CARDINAL SPARSE PARTIAL LEAST SQUARE FEATURE SELECTION AND ITS APPLICATION IN FACE RECOGNITION

Honglei Zhang, Serkan Kiranyaz, Moncef Gabbouj

Department of Signal Processing, Tampere University of Technology

ABSTRACT

Many modern computer vision systems combine high dimensional features and linear classifiers to achieve better classification accuracy. However, the excessively long features are often highly redundant; thus dramatically increases the system storage and computational load. This paper presents a novel feature selection algorithm, namely cardinal sparse partial least square algorithm, to address this deficiency in an effective way. The proposed algorithm is based on the sparse solution of partial least square regression. It aims to select a sufficiently large number of features, which can achieve good accuracy when used with linear classifiers. We applied the algorithm to a face recognition system and achieved the state-of-the-art results with significantly shorter feature vectors.

Index Terms— Feature selection, sparse partial least square, face recognition

1. INTRODUCTION

“Curse of dimensionality” are often referred by researchers when explaining the difficulties dealing with the high dimensional data in optimization, numeric analysis, machine learning, and etc [1]. Never the less, the exteriorly contradicting phrase “blessing of dimensionality” have recently appeared to describe the opposite effects. Donoho pointed out that high dimensional data are inevitable and would be surely helpful if one exploits the blessings [2]. In this paper, we present a feature selection algorithm, namely the cardinal sparse partial least square algorithm, which exploits the blessings by keeping the number of selected features large.

It is well known that two manifolds that are generated by continuous functions can be better separated by a hyperplane if they are projected into higher dimensional space. Hall proved that asymptotically high dimensional data tends to lie at the vertices of a simplex [3]. He further showed that the classification accuracy converges to 1 asymptotically if certain conditions meet. The kernel trick in support vector machine is based on the same theory [4].

Based on these observations, computer vision researchers are in favor of the combination of high dimensional features and linear classifiers [5, 6]. It is easier to design this kind of system since longer features can represent various aspects of the data. High dimensional features are formulated by different techniques: concatenation of different types of features [6], spatial pyramid and partition of the images [5], and self-expansion features [7]. The dimension of these features varies from tens of thousands to several millions.

However, excessively long features increase storage load and computational complexity. Chen *et al.* showed that a system saturates at certain length of features–adding new features shows no or very little impact [8]. Dimension reduction methods–including feature selection and feature transformation–are necessary to reduce the system complexity with little or no accuracy loss.

Feature selection is normally used when data contains irrelevant or redundant features. Irrelevant features are common in bioinformatic data, in which a very small amount of the features are associated with the classification or regression response; on the contrary, features in computer vision problems are often highly redundant. Comparing to the aforementioned feature transformation techniques, feature selection algorithms are able to generate longer feature vectors, which is important for classification accuracy.

Feature selection is a combinatorial problem which is NP-hard. Applying sparsity constraints, often an L_1 norm of the coefficients in the objective function, yields a good approximation. Jolliffe *et al.* gave a sparse solution of the principal component analysis [9]. Inspired by their work, Qiao *et al.* applied sparsity to the linear discriminant analysis [10]; Chun and Keles applied sparsity to the partial least square (PLS) regression [11].

The algorithm proposed in [11] optimizes the target function by a predefined threshold value, which is not directly related to the number of selected features. This makes it difficult to determine the number of selected features by observing the saturation effect. It aslo shows high computational complexity, especially when the number of features and the number of samples are large.

We adapt the sparse PLS algorithm and propose a novel and efficient method–cardinal sparse partial least square (SPLS)–that is more suitable for computer vision problems by directly targeting the number of selected features. The performance of the cardinal SPLS algorithm are illustrated in a face recognition system. The rest of the paper is organized as follows. In section 2, we give brief introduction of PLS and sparse PLS algorithm. The proposed cardinal SPLS algorithm is presented in section 3. Experiments using simulated data and face recognition data are presented in section 4. Finally, conclusive remarks and topics for our future work are discussed in section 5.

2. SPARSE PARTIAL LEAST SQUARE

2.1. Partial Least Square Regression

Partial least square regression (PLSR) models the relationship between the predictors[1] and the response variables by means of latent variables. It is a linear regression model that projects predictors into a new set of latent variables in a lower dimensional subspace. The projection optimizes the discriminant property of the latent variables.

Let X be the $N \times p$ matrix of zero-mean predictor sample data and Y be the $N \times q$ matrix of zero-mean response sample data, where N is the number of samples, p is the number of the predictors, and q is the dimension of the response variables. PLS decomposes X and Y into the form

$$\begin{array}{l} X = TP^T + E \\ Y = TQ^T + F \end{array}, \tag{1}$$

where the $N \times K$ matrix T is the latent component matrix of X and Y, K is the number of latent variables, P and Q are the orthogonal loading matrices, E and F are the residual matrices. The superscript T denotes the matrix transpose.

