum and Langley, 1997; Kohavi and John, 1997; Guyon and Elissef, 2003).

In this paper the inference algorithm is not employed directly in the feature selection process but instead general properties are being gathered which indirectly indicate whether a feature subset would be appropriate or not. Specifically, we use clustering as the predictor and use spectral properties of the candidate feature subset to guide the search. This leads to a "direct" approach where the search is conducted on the basis of optimizing desired spectral properties rather than on the basis of explicit clustering and prediction cycles. The search is conducted by the solution of a least-squares optimization function using a weighting scheme for the ranking of features. *A remarkable property of the energy function is that the feature weights come out positive as a result of a "biased non-negativity" of a key matrix in the process and sharply decay at the border between relevant and non-relevant features*. These properties make the algorithm ideal for "feature weighting" applications and for feature selection as the boundary between relevant and non-relevant features is typically clearly expressed by the decaying property of the feature weights. The algorithm, called $Q-\alpha$, is iterative, very efficient and achieves remarkable performance on a variety of experiments we have conducted.

There are many benefits of our approach: First, we avoid the expensive computations associated with Embedded and Wrapper approaches, yet still make use of a predictor to guide the feature selection. Second, the framework can handle both unsupervised and supervised inference within the same framework and handle any number of classes. In other words, since the underlying inference is based on clustering class labels are not necessary, but on the other hand, when class labels are provided they can be used by the algorithm to provide better feature selections. Third, the algorithm is couched within a least-squares framework — and least-squares problems are the best understood and easiest to handle. Finally, the performance (accuracy) of the algorithm is very good on a large number of experiments we have conducted.

2. Algebraic Definition of Relevancy

A key issue in designing a feature selection algorithm in the context of an inference is defining the notion of relevancy. Definitions of relevancy proposed in the past (Blum and Langley, 1997; Kohavi and John, 1997) lead naturally to a explicit enumeration of feature subsets which we would like to avoid. Instead, we take an algebraic approach and measure the relevance of a subset of features against its influence on the cluster arrangement of the data points with the goal of introducing an energy function which receives its optimal value on the desired feature selection. We will consider two measures of relevancy based on spectral properties where the first is based on the Standard spectrum and the second on the Laplacian spectrum.

2.1 The Standard Spectrum

Consider a $n \times q$ data set M consisting of q samples (columns) over n-dimensional feature space R^n representing n features $x_1, ..., x_n$ over q samples. Let the row vectors of M be denoted by $\mathbf{m}_1^\top, ..., \mathbf{m}_n^\top$ pre-processed such that each row is centered around zero and is of unit L_2 norm

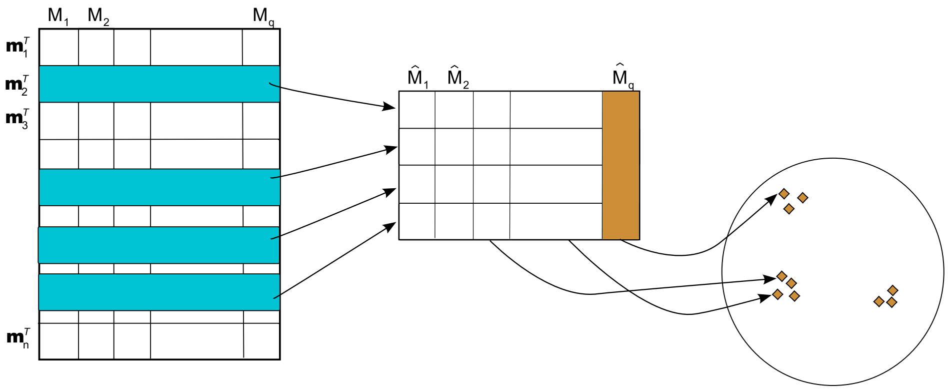


Figure 1: An illustration of variable-selection using our matrix notation. The large array on the left represents the matrix M, which contains q columns that represent the q data- points $(M_1, ..., M_q)$. Each row of this matrix is a feature-vector $\mathbf{m}_1^\top, ..., \mathbf{m}_n^\top$. In an idealized variable selection process, rows of the matrix M are selected to construct the matrix $\hat{M}$ (middle), whose columns form well coherent clusters.

$\|\mathbf{m}_i\| = 1$. Let $\mathcal{S} = \{x_{i_1}, ..., x_{i_l}\}$ be a subset of (relevant) features from the set of n features and let $\alpha_i \in \{0, 1\}$ be the indicator value associated with feature x_i, i.e., $\alpha_i = 1$ if $x_i \in \mathcal{S}$ and zero otherwise (see Fig. 1). Let A_s be the corresponding *affinity* matrix whose (i, j) entries are the inner-product between the i'th and j'th data points restricted to the selected coordinate features, i.e., $A_s = \sum_{i=1}^{n} \alpha_i \mathbf{m}_i \mathbf{m}_i^\top$ where $\mathbf{m}_i \mathbf{m}_i^\top$ is the rank-1 matrix defined by the outer-product between $\mathbf{m}_i$ and itself. Finally, let Q_s be a $q \times k$ matrix whose columns are the first k eigenvectors of A_s associated with the leading (highest) eigenvalues $\lambda_1 \geq ... \geq \lambda_k$.

We define "relevancy" as directly related to the clustering quality of the data points restricted to the selected coordinates. In other words, we would like to measure the quality of the subset $\mathcal{S}$ in terms of cluster coherence of the first k clusters, i.e., we make a direct linkage between cluster coherence of the projected data points and relevance of the selected coordinates.

We measure cluster coherence by analyzing the (standard) spectral properties of the affinity matrix A_s. Considering the affinity matrix as representing weights in an undirected graph, it is known that maximizing the quadratic form $\mathbf{x}^\top A_s \mathbf{x}$ where $\mathbf{x}$ is constrained to lie on the standard simplex ($\sum x_i = 1$ and $x_i \geq 0$) provides the identification of the maximal *clique* of the (unweighted) graph (Motzkin and Straus, 1965; Gibbons et al., 1997), or the maximal "dominant" subset of vertices of the weighted graph (Pavan and Pelillo, 2003). Likewise there is evidence (motivated by finding cuts in the graph) that solving the quadratic form above where $\mathbf{x}$ is restricted to the unit sphere provides cluster membership information (cf. Ng et al., 2001; Weiss, 1999; Perona and Freeman, 1998; Shi and Malik, 2000; Brand and Huang, 2003; Chung, 1998). In this context, the eigenvalue (the value of the quadratic form) represents the cluster coherence. In the case of k clusters, the highest k eigenvalues of A_s represent the corresponding cluster coherences and the components of an eigenvector represent the coordinate (feature) participation in the corresponding cluster. The eigenvalues decrease as the interconnections of the points within clusters get sparser (see (Sarkar and Boyer, 1998)). Therefore, we define the relevance of the subset $\mathcal{S}$ as:

$$\begin{aligned} rel(\mathcal{S}) &= trace(Q_s^\top A_s^\top A_s Q_s) \\ &= \sum_{r,s} \alpha_{i_r}\alpha_{i_s}(\mathbf{m}_{i_r}^\top \mathbf{m}_{i_s})\mathbf{m}_{i_r}^\top Q_s Q_s^\top \mathbf{m}_{i_s} \\ &= \sum_{j=1}^{k} \lambda_j^2, \end{aligned}$$

where λ_j are the leading eigenvalues of A_s. Note that the proposed measure of relevancy handles interactions among features up to a second order. To conclude, achieving a high score on the combined energy of the first k eigenvalues of A_s indicate (although indirectly) that the q input points projected onto the l-dimensional feature space are "well clustered" and that in turn suggests that $\mathcal{S}$ is a relevant subset of features.

Maximizing the relevancy above for all possible feature subsets in infeasible. Therefore, we relax the problem, i.e., instead of enumerating the feature subsets $\mathcal{S}$ and ranking them according to the value of $rel(\mathcal{S})$ we consider the prior weights $\alpha_1, ..., \alpha_n$ as unknown *real numbers* and define the following optimization function:

Definition 1 (Relevant Features Optimization) *Let M be an $n \times q$ input matrix with rows $\mathbf{m}_1^\top, ..., \mathbf{m}_n^\top$. Let $A_\alpha = \sum_{i=1}^n \alpha_i \mathbf{m}_i \mathbf{m}_i^\top$ for some unknown scalars $\alpha_1, ..., \alpha_n$. The weight vector $\alpha = (\alpha_1, ..., \alpha_n)^\top$ and the orthonormal $q \times k$ matrix Q are determined at the maximal point of the following optimization problem:*

$$\max_{Q,\alpha_i} trace(Q^\top A_\alpha^\top A_\alpha Q) \tag{1}$$

$$subject\ to \quad \sum_{i=1}^{n} \alpha_i^2 = 1, \;\; Q^\top Q = I$$

Note that the optimization function does not include the inequality constraint $\alpha_i \geq 0$ and neither a term for "encouraging" a sparse solution of the weight vector α — both of which are necessary for a "feature selection". As will be shown later in Section 4, the sparsity and positivity conditions are implicitly embedded in the nature of the optimization function and therefore "emerge" naturally with the optimal solution.

Note also that it is possible to maximize the gap $\sum_{i=1}^k \lambda_i^2 - \sum_{j=k+1}^q \lambda_j^2$ by defining $Q = [Q_1|Q_2]$ where Q_1 contains the first k eigenvectors and Q_2 the remaining $q-k$ eigenvectors (sorted by decreasing eigenvalues) and the criterion function (1) would be replaced by:

$$\max_{Q=[Q_1|Q_2],\alpha_i} trace(Q_1^\top A_\alpha^\top A_\alpha Q_1) - trace(Q_2^\top A_\alpha^\top A_\alpha Q_2).$$

We will describe in Section 3 an efficient algorithm for finding a local maximum of the optimization (1) and later address the issue of sparsity and positivity of the resulting weight vector α. The algorithms are trivially modified to handle the gap maximization criterion and those will not be further elaborated here. We will describe next the problem formulation using an additive normalization (the Laplacian) of the affinity matrix.

2.2 The Laplacian Spectrum

Given the standard affinity matrix A, consider the Laplacian matrix: $L = A - D + d_{max}I$ where D is a diagonal matrix $D = diag(\sum_j a_{ij})$ and d_{max} is a scalar larger or equal to the maximal element of D[1]. The matrix L normalizes A in an additive manner and there is much evidence to support such a normalization both in the context of graph partitioning (Mohar, 1991; Hall, 1970) and spectral clustering (Weiss, 1999; Ng et al., 2001).

It is possible to reformulate the feature selection problem (1) using the Laplacian as follows. Let $A_i = \mathbf{m}_i \mathbf{m}_i^\top$ and $D_i = diag(\mathbf{m}_i \mathbf{m}_i^\top \mathbf{1})$. We define $L_\alpha = \sum_i \alpha_i L_i$ where $L_i = A_i - D_i + d_{max}I$. We have, therefore:

$$L_\alpha = A_\alpha - D_\alpha + (\sum_i \alpha_i) d_{max} I,$$

where $D_\alpha = diag(A_\alpha^\top \mathbf{1})$. Note that since α is a unit norm vector that contains positive elements, then $\sum_i \alpha_i > 1$. The feature selection problem is identical to (1) where L_α replaces A_α.

3. An Efficient Algorithm

We wish to find an optimal solution for the non-linear problem (1). We will focus on the Standard spectrum matrix A_α and later discuss the modifications required for L_α. If the weight vector α is known, then the solution for the matrix Q is readily available by employing a Singular Value Decomposition (SVD) of the symmetric (and positive definite) matrix A_α. Conversely, if Q is known then α is readily determined as shown next. We already saw that

$$\begin{aligned} trace(Q^\top A_\alpha^\top A_\alpha Q) &= \sum_{i,j} \alpha_i \alpha_j (\mathbf{m}_i^\top \mathbf{m}_j) \mathbf{m}_i^\top Q Q^\top \mathbf{m}_j \\ &= \alpha^\top G \alpha \end{aligned}$$

where $G_{ij} = (\mathbf{m}_i^\top \mathbf{m}_j) \mathbf{m}_i^\top Q Q^\top \mathbf{m}_j$ is symmetric and positive definite. The optimal α is therefore the solution of the optimization problem:

$$\max_\alpha \alpha^\top G \alpha \quad subject\ to\ \alpha^\top \alpha = 1,$$

which results in α being the leading eigenvector of G, i.e., the one associated with its largest eigenvalue. A possible scheme, guaranteed to converge to a local maxima, is to start with some initial guess for α and iteratively interleave the computation of Q given α and the computation of α given Q until convergence. We refer to this scheme as the **Basic** $Q - \alpha$ **Method**.

In practice, the number of iterations is rather small — typically between 5 to 10. The runtime complexity as a function of the number of features n is therefore governed by the complexity of finding the leading eigenvector of G — typically in the order of n^2 assuming a “reasonable” spectral gap (for example, if G were a random matrix then the spectral gap is large — asymptotically in the order of $\sqrt{n}$ — as we know from the semi-circle law (Wigner, 1958)). A quadratic complexity is the best that one can expect when performing feature selection in an unsupervised manner since all pairs of feature vectors need to be compared to each other.

