Expectation-Maximization for Sparse and Non-Negative PCA

Christian D. Sigg CHRSIGG@INF.ETHZ.CH
Joachim M. Buhmann JBUHMANN@INF.ETHZ.CH
Institute of Computational Science, ETH Zurich, 8092 Zurich, Switzerland

Abstract

We study the problem of finding the dominant eigenvector of the sample covariance matrix, under additional constraints on the vector: a cardinality constraint limits the number of non-zero elements, and non-negativity forces the elements to have equal sign. This problem is known as sparse and non-negative principal component analysis (PCA), and has many applications including dimensionality reduction and feature selection. Based on expectation-maximization for probabilistic PCA, we present an algorithm for any combination of these constraints. Its complexity is at most quadratic in the number of dimensions of the data. We demonstrate significant improvements in performance and computational efficiency compared to other constrained PCA algorithms, on large data sets from biology and computer vision. Finally, we show the usefulness of non-negative sparse PCA for unsupervised feature selection in a gene clustering task.

1. Introduction

Principal component analysis (PCA) provides a lower dimensional approximation of high dimensional data, where the reconstruction error (measured by Euclidean distance) is minimal. The first principal component (PC) is the solution to

$$\arg\max_{\mathbf{w}} \mathbf{w}^\top \mathbf{C}\mathbf{w}, \text{ subject to } \|\mathbf{w}\|_2 = 1, \qquad (1)$$

where $\mathbf{C} \in \mathbb{R}^{D\times D}$ is the positive semi-definite covariance matrix of the data. It is straightforward to show that the first PC is the dominant eigenvector of $\mathbf{C}$, i.e. the eigenvector corresponding to the largest eigenvalue. The first PC maximizes the variance of the projected data, while the second PC again maximizes the variance, under the constraint that it is orthogonal to the first, and so on.

Constrained PCA and its Applications. We consider problem (1) under two additional constraints on $\mathbf{w}$: Sparsity $\|\mathbf{w}\|_0 \leq K$[1] and non-negativity $\mathbf{w} \succeq \mathbf{0}$. Constraining PCA permits a trade-off between maximizing *statistical fidelity* on the one hand, and facilitating *interpretability* and *applicability* on the other (d'Aspremont et al., 2007). Although it is often the case that PCA provides a good approximation with few PCs, each component is usually a linear combination of all original features. Enforcing *sparsity* facilitates identification of the relevant influence factors and is therefore an unsupervised feature selection method. In applications where a fixed penalty is associated with each included dimension (e.g. transaction costs in finance), a small loss in variance for a large reduction in cardinality can lead to an overall better solution. Enforcing *non-negativity* renders PCA applicable to domains where only positive influence of features is deemed appropriate (e.g. due to the underlying physical process). Moreover, the total variance is explained *additively* by each component, instead of the mixed sign structure of unconstrained PCA. Often non-negative solutions already show some degree of sparsity, but a combination of both constraints enables precise control of the cardinality. Sparse PCA has been successfully applied to gene ranking (d'Aspremont et al., 2007), and non-negative sparse PCA has been compared favorably to non-negative matrix factorization for image parts extraction (Zass & Shashua, 2006).

Related Work. Problem (1) is a *concave programming* problem, and is NP-hard if either sparsity or non-negativity is enforced (Horst et al., 2000). Although an efficient global optimizer is therefore unlikely, local optimizers often find good or even optimal solutions in practice, and global optimality can be tested

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008.

[1] See final paragraph of this section for a definition of our notation.

in $O(D^3)$ (d'Aspremont et al., 2007), where D is the dimensionality of the data. As is evident from writing the objective function of (1) as

$$\mathbf{w}^\top \mathbf{C}\mathbf{w} = \sum_{i=1}^{D}\sum_{j=1}^{D} C_{ij} w_i w_j, \tag{2}$$

setting w_k to zero excludes the k-th column and row of $\mathbf{C}$ from the summation. For a *given* sparsity pattern $\mathcal{S} = \{i|w_i \neq 0\}$, the optimal solution is the dominant eigenvector of the corresponding submatrix of $\mathbf{C}$. For sparse PCA, the computationally hard part is therefore to identify the optimal sparsity pattern, and any solution can potentially be improved by keeping $\mathcal{S}$ only and recomputing the weights, a process called *variational renormalization* by Moghaddam et al. (2006).

Sparse PCA methods can be characterized by the following two paradigms:

1. Relaxation of the hard cardinality constraint $\|\mathbf{w}\|_0 \leq K$ into a convex constraint $\|\mathbf{w}\|_1 \leq B$, thus approximating the combinatorial problem by continuous optimization of (1) on a convex feasible region.

2. Direct combinatorial optimization of $\mathcal{S}$. Due to the potentially exponential runtime of exact methods, heuristics such as greedy search have to be employed for large values of D.

