Convex Principal Feature Selection

Mahdokht Masaeli[1], Yan Yan[1], Ying Cui[1,2], Glenn Fung[3], Jennifer G. Dy[1]
[1]Department of Electrical and Computer Engineering, Northeastern University, Boston, MA
[2]Advertising Sciences, Yahoo! Labs, Silicon Valley, CA, USA
[3]Computer Aided Diagnosis and Therapy, Siemens Medical Solutions, USA
{masaeli.m, yan.y, cui.yi}@neu.edu, glenn.fung@siemens.com, jdy@ece.neu.edu

Abstract

A popular approach for dimensionality reduction and data analysis is principal component analysis (PCA). A limiting factor with PCA is that it does not inform us on which of the original features are important. There is a recent interest in sparse PCA (SPCA). By applying an L_1 regularizer to PCA, a sparse transformation is achieved. However, true feature selection may not be achieved as non-sparse coefficients may be distributed over several features. Feature selection is an NP-hard combinatorial optimization problem. This paper relaxes and re-formulates the feature selection problem as a convex continuous optimization problem that minimizes a mean-squared-reconstruction error (a criterion optimized by PCA) and considers feature redundancy into account (an important property in PCA and feature selection). We call this new method *Convex Principal Feature Selection (CPFS)*. Experiments show that *CPFS* performed better than SPCA in selecting features that maximize variance or minimize the mean-squared-reconstruction error.

Keywords: feature selection, dimensionality reduction, principal component analysis, convex optimization, sparsity.

1 Introduction

Principal component analysis (PCA) is a popular tool for data analysis and dimensionality reduction. Dimensionality reduction transformations can have a variety of optimality criteria, and ten of them discussed in [1] lead to the principal component solutions, which is one reason for the popularity of this analysis. Two other reasons can be attributed to: (1) PCA finds the global solution; and (2) the principal components are orthogonal (uncorrelated). However, one limitation with PCA is that it does not explicitly inform us on which features are important.

Feature selection selects a subset of features from the original set; whereas, feature transformation methods (such as, PCA) apply a linear or nonlinear function on the original features. In some applications it is desirable not only to reduce the dimension of the space, but also to reduce the number of variables that are to be considered or measured in the future and to keep the original meaning of the features. In some cases, one just want to know which features are important. In these situations, feature selection is desired.

Feature transformation can be expressed as an optimization problem over a continuous feature space solution. Feature selection, on the other hand, is an NP-hard optimization problem over a discrete space. An exhaustive search of 2^d possible feature subsets (where d is the number of features) is computationally impractical. More realistic search strategies such as greedy approaches (e.g., sequential forward/backward search [2]) can lead to local optima. Random search methods, such as genetic algorithms, add some randomness in the search procedure to help escape from local optima. Exhaustive, greedy and random searches are subset search methods because they evaluate each candidate subset with respect to some evaluation criterion. In some cases when the dimensionality is very high, one can only afford an individual search. Individual search methods evaluate each feature individually according to a criterion or a condition [3]. They then select features, which either satisfy the condition or are top-ranked.

In this paper, instead of solving feature selection as a discrete optimization, we relax and formulate the problem as a continuous optimization problem that minimizes a mean-squared-reconstruction error (a criterion optimized by PCA) and considers feature redundancy into account (an important property in PCA and feature selection). We call our new method *Convex Principal Feature Selection (CPFS)*. Our experiments show that CPFS performed better than other feature selection algorithms based on PCA in selecting features that maximize variance or minimize the mean-squared-reconstruction error.

The rest of this paper is organized as follows. In Section 2, we provide a survey of related work on feature selection based on PCA. Section 3 presents a brief review on PCA and singular value decomposition. We formulate our continuous feature selection approach as a continuous convex optimization problem in Section 4. In Section 5, we describe our algorithm and the implementation details for CPFS. We explain the limitations of sparse principal component analysis as a feature selection approach in Section 6. We report our experimental results in Section 7. And, finally we present our conclusions in Section 8.

2 Review of Related Work

There is a growing interest in sparsifying PCA (SPCA) [4] modeled after LASSO which sparsifies coefficients in regression through an L_1 regularizer [5]. SPCA finds a sparse set of features that explains the principal components (PC). However, even when several coefficients are zero, several or all features may have non-zero components in explaining different PCs (i.e., several features are still needed and efficient feature reduction may not be achieved). We provide an illustration of this limitation in Section 6. Other related methods utilize the results of PCA and then perform heuristic search to select features. Jolliffe [6] selects variables with the highest coefficient (or loading) in absolute value of the first q principal eigenvectors. It can be implemented both iteratively and non-iteratively. Another approach is by Krzanowski [7], which tries to minimize the error between the principal components (PC) calculated with all the original features and the ones in the selected feature subset, via sequential forward search and backward elimination and procrustes analysis (minimize sum-squared error under translation, rotation and reflection). Mao [8] provided a modified faster version of Krzanowski's method by minimizing the least-square error between the feature subset and the original PCs. The iterative PCA approach by Jolliffe does not take redundancy into account. The other methods do but not explicitly, contrary to our approach. Moreover, Krzanowski and Mao apply sequential search techniques which are slow. Lu et al. [9] pointed out the importance of removing redundancy in image recognition applications and performed Kmeans clustering to the loadings of the first several PCs, and selected the features closest to each clusters' centroid, called principal feature analysis (PFA). This method depends on the performance of the clustering method. Cui and Dy [10] presented a mean-squared-error objective for feature selection that incorporates redundancy into account and proposed a Gram-Schmidt orthogonalization approach for selecting principal features. The work by Lu et al. and by Cui and Dy, both consider the goal of minimizing reconstruction error and minimizing redundancy. However, their approach only guarantees local optimum. Whereas this paper, presents a convex relaxation to the principal feature selection problem and is thus guaranteed a global solution to this relaxed formulation.