The latent component matrix T is derived from $T = XW$, where the $p \times K$ matrix W is the PLS projection direction matrix whose columns are the PLS projection directions. The k-th direction vector $\hat{w}_k$ is found by solving the optimization problem

$$\begin{array}{ll} \text{maximize} & w^T M w \\ \text{subject to} & w^T w = 1 \\ & w^T S_{XX} \hat{w}_l = 0 \quad l = 1, \dots k-1 \end{array}, \tag{2}$$

where $M = X^T Y Y^T X$ and $S_{XX} = X^T X$ is the covariance matrix of the predictors.

2.2. Sparse PLS

Sparsity can be achieved by adding L_1 norm term into (2). However the problem is not convex, thus no direct numeric solution is available. By introducing a surrogate variable c, Chun and Keles [11] transformed the objective function into the form

$$\begin{array}{ll} \text{minimize} & -\kappa w^T M w \\ & +(1-\kappa)(c-w)^T M (c-w) \\ & +\lambda_1 \left\|c\right\|_1 + \lambda_2 \left\|c\right\|_2^2 \\ \text{subject to} & w^T w = 1 \end{array}, \tag{3}$$

where the variables are w and c. For simplicity we chose $\kappa = 1/2$. The problem can be approximately solved by alternatively optimizing the object function with regard to variable c and w. When c is fixed, the analytical solution for w is available.

When w is fixed, the objective function becomes a standard elastic net form:

$$\begin{array}{ll} \text{minimize} & \left(Z^T c - Z^T w\right)^T \left(Z^T c - Z^T w\right) + \\ & +\lambda_1 \left\|c\right\|_1 + \lambda_2 \left\|c\right\|_2^2 \end{array}, \tag{4}$$

where $Z = X^T Y$.

If λ_2 is large, the solution of c can be found by a soft thresholded estimator.

The soft-thresholding function is given by

$$\hat{c} = \operatorname{sign}(c) \left(|c| - \eta \max(|c|)\right)_+ , \tag{5}$$

where $(x)_+$ is defined as

$$(x)_+ = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}. \tag{6}$$

Here the parameter η, $\eta \in (0,1)$, is a replacement of the parameter λ_1 which controls the sparsity of w.

After $\hat{w}_k$ is found from the optimization problem (3), matrices X and Y are deflated. The procedure iterates until all projection directions are found. The iteration number is the number of latent variables K.

For clarity and simplicity, we denote this algorithm as SPLS-η.

3. CARDINAL SPLS ALGORITHM

3.1. Cardinal SPLS

Recall the computer vision systems we described in the section 1, our goal is to select a sufficiently large number of features. In practice, this goal is difficult to achieve with the SPLS-η algorithm, because the parameter η is not proportional to the number of selected features. SPLS-η does not specify whether and how different η value should be used in each iteration.

We first rewrite the equation (4). Let $A^{(k-1)}$ be the selected feature set after the iteration step $k-1$. At iteration k, we shall not penalize the objective function for selecting features that have been already selected in $A^{(k-1)}$. Denote $B^{(k)}$ as the weight matrix where $B_{ii}^{(k)} = 1$ if $i \notin A^{(k-1)}$; all other elements in $B^{(k)}$ are zeros. To get a sparse solution of c at iteration k, the objective function becomes

$$\begin{array}{ll} \text{minimize} & \left(Z^T c - Z^T w\right)^T \left(Z^T c - Z^T w\right) + \\ & +\lambda_1 \left\|B^{(k)} c\right\|_1 + \lambda_2 \left\|c\right\|_2^2 \end{array}. \tag{7}$$

[1] Note, in this paper the term "predictor" has the same meaning as the term "feature". They are both used because of the preference in statistics and machine learning.

Imposing $\lambda_2 = \infty$, the soft-thresholding function 5 becomes

$$\hat{c} = \operatorname{sign}(c) \left(|c| - s(c, n^{(k)}, A^{(k-1)}) \right)_{+,} \tag{8}$$

where $n^{(k)}$ is the number of features that we select at iteration k, and $s\left(c, n^{(k)}, A^{(k-1)}\right)$ is the $n^{(k)}$-th largest value of the set $\left\{x | x = |c_i|, i \notin A^{(k-1)}\right\}$.

Next, we will calculate $n^{(k)}$ given n–the cardinality of the target feature set. We assume that each feature contribute equally to the object function at the iteration step k. Note that earlier selected features contribute to later iterations, we derive the following equation to determine $n^{(k)}$:

$$n^{(k)} = \left[\frac{2(K-k+1)}{K(K+1)} \cdot n \right], \tag{9}$$

where $[\cdot]$ denotes the nearest integer function, K is the number of latent variables.