Cadima and Jolliffe (1995) proposed thresholding the $(D-K)$ smallest elements of the dominant eigenvector to zero, which has complexity $O(D^2)$. Better results have been achieved by the SPCA algorithm of Zou et al. (2004), which is based on iterative elastic net regression. Combinatorial optimization was introduced by Moghaddam et al. (2006), who derived an exact branch-and-bound method and a greedy algorithm, that computes the full sparsity path $1 \leq K \leq D$ in $O(D^4)$. Based on a semi-definite relaxation of the sparse PCA problem, d'Aspremont et al. (2007) proposed PathSPCA, which reduces the complexity of each greedy step to $O(D^2)$, and renders computation of the full regularization path possible in $O(D^3)$. Finally, Sriperumbudur et al. (2007) formulate sparse PCA as a d.c. program (Horst et al., 2000) and provide an iterative algorithm called DC-PCA, where each iteration consists of solving a quadratically constrained QP with complexity $O(D^3)$.

Non-negative (sparse) PCA was proposed by Zass and Shashua (2006). In contrast to the methods discussed so far, their algorithm (called NSPCA) optimizes the cumulative variance of L components jointly, versus a sequential approach that computes one component after another. Orthonormality of the components is enforced by a penalty in the objective function (see section 4 for a discussion about orthogonality for non-negative components), and the desired sparsity is again expressed in terms of the whole set of L components.

Our Contribution. To our knowledge, there is no algorithm either for sparse or non-negative sparse PCA that achieves competitive results in less than $O(D^3)$. In this paper, we propose an $O(D^2)$ algorithm that enforces sparsity, or non-negativity or both constraints simultaneously in the same framework, which is rooted in expectation-maximization for a probabilistic generative model of PCA (see next section). As for the combinatorial algorithms, the desired cardinality can be expressed directly as $K = |\mathcal{S}|$, instead of a bound B on the l_1 norm of $\mathbf{w}$ (which requires searching for the appropriate value). Although computing the full regularization path is also of order $O(D^3)$, our method *directly* computes a solution for any K in $O(D^2)$, in contrast to forward greedy search which needs to build up a solution incrementally. As is the case with SPCA, our method works on the data matrix $\mathbf{X} \in \mathbb{R}^{N\times D}$ (N is the number of samples), instead of the covariance matrix $\mathbf{C}$. To summarize, the low complexity combined with an efficient treatment of the $D \gg N$ case enables an application of our method to large data sets of high dimensionality.

Notation. Vectors are indexed as $\mathbf{w}_{(t)}$, and elements of vectors as w_i. $\|\mathbf{w}\|_1 = \sum_i |w_i|$ and $\|\mathbf{w}\|_0 = |\mathcal{S}|$, where $\mathcal{S} = \{i|w_i \neq 0\}$. $\|\mathbf{w}\|_0$ is also called the *cardinality* of $\mathbf{w}$. $\mathbf{I}$ is the identity matrix, $\mathbf{0}$ a vector of zero elements, and $\mathbf{w} \succeq \mathbf{0} \Leftrightarrow \forall i : w_i \geq 0$. $\mathbf{x} \circ \mathbf{y}$ denotes element-wise multiplication of $\mathbf{x}$ and $\mathbf{y}$, and $\mathrm{tr}(\mathbf{X}) = \sum_i X_{ii}$ is the trace of matrix $\mathbf{X}$. $\mathbb{E}[.]$ is the expectation operator, and $\mathcal{N}$ denotes a Gaussian distribution.

2. EM for Probabilistic PCA

Tipping and Bishop (1999) and independently Roweis (1998) proposed a generative model for PCA, where the full covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{D\times D}$ of the Gaussian distribution is approximated by its first L eigenvectors (in terms of magnitude of the respective eigenvalues). The latent variable $\mathbf{y} \in \mathbb{R}^L$ (in the principal component subspace) is distributed according to a zero mean, unit covariance Gaussian

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{3}$$

The observation $\mathbf{x} \in \mathbb{R}^D$, conditioned on the value of the latent variable $\mathbf{y}$, is linear-Gaussian distributed

according to

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{W}\mathbf{y} + \boldsymbol{\mu}, \sigma^2\mathbf{I}), \tag{4}$$

where the matrix $\mathbf{W} \in \mathbb{R}^{D\times L}$ spans the principal subspace, and $\boldsymbol{\mu} \in \mathbb{R}^D$ is the mean of the data. To simplify the presentation, we will assume centered data from now on.

The EM equations for probabilistic PCA have the following form. The E-step keeps track of

$$\mathbb{E}[\mathbf{y}_{(n)}] = \mathbf{M}_{(t)}^{-1}\mathbf{W}_{(t)}^\top\mathbf{x}_{(n)} \tag{5}$$

$$\mathbb{E}[\mathbf{y}_{(n)}\mathbf{y}_{(n)}^\top] = \sigma_t^2\mathbf{M}_{(t)}^{-1} + \mathbb{E}[\mathbf{y}_{(n)}]\mathbb{E}[\mathbf{y}_{(n)}]^\top, \tag{6}$$