3 Background Review of PCA and SVD

This section provides a brief review of Principal Component Analysis (PCA) and the Singular Value Decomposition (SVD). At the same time, we introduce basic notations, assumptions, and definitions used throughout the paper.

Let X denote an $n \times p$ data matrix, where n is the number of data samples, which can be regarded as n realizations (observations) of a p-dimensional random vector x_i, $i = 1, 2, \ldots, n$. Throughout this paper, we assume without loss of generality that the columns of X are zero-centered. The goal of **PCA** is to find a linear transformation, $B \in R^{p\times q}$, from X to a q-dimensional vector space, $Y = XB$, where $q \leq p$, that minimizes the sum-squared-reconstruction error, subject to the constraint that the column vectors of $B = [b_1, b_2, \cdots, b_q]$, $b_i \in R^{p\times 1}$, are orthonormal:

$$\begin{array}{lll} & \min\limits_{B\in R^{p\times q}} & \|X - XBB^T\|^2_{frob} \\ (3.1) & \text{s.t.} & b_i^T b_j = 0, \text{ for } i \neq j \\ & & b_i^T b_i = 1 \end{array}$$

where $frob$ stands for the matrix Frobenius norm, and T stands for the matrix transpose. The optimal linear mapping, B, in the least-squares sense is the one formed by the eigenvectors of the covariance matrix Σ of X (or the correlation matrix assuming X has zero-mean as assumed in this paper). $\Sigma = \frac{1}{n}X^T X$. Other criteria, including maximizing the variance, maximizing the data scatter, maximizing the population entropy, also lead to the PCA solution.

The PCA solution can also be derived from the singular value decomposition (**SVD**) of X,

$$X = UDV^T.$$

Here, both U and V are unitary matrices, and D is a diagonal matrix of the same dimension as X with non-negative diagonal elements d_i, called the *singular values*. PCA transforms X to Y by the following equation

$$Y = X\tilde{V} = \tilde{U}D,$$

where $\tilde{V}$, the $p \times q$ weight matrix, consists of q eigenvectors corresponding to the first q largest eigenvalues of the correlation matrix S_X, or $\tilde{V}$ corresponds to the first q column vectors, $[V_1, \ldots, V_q]$, of matrix V corresponding to the q largest singular values in the SVD decomposition. Similarly, $\tilde{U}$ corresponds to the first q

column vectors, $[U_1, \ldots, U_q]$, of U. The column vectors of Y, $Y_j = XV_j = U_jD$, are the *principal components*. The column vectors, U_j, are called the principal components of unit length. V_j are called the *loadings* of the principal component.

4 Convex Principal Feature Selection

A limitation with PCA is that it does not perform feature selection; all features still need to be computed/measured to transform the data in the reduced space. In this section, we reformulate the PCA problem into a feature selection problem. The key to this formulation is in constraining the structure of the transformation matrix.

Given a data sample,

$$X = [x_1, x_2, \ldots, x_n]^T = [f_1, f_2, \ldots, f_p],$$

with p features, denoted as f_i, $i = 1, 2, \ldots, p$, and each feature is an $n \times 1$ column vector. The representation of X in terms of its columns f_i will be important in describing our feature selection method. The goal in feature selection is to select a subset of q features, where $q < p$, that optimizes a feature evaluation criterion.

Similar to PCA, we base our Convex Principal Feature Selection (CPFS) formulation on minimizing the reconstruction error. We translate PCA into a feature selection formulation as follows. If the selected subset F_{sel} is

$$F_{sel} = \{f_{q_1}, f_{q_2}, \ldots, f_{q_q}\},$$

then to reconstruct X means that: (1) every column in the reconstructed X, denoted as $\tilde{X}$, that corresponds to $f_{q_j} \in F_{sel}$ should be equal to the one in X, and (2) the columns in $\tilde{X}$ that corresponds to features not selected should be the projections of the unselected features to the subspace spanned by F_{sel}. This accounts for feature redundancy. The reconstruction matrix $\tilde{X}$ is described as:

$$\begin{aligned} \tilde{X} &= [\tilde{f}_1, \tilde{f}_2, \ldots, \tilde{f}_p] \\ &= [f_1, f_2, \ldots, f_p]A \end{aligned}$$

Let a_{ij} be the elements of matrix A. If feature j is selected by the algorithm, a_{jj} should be equal to one and the other elements in the jth column should be equal to zero, indicating that the reconstruction of feature j is itself. Also, if feature i is not selected by the algorithm, the ith row of the A matrix should be all zeros, showing that feature i does not contribute to reconstructing any feature. Therefore, forcing the A matrix to have more zero rows can be interpreted as selecting less features. Here, using this idea, we enforce sparsity on the rows of A by adding a regularization term to the reconstruction error, $\sum_{i=1}^{p} s_i$, where s_i is the maximum value of the elements in the i-th row of matrix A. The constraints should therefore be such that for every i and j, the absolute value of a_{ij} is smaller than or equal to s_i, where all s_i are non-negative. To make the constraints linear, we can express the constraint $|a_{ij}| \leq s_i$ equivalently as $-s_i \leq a_{ij} \leq s_i$. Note that our regularization is essentially an L_1 regularization on $s = \{s_1, \ldots, s_i, \ldots, s_p\}$, where s_i is the L_∞ of the i-th row of A.