[1] Note that in applications of algebraic graph theory the Laplacian is defined as $D - A$. The reason for the somewhat different definition is that we wish to maintain the order of eigenvectors as in those of A (where the eigenvectors associated with the largest eigenvalues come first).

A more advanced scheme with superior convergence rate and more importantly accuracy of results (based on empirical evidence) is to embed the computation of α within the "orthogonal iteration" (Golub and Loan, 1996) cycle for computing the largest k eigenvectors, described below:

Definition 2 (Standard Power-Embedded $Q - \alpha$ Method) *Let M be an $n \times q$ input matrix with rows $\mathbf{m}_1^\top, ..., \mathbf{m}_n^\top$, and some orthonormal $q \times k$ matrix $Q^{(0)}$, i.e., $Q^{(0)^\top} Q^{(0)} = I$. Perform the following steps through a cycle of iterations with index $r = 1, 2, ...$*

1. *Let $G^{(r)}$ be a matrix whose (i,j) components are $(\mathbf{m}_i^\top \mathbf{m}_j)\mathbf{m}_i^\top Q^{(r-1)} Q^{(r-1)^\top} \mathbf{m}_j$.*
2. *Let $\alpha^{(r)}$ be the largest eigenvector of $G^{(r)}$.*
3. *Let $A^{(r)} = \sum_{i=1}^n \alpha_i^{(r)} \mathbf{m}_i \mathbf{m}_i^\top$.*
4. *Let $Z^{(r)} = A^{(r)} Q^{(r-1)}$.*
5. *$Z^{(r)} \xrightarrow{QR} Q^{(r)} R^{(r)}$.*
6. *Increment index r and go to step 1.*

The method is considerably more efficient than the basic scheme above and achieves very good performance (accuracy). Note that steps 4,5 of the algorithm consist of the "orthogonal iteration" module, i.e., if we were to repeat steps 4,5 *only* we would converge onto the eigenvectors of $A^{(r)}$. However, note that the algorithm does not repeat steps 4,5 in isolation and instead recomputes the weight vector α (steps 1,2,3) before applying another cycle of steps 4,5. We show below that the recomputation of α does not alter the convergence property of the orthogonal iteration scheme, thus the overall scheme converges to a local maxima:

Proposition 3 (Convergence of Power-Embedded $Q - \alpha$) *The Power Embedded $Q - \alpha$ method convergence to a local maxima of the criterion function (1).*

Proof: We will prove the claim for the case $k = 1$, i.e., the scheme optimizes over the weight vector α and the largest eigenvector $\mathbf{q}$ of A_α.

Because the computation of α is analytic (the largest eigenvector of G) and because the optimization energy is bounded from above, it is sufficient to show that the computation of $\mathbf{q}$ monotonically increases the criterion function. It is therefore sufficient to show that:

$$\mathbf{q}^{(r)} A^2 \mathbf{q}^{(r)} \geq \mathbf{q}^{(r-1)} A^2 \mathbf{q}^{(r-1)}, \tag{2}$$

for all symmetric matrices A. Since steps 4,5 of the algorithm are equivalent to the step:

$$\mathbf{q}^{(r)} = \frac{A\mathbf{q}^{(r-1)}}{\|A\mathbf{q}^{(r-1)}\|},$$

we can substitute the right hand side into (2) and obtain the condition:

$$\mathbf{q}^\top A^2 \mathbf{q} \leq \frac{\mathbf{q}^\top A^4 \mathbf{q}}{\mathbf{q}^\top A^2 \mathbf{q}}, \tag{3}$$

which needs to be shown to hold for all symmetric matrices A and unit vectors $\mathbf{q}$. Let $\mathbf{q} = \sum_i \gamma_i \mathbf{v}_i$ be represented with respect to the orthonormal set of eigenvectors $\mathbf{v}_i$ of the matrix A. Then, $A\mathbf{q} =$

$\sum_i \gamma_i \lambda_i \mathbf{v}_i$ where λ_i are the corresponding eigenvalues. Since $\mathbf{q}^\top A^2 \mathbf{q} \geq \mathbf{0}$, it is sufficient to show that: $\|A\mathbf{q}\|^4 \leq \|A^2\mathbf{q}\|^2$, or equivalently:

$$(\sum_i \gamma_i^2 \lambda_i^2)^2 \leq \sum_i \gamma_i^2 \lambda_i^4. \tag{4}$$

Let $\mu_i = \lambda_i^2$ and let $f(x) = x^2$. We then have:

$$f(\sum_i \gamma_i^2 \mu_i) \leq \sum_i \gamma_i^2 f(\lambda_i^2),$$

which follows from convexity of $f(x)$ and the fact that $\sum_i \gamma_i^2 = 1$. □

A faster converging algorithm is possible by employing the "Ritz" acceleration (Golub and Loan, 1996) to the basic power method as follows:

Definition 4 ($Q - \alpha$ with Ritz Acceleration) *Let M be an $n \times q$ input matrix with rows $\mathbf{m}_1^\top, ..., \mathbf{m}_n^\top$, and some orthonormal $n \times k$ matrix $Q^{(0)}$, i.e., $Q^{(0)^\top} Q^{(0)} = I$. Perform the following steps through a cycle of iterations with index $r = 1, 2, ...$*

1. *Let $G^{(r)}, \alpha^{(r)}$ and $A^{(r)}$ be defined as in the Standard Power-Embedded $Q - \alpha$ algorithm.*
2. *$Z^{(r)} = A^{(r)} Q^{(r-1)}$.*
3. *$Z^{(r)} \xrightarrow{QR} \bar{Q}^{(r)} R^{(r)}$.*
4. *Let $\bar{G}^{(r)}$ be a matrix whose (i, j) components are $\mathbf{m}_i^\top \bar{Q}^{(r)^\top} \bar{Q}^{(r)} \mathbf{m}_j$.*
5. *Recompute $\alpha^{(r)}$ as the largest eigenvector of $\bar{G}^{(r)}$, and recompute $A^{(r)}$ accordingly.*
6. *Let $S^{(r)} = \bar{Q}^{(r)^\top} A^{(r)} \bar{Q}^{(r)}$.*
7. *Perform SVD on $S^{(r)}$: $[U^{(r)^\top} S^{(r)} U^{(r)}] = svd(S^{(r)})$.*
8. *$Q^{(r)} = \bar{Q}^{(r)} U^{(r)}$.*
9. *Increment index r and go to step 1.*

The $Q - \alpha$ algorithm for the Laplacian spectrum L_α follows the Standard spectrum with the necessary modifications described below.

Definition 5 (Laplacian Power-Embedded $Q - \alpha$ Method) *In addition to the definition of the Standard method, let $d_i = \max diag(\mathbf{m}_i \mathbf{m}_i^\top)$ and $L_i^{(0)} = \mathbf{m}_i \mathbf{m}_i^\top - diag(\mathbf{m}_i \mathbf{m}_i^\top \mathbf{1}) + d_i I$. Perform the following steps with index $r = 1, 2, ...$*

1. *Let $F^{(r)}$ be a matrix whose (i, j) components are $trace(Q^{(r-1)^\top} L_i^{(r-1)\top} L_j^{(r-1)} Q^{(r-1)})$.*
2. *Let $\alpha^{(r)}$ be the largest eigenvector of $F^{(r)}$.*
3. *Let $d^{(r)} = (\max diag(\sum_{i=1}^n \alpha_i^{(r)} \mathbf{m}_i \mathbf{m}_i^\top))/(\sum_{i=1}^n \alpha_i)$*
4. *For each i let $L_i^{(r)} = \mathbf{m}_i \mathbf{m}_i^\top - diag(\mathbf{m}_i \mathbf{m}_i^\top \mathbf{1}) + d^{(r)} I$*
5. *Let $L^{(r)} = \sum_{i=1}^n \alpha_i^{(r)} L_i^{(r)}$.*
6. *Let $Z^{(r)} = L^{(r)} Q^{(r-1)}$.*
7. *$Z^{(r)} \xrightarrow{QR} Q^{(r)} R^{(r)}$.*
8. *Increment index r and go to step 1.*

3.1 The Supervised Case

The $Q-\alpha$ algorithms and the general approach can be extended to handle data with class labels. One of the strengths of our approach is that the feature selection method can handle both unsupervised and supervised data sets. In a nutshell, the supervised case is handled as follows. Given c classes, we are given c data matrices $M^l, l = 1, ..., c$, each of size $n \times q^l$.

Definition 6 (Supervised Relevant Features Optimization) *Let M^l be an $n \times q^l$ input matrices with rows $\mathbf{m}_1^{l\top}, ..., \mathbf{m}_n^{l\top}$. Let $A_\alpha^{gh} = \sum_{i=1}^n \alpha_i \mathbf{m}_i^g \mathbf{m}_i^{h\top}$ for some unknown scalars $\alpha_1, ..., \alpha_n$. The weight vector $\alpha = (\alpha_1, ..., \alpha_n)^\top$ and the orthonormal $q^h \times k^{gh}$ matrices Q^{gh} are determined at the maximal point of the following optimization problem:*

$$\begin{aligned} \max_{Q^{gh},\alpha_i} & \sum_l trace(Q^{ll\top} A_\alpha^{ll\top} A_\alpha^{ll} Q^{ll}) \\ & -\gamma \sum_{g\neq h} trace(Q^{gh\top} A_\alpha^{gh\top} A_\alpha^{gh} Q^{gh}) \\ subject\ to & \sum_{i=1}^n \alpha_i^2 = 1,\ \ Q^{gh\top} Q^{gh} = I \end{aligned} \tag{5}$$

Where the weight γ and the parameters k^{gh} are determined manually (see below).

The criterion function seeks a weight vector α such that the resulting affinity matrix of all the data points (sorted) would be semi-block-diagonal, i.e., high inter-class eigenvalue energy and low intra-class energy. Therefore, we would like to minimize of the intra-class eigenvalue energy $trace(Q^{gh\top} A_\alpha^{gh\top} A_\alpha^{gh} Q^{gh})$ (off-block-diagonal blocks) and maximize the inter-class eigenvalue energy $trace(Q^{ll\top} A_\alpha^{ll\top} A_\alpha^{ll} Q^{ll})$. The parameters k^{gh} control the complexity of each affinity matrix. A typical choice of the parameters would be $k^{gh} = 2$ when $g = h$, $k^{gh} = 1$ otherwise, and $\gamma = 0.5$.

The solution to the optimization function follows step-by-step the $Q - \alpha$ algorithms. At each cycle Q^{gh} is computed using the current estimates A_α^{gh} and α is optimized by maximizing the expression:

$$\sum_l \alpha^\top G^{ll} \alpha - \gamma \sum_{g\neq h} \alpha^\top G^{gh} \alpha = \alpha^\top \mathcal{G} \alpha \quad ,$$

where $G_{ij}^{gh} = (\mathbf{m}_i^{g\top} \mathbf{m}_j^g) \mathbf{m}_i^{h\top} Q^{gh\top} Q^{gh} \mathbf{m}_j^h$ and $\mathcal{G} = \sum_l G^{ll} - \gamma \sum_{g\neq h} G^{gh}$. We analyze next the properties of the unsupervised $Q - \alpha$ algorithm with regard to sparsity and positivity of the weight vector α and then proceed to experimental analysis.

4. Sparsity and Positivity of α

The optimization criteria (1) is formulated as a least-squares problem and as such there does not seem to be any apparent guarantee that the weights $\alpha_1, ..., \alpha_n$ would come out *non-negative* (same sign condition), and in particular *sparse* when there exists a sparse solution (i.e., there is a relevant subset of features which induces a coherent clustering).

The positivity of the weights is a critical requirement for the $Q-\alpha$ to form a "feature weighting" scheme. In other words, if one could guarantee that the weights would come out non-negative then $Q - \alpha$ would provide feature weights which could be used for selection or for simply weighting the features as they are being fed into the inference engine of choice. If in addition the feature weights exhibit a “sparse” profile, i.e., the gap between the high and low values of the weights is high, then the weights could be used for selecting the relevant features as well. We will refer to the gap between the high and low weights as “sparsity gap” and discuss later in the paper the value of the gap in simplified domains. With the risk of abusing standard terminology, we will refer to the property of having the weight vector concentrate its (high) values around a number of coordinates as a sparsity feature. Typically, for our algorithm, none of the values of the weight vector strictly vanish.

For most feature weighting schemes, the conditions of positivity and sparsity should be specifically presented into the optimization criterion one way or the other. The possible means for doing so include introduction of inequality constraints, use of L_0 or L_1 norms, adding specific terms to the optimization function to “encourage” sparse solutions or use a multiplicative scheme of iterations which preserve the sign of the variables throughout the iterations (for a very partial list see Olshausen and Field, 1996; Kivinen and Warmuth, 1997; Lee and Seung, 1999; Vapnik, 1998). It is therefore somewhat surprising, if not remarkable, that the least-squares formulation of the feature selection problem could consistently converge onto same-sign and sparse solutions.