where $\mathbf{M} \in \mathbb{R}^{L\times L}$ is defined as

$$\mathbf{M} = \mathbf{W}^\top\mathbf{W} + \sigma^2\mathbf{I}. \tag{7}$$

The M-step equations are

$$\mathbf{W}_{(t+1)} = \left[\sum_{n=1}^N \mathbf{x}_{(n)}\mathbb{E}[\mathbf{y}_{(n)}]^\top\right]\left[\sum_{n=1}^N \mathbb{E}[\mathbf{y}_{(n)}\mathbf{y}_{(n)}^\top]\right]^{-1} \tag{8}$$

$$\sigma_{t+1}^2 = \frac{1}{ND}\sum_{n=1}^N \Big[\|\mathbf{x}_{(n)}\|_2^2 - 2\mathbb{E}[\mathbf{y}_{(n)}]^\top\mathbf{W}_{(t+1)}^\top\mathbf{x}_{(n)} + \operatorname{tr}\Big(\mathbb{E}[\mathbf{y}_{(n)}\mathbf{y}_{(n)}^\top]\mathbf{W}_{(t+1)}^\top\mathbf{W}_{(t+1)}\Big)\Big]. \tag{9}$$

In order to efficiently incorporate constraints into the EM algorithm (see next section), we make three simplifications: take the limit $\sigma^2 \to 0$, consider a one-dimensional subspace and normalize $\|\mathbf{w}_{(t)}\|_2$ to unity. The first simplification reduces probabilistic PCA to standard PCA. Computing several components will be treated in section 4, and the unity constraint on $\|\mathbf{w}_{(t)}\|_2$ is easily restored after each EM iteration. The E-step now amounts to

$$\mathbb{E}[y_n] = \mathbf{w}_{(t)}^\top\mathbf{x}_{(n)}, \tag{10}$$

and the M-step is

$$\mathbf{w}_{(t+1)} = \frac{\sum_{n=1}^N \mathbf{x}_{(n)}\mathbb{E}[y_n]}{\sum_{n=1}^N \mathbb{E}[y_n]^2}. \tag{11}$$

These two equations have the following interpretation (Roweis, 1998): The E-step orthogonally projects the data onto the current estimate of the subspace, while the M-step re-estimates the projection to minimize squared reconstruction error for fixed subspace coordinates. We summarize this result in algorithm 1, which iteratively computes the solution to eq. (1). Due to the fact that so far only $\|\mathbf{w}\|_2 = 1$ is enforced, convergence to the global optimum doesn't depend on the initial estimate $\mathbf{w}_{(1)}$. This will no longer be the case for additional constraints.

Algorithm 1 Iterative Computation of First PC

Input: Data $\mathbf{X} \in \mathbb{R}^{N\times D}$, initial estimate $\mathbf{w}_{(1)}$, ε
Algorithm:
$t \leftarrow 1$
repeat
  $\mathbf{y} = \mathbf{X}\mathbf{w}_{(t)}$
  $\mathbf{w}_{(t+1)} = \arg\min_{\mathbf{w}} \sum_{n=1}^N \|\mathbf{x}_{(n)} - y_n\mathbf{w}\|_2^2$
  $\mathbf{w}_{(t+1)} \leftarrow \mathbf{w}_{(t+1)}/\|\mathbf{w}_{(t+1)}\|_2$
  $t \leftarrow t + 1$
until $|\mathbf{w}_{(t+1)}^\top\mathbf{w}_{(t)}| > 1 - \varepsilon$
Output: $\mathbf{w}$

3. Constrained PCA

Consider the minimization step in algorithm 1, which can be written as

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} J(\mathbf{w}) := h\mathbf{w}^\top\mathbf{w} - 2\mathbf{f}^\top\mathbf{w}, \tag{12}$$

with $h = \sum_{n=1}^N y_n^2$ and $\mathbf{f} = \sum_{n=1}^N y_n\mathbf{x}_{(n)}$. Eq. (12) is a quadratic program (QP), and is convex due to the non-negativity of h. Furthermore, because the Hessian is a scaled identity matrix, the problem is also isotropic. The unique global optimum is found by analytical differentiation of the objective function

$$\nabla J \stackrel{!}{=} 0 \Rightarrow \mathbf{w}^* = \frac{\mathbf{f}}{h}, \tag{13}$$

which of course is identical to eq. (11).

3.1. Sparsity

It is well known (Tibshirani, 1996) that solving a QP under an additional constraint on $\|\mathbf{w}\|_1$ favors a sparse solution. This constraint corresponds to restricting the feasible region to an l_1 diamond:

$$\mathbf{w}^\circ = \arg\min_{\mathbf{w}} \left(h\mathbf{w}^\top\mathbf{w} - 2\mathbf{f}^\top\mathbf{w}\right) \quad \text{s.t.} \quad \|\mathbf{w}\|_1 \le B, \tag{14}$$

where the upper bound B is chosen such that $\mathbf{w}^\circ$ has the desired cardinality. The l_1 constrained QP is again convex, and because the objective function is isotropic, it implies that $\mathbf{w}^\circ$ is the feasible point minimizing l_2 distance to the unconstrained optimum $\mathbf{w}^*$.