The resulting optimization objective function is:

$$(4.2) \qquad \min_{A \in R^{p \times p}} \|X - XA\|_{frob}^2 + \lambda \sum_{j=1}^{d} \|\mathbf{a}_j\|_{\ell_\infty}$$

where the first term is the reconstruction error, $E = \|X - \tilde{X}\|_{frob}^2 = \|X - XA\|_{frob}^2$, $\tilde{X} = XA$ is the reconstruction of the reduced space in $R^{n \times q}$ to the original space in $R^{n \times d}$, and A corresponds to the BB^T of the PCA formulation in Equation 3.1. λ controls the trade-off between reconstruction error and sparsity. Increasing λ means forcing more s_i to be small, which will result in removing more features. For an extreme case, with $\lambda = 0$, which means selecting all of the features, the optimum A will be the identity matrix. Conversely, for very large λ, no features are selected. Therefore, ranging λ from zero to infinity can be interpreted as ranging the number of selected features from p to zero.

Compared to PCA, instead of selecting q PCs, CPFS selects q features. By constraining the structure of A, the solution set would translate from a subset of PCs to a subset of features, which leads to a minimum reconstruction error for a feature selection problem. Feature selection aims to keep relevant (with respect to a task or criterion) features and remove redundancy. With respect to the minimum reconstruction error criterion, the goal in feature selection is to select features that minimize error (also equivalent to maximizing variance), and at the same time the subset of features should not be redundant. The overall CPFS formulation accounts for both (minimizing error and redundancy). The redundancy constraint is analogous to the orthogonality constraint in PCA.

5 Implementation Details

Formulation 4.2 is now a continuous optimization problem. In particular, it is convex and quadratic. One can solve it with any favorite optimization algorithm [11, 12]. To optimize for the ℓ_∞ norm, we utilize a dummy variable s_j to represent the maximum absolute value of the elements of row $\mathbf{a}_j$. This means that the absolute value of every element in row j, $|a_{jk}|$ should be smaller than or equal to $s_j \geq 0$. In other words, minimizing the ℓ_∞ norm term,

$$\min_{A \in R^{d \times d}} \lambda \sum_{j=1}^{d} \|\mathbf{a}_j\|_{\ell_\infty} \tag{5.3}$$

can be rephrased as

$$\begin{aligned} \min_{s_j} \lambda \sum_{j=1}^{d} s_j & \\ \text{s.t.} \quad |a_{ij}| \leq s_j, \quad \forall i \end{aligned} \tag{5.4}$$

The optimization problem can therefore be written as:

$$\begin{aligned} \min_{A \in R^{p \times p}} \quad & \|X - \tilde{X}\|^2_{frob} + \lambda \textstyle\sum_{i=1}^{p} s_i \\ \text{s.t.} \quad & -s_i \leq a_{ij} \leq s_i, \forall i, j \end{aligned} \tag{5.5}$$

which is a simple quadratic programming problem with linear constraints. We optimized this objective function using a projected quasi-Newton method based on the L-BFGS algorithm [13].

6 Sparse Principal Component Analysis and CPFS

In this section, we describe sparse principal component analysis (SPCA) and compare it with the newly introduced method, convex principal feature selection (CPFS).

SPCA is not exactly feature selection. Its goal is different from ours. The goal in SPCA is to make the PCs interpretable by finding sparse loadings. Sparsity allows one to determine which features explains which PCs. CPFS, on the other hand, applies a direct approach to feature selection. Our goal is to determine which set of q original features captures the variance of the data most and at the same time are as non-redundant as possible. Another way of saying this is that we find a set of q features that minimizes the reconstruction error between the original data and the data in the spanning space of the selected features.

The SPCA criterion is as follows:

$$\begin{aligned} \min_{\alpha, \beta} \quad & \textstyle\sum_{i=1}^{n} \|X_i - \alpha\beta^T X_i\|^2 + \lambda \sum_{j=1}^{k} |\beta_j|^2 \\ & + \textstyle\sum_{j=1}^{k} \lambda_{1,j} |\beta_j|_1 \\ \text{s.t.} \quad & \alpha^T \alpha = I_k \end{aligned} \tag{6.6}$$

where k is the number of principal components, I_k is an identity matrix of size $k \times k$, λ controls the weight on the ridge regression (norm-2) penalty, and $\lambda_{1,j}$ are weight parameters to control the sparsity. α and β are $p \times k$ matrices. Let $\hat{\alpha}$ and $\hat{\beta}$ be the parameters that minimize the SPCA criterion in Equation 6.6. $\hat{\beta}_i \approx V_i$ (the loadings) for $i = 1, \ldots, k$.