Before we proceed with the technical issues, it is worthwhile to make qualitative arguments (which were the basis of developing this approach to begin with) as to the underlying reason for sparsity. Consider rewriting the optimization criterion (1) by an equivalent criterion:

$$\min_{\alpha,Q} \left\{ \|A_\alpha - QQ^\top A_\alpha\|_F^2 - \|A_\alpha\|_F^2 \right\} \tag{6}$$

where $\|\cdot\|_F^2$ is the square Frobenius norm of a matrix defined as the sum of squares of all entries of the matrix. The first term of (6) measures the distance between the columns of A_α and the projection of those columns onto a k-dimensional subspace (note that $QQ^\top$ is a projection matrix). This term receives a low value if indeed A_α has a small (k) number of dominant eigenvectors, i.e., the spectral properties of the feature subset represented by A_α are indicative to a good clustering score. Since $A_\alpha = \sum_i \alpha_i \mathbf{m}_i \mathbf{m}_i^\top$ is represented by the sum of rank-1 matrices one can combine only a *small* number of them if the first term is desired to be small. The second term (which may be viewed also as a regularization term) encourages addition of more rank-1 matrices to the sum provided they are *redundant*, i.e., are already spanned by the previously selected rank-1 matrices. This makes the point that the feature selection scheme looks for relevant features but not necessarily the minimal set of relevant features. To summarize, from a qualitative point of view the selection of values for the weights α_i is directly related to the rank of the affinity matrix A_α which should be small if indeed A_α arises from a clustered configuration of data points. A uniform spread of values α_i would result in a high rank for A_α, thus the criteria function encourages a non-uniform (i.e., sparse) spread of weight values.

The argument presented above to facilitate clarity of the approach and should not be taken as a proof for sparsity. The positivity and sparsity issues are approached in the sequel from a different angle which provides a more analytic handle to the underlying search process than the qualitative argument above.

4.1 Positivity of α

The key for the emergence of a sparse and positive α has to do with the way the entries of the matrix G are defined. Recall that $G_{ij} = (\mathbf{m}_i^\top \mathbf{m}_j)\mathbf{m}_i^\top QQ^\top \mathbf{m}_j$ and that α comes out as the leading eigenvector of G (at each iteration). If G were to be non-negative (and irreducible), then from the

Perron-Frobenius theorem the leading eigenvector is guaranteed to be non-negative (or same-sign). However, this is not the case and G in general has negative entries as well as positive ones. However, from a probabilistic point of view the probability that the leading eigenvector of G will come out positive rapidly approaches 1 with the growth of the number of features — this under a couple of simplifying assumptions.

The simplifying assumptions we will make in order to derive a probabilistic argument, is first that the entries of the upper triangular part of G are independent. The second simplifying approximation is that the columns of Q are sampled uniformly over the unit hypersphere. Although the independence and uniformity assumptions are indeed an idealization of the true nature of G and Q, they nevertheless allow us to derive a powerful probabilistic argument which shows in a rigorous manner that the weights α_i are non-negative with probability 1 — a statement which agrees with practice over extensive experimentations which we have performed.

The probabilistic approach follows from the observation that each entry of G consists of a sum of products of three inner-products:

$$G_{ij} = \sum_{l=1}^{k} (\mathbf{m}_i^\top \mathbf{q}_l)(\mathbf{m}_j^\top \mathbf{q}_l)(\mathbf{m}_i^\top \mathbf{m}_j).$$

In general, a product of the form $f = (\mathbf{a}^\top \mathbf{b})(\mathbf{a}^\top \mathbf{c})(\mathbf{b}^\top \mathbf{c})$, where $\|\mathbf{a}\| = \|\mathbf{b}\| = \|\mathbf{c}\| = 1$ satisfies $-1/8 \leq f \leq 1$ where $f = 1$ when $\mathbf{a} = \mathbf{b} = \mathbf{c}$. Since $f \geq -1/8$ (will be proven below) there is an asymmetry on the expected value of f, i.e., the expected values of the entries of G are biased towards a positive value — and we should expect a bias towards a positive leading eigenvector of G. We will derive below the expectation on the entries of G (assuming independence and uniformity) and prove the main theorem showing that a random matrix whose entries are sampled i.i.d. form some distribution with positive mean and bounded variance has a positive leading eigenvector with probability 1 when the number of features n is sufficiently large. The details are below.

Proposition 7 *The minimal value of $f = (\mathbf{a}^\top \mathbf{b})(\mathbf{a}^\top \mathbf{c})(\mathbf{b}^\top \mathbf{c})$ where $\mathbf{a}, \mathbf{b}, \mathbf{c} \in R^q$ are defined over the unit hypersphere is $-1/8$.*

Proof: See appendix.

Proposition 8 *The expected value of $f = (\mathbf{a}^\top \mathbf{b})(\mathbf{a}^\top \mathbf{c})(\mathbf{b}^\top \mathbf{c})$ where $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \Re^q$ and $\mathbf{c}$ is uniformly sampled over the unit hypersphere is $(1/q)(\mathbf{a}^\top \mathbf{b})^2$*

Proof: See appendix.

To get a rough estimate on the values in the matrix G we can further assume that **a** and **b** are also evenly distributed on the q-dim sphere. In this case the expectation of $(\mathbf{a}^\top \mathbf{b})^2$ is $1/q$. To see this observe that the expectation $E(\mathbf{a}^\top \mathbf{b})^2 = \int\int (\mathbf{a}^\top \mathbf{b})^2 d\sigma(\mathbf{a}) d\sigma(\mathbf{b}) = \int \mathbf{a}^\top (\int \mathbf{b}\mathbf{b}^\top d\sigma(\mathbf{b}))\mathbf{a} d\sigma(\mathbf{a}) = \int \mathbf{a}^\top ((1/q)I)\mathbf{a} d\sigma(\mathbf{a})$ where I is the identity matrix in $\Re^q$.

Each entry G_{ij} is a sum of k such terms, $G_{ij} = \sum_{l=1}^{k} (\mathbf{m}_i^\top \mathbf{q}_l)(\mathbf{m}_j^\top \mathbf{q}_l)(\mathbf{m}_i^\top \mathbf{m}_j)$. If the features are irrelevant, we can expect the correlation with the vector $\mathbf{q}_1$ to be similar to correlation with a "typical" random vector. In this case the above proposition applies. However, when $k > 1$ there are interrelations between the elements in the sum resulting from the orthogonality of the columns of Q. The following proposition shows that the expectation is still larger than zero.

Proposition 9 *The expected value of $f = \sum_{i=1}^{k} (\mathbf{a}^\top \mathbf{b})(\mathbf{a}^\top \mathbf{c}_i)(\mathbf{b}^\top \mathbf{c}_i)$ where $\mathbf{a}, \mathbf{b} \in \Re^q$ and $\mathbf{c}_i$ are orthonormal vectors uniformly sampled over the unit hypersphere in $\Re^q$ is $(k/q)(\mathbf{a}^\top \mathbf{b})^2$.*

Proof: See appendix.

The body of results on spectral properties of random matrices (see for example Mehta, 1991) deals with the distribution of eigenvalues. For example, the corner-stone theorem known as Wigner's *semicircle* theorem (Wigner, 1958) is about the asymptotic distribution of eigenvalues with the following result: "Given a symmetric $n \times n$ matrix whose entries are bounded independent random variables with mean μ and variance σ^2, then for any $c > 2\sigma$, with probability $1-o(1)$ all eigenvalues except for at most $o(n)$ belong to $\Theta(c\sqrt{n})$, i.e., lie in the interval $\mathcal{I} = (-c\sqrt{n}, c\sqrt{n})$."

The notation $f(n) = o(g(n))$ stands for $\lim_{n \to \infty} f(n)/g(n) = 0$, i.e., $f(n)$ becomes insignificant relative to $g(n)$ with the growth of n. This is a short-hand notation (which we will use in the sequel) to the formal statement: "$\forall \epsilon > 0$, $\exists n_0$ s.t. $\forall n > n_0$ the statement holds with probability $1 - \epsilon$."

It is also known that when $\mu = 0$ all the eigenvalues belong to the interval $\mathcal{I}$ (with probability $1 - o(1)$), while for the case $\mu > 0$ only the leading eigenvalue λ_1 is outside of $\mathcal{I}$ and

$$\lambda_1 = \frac{1}{n} \sum_{i,j} G_{ij} + \frac{\sigma^2}{\mu} + O(\frac{1}{\sqrt{n}}),$$

i.e., λ_1 asymptotically has a normal distribution with mean $\mu n + \sigma^2/\mu$ (Furedi and Komlos, 1981). Our task is to derive the asymptotic behavior of the leading eigenvector when $\mu > 0$ under the assumption that the entries of G are i.i.d. random variables. We will first prove the theorem below, which deals with Gaussian random variables, and then extend it to bounded random variables:

Theorem 10 (Probabilistic Perron-Frobenius) *Let $G = g_{ij}$ be a real symmetric $n \times n$ matrix whose entries for $i \geq j$ are independent identically and normally distributed random variables with mean $\mu > 0$ and variance σ^2. Then, for any $\epsilon > 0$ there exist n_o such that for all $n > n_0$ the leading eigenvector* **v** *of G is positive with probability of at least $1 - \epsilon$.*

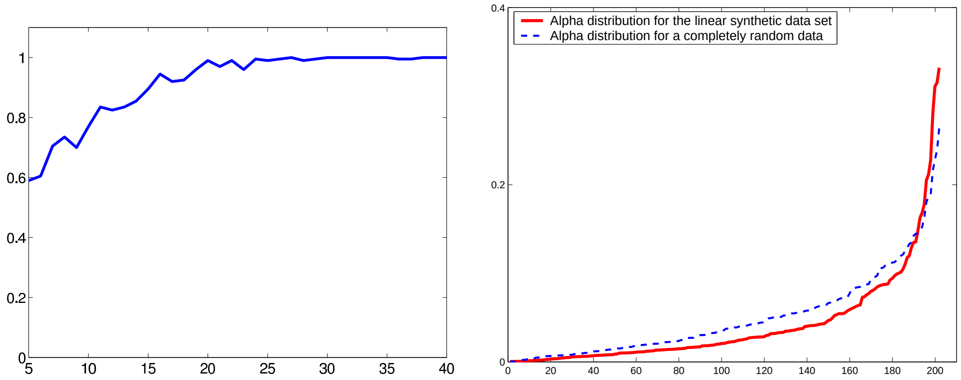
Proof: see appendix.

Fig. 2(a) displays a simulation result plotting the probability of positive leading eigenvector of G (with $\mu = 1/6$ and $\sigma = \sqrt{2}/6$) as a function of n. One can see that for $n > 20$ the probability becomes very close to 1 (above 0.99). Simulations with $\mu = 0.1$ and $\sigma = 1$ show that the probability is above 0.99 starting from $n = 500$.

Theorem 10 used independent Gaussian random variables as a model to the matrix G. This might seem a bit artificial, since the variables of the matrix G are dependent and bounded. While the independence assumption between all the elements in the upper triangular part of G is hard to remove, the use of Gaussian variables is not essential; as stated above the semi circle law holds for matrices with elements that are not necessarily Gaussian.

The only place where we actually used the "Gaussianity" property was in the assumed structure of the variable **g**. Since **g** contains normal i.i.d distributions, we deducted that $||\mathbf{g}|| = \Theta(\sqrt{n})$ and that the probability of $||\mathbf{g}|| \geq n^{3/4}$ decays exponentially. Instead of Gaussianity, we can use the property that the elements of the matrix G are bounded, and instead of Gaussian tail bounds we can use Hoeffding's tail inequality (Hoeffding, 1963). We will use the one sided inequality [2]

[2] This is the inequality one gets while proving Hoeffding's inequality. It differs from the canonical inequality in that the one sided case has a factor of 2 improvement.



(a) (b)

Figure 2: (a) Probability, as computed using simulations, of positive leading eigenvector of a symmetric matrix G with i.i.d elements drawn from the Gaussian distribution with $\mu = 1/6$ and $\sigma = \sqrt{2}/6$. The probability is very close to 1 starting from $n = 20$. (b) Positivity and sparsity demonstrated on the synthetic feature selection problem described in Section 6 (6 relevant features out of 202) and of a random data matrix. The alpha weight vector (sorted for display) comes out positive and sparse.

Lemma 11 (Hoeffding's one-sided tail inequality) *Let $X_1, X_2, .., X_n$ be bounded independent random variables such that $X_i \in [a_i, b_i]$. Then for $S_n = X_1 + X_2 + ... + X_n$ the following inequality holds*

$$Pr(S_n - \mathrm{E}\, S_n \geq t) \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right)$$

Using Hoeffding's lemma, the following lemma could be used to bound the norm of $\mathbf{g}$.

Lemma 12 *Let $\mathbf{g}$ be a random $n-$vector of i.i.d bounded variables, i.e., for each $i = 1..n$, $|\mathbf{g}_i| \leq M$. The following holds for some constant C:*

$$P(\|\mathbf{g}\|^2 \geq Dn^{1/2+\epsilon}) \leq \exp\left(-\frac{C^2D^2n^{2\epsilon}}{M^2}\right)$$

Proof: see appendix.

By letting $D = 1$ and $\epsilon = 1$, one gets that the probability $P(\|\mathbf{g}\| \geq n^{3/4}) = P(\|\mathbf{g}\|^2 \geq n^{3/2}) = P(\frac{1}{n}\|\mathbf{g}\|^2 \geq n^{1/2})$ is smaller than $e^{-\frac{c^2n^2}{M^2}}$. This is similar to the Gaussian case, and is sufficient to prove Theorem 10 in the case in which bounded variables are used instead of Gaussian variables.