We derive an efficient and optimal algorithm for eq. (14), where the desired cardinality can be specified directly by the number K of non-zero dimensions. Observe that $\mathbf{w}^\circ$ must have the same sign structure as $\mathbf{f}$, therefore we can transform the problem such that both $\mathbf{w}^*$ and $\mathbf{w}^\circ$ come to lie in the non-negative orthant. The algorithm (illustrated in fig. 1) approaches $\mathbf{w}^\circ$

with axis-aligned steps in the direction of the largest element of the negative gradient

$$-\nabla J(\mathbf{w}) \propto \mathbf{w}^* - \mathbf{w}, \tag{15}$$

until the boundary of the feasible region is hit or the gradient vanishes. Because the elements of $\mathbf{w}$ become positive one after another, and their magnitude increases monotonically, B is set implicitly by terminating the gradient descent once the cardinality of the solution vector is K. Finally, the solution is transformed back into the original orthant of $\mathbf{w}^*$.

Proposition 3.1 *Axis-aligned gradient descent with infinitesimal stepsize terminates at the optimal feasible point* $\mathbf{w}^\circ$.

Proof. Optimality is trivial if $\mathbf{w}^*$ lies within the feasible region, so we consider the case where the l_1 constraint is active. The objective function in eq. (14) is equivalent to

$$\|\mathbf{w}^* - \mathbf{w}\|_2^2 = \sum_{d=1}^{D} \left(w_d^* - w_d\right)^2. \tag{16}$$

The gradient descent procedure invests all available coefficient weight B into decreasing the largest term(s) of this sum, which follows from eq. (15). We show equivalence of $\mathbf{w}^\circ$ to the gradient descent solution $\mathbf{v}$ by contradiction. Suppose the computation of $\mathbf{w}^\circ$ follows a different strategy, so at least one summation term $(w_l^* - w_l^\circ)^2$ is larger than $\max_d (w_d^* - v_d)^2$. However, subtracting a small amount from w_s° $(s \neq l)$ and adding it to w_l° doesn't change $\|\mathbf{w}^\circ\|_1$ but decreases the objective, which is a contradiction. □

Implementation of axis-aligned gradient descent amounts to sorting the elements of $-\nabla J(\mathbf{w})$ in descending order (an $O(D \log D)$ operation), and iterating over its first K elements. At each iteration $k \in \{1, \ldots, K\}$, the first k elements of $\mathbf{w}$ are manipulated, resulting in complexity $O(K^2)$ for the whole loop. Algorithm 2 provides a full specification of the method. Because EM is a local optimizer, the initial direction $\mathbf{w}_{(1)}$ must be chosen carefully to achieve good results. For sparse PCA, initialization with the unconstrained first principal component gave best results (see section 5). Initialization is therefore the most expensive operation of the algorithm with its $O(D^2)$ complexity. For the $D \gg N$ case, it can be reduced to $O(N^2)$ by working with $\mathbf{X}\mathbf{X}^\top$ instead of $\mathbf{X}^\top\mathbf{X}$. As initialization is independent of K, $\mathbf{w}_{(1)}$ can be cached and re-used when varying the sparsity parameter. The number of EM iterations t until convergence also depends on D and K, but our experiments (see section 5) suggest that dependence is weak and sub-linear. On average, $t < 10$ iterations were sufficient to achieve convergence.

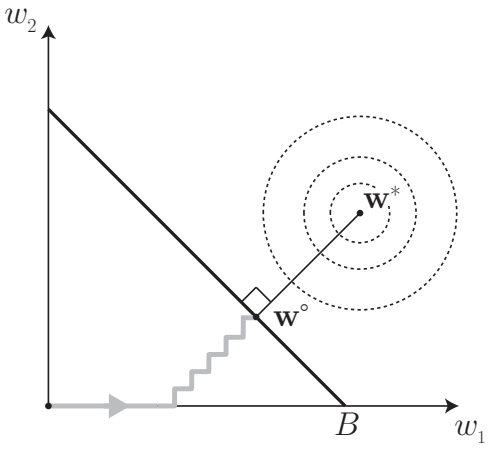

Figure 1. Starting at the origin, $\mathbf{w}^\circ$ is approached by axis-aligned steps in the direction of the largest element of the negative gradient. As dimensions enter the solution vector one after another, and the corresponding weights w_i increase monotonically, the bound B is set implicitly by terminating once $\|\mathbf{w}\|_0 = K$.

3.2. Non-Negativity

Enforcing non-negativity is achieved in the same way as sparsity. Here, the the feasible region is constrained to the non-negative orthant, which is again a convex domain:

$$\begin{aligned} \mathbf{w}^\circ \quad &= \quad \arg\min_{\mathbf{w}} \left(h\mathbf{w}^\top\mathbf{w} - 2\mathbf{f}^\top\mathbf{w}\right) \\ &\text{s.t.} \quad \mathbf{w} \succeq \mathbf{0}. \end{aligned} \tag{17}$$

Eq. (17) implies that choosing $w_i = 0$ for $f_i < 0$ is optimal. The non-negativity constraint can then be dropped, and optimization for the other elements of $\mathbf{w}$ proceeds as before.