SPCA is a good way to sparsify the loadings of principal components or determine which features correspond to each PC, however it is not appropriate for global feature selection (i.e., find a set of features to represent all the PCs). To illustrate this, we provide an example using the Glass data from the UCI repository [14]. Glass data has a total of nine features, 214 samples, and six classes. Table 1 presents the PC loadings obtained by applying SPCA with a sparsity of five for each PC and the number of PCs, k, set to two. Note that although each PC have sparse loadings, all features have non-zero loadings to explain both PCs. In this case, all features are still needed and no reduction in features is achieved. SPCA has a tendency to spread the non-zero loadings to different features in different PCs because the sparse PCs are constrained to be orthogonal.

Table 1: PC Loadings Applied to Glass Data Using SPCA with Sparsity 5 and 2 PCs.

Loadings	PC1	PC2
Feature 1	0	-0.76
Feature 2	0.35	0
Feature 3	-0.60	0
Feature 4	0.47	0
Feature 5	0	0.03
Feature 6	0	0.11
Feature 7	0	-0.64
Feature 8	0.53	0
Feature 9	-0.14	-0.07
Feature 10	0.42	0

From Table 1, it is not clear how one can utilize SPCA to select features. Let us say, we wish to retain two features. Which two features should we keep? Features 3 and 8 based on the loadings in PC1 or features 3 and 1 the top loadings of PC1 and PC2 respectively? Another complication is that in SPCA one can tweak the sparsity parameter and the number of components to keep. Changing those parameters modifies the loadings and the features with the non-zero loadings as shown in Table 2.

Now, if we consider the A matrix that optimizes our objective function, in Table 6 we can see that the sparse loadings of our A matrix directly identifies which features to keep. Here λ is set to two hundred to keep

Table 2: PC Loadings Applied to Glass Data Using SPCA with Sparsity 2 and 5 PCs.

	PC1	PC2	PC3	PC4	PC5
F1	-0.4423	0	0	0	0
F2	0	0	0.5881	0	0.1360
F3	0	-0.9093	0	0	0
F4	0	0.4161	0	0	0
F5	0	0	0	0.9519	0
F6	0	0	-0.8088	0	0
F7	-0.8969	0	0	0	0
F8	0	0	0	-0.3065	0
F9	0	0	0	0	-0.9907
F10	0	0.3972	0	0	0

five features. Rows 1, 2, 5, 9 and 10 are almost zero and are highlighted in blue font for clarity. The diagonal elements corresponding to the other features are close to one, demonstrating that features 1, 2, 5, 9 and 10 are the ones to be removed, keeping the other five features.

7 Experiments and Discussion of Results

In this section, we investigate whether or not CPFS can successfully capture a smaller reconstruction error of the data sample matrix for compression purposes compared to other PCA-based feature selection methods. Measuring reconstruction error retained is desirable since minimizing the reconstruction error of the original data reveals the amount of loss attained by the different feature selection methods.

In particular, we compare CPFS to SPCA, two eigenvector-loading-based methods by Jolliffe [6], iterative (I) and non-iterative (NI), and Principal feature analysis (PFA) [9]. *Jolliffe_I* iteratively computes the first eigenvector. It starts by selecting the feature with the largest loading in the first PC; removes the chosen feature, re-computes the first eigenvector; and repeat the process until q features are selected. *Jolliffe_NI* computes all the eigenvectors at once and selects q features corresponding to the largest loading in the first q eigenvectors. As mentioned in Section 6, it is not clear how to translate SPCA to perform feature selection. In our experiments, we set SPCA with sparsity equal to q (the number of features to be selected) and set the number of PCs to keep equal to one (to avoid ambiguities). This version of feature selection will be aggressive in selecting features that maximize variance. The other extreme is to keep q PCs and sparsity equal to one. This version will be aggressive in removing redundancy and provides results similar to Jolliffe_NI, and are thus not provided to avoid clutter in the results. It is not clear how to select features in SPCA that is somewhere in between these two extremes. We also compared our algorithm to principal feature analysis (PFA) [9] which performs clustering in the feature space; here we use Kmeans with 10 random starts for PFA.

To evaluate the performance of these methods, we plot the reconstruction error as more and more features are selected. Note that in the SPCA literature [4], the quality of results are sometimes evaluated using the proportion of explained variance. Below, we define both the reconstruction error (E) and the proportion of explained variance (PEV).

The reconstruction error (E) is equal to:

$$\texttt{E} = \|X - XA\|_{frob}^2.$$

Let $F_{sel} = [f_{q_1}, f_{q_1}, \ldots, f_{q_q}]$ be a matrix whose columns are the columns corresponding to the set of selected features, and $F_{sel} = U_r S_r V_r^T$ be the SVD decomposition of F_{sel}. Let $X = USV^T$ be the SVD decomposition of X. The proportion of explained variance (PEV), is equal to:

$$\texttt{PEV} = \frac{tr((U_r^T X V_r)^2)}{tr(S^2)}.$$

The reconstruction error and the proportion of explained variance are simply related as follows: $\frac{E}{\|X\|_{frob}^2} = 1 - \texttt{PEV}$. The lower the error the better, and the higher the PEV the better. Since both PEV and reconstruction error provided similar information about the obtained projections, we simply present the results in terms of reconstruction error normalized by $\|X\|_{frob}^2$.