To summarize the positivity issue, the weight vector α comes out positive due to the fact that it is the leading eigenvector of a matrix whose entries have a positive mean (Propositions 7 and 8). Theorem 10 made the connection between matrices which have the property of a positive mean and the positivity of its leading eigenvector in a probabilistic setting.

4.2 Sparsity

We move our attention to the issue of the sparsity of the weight vector α. It has been observed in the past that the key for sparsity lies in the positive combination of terms (cf. Lee and Seung, 1999) — therefore there is a strong, albeit anecdotal, relationship between the positivity of α and the sparsity feature. Below, we will establish a relationship between the spectral properties of the relevant and the irrelevant feature sets, and the sparsity of the feature vector.

Let M be the (normalized) data matrix consisting of n rows. Assume that the rows of the matrix have been sorted such that the first n_1 rows are relevant features, and the next $n_2 = n - n_1$ features are irrelevant. Let the matrix containing the first n_1 rows be noted as M_1, and let the matrix containing the rest of the rows be M_2, i.e, $M = [\frac{M_1}{M_2}]$.

We study the elements of the vector α that correspond to the irrelevant features to show that these elements have a small magnitude. If these n_2 weights are low, we can expect the effect of the associated features to be small. We will next tie the average of these values to the spectral properties of M_1 and M_2.

Recall the weight vector α is the first eigenvector of the matrix $G_{ij} = (\mathbf{m}_i^\top \mathbf{m}_j)\mathbf{m}_i^\top QQ^\top \mathbf{m}_j$, where $\mathbf{m}_i$ are the rows of the matrix M, and Q is a matrix containing k orthonormal columns q_i, $i = 1..k$. Let λ be the largest eigenvalue of G.

Lemma 13 (sum of irrelevant features' weight) *Using the above definitions, let $\gamma_i, i = 1..n_2$ be the eigenvalues of $M_2 M_2^\top$.*

$$\sum_{i=n_1+1}^{n} \alpha_i \le \sqrt{\frac{\sum_{i=1}^{n_2} \gamma_i^2}{\lambda}} \ .$$

Proof:

Note that if $\sum_{i=n_1+1}^{n} \alpha_i \le 0$ the lemma holds trivially. Let $\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]$ be the vector with n_1 zeros and n_2 ones.

$$\sum_{i=n_1+1}^{n} \alpha_i = \begin{bmatrix}0\\1\end{bmatrix}^\top \alpha = \sqrt{\begin{bmatrix}0\\1\end{bmatrix}^\top \alpha\alpha^\top \begin{bmatrix}0\\1\end{bmatrix}} \le \sqrt{\frac{\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]^\top G \left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]}{\lambda}} \ ,$$

where the last inequality follows from the spectral decomposition of the positive definite matrix G, to which α is an eigenvector with an eigenvalue of λ.

Let $\hat{G}$ be the matrix containing the point-wise squares of the elements of $MM^\top$, i.e., $\hat{G}_{ij} = (\mathbf{m}_i^\top \mathbf{m}_j)^2$.

Let $\hat{Q}$ be a matrix containing $n - k$ orthonormal columns that span the space orthogonal to Q. $(\hat{G} - G)$ has a structure similar to G, but with $\hat{Q}$ instead of Q, and is also positive definite. To see this notice that $QQ^\top + \hat{Q}\hat{Q}^\top = I_n$ and that the ij element of $(\hat{G} - G)$ is therefore given by

$$\hat{G}_{ij} - G_{ij} = (\mathbf{m}_i^\top \mathbf{m}_j)^2 - (\mathbf{m}_i^\top \mathbf{m}_j)\mathbf{m}_i^\top QQ^\top \mathbf{m}_j = (\mathbf{m}_i^\top \mathbf{m}_j)\mathbf{m}_i^\top \hat{Q}\hat{Q}^\top \mathbf{m}_j \ .$$

We have

$$\begin{bmatrix}0\\1\end{bmatrix}^\top G \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\\1\end{bmatrix}^\top \hat{G} \begin{bmatrix}0\\1\end{bmatrix} - \begin{bmatrix}0\\1\end{bmatrix}^\top (G - \hat{G}) \begin{bmatrix}0\\1\end{bmatrix} \le$$

$$\le \begin{bmatrix}0\\1\end{bmatrix}^\top \hat{G} \begin{bmatrix}0\\1\end{bmatrix} = ||M_2 M_2^\top||_F^2 = \sum_{i=1}^{n_2} \gamma_i^2 \ .$$

The lemma follows. □

The denominator in the bound ($\sqrt{\lambda}$) is exactly the quantity that our algorithms maximize. The higher this value, the tighter the bound. In the ideal case, almost all of the energy in the features is contained in the space spanned by the columns of Q. Let $\left[\frac{1}{0}\right]$ be the vector of n_1 ones, followed by n_2 zeros. We have:

$$\lambda = \alpha^\top G\alpha \geq \frac{\left[\frac{1}{0}\right]^\top G \left[\frac{1}{0}\right]}{n_1} \sim \frac{\left[\frac{1}{0}\right]^\top \hat{G} \left[\frac{1}{0}\right]}{n_1} = \frac{||M_1 M_1^\top||_F^2}{n_1} \quad .$$

The bound will be tightest if all relevant features have high correlations. In this case, we can expect $\sqrt{\lambda}$ to be linear in n_1. Therefore the addition of more relevant features reduces the weights of the irrelevant features.

Without any assumption about the entries of the data matrix M, we cannot say much on the numerator of the bound in Lemma 13. However, by using random matrices, we can qualitatively evaluate this bound.

The numerator of the bound contains the term $\sum_{i=1}^{n_2} \gamma_i^2$, which is just the squared Frobenius norm of $M_2 M_2^\top$. Let $W_2 = M_2 M_2^\top$, $||W_2||_F^2 = trace(W_2 W_2^\top) = trace(W_2^2)$. The expectation of this expression (where M_2 is drawn from a random distribution), normalized by the number of rows in M_2, is called the *second moment of* W_2. More generally, if A is a random matrix of size $n \times n$, the k moment of it is defined as $m_k = \frac{1}{n} trace(A^k)$. For large n this definition coincides with the moments of the distribution of the eigenvalues of the matrix A.

Consider now matrices W of the form $W = \frac{1}{q} M M^\top$, where M is an $n \times q$ random matrix with zero mean (weakly) independent elements with a variance of 1. These matrices are called Wishart matrices. Note that the elements in the matrix M need not be Gaussian. The rows of the matrix M are not explicitly normalized. However, the scale of $\frac{1}{q}$ can be thought of as a scale of $\frac{1}{\sqrt{q}}$ for each element of M, and due to the central limit theorem we can expect for large enough values of q to have the mean of each row approximately zero and the norm of each row approximately 1. Hence, Wishart matrices well approximate our data matrices, if we are willing to assume that the elements of our data matrices are independent. For the bulk of irrelevant features, this may be a reasonable assumption.

For large n, the moments of the Wishart matrices are well approximated by the Narayana polynomials $m_k = \sum_{j=0}^{k-1} \frac{(n/j)^j}{j+1} \binom{k}{j} \binom{k-1}{j}$. In particular, the second moment is given by $1 + n/q$. Since the moment is the appropriate trace scaled by $\frac{1}{n}$, we can expect $\sqrt{\sum_{i=1}^{n_2} \gamma_i^2}$ to behave similarly to $\sqrt{n_2(1 + n_2/q)}$.

Therefore, the rate in which the bound on the sum of squares of weights of irrelevant features grow is mostly linear. The implication is that the $Q - \alpha$ algorithm is robust to many irrelevant features: to a first approximation, the bound on the average squared weight of an irrelevant feature remains mostly the same, as the number of irrelevant features increases.

In Sec. 6 we will present a number of experiments, both with synthetic and real data. Fig. 2(b) shows the weight vector α for a random data matrix M, and for a synthetic experiment (6 relevant features out of 202). One can clearly observe the positivity and sparsity of the recovered weight vector — even for a random matrix.

4.3 Sparsity and generalization

The sparsity of the returned vector of weights does more than just directly ensure that the irrelevant features are left out; it also helps the generalization ability of the returned kernel by lowering the trace of the kernel matrix.

Recall that in our optimization scheme, the vector of weights α has a norm of 1, and is expected to have all positive elements. For norm-1 vectors, the sum $\sum_i \alpha_i$ is highest when the elements of the vector are evenly distributed. Due to the sparsity of the returned vector of weights, we can expect the above sum to be much lower than what we get with a uniform weighting of the data.

Consider the matrix A_α, the linear kernel matrix based on the weights returned by the $Q - \alpha$ algorithm. A_α, which equals $\sum \alpha_i m_i m_i^\top$, is a weighted sum of rank-one matrices. Since our features are normalized to have norm-1, each such rank-one matrix $m_i m_i^\top$ has a trace of 1. Therefore, the trace of the kernel matrix A_α is exactly the sum of the elements of the vector α.

From the discussion above, the trace of the kernel matrix returned by the $Q - \alpha$ algorithm is expected to be low. This is exactly the criteria for a good kernel matrix expressed by "the trace bounds." The trace bounds are Rademacher complexity type of error bounds for classifiers that are linear combinations of kernel functions (Bousquet and Herrmann, 2003). These bounds relate the trace of the kernel matrix used for training with the generalization error of the classifiers that were trained. The lower the trace, the lower the bound on the difference between the testing error and the training error.

Although there is no immediate analog to the concept of generalization error in the unsupervised case, we can expect a similar criteria to hold for this case as well. A good kernel matrix for unsupervised learning should support the separation given by the set of true underlying labels (although unknown). It should not, however, support any random labeling. This is exactly what is measured by the Rademacher process: how well does the class of functions used for learning separate random partitions of the training set.

In supervised learning, feature selection can be a major cause of over fitting. Consider the common case where the number of features is much larger than the number of examples. In this case, it might be possible to construct a classifier with a very low training error, which employs only few selected features. This reliance on a small portion on the data when making the classification leads to poor generalization performance. This problem was pointed out, for example, by Long and Vega (2003), who suggested, in their simplest and most effective solution ("AdaBoost-NR"), to encourage redundancy in the pool of participating features. The $Q - \alpha$ algorithm, as a result of optimizing the cost function subject to the constraint that the norm of the α vector is one, has a similar property. It prefers to divide high weights between a group of correlated features, rather than to pick one promising feature out of this group and assign it a higher weight.

5. Representing Higher-order Cumulants using Kernel Methods

The information on which the $Q - \alpha$ method relies on to select features is contained in the matrix G. Recall that the criterion function underlying the $Q - \alpha$ algorithm is a sum over all pairwise feature vector relations:

$$trace(Q^\top A_\alpha^\top A_\alpha Q) = \alpha^\top G \alpha,$$

where G is defined such that $G_{ij} = (\mathbf{m}_i^\top \mathbf{m}_j)\mathbf{m}_i^\top Q Q^\top \mathbf{m}_j$. It is apparent that feature vectors interact in pairs and the interaction is *bilinear*. Consequently, cumulants of the original data matrix M

which are of higher order than two are not being considered by the feature selection scheme. For example, if M were to be decorrelated (i.e., $MM^\top$ is diagonal) the matrix G would be diagonal and the feature selection scheme would select only a single feature rather than a feature subset.

In this section we employ the so called "kernel trick" to allow for cumulants of higher orders among the feature vectors to be included in the feature selection process. Kernel methods in general have been attracting much attention in the machine learning literature — initially with the introduction of the support vector machines (Vapnik, 1998) and later took a life of their own (see Scholkopf and Smola, 2002). The common principle of kernel methods is to construct nonlinear variants of linear algorithms by substituting inner-products by nonlinear kernel functions. Under certain conditions this process can be interpreted as mapping of the original measurement vectors (so called "input space") onto some higher dimensional space (possibly infinitely high) commonly referred to as the "feature space" (which for this work is an unsuccessful choice of terminology since the word "feature" has a different meaning). Mathematically, the kernel approach is defined as follows: let $\mathbf{x}_1, ..., \mathbf{x}_l$ be vectors in the input space, say R^q, and consider a mapping $\phi(\mathbf{x}) : R^q \rightarrow \mathcal{F}$ where $\mathcal{F}$ is an inner-product space. The kernel-trick is to calculate the inner-product in $\mathcal{F}$ using a kernel function $k : R^q \times R^q \rightarrow R$, $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$, while avoiding explicit mappings (evaluation of) $\phi()$. Common choices of kernel selection include the d'th order polynomial kernels $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j + c)^d$ and the Gaussian RBF kernels $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{1}{2\sigma^2}\|\mathbf{x}_i - \mathbf{x}_j\|^2)$. If an algorithm can be restated such that the input vectors appear in terms of inner-products only, one can substitute the inner-products by such a kernel function. The resulting kernel algorithm can be interpreted as running the original algorithm on the space $\mathcal{F}$ of mapped objects $\phi(\mathbf{x})$. Kernel methods have been applied to the support vector machine (SVM), principal component analysis (PCA), ridge regression, canonical correlation analysis (CCA), QR factorization and the list goes on. We will focus below on deriving a kernel method for the $Q - \alpha$ algorithm.