The first PC is invariant to a change of sign. However, this symmetry is broken if the non-negativity constraint is enforced. As an extreme example, non-negative EM fails if the initial projection $\mathbf{w}_{(1)}$ is a dominant eigenvector that only consists of non-positive elements - the minimum of eq. (17) is the zero vector. But changing the sign of $\mathbf{w}_{(1)}$ implies that the non-negativity constraint becomes inactive, and the algorithm terminates immediately with the optimal solution. We choose to initialize EM for non-negative PCA with a random unit vector in the non-negative orthant, which exploits the benefit of random restarts.

For non-negative sparse PCA, the feasible region is defined as the intersection of the non-negative orthant

Algorithm 2 EM for Sparse PCA

Input: $\mathbf{X} \in \mathbb{R}^{N \times D}$, $K \in \{1, \ldots, D-1\}$, ε
Algorithm:
$t \leftarrow 1$
$\mathbf{w}_{(t)} \leftarrow$ first principal component of $\mathbf{X}$
repeat
  $\mathbf{y} \leftarrow \mathbf{X}\mathbf{w}_{(t)}$
  $\mathbf{w}^* \leftarrow \sum_{n=1}^{N} y_n \mathbf{x}_{(n)} / \sum_{n=1}^{N} y_n^2$
  $\mathbf{s} \leftarrow$ elements $|w_i^*|$ sorted in descending order
  $\boldsymbol{\pi} \leftarrow$ indices of sorting order
  $\mathbf{w}_{(t+1)} \leftarrow \mathbf{0}$
  for $k = 1$ **to** K **do**
    Add $(s_k - s_{K+1})$ to element k of $\mathbf{w}_{(t+1)}$
  end for
  Permute elements of $\mathbf{w}_{(t+1)}$ according to $\boldsymbol{\pi}^{-1}$
  $\mathbf{w}_{(t+1)} \leftarrow \mathbf{w}_{(t+1)} \circ \operatorname{sign}(\mathbf{w}^*) / \|\mathbf{w}_{(t+1)}\|_2$
  $t \leftarrow t + 1$
until $|\mathbf{w}_{(t+1)}^\top \mathbf{w}_{(t)}| > 1 - \varepsilon$
Output: $\mathbf{w}$

and the l_1 diamond. As the intersection of two convex sets is again convex, the combined constraints can be treated in the same framework. We establish convergence of our method in the following proposition:

Proposition 3.2 *EM for sparse and non-negative PCA converges to a local minimum of the l_2 reconstruction error.*

Proof. Given a feasible $\mathbf{w}_{(t)}$ (either by proper initialization or after one EM iteration), both the E-step and the M-step never increase l_2 reconstruction error. Orthogonal projection $\mathbf{y} = \mathbf{X}\mathbf{w}$ in the E-step is the l_2 optimal choice of subspace coordinates for given $\mathbf{w}$. Error minimization w.r.t. $\mathbf{w}$ in the M-step either recovers $\mathbf{w}_{(t)}$ as it is feasible, or provides an improved $\mathbf{w}_{(t+1)}$. □

4. Several Components

A full eigen decomposition of the covariance matrix $\mathbf{C}$ provides all r PCs, where r is the rank of $\mathbf{C}$. Sorted in descending order of eigenvalue magnitude, each eigenvector maximizes the variance of the projected data, under the constraint that it is orthogonal to all other components considered so far. For sparse PCA, we compute more than one component by means of iterative deflation: having identified the first component $\mathbf{w}_{(1)}$, project the data to its orthogonal subspace using

$$\mathbf{P} = \mathbf{I} - \mathbf{w}_{(1)}\mathbf{w}_{(1)}^\top, \tag{18}$$

re-run EM to identify $\mathbf{w}_{(2)}$, and so on. Although deflation suffers from numerical errors that accumulate over each iteration, this inaccuracy is not a serious problem as long as the desired number of components L is small compared to r (which is true in many applications of PCA).

Desiring non-negativity and orthogonality implies that each feature can be part of at most one component:

$$w_i^{(l)} > 0 \Rightarrow w_i^{(m)} = 0 \tag{19}$$

for $m \neq l$, i.e. the sparsity patterns have to be disjoint: $\mathcal{S}_l \bigcap \mathcal{S}_m = \emptyset$, for $l \neq m$ and $\mathcal{S}_l = \{i | w_i^{(l)} > 0\}$. This constraint might be too strong for some applications, where it can be relaxed to require a minimum angle between components. This *quasi*-orthogonality is enforced by adding a quadratic penalty term

$$\alpha \mathbf{w}^\top \mathbf{V}\mathbf{V}^\top \mathbf{w}, \tag{20}$$

to eq. (17), where $\mathbf{V} = \left[\mathbf{w}_{(1)}\mathbf{w}_{(2)} \cdots \mathbf{w}_{(l-1)}\right]$ contains previously identified components as columns, and α is a tuning parameter. Because $\mathbf{V}\mathbf{V}^\top$ is also positive semi-definite, the QP remains convex, but the Hessian is no longer isotropic. We have used the standard Matlab QP solver, but there exist special algorithms for this case in the literature (Sha et al., 2007).