We ran experiments on eight data sets: glass, face, HRCT, Gene_ALL, Ionosphere, Pima Indians, Bupa Liver and Boston Housing. Glass is a small data set that we used in Section 6 as an example. Face data is from the UCI KDD repository [15] consisting of 640 face images of 20 people. Each person has 32 images with image resolution 30 × 32. We remove missing data to form a 624×960 data matrix. HRCT [16] is a real-world data set from high-resolution computed-tomography images of the lungs (HRCT), with eight disease classes, 1545 instances and 183 features. Ionosphere, Pima Indians, Bupa Liver and Boston Housing are from the UCI repository [17]. The Ionosphere data consists of 351 samples and 34 features; the Pima Indians data has 768 samples and eight features; Bupa Liver contains 345 samples and six features; and Boston Housing has 506 samples with 13 features. Gene_ALL [18] is from Kent Ridge Bio-medical data set repository. The original Gene_ALL data contains 6817 human genes

Features	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
F1	0	0	0	0	-0.0011	0	0	0	0	0
F2	0.0017	0	0	0	0	0	0	0	0	0.0022
F3	0.0202	0.1026	**0.7248**	0.0211	0.0784	0.0344	0.0170	0.0021	0.0953	0.0148
F4	0.1991	0.3082	0	**0.8073**	0.0611	0.1016	0.0104	0.0821	0.0486	-0.1894
F5	0	0	0	0	0.0010	0	0	0	0.0033	0
F6	-0.1743	-0.0880	0	-0.0011	0.0105	**0.7229**	0.0116	0.0072	-0.0578	0.0896
F7	0.0251	0.0340	0.0013	0	0.0283	0	**0.5290**	0.0907	0.0088	0.0115
F8	-0.1190	0.1407	-0.0014	0.0170	0.0069	0.0125	0.0043	**0.6833**	0.0061	0.0723
F9	0	0	0.0047	0	0	0	0	0	0.0059	0.0026
F10	0	0	0	0	0.0012	0	0	0	0	0.0024

Table 3: The weight matrix A with $\lambda = 200$ (i.e., five features selected). The non-zero diagonal elements are shown in bold font. The zero rows are highlighted in blue font. These results show that features 1, 2, 5, 9 and 10 are effectively removed.

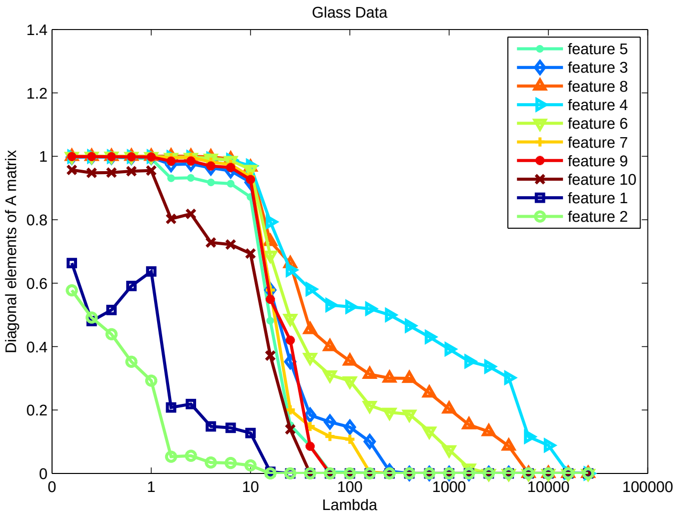


Figure 1: The diagonal elements of the weight matrix A as a function of λ for the Glass data.

over 7129 probes. After pre-processing, there are 327 observations, 472 features, and seven classes. These data sets provide us with a wide range of feature dimensions. In the face and gene data, there are more features than samples, $p > n$.

As discussed earlier in Section 4, row sparsity and correspondingly the number of features selected is controlled by λ. In Figure 1, we plot the values of the diagonal elements of the weight matrix A as a function of λ for the Glass data. The coefficient in each diagonal corresponds to the importance of its associated feature. Observe that as λ increases, more feature weights go to zero (i.e., more features are removed). The graph also reveal a ranking of the features. It shows that feature two is the least important, since it goes to zero the fastest compared to the rest. Moreover, such a graph indicates what λ corresponds to the number of selected features.