5.1 Kernel $Q - \alpha$

We will consider mapping the rows $\mathbf{m}_i^\top$ of the data matrix M such that the rows of the mapped data matrix become $\phi(\mathbf{m}_1)^\top, ..., \phi(\mathbf{m}_n)^\top$. Since the entries of G consist of inner-products between pairs of mapped feature vectors, the interaction will be no longer bilinear and will contain higher-order cumulants whose nature depends on the choice of the kernel function.

Replacing the rows of M with their mapped version introduces some challenges before we could apply the kernel trick. The affinity matrix $A_\alpha = \sum_i \alpha_i \phi(\mathbf{m}_i)\phi(\mathbf{m}_i)^\top$ cannot be explicitly evaluated because A_α is defined by *outer-products* rather than inner-products of the mapped feature vectors $\phi(\mathbf{m}_i)$. The matrix Q holding the eigenvectors of A_α cannot be explicitly evaluated as well and likewise the matrix $Z = A_\alpha Q$ (in step 4). As a result, kernelizing the $Q - \alpha$ algorithm requires one to represent α without explicitly representing A_α and Q both of which were instrumental in the original algorithm. Moreover, the introduction of the kernel should be done in such a manner to preserve the key property of the original $Q - \alpha$ algorithm of producing a sparse solution.

Let $V = MM^\top$ be the $n \times n$ matrix whose entries are evaluated using the kernel $v_{ij} = k(\mathbf{m}_i, \mathbf{m}_j)$. Let $Q = M^\top E$ for some $n \times k$ (recall k being the number of clusters in the data) matrix E. Let $D_\alpha = diag(\alpha_1, ..., \alpha_n)$ and thus $A_\alpha = M^\top D_\alpha M$ and $Z = A_\alpha Q = M^\top D_\alpha V E$. The matrix Z cannot be explicitly evaluated but $Z^\top Z = E^\top V D_\alpha V D_\alpha V E$ can be evaluated. The matrix G can be expressed with regard to E instead of Q:

$$G_{ij} = (\phi(\mathbf{m}_i)^\top \phi(\mathbf{m}_j))\phi(\mathbf{m}_i)^\top Q Q^\top \phi(\mathbf{m}_j)$$

$$
\begin{aligned}
&= k(\mathbf{m}_i,\mathbf{m}_j)\phi(\mathbf{m}_i)^\top(M^\top E)(M^\top E)^\top\phi(\mathbf{m}_j)\\
&= k(\mathbf{m}_i,\mathbf{m}_j)\mathbf{v}_i^\top EE^\top\mathbf{v}_j
\end{aligned}
$$

where $\mathbf{v}_1,...,\mathbf{v}_n$ are the columns of V. Step 5 of the $Q-\alpha$ algorithm consists of a QR factorization of Z. Although Z is uncomputable it is possible to compute R and R^{-1} directly from the entries of $Z^\top Z$ without computing Q using the Kernel Gram-Schmidt described by Wolf and Shashua (2003). Since $Q = ZR^{-1} = M^\top D_\alpha VER^{-1}$ the update step is simply to replace E with ER^{-1} and start the cycle again. In other words, rather than updating Q we update E and from E we obtain G and from there the newly updated α. The kernel $Q-\alpha$ is summarized below:

Definition 14 (Kernel $Q-\alpha$) *Let M be an uncomputable matrix with rows $\phi(\mathbf{m}_1)^\top,...,\phi(\mathbf{m}_n)^\top$ where $\phi():R^n \longrightarrow \mathcal{F}$ is a mapping from input space to a feature space and which is endowed with a kernel function $\phi(\mathbf{m}_i)^\top\phi(\mathbf{m}_j) = k(\mathbf{m}_i,\mathbf{m}_j)$. Therefore the matrix $V = MM^\top$ is a computable $n\times n$ matrix. Let $E^{(0)}$ be an $n\times k$ matrix selected such that $M^\top E^{(0)}$ has orthonormal columns. Perform the following steps through a cycle of iterations with index $r = 1,2,...$*

1. *Let $G^{(r)}$ be a $n\times n$ matrix whose (i,j) components are $k(\mathbf{m}_i,\mathbf{m}_j)\mathbf{v}_i^\top E^{(r-1)}E^{(r-1)^\top}\mathbf{v}_j$.*
2. *Let $\alpha^{(r)}$ be the largest eigenvector of $G^{(r)}$, and let $D^{(r)} = diag(\alpha_1^{(r)},...,\alpha_n^{(r)})$.*
3. *Let $Z^{(r)}$ be an uncomputable matrix*

$$Z^{(r)} = (M^\top D^{(r)}M)(M^\top E^{(r-1)}) = M^\top D^{(r)}VE^{(r-1)}.$$

   Note that $Z^{(r)^\top}Z^{(r)}$ is a computable $k\times k$ matrix.

4. *$Z^{(r)} \xrightarrow{QR} QR$. It is possible to compute directly R, R^{-1} from the entries of $Z^{(r)^\top}Z^{(r)}$ without explicitly computing the matrix Q (see (Wolf and Shashua, 2003)).*
5. *Let $E^{(r)} = E^{(r-1)}R^{-1}$.*
6. *Increment index r and go to step 1.*

The result of the algorithm is the weight vector α and the design matrix G which contains all the data about the features.

6. Experiments

We have validated the effectiveness of the proposed algorithms on a variety of datasets. Our main focus in the experiments below is in the unsupervised domain, which has received much less attention in the feature selection literature than the supervised one.

Synthetic Data

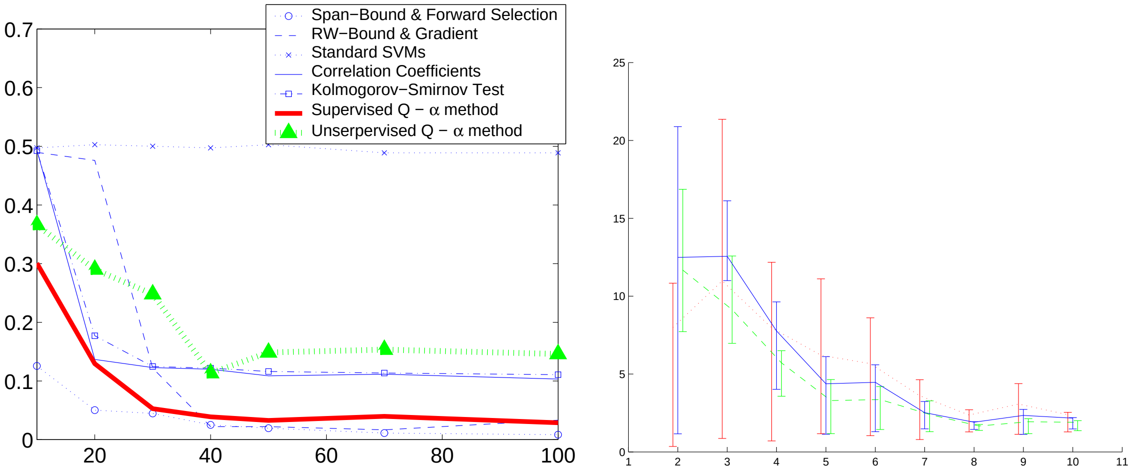
We compared the $Q-\alpha$ algorithm with three classical filter methods (Pearson correlation coefficients, Fisher criterion score and the Kolmogorov-Smirnoff test), standard SVM and the wrapper method using SVM of Weston et al. (2001). The data set we used follow precisely the one described by Weston et al., which was designed for supervised 2-class inference. Two experiments were designed, one with 6 relevant features out of 202 referred to as “linear” problem, and the other experiment with 2 relevant features out of 52 designed in a more complex manner and referred to as “non-linear” problem. In the linear data the class label $y\in\{-1,1\}$ was drawn at equal probability.

The first six features were drawn as $x_i = yN(i, 1)$, $i = 1..3$, and $x_j = N(0, 1)$, $j = 4..6$ at probability 0.7, otherwise they were drawn as $x_i = N(0, 1)$, $i = 1..3$, and $x_j = yN(i - 3, 1)$, $j = 4..6$. The remaining 196 dimensions were drawn from $N(0, 20)$. The reader is referred to (Weston et al., 2001) for details of the non-linear experiment. We ran $Q - alpha$ on the two problems once with known classes (supervised version) and with unknown class labels (unsupervised version). In the supervised case the selected features were used to train an SVM and in the unsupervised case the class labels were not used for the $Q - \alpha$ feature selection but were used for the SVM training. The unsupervised test appears artificial but is important for appreciating the strength of the approach as the results of the unsupervised are only slightly inferior to the supervised test. For each size of training set we report the average test error on 500 samples over 30 runs. In Fig. 3(a) we *overlay* the $Q - \alpha$ results (prediction error of the SVM on a testing set) on the figure obtained by Weston et al.. The performance of the supervised $Q - \alpha$ closely agrees with the performance of the wrapper SVM feature selection algorithms. The performance of the unsupervised version does not fall much behind.

Since our method can handle more than two classes we investigated the scaling-up capabilities of the algorithm as we increase the number of classes in an unsupervised setting. For $n_c = 2, 3, ...$ classes we sampled n_c cluster centers in 5D space (5 coordinates per center) in the 5D cube where each cluster center coordinate is uniformly sampled in the interval $[-1, 1]$. For each cluster we also uniformly samples a diagonal covariance matrix with elements taken from the interval $[0, .02]$. Around each of the n_c class centers we sampled $\lceil \frac{60}{n_c} \rceil$ points according to a normal distribution whose mean is the class center and with a the random covariance matrix. We added 120 additional coordinates drawn similarly around n_c centers sampled uniformly inside the 120D hypercube with edges of length 2, according to the same rules. Each such added coordinate was permuted by a random permutation to break the correlation between the dimensions. Thus each of the 60 points lives in a 125-dimensional space out of which only the first five dimensions are relevant. We ran the $Q - \alpha$ algorithm on the data matrix and obtained the weight vector α and computed the sparsity gap - i.e the ratio between the average weight of the first five features and the average weight of the rest 120 features. Ideally the ratio should be high if the algorithm consistently succeeds in selecting the first three coordinates as the relevant ones.

Fig. 3(b) illustrates the results of this experiment in a graph whose x-axis runs over the number of classes k and the y-axis displays the sparsity gap (the ratio discussed above). Each experiment was repeated 20 times and the results in the plot are the average of the 20 runs and the 25 and 75 percentiles. In general the error bars for small number of classes are large indicating that some experiments are much more difficult than others. This is probably a results of the cluster centers being close to one another in some of the experiments.

There are three plots on the graph. The solid blue describes the result obtained when choosing $k = n_c$. For small number of classes this gives the best results. The dashed green plot describes the results obtained while choosing $k = n_c + 2$. This choice seems to result with a smaller variance between experiments. The explanation might be that variance is a results of the fact that in some experiments the cluster centers are close, making the separation difficult. Taking a large value of k captures more complex details about the cluster structure. For example: when two clusters have close centers the resulting distribution might look like one strong cluster in the middle, and some cluster tails around it. The red plot is the one obtained when under estimating the number of clusters and taking $k = \max(1, n_c - 2)$. This has the largest variance, but the best (in average) when the number of clusters is large. The reason might be that focusing on the clusters which are well



(a) (b)

Figure 3: (a) Comparison of feature selection methods following (Weston et al., 2001). Performance curves of $Q-\alpha$ were overlaid on the figure adapted from (Weston et al., 2001). The x-axis is the number of training points and the y-axis is the test error as a fraction of test points. The thick solid lines correspond to the $Q-\alpha$ supervised and unsupervised methods (see text for details). (b) Performance of a test with five relevant features and 120 irrelevant ones with n_c clusters represented by the x-axis of the graph. The y-axis represents the sparsity gap (see text for details).The three graphs are solid blue for $k=n_c$, dashed green for $k=n_c+2$ and dotted red for $k=\max(1,n_c-2)$. One can see that the unsupervised $Q-\alpha$ sustained good performance up to 6 classes in this settings.

separated is better than trying to capture information from all clusters. This is however, a "risky" strategy leading to a large variance.

One can see that the algorithm performed well until $k=6$. After that the sparsity ratio is still larger than one most of the time, but separation is not easy. It is possible to get better performance in average by underestimating k by more than 2 at the price of a higher variance. Good performance up to 6 clusters and a sparsity gap around $5-10$ are not "magical numbers". For other feature selection problems (e.g., a different number of points per cluster, other sampling probabilities, etc.) we can get good performance for more classes or for less depending on the complexity of the problem.

Real Image Unsupervised Feature Selection

The strength of the $Q-\alpha$ method is that it applies for unsupervised settings as well as supervised. An interesting unsupervised feature selection problem in the context of visual processing is the one of automatic selection of relevant features which discriminate among perceptual classes. Assume one is given a collection of images where some of them contain pictures of a certain object class (say, green frogs, the *Rana clamitans* specie) and other images contain pictures of a different class of objects (say, American toads) — see Fig. 4. We would like to automatically, in an unsupervised manner, select the relevant features such that a new picture could be classified to the correct class membership.