5. Experimental Results

We report performance and efficiency of our method in comparison to three algorithms: SPCA[2] and PathSPCA[3] for cardinality constrained PCA, and NSPCA[4] for non-negative sparse PCA. SPCA was chosen because it has conceptual similarities to our algorithm: both are iterative methods that solve an l_1 constrained convex program, and both use the data matrix instead of the covariance matrix. PathSPCA was chosen because it is (to our knowledge) the most efficient combinatorial algorithm. We are not aware of any other non-negative PCA algorithm besides NSPCA.

The data sets considered in the evaluation are the following:

1. CBCL face images (Sung, 1996): 2429 gray scale images of size 19×19 pixels, which have been used in the evaluation of (Zass & Shashua, 2006).
2. Leukemia data (Armstrong et al., 2002): Expression profiles of 12582 genes from 72 patients. Sim-

[2]We use the Matlab implementation of SPCA by Karl Sjöstrand, available at `http://www2.imm.dtu.dk/~kas/software/spca/index.html`.

[3]Available from the authors at `http://www.princeton.edu/~aspremon/PathSPCA.htm`.

[4]Available from the authors at `http://www.cs.huji.ac.il/~zass/`.

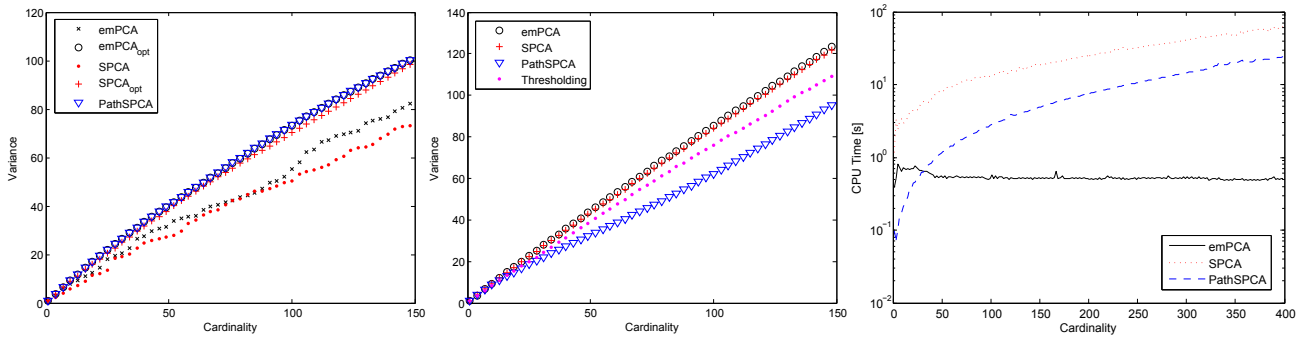

Figure 2. Left: Variance versus cardinality trade-off curves for the face image data. "opt" subscripts denote variance after recomputing optimal weights for a given sparsity pattern (which is not necessary for PathSPCA). *Middle:* Variance versus cardinality trade-off curves for the gene expression data. Performance of simple thresholding was included for reference. *Right:* Running times of Matlab implementations on the gene expression data, which include renormalization for SPCA and emPCA.

ilar data sets have been used in the evaluation of (Zou et al., 2004) and (d'Aspremont et al., 2007).

The two data sets cover the $N > D$ and $D \gg N$ case and are large enough, such that differences in computational complexity can be established with confidence. For the experiments of section 5.1 and 5.2, the features were standardized to unit variance. For unsupervised gene selection (section 5.3), not standardizing the variance led to significantly better results.

5.1. Sparse PCA

Figure 2 (left) plots explained variance versus cardinality for SPCA, PathSPCA and our algorithm (called emPCA) on the face image data set. Variational renormalization is necessary for SPCA and emPCA to close the performance gap to PathSPCA, which computes optimal weights for a specific sparsity pattern by construction. Figure 2 (middle) shows analogous results for the gene expression data. As a reference, we have also plotted results for simple thresholding (after renormalization).

We have also measured running times of Matlab implementations of the algorithms. CPU time was measured using Matlab's `tic` and `toc` timer constructs, running on an Intel Core 2 Duo processor at 2.2GHz with 3GB of RAM. Our focus is not to report absolute numbers, but rather demonstrate the dependency on the choice of K. Figure 2 (right) plots the running times versus cardinality on the gene expression data. The PathSPCA curve is well explained by the incremental forward greedy search. SPCA is harder to analyze, due to its active set optimization scheme: at each iteration of the algorithm, active features are reexamined and possibly excluded, but might be added again later on. emPCA is only marginally affected by the choice of K, but shows an increased number of EM iterations for $10 \leq K \leq 25$, which was observed on other data sets as well.