Figures 2 to 9 display the reconstruction error as a function of the number of features retained for each of the data sets and the different feature selection methods. The smaller the error for each number of retained features, q, the better. Also, in Table 7, we provide a summary of the results for all of the data sets for the different methods, with the best error value for each data shown in bold. These results show that CPFS is consistently the best compared to SPCA, Jolliffe_NI and Jolliffe_I, and PFA. Jolliffe_I iterative was consistently the worst. Jolliffe_NI, which considers redundancy and variance, is usually better than SPCA which aggressively selects features that maximizes variance when the number of retained features is small. The performance of PFA is close to Jolliffe_NI due to the equivalence of PCA and k-means, but PFA is slightly worse because k-means can get stuck at local minima. *In summary, CPFS efficiently selects features that provide a good balance in keeping the minimum reconstruction error while at the same time minimizing feature redundancy.*

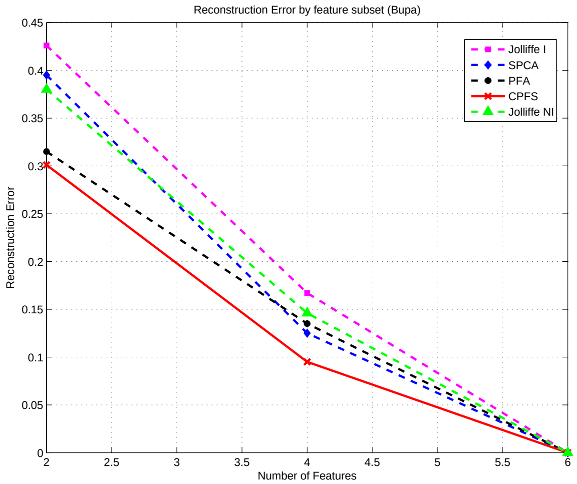


Figure 2: The reconstruction error captured versus the number of features on the Bupa data.

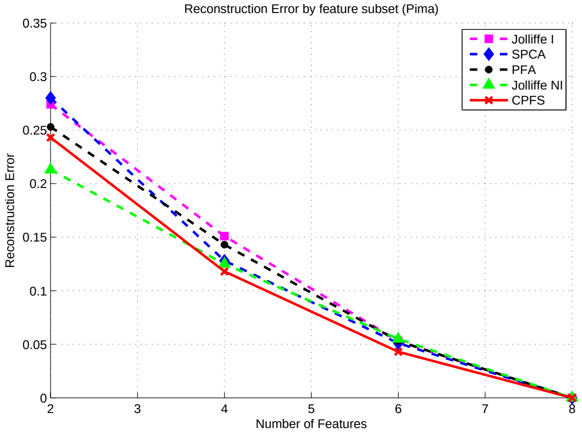


Figure 3: The reconstruction error captured versus the number of features on the Pima data.

7.1 Features Selected on Face Data PCA is commonly used for pre-processing data (for example, to build templates). However, it does not inform us on which features are important. Let us say we wish to know which pixels are important for capturing face pictures. This can be useful for tracking faces or for designing placement of sensors. Here we applied CPFS on the face data containing 640 face images of 20 different people with different poses. Figure 10 shows the result of feature selection. The dots show the selected pixels (features) for four different numbers of selected features (10, 20, 30 and 40). To provide an idea on what the selected pixels correspond to on a face image, we display the dots over one of the face images in this database. Note that the pixels selected fall on the face. Moreover, most of them are located on one side of the face; since face images are symmetric, the right side is redundant to the information on the left.

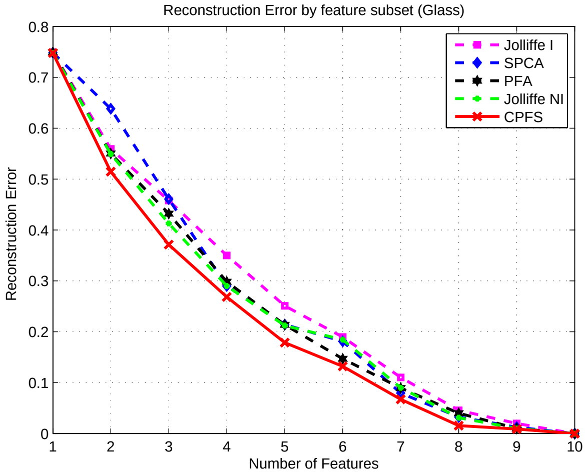


Figure 4: The reconstruction error captured versus the number of features on the Glass data.

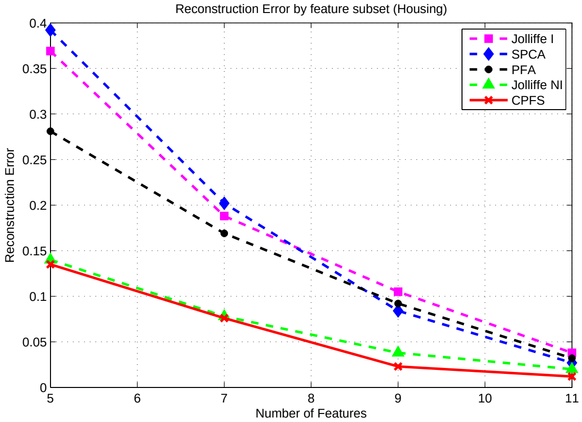
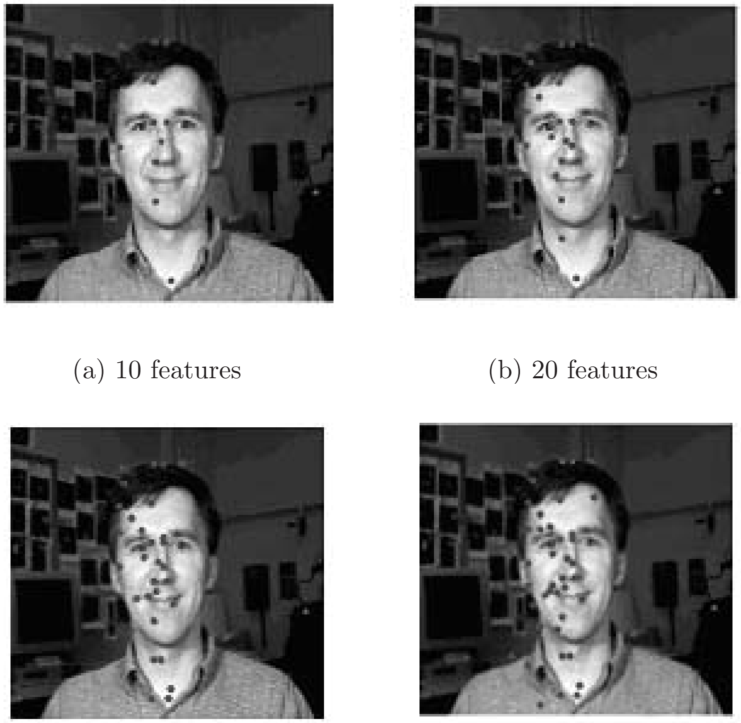