The features were computed by matching patches of equal size of 20×20 pixels in the following manner. Assuming that the object of interest lies in the vicinity of the image center, we defined 9 "template" patches arranged in a 3×3 block centered at the image. for example, in one experiment, we had 27 images (18 from one class and 9 from the other), which in turn defines $27 * 9 = 243$ feature coordinates. Each image was sampled by 49 "candidate" patches (covering the entire image) where each of the 243 template patches was matched against the 49 patches in its respective image and the score of the best match was recorded in 243×27 data matrix. The matching between a pair of patches was based on L_1-distance between the respective color histograms in HSV space. We ran the $Q - \alpha$ algorithm with $k = 2$. The resulting α weight vector forms a feature selection from which we create a submatrix of data points and construct its affinity matrix and the associated matrix of eigenvectors Q. The rows of the Q matrix were clustered using k-means into two clusters.

This experiment was done in an unsupervised settings. As a measure of performance we used the percent of samples with labels matching the correct labeling (the maximum over the two flips of the class labels). Performance varied between 80% to 90% correct assignments over many experiments over several object classes (including elephants, sea elephants, and so forth). Images where taken from CalPhotos: Animals (http://elib.cs.berkeley.edu/photos/fauna/). For each class we took all images in which the animal appears, e.g., we removed all tadpoles images from the green frog class. This performance was compared to spectral clustering using all the features (243 in the above examples) which provided a range of 55% to 65% correct classification.

Fig. 5(a) and Fig. 5(b) show the 20 most relevant templates selected for the two classes, and Fig. 5(c) shows the alpha values. Note that the α weights are positive as predicted from Theorem 10 and that only few of the features have very high weights.

KERNEL $Q - \alpha$ EXPERIMENTS

One of the possible scenarios for which a polynomial (for example) kernel is useful is when hidden variables affect the original feature measurements and thus create non-linear interactions among the feature vectors. We consider the situation in which the original measurement matrix M is multiplied, element wise, with a hidden variable matrix whose entries are ± 1. The value of the hidden state was changed randomly every 8 measurements and independently for each feature. This scheme simulates measurements taken in "sessions" where a session lasts for 8 sample data points. As a result, the expectation of the inner product between any two feature vectors is zero yet any two feature vectors contain higher-order interactions which could come to bear using a polynomial kernel.

The kernel we used in this experiment was a sum of second-order polynomial kernels each over a portion of 8 entries of the feature vector:

$$k(\mathbf{m}_i, \mathbf{m}_j) = \sum_k (m_i^{k\top} m_j^k)^2,$$

where m_i^k represents the k'th section of 8 successive entries of the feature vector $\mathbf{m}_i$. The original data was composed out 120 sample points with 60 coordinates out of which 12 were relevant and 48 were irrelevant. The relevant features were generated from three clusters, each containing 40 points. The points of a cluster were Normally distributed with a mean vector drawn uniformly from the unit hypercube in $\mathcal{R}^{12}$ and with a diagonal covariance matrix with entries uniformly distributed in the range [λ,2λ], where λ is a parameter of the experiment. A 2D slice out of the relevant 12

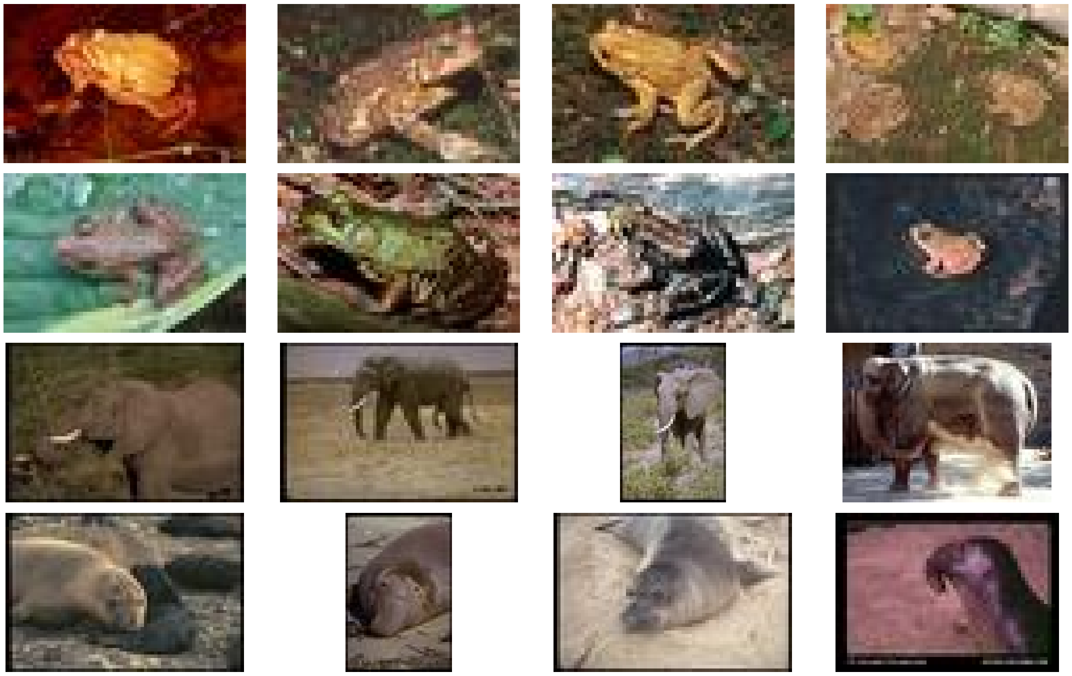
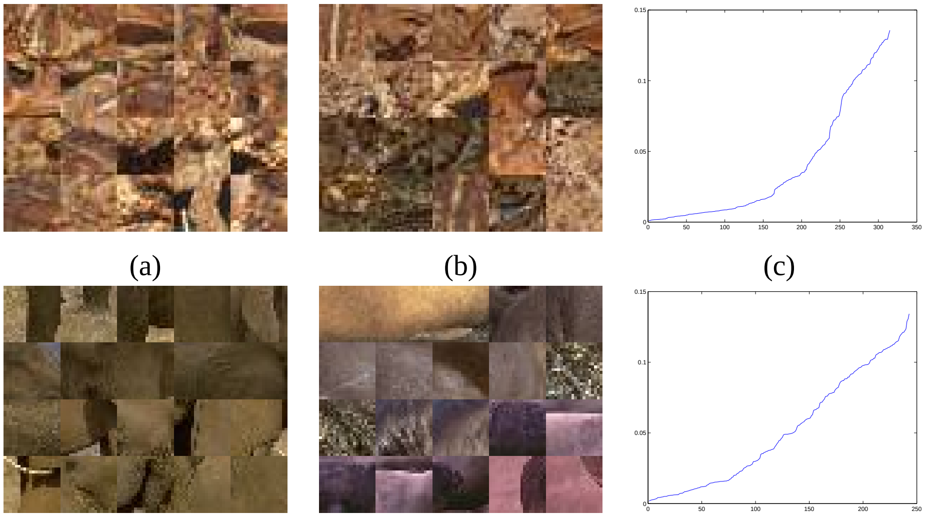


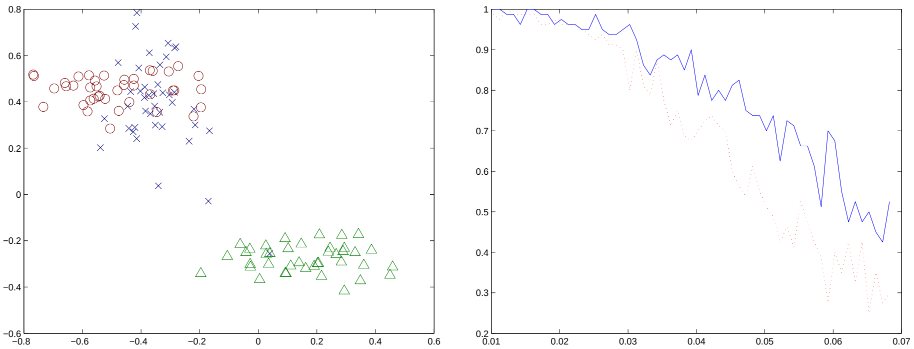
Figure 4: Image samples of several animal classes — American toad (top row) and Green frogs (*Rana clamitans*), elephants, and sea elephants. The objects appear in various positions, illumination, context and size.



(a) (b) (c)

(d) (e) (f)

Figure 5: Unsupervised feature selection for automatic object discrimination from images. (a),(b) the first 20 features from pictures containing the American frog and the Green frog ranked by the α weight vector. (c) the (sorted) α values. (d),(e),(f) similar to the elephant and sea elephant.



(a) (b)

Figure 6: (a) 2D slice out of the relevant features in the original data matrix used in the synthetic experiment, showing three clusters. (b) A graph showing the success rate for the 2nd order polynomial kernel (solid blue), and for a preprocessing of the data (dashed red). The results are shown over the parameter λ specifying the variance of the original dataset (see text). The success rate of the regular $Q - \alpha$ algorithm was constantly zero and is not shown.

dimensions is shown in figure 6(a). The irrelevant features were generated in a similar manner, where for each irrelevant feature the sample points were permuted independently in order to break the interactions between the irrelevant features. This way it is impossible to distinguish between a single relevant feature and a single irrelevant feature.

We considered an experiment to be successful if among the 12 features with the highest α values, at least 10 were from the relevant features subset. The graph in figure 6(b) shows the success rate for the kernel $Q - \alpha$ algorithm averaged over 80 runs. It also shows, for comparison, the success rate for experiments conducted by taking the square of every element in the measurements matrix followed by running the original $Q-\alpha$ algorithm. The success rate for the original $Q-\alpha$ algorithm on the unprocessed measurements was constantly zero and is not shown in the graph.

GENOMICS

Synthetic data We have tested our algorithm against the synthetic model of gene expression data (“microarrays”) given in (Ben-Dor et al., 2001). This synthetic model has 6 parameters m, a, b, e, d, s, explained below. a samples are drawn from class A, and b samples are drawn from class B. Each sample has m dimensions - em samples are drawn randomly using the distribution $N(0, s)$. The rest of the $(1 - e)m$ features are drawn using either $N(\mu_A, \mu_A s)$ or $N(\mu_B, \mu_B s)$, depending on the class of the sample. The means of the distributions μ_A and μB are uniformly chosen from the interval $[-1.5d, 1.5d]$.

In (Ben-Dor et al., 2001) the parameters of the model were estimated to best fit the gene expressions of the leukemia dataset: $m = 600, a = 25, b = 47, e = 0.72, d = 555, s = 0.75$ [3]. Similarly to (Ben-Dor et al., 2001), we varied one of the parameters m, d, e, s while fixing the other

[3] the leukemia dataset has over 7000 gene expressions but contains much redundancy. (Ben-Dor et al., 2001) estimated the effective number of features to be 600 and we follow their choice parameters to allow comparison. Note below that the problem becomes easier as the number of features increase as long as the ratio of relevant features is fixed

parameters to the values specified above. This enabled us to compare the performance of the $Q - \alpha$ algorithm to the performance of their Max-Surprise algorithm (MSA).

Our algorithm was completely robust to the number of features m. It always chose the correct features using as few as 5 features. MSA needed at least 250 features, since it used the redundancy in the features in order to locate the informative features. Both algorithms are invariant to the distance between the means of the distributions determined by d, and perform well for $d \in [1, 1000]$. The percentage of irrelevant features, e, can reach 95% for MSA and 99.5% for our algorithm. Such performance suggests that the data set is not very difficult.

The parameter s effects the spread of each class. While MSA was able to handle values of s reaching 2, our algorithm was robust to s, and was at least 30 times more likely to choose a relevant feature than an irrelevant one, even for $s > 1000$.

Real genomics datasets We evaluated the performance of the $Q - \alpha$ algorithm for the problem of gene selection on four datasets containing treatment outcome or status studies (see Wolf et al., 2005, for the full report). The first was a study of treatment outcome of patients with diffuse large cell lymphoma (DLCL), referred to as “lymphoma” (Shipp et al., 2002). The dimensionality of this dataset was $7,129$ and there were 32 samples with good successful outcome and 26 with unsuccessful outcome. The second was a study of treatment outcome of patients with childhood medulloblastomas (Pomeroy et al., 2002), referred to as “brain”. The dimensionality of this dataset was $7,129$ and there were 39 samples with good successful outcome and 21 with unsuccessful outcome. The third was a study of the metastasis status of patients with breast tumors (van ’t Veer et al., 2002), referred to as “breast met”. The dimensionality of this dataset was $24,624$ and there were 44 samples where the patients were disease free for 5 years after onset and 34 samples where the tumors metastasized within five years. The fourth is an unpublished study of breast tumors (Ramaswamy) for which corresponding lymph nodes either were cancerous or not, referred to as “lymph status”. The dimensionality of this dataset is $12,600$ with 47 samples positive for lymph status and 43 negative for lymph status.