5.2. Non-Negative PCA

The impact of the non-negativity constraint on the explained variance depends on the sign structure of $\mathbf{w}^*$. Because the first principal component for the face image data happens to lie in the non-negative orthant, we projected the data onto its orthogonal subspace such that the constraint becomes active. Figure 3 (left) shows the variance versus cardinality trade-off curves for non-negative sparse PCA. For NSPCA, the sparsity penalty β was determined for each K using bisection search, which was aborted when the relative length of the parameter search interval was below a threshold of 10^{-5}. Both the variance achieved and the number of cardinalities for which a solution was found strongly depend on the value of α, which corresponds to a unit norm penalty (for the case of a single component). For smaller values of α the performance of NSPCA is comparable to emPCA, but only solutions close to the full cardinality are found. Increasing the magnitude of α makes it possible to sweep the whole cardinality path, but the performance degrades.

Because both algorithms are initialized randomly, we chose the best result after ten restarts. Running times for both methods showed no strong dependency on K. Average times for $K \in \{1, \ldots, 100\}$ were $0.4s$ for emPCA ($0.15s$ standard deviation) and $24s$ for NSPCA ($14.7s$ standard deviation).

We already motivated in section 4 that requiring orthogonality between several non-negative components can be restrictive. If the first PC happens to lie in the non-negative orthant, the constraints have to be mod-

ified such that more than one component can satisfy them. We have explored the following two strategies:

1. Enforcing orthogonality, but constraining the cardinality of each component.
2. Relaxing the orthogonality constraint, by enforcing a minimum angle between components instead.

There is a methodological difficulty in comparing the performance of NSPCA and emPCA. The former maximizes cumulative variance of all components jointly, while our algorithm computes them sequentially, maximizing the variance under the constraint that subsequent components are orthogonal to previous ones (see section 4). We therefore expect emPCA to capture more variance in the first components, while NSPCA is expected to capture larger cumulative variance. Figure 3 (middle) shows the results of applying the first strategy to the face image data. The NSPCA sparsity penalty β was tuned to achieve a joint cardinality of 200 for all components. For emPCA we distributed the active features evenly among components by setting $K = 20$ for all of them. As in figure 3 (left), emPCA captures significantly more of the variance, suggesting that the way NSPCA incorporates sparsity seriously degrades performance. This observation was confirmed for various values of K and L.

Finally, figure 3 (right) reports results for the second strategy, where a minimum angle of 85 degrees was enforced between components. Here, the complementary objectives of NSPCA and emPCA match with our prior expectations. Again, various values for L and minimum angle lead to essentially the same behavior.

5.3. Unsupervised Gene Selection

We apply emPCA to select a subset of genes of the leukemia data, and measure subset relevance by following the evaluation methodology of Varshavsky et al. (2006). For each gene subset, we cluster the data using k-means ($k = 3$), and compare the cluster assignments to the true labeling of the data, which differentiates between three types of leukemia (ALL, AML and MLL). Agreement is measured using Jaccard scores (Varshavsky et al., 2006), where a value of one signifies perfect correspondence between cluster assignment and label. We compare emPCA to simple ranking of the CE criterion as proposed by the authors, which has shown competitive performance to other popular gene selection methods. Figure 4 shows that selecting 70 genes according to the first non-negative sparse PC results in a significantly better Jaccard score than a clustering of the full data set.

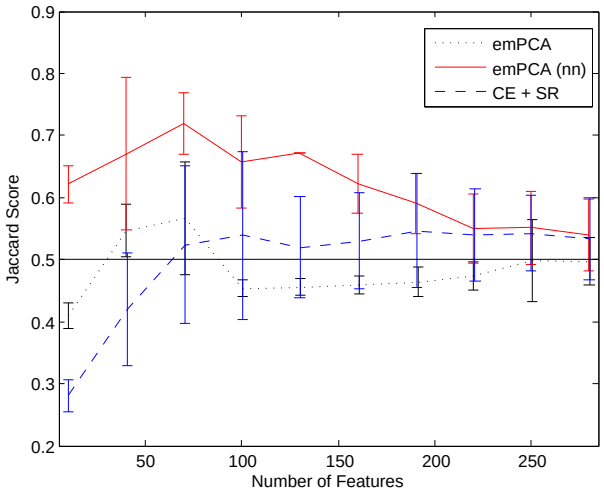

Figure 4. Mean and standard deviation for Jaccard scores after subset selection and k-means clustering ($k = 3$), averaged over 100 random initializations of the centroids. The full data score is shown as the solid line. A small amount of jitter has been added to better distinguish error bars.

6. Conclusions

We have presented a novel algorithm for constrained principal component analysis, based on expectation-maximization for probabilistic PCA. Our method is applicable to a broad range of problems: it includes sparsity, non-negativity or both kinds of constraints, it has an efficient formulation for $N > D$ and $D \gg N$ type of data, and it enforces either strict or quasi-orthogonality between successive components. Desired sparsity is directly specified in the number of non-zero elements, instead of a bound on the l_1 norm of the vector. We have demonstrated on popular data sets from biology and computer vision that our method achieves competitive results for sparse problems, and that it shows significant improvements for non-negative sparse problems. Its unmatched computational efficiency enables a constrained principal component analysis of substantially larger data sets and lower requirements on available computation time.