Figure 5: The reconstruction error captured versus the number of features on the Housing data.

7.2 Features Selected on Text Data We can also apply CPFS on text data to summarize the contents of a

Table 4: 5-fold Cross-Validated Classification Errors on Real Data

	# OF ORIGINAL	# OF SELECTED	CPFS	JOLLIFFE NI	JOLLIFFE I	SPCA	PFA
	FEATURES	FEATURES					
BUPA	6	2	**0.301**	0.380	0.426	0.395	0.315
PIMA	8	2	0.243	**0.213**	0.274	0.280	0.253
GLASS	10	5	**0.179**	0.212	0.251	0.214	0.215
HOUSING	13	5	**0.135**	0.140	0.369	0.392	0.281
IONOSPHERE	34	20	**0.021**	0.028	0.047	0.032	0.046
HRCT	183	30	**0.094**	0.136	0.225	0.151	0.149
GENE-ALL	472	90	**0.012**	0.014	0.022	0.016	0.016
FACE	960	90	**0.100**	0.137	0.249	0.188	0.138



(a) 10 features

(b) 20 features

(c) 30 features

(d) 40 features

Figure 10: The first 10, 20, 30 and 40 selected features (pixels) for the face image data.

group of documents based on non-redundant words that are most informative of that group. As an example, we applied our algorithm on the NIPS text data from the UCI repository [14]. It contains actual words from 1500 documents from full papers at `books.nips.cc`. The number of words in the NIPS data vocabulary after the removal of stop-words and only keeping words that occur more than ten times, is 12419. This collection has a total of about $6,400,000$ words. The first 30 words (features) selected by CPFS are: {*computation, dimentional, family, language, measured, equation, label, partitioned, noiseless, evaluation, coding, voltage, cost, initialization, vector, pattern, problem, error, model, network, training, neural, data, algorithm, unit, input, function, learning, dominant and face*}. Note that these words are reasonable representatives for this collection of documents.

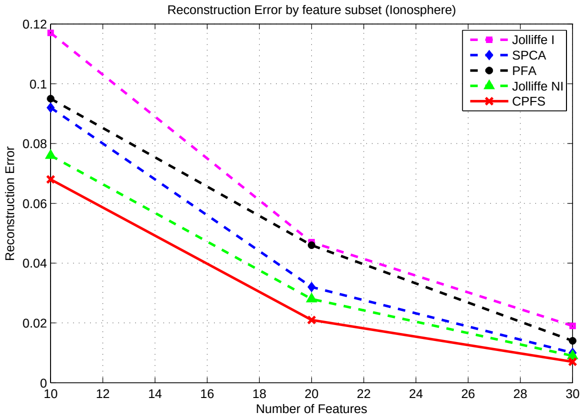


Figure 6: The reconstruction error captured versus the number of features on the Ionosphere data.

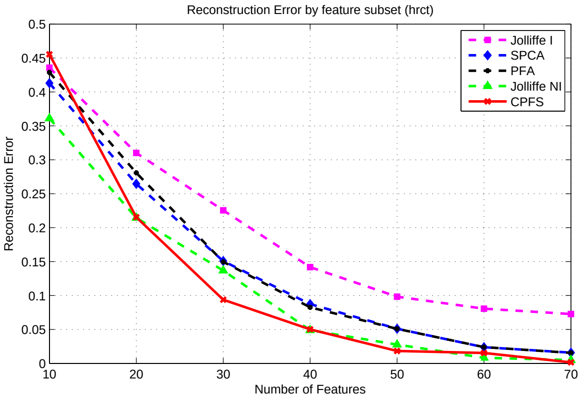


Figure 8: The reconstruction error captured versus the number of features on the high-resolution computed tomography images of the lungs (HRCT) data.