For the four datasets with label information classification accuracy was used as a measure of the goodness of the (unsupervised) $Q - \alpha$ algorithm. We compared the leave-one-out error on these datasets with that achieved by both supervised and unsupervised methods of gene selection. The supervised methods used were signal-to-noise (SNR) (Golub et al., 1999), radius-margin bounds (RMB) (Chapelle et al., 2002; Weston et al., 2001), and recursive feature elimination (RFE) (Guyon et al., 2002). The unsupervised methods used were PCA and gene shaving (GS) (Hall, 2000). In the unsupervised mode the class labels were ignored — and thus in general one should expect the supervised approaches to produce superior results than the unsupervised ones. A linear support vector machine classifier was used for all the gene selection methods Parameters for SNR, RFE, and RMB were chosen to minimize the leave-one-out error. For the $Q - \alpha$ algorithm we took $k = 6$ for all experiments, to allow for more complex structures than just two clusters. For the breast me dataset and for the lymph status dataset we took only the first $7,000$ features to reduce the computation complexity.

A summary of the results appear in table 1. The $Q - \alpha$ algorithm considerably out-performs all other unsupervised methods. Furthermore, and somewhat intriguing, is that the $Q - \alpha$ algorithm is competitive with the other supervised algorithm (despite the fact that the labels were not taken into account in the course of running the algorithm) and performs *significantly better* on the lymph status of breast tumors as compared to all other gene selection approaches — including the supervised methods.

Method	brain	lymph status [1]	breast met.[1]	lymp-. homa
RAW	32	44	34	27
PCA5	22	47	33	40
PCA10	26	47	26	27
PCA20	25	47	25	29
PCA30	31	47	31	33
PCA40	31	47	31	33
PCA50	30	47	30	33
GS5	20	45	32	33
GS10	24	43	31	30
GS20	28	47	32	31
GS30	30	44	33	33
$Q - \alpha$	15	19	22	15
SNR	16	42	29	18
RFE	14	38	26	14
RMB	13	39	24	14

Table 1: The table entries show the Leave-one-out classification errors for the supervised and unsupervised algorithms on the various datasets. In both PCAN and GSN the number N the number of components used. [1] Only the first $7,000$ genes were used.

7. Conclusions

In this work we presented an algebraic approach to variable weighting, which is based on maximizing a score based on the spectral properties of the kernel matrix. The approach has the advantage of being suitable to unsupervised feature selection, but can also be applied in the supervised settings.

It is interesting to compare the algebraic approach presented in this work to probabilistic approaches which take a ”holistic” view of the data such as the information bottleneck (Tishby et al., 1999) and the infomax (Linsker, 1988; Vasconcelos, 2003). The gap that exists between the algebraic and the probabilistic tools of machine learning make a direct comparison to information-based feature selection criteria a subject for future work. However, it is evident that algebraic methods have the advantages of not requiring the estimation of probability distributions, of being more suitable for application on continuous data and, in general, for being easier to optimize for. We conducted a limited experimental comparison to an information-bottleneck method called sufficient dimensionality-reduction (Shashua and Wolf, 2004), and more work is required.

The emergence of sparsity and positiveness in our simple least square optimization function, is a surprising result, that might indicated the possibility of similar results in other algebraic methods of machine learning. For example, it might be interesting to examine if the vector of examples' weights returned by the regularized least squares classification method (Rifkin et al., 2003) would be considered sparse by our definition of sparseness. Regularized least squares method are similar to Support Vector Machines in many ways, only SVMs are known to produce sparse solutions.

As a last remark, we would like to point out that the methods presented in this work are extremely flexible and can be extended. For example, to the case of semi-supervised learning (Shashua and Wolf, 2004).

Acknowledgments

We would like to thank anonymous JMLR reviewers for many useful suggestions. We would like to thank Ofer Zeitouni, Michael Ben-Or and Alex Samorodnitsky for assistance with the proof of Theorem 1. We would like to thank Sayan Mukherjee for providing many of the experimental results on the genomics datasets.

A.S. thanks the financial contribution from Israeli Science Foundation grant No. 189/03 and contribution from the German Israel Foundation grant No. 773/2004.

References

H. Almuallim and T .G Dietterich. Learning boolean concepts in the presence of many irrelevant features. *AI*, 69(1-2):279–305, 1991.

A. Ben-Dor, N. Friedman, and Z. Yakhini. Class discovery in gene expression data. In *RECOMB*, 2001.

A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *AI*, 97 (1-2):245–271, 1997.

O. Bousquet and D.J.L. Herrmann. On the complexity of learning the kernel matrix. In *NIPS*, 2003.

P.S. Bradley and O.L. Mangasarian. Feature selection via concave minimization and support vector machines. In *ICML*, 1998.

M. Brand and K. Huang. A unifying theorem for spectral embedding and clustering. In *Ninth Int. Workshop on AI and Statistics*, 2003.

O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1-3):131–159, 2002.

F.R.K. Chung. *Spectral Graph Theory*. AMS, 1998.

Z. Furedi and J. Komlos. The eigenvalues of random symmetric matrices. *Combinatorica*, 1(3): 233–241, 1981.

L.E. Gibbons, D.W. Hearn, P.M. Pardalos, and M.V. Ramana. Continuous characterizations of the maximum clique problem. *Math. Oper. Res*, 22:754–768, 1997.

G. Golub and C.F.V. Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, 1996.

T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, and E.S. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286: 531–537, 1999.

I. Guyon and A. Elissef. An introduction to variable and feature selection. *Journal of Machine Learning Research, Special issue on special feature*, 3:389–422, 2003.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002.

K. Hall. An r-dimensional quadratic placement algorithm. *Management Science*, 17(3):219–229, 1970.

K. Hall. 'gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biology*, 1(2):1–21, 2000.

W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

K. Kira and L. Rendell. A practical approach to feature selection. In *Ninth Int, Workshop on Machine Learning*, 1992.

J. Kivinen and M.K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Journal of Information and Computation*, 132(1):1–63, 1997.

R. Kohavi and G. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2): 273–324, 1997.

P. Langley and W. Iba. Average-case analysis of a nearest neighbor algorithm. In *13th Int. Joint Conf. on Artificial Intelligence*, 1993.

D.D. Lee and H.S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

D.D. Lewis. Feature selection and feature extraction for text categorization. In *Speech and Natural Language Workshop*, 1992.

R. Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988. ISSN 0018-9162. doi: http://dx.doi.org/10.1109/2.36.

P.M. Long and V.B. Vega. Boosting and microarray data. *Machine Learning*, 52:31–44, 2003.

M.L. Mehta. *Random Matrices*. Academic Press, 1991.

B. Mohar. The laplacian spectrum of graphs. In Y. Alavi et al., editor, *Graph Theory, Combinatorics and Applications*. Wiley, New York, 1991.

T.S. Motzkin and E.G. Straus. Maxima for graphs and a new proof of a theorem by turan. *Canadian Journal of Math.*, 17:533–540, 1965.

A.Y. Ng, M.I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2001.

B.A. Olshausen and D.J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(13):607–609, 1996.

M. Pavan and M. Pelillo. A new graph-theoretic approach to clustering and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

P. Perona and W.T. Freeman. A factorization approach to grouping. In *European Conference of Computer Vision (ECCV)*, 1998.

S.L. Pomeroy, P. Tamayo, M. Gaasenbeek, L.M. Sturla, M. Angelo, M.E. McLaughlin, J.Y.H. Kim, L.C. Goumnerova, P.McL. Black, C. Lau, J.C. Allen, D. Zagzag, J.M.Olson, T. Curran, C. Wetmore, J.A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A.Califano, G.Stolovitzky, D.N. Louis, J.P. Mesirov, E.S. Lander, and T.R. Golub. Gene expression-based classification and outcome prediction of central nervous system embryonal tumors. *Nature*, 415(24):436–442, 2002.

S. Ramaswamy. Personal communication.

R. Rifkin, G. Yeo, and T. Poggio. *Regularized Least Squares Classification*, volume 190 of *NATO Science Series III: Computer and Systems Sciences*. IOS Press, Amsterdam, 2003.

S. Sarkar and K.L. Boyer. Quantitative measures of change based on feature organization: eigenvalues and eigenvectors. *CVIU*, 71(1):110–136, 1998.

B. Scholkopf and A.J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

A. Shashua and L. Wolf. Kernel feature selection with side data using a spectral approach. In T. Pajdla and J. Matas, editors, *ECCV (3)*, volume 3023 of *Lecture Notes in Computer Science*, pages 39–53. Springer, 2004. ISBN 3-540-21982-X.

J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, 2000.

M.A. Shipp, K.N. Ross, P. Tamayo, A.P. Weng, J.L Kutok, R.C Aguiar, M. Gaasenbeek, M. Angelo, M. Reich, G.S. Pinkus, T.S. Ray, M.A Koval, K.W Last, A. Norton, T.A. Lister, J. Mesirov, D.S. Neuberg, E.S. Lander, J.C. Aster, and T.R. Golub. Diffuse large b-cell lymphoma outcome prediction by gene expression profiling and supervised machine learning. *Nature Medicine*, 8(1): 68–74, 2002.

N. Tishby, F. Pereira, and W. Bialek. The information bottleneck method. In *Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

L.J. van 't Veer, H. Dai, M.J. van de Vijver, Y.D. He, A.A Hart, M. Mao, H.L. Peterse, K. van der Kooy, M.J. Marton, A.T. Witteveen, G.J. Schreiber, R.M. Kerkhoven, C. Roberts, P.S. Linsley, R. Bernards, and S.H. Friend. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415(6871):530–536, 2002.

V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1998.

N. Vasconcelos. Feature selection by maximum marginal diversity: optimality and implications for visual recognition. In *CVPR (1)*, pages 762–772. IEEE Computer Society, 2003. ISBN 0-7695-1900-8.

P. Viola and M. Jones. Robust real-time object detection. Technical Report CRL-2001-1, Compaq Cambridge Research Lab, 2001.

Y. Weiss. Segmentation using eigenvectors: A unifying view. In *ICCV*, 1999.

J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for svms. *NIPS*, 2001.

E.P. Wigner. On the distribution of the roots of certain symmetric matrices. *Ann. of Math.(2)*, 67: 325–327, 1958.

L. Wolf and A. Shashua. Kernel principal angles for classification machines with applications to image sequence interpretation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

L. Wolf, A. Shashua, and S. Mukherjee. Gene selection via a spectral approach. In *post CVPR IEEE Workshop on Computer Vision Methods for Bioinformatics (CVMB)*, 2005.

Appendix A. Positivity of α

The proofs for the claims and theorems made in Section 7 are presented below.
Proposition 7 *The minimal value of* $f = (\mathbf{a}^\top \mathbf{b})(\mathbf{a}^\top \mathbf{c})(\mathbf{b}^\top \mathbf{c})$ *where* $\mathbf{a}, \mathbf{b}, \mathbf{c} \in R^q$ *are defined over the unit hypersphere is* $-1/8$*.*

Proof: The QR decomposition of 3 points on the unit hypersphere takes the form:

$$[\mathbf{a}, \mathbf{b}, \mathbf{c}] = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3] \left[\begin{array}{ccc} 1 & \cos(\beta) & \cos(\gamma_1) \\ 0 & \sin(\beta) & \sin(\gamma_1)\cos(\gamma_2) \\ 0 & 0 & sin(\gamma_1) sin(\gamma_2) \end{array} \right] \tag{7}$$

where $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3 \in R^n$ are three orthogonal vectors.

The problem, therefore, becomes the problem of minimizing

$$f = \cos(\beta)\cos(\gamma_1)\left(\cos(\beta)\cos(\gamma_1) + \sin(\beta)\sin(\gamma_1)\cos(\gamma_2)\right) \tag{8}$$

with respect to $\beta, \gamma_1, \gamma_2$. Since γ_2 appears only in the $cos(\gamma_2)$ expression, it can take only the values of 1 or -1 at the minimum energy point. By symmetry we can assume it to be -1, and the problem reduces to the problem of minimizing $1/2\cos(\beta+\gamma_1)(\cos(\beta+\gamma_1) + cos(\beta-\gamma_1))$. The minimum occurs when $cos(\beta-\gamma_1)$ is either 1 or -1. Both problems $1/2cos(u)(cos(u)-1)$ and $1/2cos(u)(cos(u)+1)$ have a minimum of $-1/8$.□
Proposition 8 *The expected value of* $f = (\mathbf{a}^\top \mathbf{b})(\mathbf{a}^\top \mathbf{c})(\mathbf{b}^\top \mathbf{c})$ *where* $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \Re^q$ *and* $\mathbf{c}$ *is uniformly sampled over the unit hypersphere is* $(1/q)(\mathbf{a}^\top \mathbf{b})^2$

Proof: This expectation is given by the following integral

$$\int (\mathbf{a}^\top \mathbf{b})(\mathbf{a}^\top \mathbf{c})(\mathbf{c}^\top \mathbf{b}) d\sigma(\mathbf{c}) = (\mathbf{a}^\top \mathbf{b})\mathbf{a}^\top (\int \mathbf{c}\mathbf{c}^\top d\sigma(\mathbf{c}))\mathbf{b}.$$