Although our algorithm is rooted in expectation-maximization for a generative model of PCA, constraints are added at the optimization stage. In the future, we will study how to include them in the model itself, which would enable a Bayesian analysis and data-driven determination of the proper sparsity and number of components. Secondly, we intend to examine whether our algorithm can be extended to the related problem of constrained linear discriminant analysis.

Matlab code for emPCA is available at `http://www.inf.ethz.ch/personal/chrsigg/icml2008`.

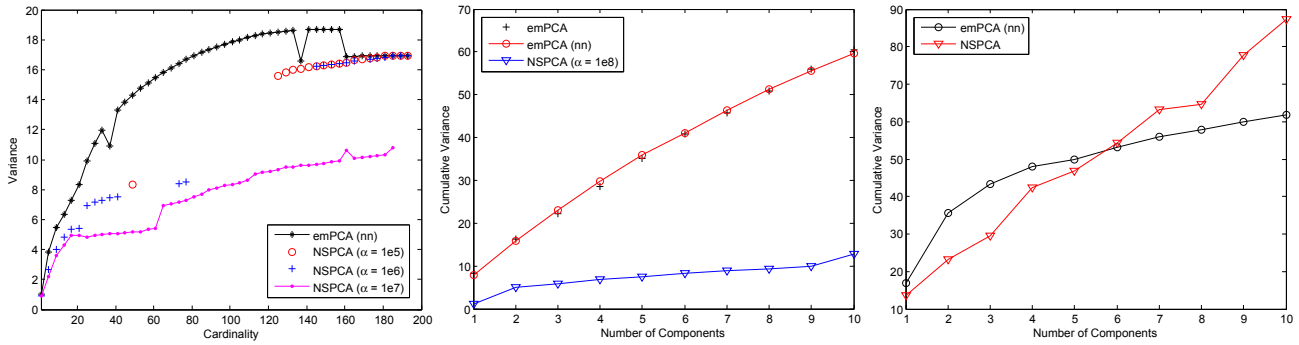

Figure 3. Left: Variance versus cardinality trade-off curves for non-negative sparse PCA methods on face image data. For NSPCA, the sparsity penalty β was determined using bisection search (see text). Values indicate better result after ten random restarts. *Middle:* Cumulative variance versus number of orthogonal components. For NSPCA, β was tuned to achieve a joint cardinality of 200 for all components. For emPCA, we set $K = 20$ for every component. emPCA (without non-negativity constraints) is plotted for reference. *Right:* Cumulative variance versus number of quasi-orthogonal components. A minimum angle of 85 degrees was enforced between components.

Acknowledgements

We thank Wolfgang Einhäuser-Treyer, Peter Orbanz and the anonymous reviewers for their valuable comments on the manuscript. This work was in part funded by CTI grant 8539.2;2 ESPP-ES.

References

Armstrong, S., Staunton, J., Silverman, L., Pieters, R., den Boer, M., Minden, M., Sallan, S., Lander, E., Golub, T., & Korsmeyer, S. (2002). MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nature Genetics*, *30*, 41–47.

Cadima, J., & Jolliffe, I. (1995). Loadings and correlations in the interpretation of principal components. *Applied Statistics*, 203–214.

d'Aspremont, A., Bach, F., & El Ghaoui, L. (2007). Full regularization path for sparse principal component analysis. *Proceedings of the International Conference on Machine Learning*.

Horst, R., Pardalos, P., & Thoai, N. (2000). *Introduction to global optimization*. Kluwer Acad. Publ.

Moghaddam, B., Weiss, Y., & Avidan, S. (2006). Spectral bounds for sparse PCA: Exact and greedy algorithms. *Advances in Neural Information Processing Systems*.

Roweis, S. (1998). EM algorithms for PCA and sensible PCA. *Advances in Neural Information Processing Systems*.

Sha, F., Lin, Y., Saul, L., & Lee, D. (2007). Multiplicative Updates for Nonnegative Quadratic Programming. *Neural Computation*, *19*, 2004–2031.

Sriperumbudur, B., Torres, D., & Lanckriet, G. (2007). Sparse eigen methods by d.c. programming. *Proceedings of the International Conference on Machine Learning*.

Sung, K.-K. (1996). *Learning and example selection for object and pattern recognition*. Doctoral dissertation, MIT, Artificial Intelligence Laboratory and Center for Biological and Computational Learning, Cambridge, MA.

Tibshirani, R. (1996). Regression shrinkage and selection via the LASSO. *Journal of the Royal statistical society, series B*, *58*, 267–288.

Tipping, M., & Bishop, C. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, *21*, 611–622.

Varshavsky, R., Gottlieb, A., Linial, M., & Horn, D. (2006). Novel Unsupervised Feature Filtering of Biological Data. *Bioinformatics*, *22*.

Zass, R., & Shashua, A. (2006). Nonnegative sparse PCA. *Advances in Neural Information Processing Systems*.

Zou, H., Hastie, T., & Tibshirani, R. (2004). Sparse principal component analysis. *Journal of Computational and Graphical Statistics*.