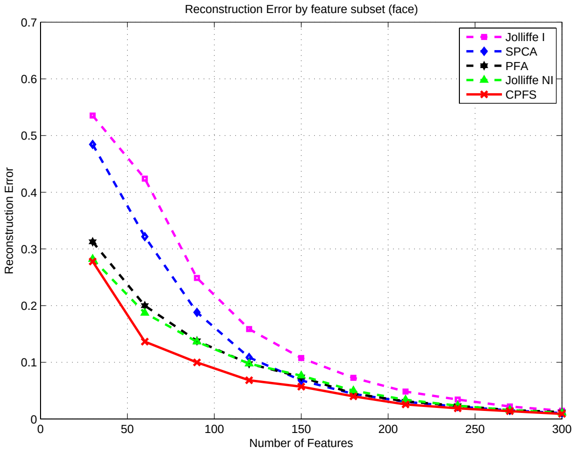


Figure 7: The reconstruction error captured versus the number of features on the Face data.

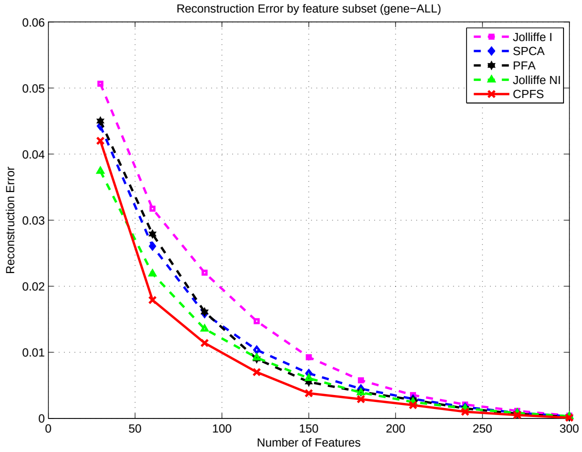


Figure 9: The reconstruction error captured versus the number of features on the Gene_ALL data.

8 Conclusion

Feature selection is an NP-hard combinatorial optimization problem. In this paper, we presented a new method, Convex Principal Feature Selection (CPFS), that relaxes the feature selection problem into a continuous convex optimization problem. As such, we avoided a computationally intensive combinatorial search. In particular, we applied CPFS to re-formulate the popular transformation-based dimensionality reduction algorithm, PCA, into a feature selection problem.

SPCA is a recent approach for sparsifying PCA. Although SPCA allows one to determine which features explain each principal component, it is not designed for feature selection. We pointed out that indeed the nonzero coefficients in SPCA may be spread out over all features in different PCs. Our experiments show that CPFS is consistently the best or close to the best in selecting features that minimize the reconstruction error in the data compared to SPCA, PFA, and iterative and non-iterative loading-based methods. CPFS performed well because it provided a good balance in selecting features that minimize the reconstruction error (equivalently, maximize the variance) and minimize redundancy, and is guaranteed to reach the global solution to our relaxed formulation of the problem.

Moreover, we were able to re-formulate a transformation-based dimensionality reduction algorithm to feature selection by constraining the structure of the weight matrix A in the CPFS formulation. This technique can also be utilized in other dimensionality reduction methods and would be an interesting direction for future work.

Acknowledgments

This work is supported by NSF Career IIS-0347532.

References

[1] G.P. McCabe. Principal variables. *Technometrics*, 26:127–134, 1984.

[2] J. Kittler. Feature set search algorithms. In *Pattern Recognition and Signal Processing*, pages 41–60, 1978.

[3] Isabelle Guyon and André; Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[4] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):262–286, 2006.

[5] R Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

[6] I.T. Jolliffe. *Principal Component Analysis*. Springer, second edition edition, 2002.

[7] W. J. Krzanowski. Selection of variables to preserve multivariate data structure, using principal components. *Applied Statistics*, 36(1):22–33, 1987.

[8] K. Z. Mao. Identifying critical variables of principal components for unsupervised feature selection. *IEEE transactions on system, man, and cyberbetics-part B: cyberbetics*, 35(2):339–344, 2005.

[9] Y. Lu, I. Cohen, X. S. Zhou, and Q. Tian. Feature selection using principal feature analysis. In *Proceedings of the 15th international conference on Multimedia*, pages 301 – 304, 2007.

[10] Y. Cui and J. G. Dy. Orthogonal principal feature selection via component analysis. In *Sparse Optimization and Variable Selection Workshop at the International Conference on Machine Learning*, Helsinki, Finland, July 2008.

[11] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[12] M. S. Bazaraa, D. S. Hanif, and C. M. Shetty. *Nonlinear Programming Theory and Algorithms*. Wiley-Interscience, 2006.

[13] Jorge Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980.

[14] C.L. Blake and C.J. Merz. UCI repository of machine learning databases. In *http://www.ics.uci.edu/∼mlearn/MLRepository.html*, 1998.

[15] S. D. Bay. The UCI KDD archive, 1999.

[16] J. G. Dy, C. E. Brodley, A. Kak, L. S. Broderick, and A. M. Aisen. Unsupervised feature selection applied to content-based retrieval of lung images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(3):373–378, March 2003.

[17] P. Murphy and D. Aha. Uci repository of machine learning databases. Technical Report Technical Report, University of California, Irvine, 1994.

[18] J. H. Golub and D. K. Slonim. Molecular clasification of cancer: Class discorvery and class prediction by gene expression monitoring. *Science*, 286:537–537, 1999.