$\mathbf{c}$ is taken from a uniform probability and in particular from a symmetric probability, i.e., where the probability of $\mathbf{c}$ and of remains the same under sign flipping of any subset of its entries (e.g., $p(\sqrt(2)[.5,.5,0,0] = p(\sqrt(2)[-.5,.5,0,0]))$. Therefore, $\int \mathbf{c}\mathbf{c}^\top d\sigma(\mathbf{c})$ is a multiplication of the identity matrix. From linearity of the trace and from the equality $trace(\mathbf{c}\mathbf{c}^\top) = \mathbf{c}^\top\mathbf{c}$ the trace of this matrix is $\int \mathbf{c}^\top \mathbf{c} d\sigma(\mathbf{c}) = 1$. The matrix $\int \mathbf{c}\mathbf{c}^\top d\sigma(\mathbf{c})$, therefore, is $1/q$ times the identity matrix in $\Re^q$. The expectation $\int (\mathbf{a}^\top \mathbf{b})(\mathbf{a}^\top \mathbf{c})(\mathbf{c}^\top \mathbf{b}) d\sigma(\mathbf{c})$ then equals $(1/q)(\mathbf{a}^\top \mathbf{b})^2$. □

Proposition 9 *The expected value of $f = \sum_{i=1}^k (\mathbf{a}^\top \mathbf{b})(\mathbf{a}^\top \mathbf{c}_i)(\mathbf{b}^\top \mathbf{c}_i)$ where $\mathbf{a}, \mathbf{b} \in \Re^q$ and $\mathbf{c}_i$ are orthonormal vectors uniformly sampled over the unit hypersphere in $\Re^q$ is $(k/q)(\mathbf{a}^\top \mathbf{b})^2$.*

Proof: This expectation is given by the following integral

$$(\mathbf{a}^\top \mathbf{b})\mathbf{a}^\top (\sum_{i=1}^k \int \mathbf{c}_i \mathbf{c}_i^\top d\sigma(\mathbf{c}_i | \mathbf{c}_1..\mathbf{c}_{i-1}))\mathbf{b} \ ,$$

where the main difference from the proof of Prop. 8 is that now the probability distribution of $\mathbf{c}_i$ is dependent on all the previous $\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_{i-1}$. Nevertheless, if $\mathbf{c}_i$ are uniformly sampled subject to the orthogonality constraint, the sum of integrals $J = \sum_{i=1}^k \int \mathbf{c}_i \mathbf{c}_i^\top d\sigma(\mathbf{c}_i | \mathbf{c}_1..\mathbf{c}_{i-1})$ is a product over the identity matrix in $\Re^q$. To see this, consider products of the form $v^\top J v$. From symmetry this product must be the same for every $v \in \Re^q$. i.e, since $v^\top J v$ depends only on dot products (the distribution $d\sigma(\mathbf{c}_i | \mathbf{c}_1..\mathbf{c}_{i-1})$ is a uniform distribution subject to constraints on dot products), it is invariant to a unitary transformation; in particular since any vector can be rotated to any other vector we get that it is not dependent on v. We have $trace(J) = k$ satisfying the proposition, as $trace(\sum_{i=1}^k \int \mathbf{c}_i \mathbf{c}_i^\top d\sigma(\mathbf{c}_i | \mathbf{c}_1..\mathbf{c}_{i-1})) = \sum_{i=1}^k \int \mathbf{c}_i^\top \mathbf{c}_i d\sigma(\mathbf{c}_i | \mathbf{c}_1..\mathbf{c}_{i-1})$ = k. □

Theorem 10 (Probabilistic Perron-Frobenius) *Let $G = g_{ij}$ be a real symmetric $n \times n$ matrix whose entries for $i \geq j$ are independent identically and normally distributed random variables with mean $\mu > 0$ and variance σ^2. Then, for any $\epsilon > 0$ there exist n_o such that for all $n > n_0$ the leading eigenvector $\mathbf{v}$ of G is positive with probability of at least $1 - \epsilon$.*

Preliminaries: Let $G = \mu J + \sigma S$ where $J = \mathbf{1}\mathbf{1}^\top$ and S_{ij} are i.i.d. sampled according to $N(0, 1)$. Let $\mathbf{e} = \frac{1}{\sqrt{n}}\mathbf{1}$. and let $\mathbf{v}, \mathbf{v}_2, ..., \mathbf{v}_n$ and $\lambda \geq \lambda_2 \geq ... \geq \lambda_n$ be the spectrum of G. From the semicircle law (Wigner, 1958) and from (Furedi and Komlos, 1981) it is known that $\lambda_i = \Theta(\sqrt{n})$ for $i = 2, 3..., n$.

The following auxiliary claims would be useful for proving the main theorem.

Lemma 15 (Bounds on Leading Eigenvalue) *Under the conditions of Theorem 10 above, with probability $1 - o(1)$ the leading eigenvalue λ of G falls into the following interval:*

$$\mu n - \Theta(1) \leq \lambda \leq \mu n + \Theta(\sqrt{n}).$$

Proof: From the definition of the leading eigenvalue we have:

$$\begin{aligned} \lambda &= \max_{\|\mathbf{x}\|=1} \mathbf{x}^\top G \mathbf{x} = \mu(\sum_i x_i)^2 + \sigma \max_{\|\mathbf{x}\|=1} \mathbf{x}^\top S \mathbf{x} \\ &\leq \mu n + \Theta(\sqrt{n}) \end{aligned}$$

where from the semicircle law $\max_{\|\mathbf{x}\|=1} \mathbf{x}^\top S \mathbf{x} = \Theta(\sqrt{n})$ and from Cauchy-Schwartz inequality $(\sum_i x_i)^2 \leq n(\sum_i x_i^2) = n$. The lower bound follows from:

$$\begin{aligned}\lambda &\geq \mathbf{e}^\top G\mathbf{e} = \mu n + \sigma \mathbf{e}^\top S\mathbf{e} \\ &= \mu n + \sum_{i,j} S_{ij}/n \geq \mu n - \Theta(1)\end{aligned}$$

Lemma 16 *Under the conditions of Theorem 10 above, with probability* $1 - o(1)$ *we have the following bound:*

$$\sum_i v_i \geq \sqrt{n} - c \tag{9}$$

for some constant c *where* v_i *are the entries of the leading eigenvector* **v** *of* G*.*

Proof: Let $\mathbf{e} = a\mathbf{v} + \sum_{i=2}^n a_i\mathbf{v}_i$. Since the eigenvectors and **e** are of unit norm we have $a^2 + \sum_{i=2}^n a_i^2 = 1$ and without lost of generality we can assume $a > 0$. We have therefore $\mathbf{e}^\top G\mathbf{e} = a^2\lambda + \sum_i \lambda_i a_i^2$. Since $\lambda_i = \Theta(\sqrt{n})$ for $i = 2, ..., n$ and $a^2 + \sum_i a_i^2 = 1$ we have:

$$\mathbf{e}^\top G\mathbf{e} \leq a^2\lambda + \Theta(\sqrt{n}).$$

Using the bound derived above of $\mathbf{e}^\top G\mathbf{e} \geq \mu n - o(1)$ and Lemma 15, we have:

$$\begin{aligned}&\mu n - o(1) \leq \lambda a_1^2 + \Theta(\sqrt{n}) \\ &\frac{\mu n - \Theta(\sqrt{n})}{\mu n + \Theta(\sqrt{n})} \leq a^2 \leq a\end{aligned}$$

from which we can conclude (with further manipulation):

$$1 - \frac{2\Theta(\sqrt{n})}{\mu n} = 1 - \frac{1}{\mu\Theta(\sqrt{n})} \leq a.$$

Consider now that a is the angle between **e** and **v**:

$$\frac{1}{\sqrt{n}}\sum_i v_i = \mathbf{e}^\top\mathbf{v} = a \geq 1 - \frac{1}{\mu\Theta(\sqrt{n})},$$

from which we obtain:

$$\sum_i v_i \geq \sqrt{n} - c,$$

for some constant c. □

As a result so far, we have that

$$\begin{aligned}\lambda v_i &= (G\mathbf{v})_i = \mu\sum_i v_i + \sigma(S\mathbf{v})_i \\ &\geq \mu\sqrt{n} - C + \sigma\mathbf{g}^\top\mathbf{v}\end{aligned}$$

where $C = \mu c$ is a constant **g** is some n-dimensional normally distributed i.i.d random vector. We would be done if we could show that the probability of the event $\mathbf{g}^\top\mathbf{v} > (1/\sigma)\mu\sqrt{n}$ occurs with probability $o(1)$, i.e., decays with the growth of n. The problem is that since **g** stands for a row of S and because **v** depends on S we cannot make the assumption that **g** and **v** are independent —

thus a straightforward tail bound would not be appropriate. The remainder of the proof below was contributed by Ofer Zeitouni where care is taken to decouple the dependency between **g** and **v**.

Proof of Theorem 10: Let $D(c)$ be the set of vectors in R^n satisfying Lemma 16:

$$D(c) = \left\{ \mathbf{v} \in R^n \ : \ \|\mathbf{v}\| = 1, \ \sum_i v_i \geq \sqrt{n} - c \right\},$$

and let $\mathbf{g} \in R^n$ be a vector of i.i.d. standard Normal distribution $N(0,1)$. We would like to analyze the probability of the event

$F(g) = \left\{ \exists \mathbf{v} \in D(c) \ \ s.t. \ \mathbf{g}^\top \mathbf{v} \geq \frac{\mu}{\sigma} \sqrt{n} \right\}$ $\mathbf{g} \in R^n$, in the case where $g_i \sim N(0,1)$.

In particular we would like to show that the probability $P_{g_i \sim N(0,1)}(F(g))$ belongs to $o(1)$, i.e., decays with the growth of n.

Let $\mathbf{v} = \mathbf{e} + \mathbf{f}$ where $\mathbf{e} = \frac{1}{\sqrt{n}}\mathbf{1}$ was defined above and **f** is the residual. From the constraint $\|\mathbf{v}\|^2 = 1$ we obtain a constraint on **f**:

$$\frac{2}{\sqrt{n}} \sum_i f_i + \sum_i f_i^2 = 0 \tag{10}$$

Given that $\mathbf{v} \in D(c)$ we obtain:

$$\sum_i v_i = \sqrt{n}\mathbf{v}^\top \mathbf{e} = \sqrt{n} + \sum_i f_i \geq \sqrt{n} - c,$$

from which obtain another constraint on **f**:

$$-\sum_i f_i \leq c \tag{11}$$

Combining both constraints (10) and (11) we arrive at:

$$\|\mathbf{f}\|^2 \leq \frac{2c}{\sqrt{n}} \tag{12}$$

The expression $\mathbf{g}^\top \mathbf{v}$ can be broken down to a sum of two terms $\mathbf{g}^\top \mathbf{e}$ and $\mathbf{g}^\top \mathbf{f}$. The first of these two terms is $o(1)$ by the law of large numbers, and so:

$$\begin{aligned} \mathbf{g}^\top \mathbf{v} &= \mathbf{g}^\top \mathbf{e} + \mathbf{g}^\top \mathbf{f} \leq o(1) + \|\mathbf{g}\|\|\mathbf{f}\| \\ &\leq o(1) + \|\mathbf{g}\| \left(\frac{\sqrt{2c}}{n^{1/4}} \right) \end{aligned}$$

$\|\mathbf{g}\|$ distributes according to the χ distribution with n degrees of freedom, which concentrates around $\sqrt{n}$. Therefore, with probability $1 - o(1)$, $\|\mathbf{g}\| = \Theta(\sqrt{n})$. The probability that $\mathbf{g}^\top \mathbf{v} \geq \Theta(\sqrt{n})$ is proportional to the probability that $\|\mathbf{g}\| \geq n^{3/4}$, which by the Gaussian tail bound decays exponentially with the growth of n. Since the probability that each entry of v is negative decays exponentially, i.e., $p(v_i < 0) < e^{-Cn}$, for some constant C, then by the union-bound the union of such events $p(v_1 < 0 \cup .... \cup v_n < 0)$ is bounded from above by ne^{-Cn} which decays exponentially with the growth of n. □

Lemma 12 *Let* **g** *be a random* $n-$*vector of i.i.d bounded variables, i.e., for each* $i = 1..n$, $|\mathbf{g}_i| \leq M$. *The following holds for some constant* C*:*

$$P(\|\mathbf{g}\|^2 \geq Dn^{1/2+\epsilon}) \leq \exp\left(-\frac{C^2D^2n^{2\epsilon}}{M^2}\right)$$

Proof: We will apply Hoeffding's inequality to the random variable $\frac{1}{n}\|\mathbf{g}\|^2$, which has a mean μ that does not depend on n.

Assume $\gamma \geq Dn^{-1/2+\epsilon}$, where $\epsilon > 1/2$. For some $n > \hat{n}$, and for some c, $\gamma - \mu \geq c\gamma$. We get:

$$P(\frac{\|\mathbf{g}\|^2}{n} \geq \gamma) = P(\frac{\|\mathbf{g}\|^2}{n} - \mu \geq \gamma - \mu) \leq P(\frac{\|\mathbf{g}\|^2}{n} - \mu \geq c\gamma) \ \ .$$

Now, we can apply Hoeffding's one sided inequality and get:

$$P(\frac{1}{n}\|\mathbf{g}\|^2 - \mu \geq c\gamma) \leq \exp\left(-\frac{c^2n\gamma^2}{M^2}\right) \leq \exp\left(-\frac{c^2D^2n^{2\epsilon}}{M^2}\right) \ \ .$$

□