The proposed algorithm is described in algorithm 1. For simplicity, our algorithm is based on SIMPLS paradigm [12].

Algorithm 1 Cardinal Sparse PLS Feature Selection

```
given training data $X$ and $Y$, the target number of features $n$,
and the number of latent variables $K$
normalize $X$ and $Y$ with zero means
iterate $k = 1, \ldots, K$
  1. $Z = X^T Y$
  2. calculate $n^{(k)}$ by (9)
  3. find $c^{(k)}$ and $w^{(k)}$ by solving (3) using (7) and soft-thresholding function (8)
  4. $A^{(k)} = A^{(k-1)} \cup \left\{ i \mid c_i^{(k)} \neq 0 \right\}$
  5. calculate $W^{(k)}$ by PLS regression using $X_{A^{(k)}}$ and $Y$, where $X_{A^{(k)}}$ denotes the predictor data matrix including only selected features in $A^{(k)}$
  6. deflate $X$ by $X_{A^{(k)}} = X_{A^{(k)}} \left(I - P(P^T P)^{-1} P^T\right)$, where $P = X_{A^{(k)}}^T X_{A^{(k)}} W \left(W^T X_{A^{(k)}}^T X_{A^{(k)}} W\right)^{-1}$
```

3.2. Selection of the K

The experimental results show that the cardinal SPLS algorithm performs equally well when $K \geq 3$. For $K \geq 10$, the performance varies only slightly. This observation can be explained from equation (9). When $k \to K$, $n^{(k)} \approx \frac{2}{K^2} n$. For large K, only a few variables are selected in later iterations. In practice we can simply choose K between 3 and 10. Larger K values increase the computational complexity without much benefit. This guidance aligns with the empirical practices that have been suggested by other researchers [13, 14].

4. EXPERIMENTAL RESULTS

4.1. Simulated Data

Our experimental data set contains 100 data samples. The feature vector has dimension of 1000, among which the first 200 variables are the true features which are generated from an underlying true data model, and the rest 800 features are iid Gaussian noise generated from $\mathcal{N}(0, 0.3)$. The true data model is a linear model:

$$\begin{aligned} x_i &= z_i A + e_i \\ y_i &= z_i B + f_i \end{aligned} \qquad i = 1, \ldots, 100, \tag{10}$$

where $z_i \in \mathbb{R}^{10}$ are the latent variables that are generated from $\mathcal{N}(0, 1)$, $A \in \mathbb{R}^{10 \times 200}$ and $B \in \mathbb{R}^{10 \times 5}$ are the conversion matrices, e_i and f_i are addition iid Gaussian noise that are generated from $\mathcal{N}(0, 0.5)$. Matrices A and B are generated from $\mathcal{N}(0, 0.2)$ and $\mathcal{N}(0, 0.3)$ during initialization.

We compared the cardinal SPLS algorithm with three other variable selection algorithms: naive variance, group LASSO [15], sparse PCA [16] and SPLS-η. The naive variance variable selection method simply selects predictors with the highest variance value. All competing algorithms selected around 200 predictors from the same simulated data. To illustrate the effect of signal to noise ratio, we chose 3 different standard deviation values of e_i and f_i (0.5, 0.75 and 1). We repeated the experiments 100 times and the average recall values of each algorithm are listed in Table 1.

Table 1. Recall Performance on Simulated Data

METHOD	$\sigma = 0.5$	$\sigma = 0.75$	$\sigma = 1$	$\sigma = 1.5$
Naive Variance	**0.977**	0.422	0.026	0
Group Lasso	0.599	0.483	0.524	0.219
Sparse PCA	0.722	0.620	0.491	0.051
SPLS-η	0.827	0.665	0.556	0.358
Cardinal SPLS	0.869	**0.738**	**0.595**	**0.372**

Table 1 clearly demonstrates the superiority of the cardinal SPLS algorithm against other methods.

We also evaluated the computational complexity of different algorithms using the simulated data. Each algorithm selected 2K features from the original 5K, 10K and 50K features. Table 2 shows the computational run time in seconds.

Table 2. Computational Run Time on Simulated Data

METHOD	5K	10K	50K
Group Lasso	8.0	12.9	60.3
Sparse PCA	135	138	157
SPLS-η	4.9	10.1	117.1
Cardinal SPLS	**0.7**	**0.8**	**1.5**

4.2. Application in Face Recognition

4.2.1. Frontal Face Recognition System

In this section we demonstrate the performance of the cardinal SPLS feature selection algorithm in a face recognition system. The system recognize a query face image given a gallery image dataset.

We used local Gabor binary patterns (LGBP) feature descriptor [17] and a linear model classifier in our experiments. The models are built in this way: for each person, we train a model–linear classifier–using his/her image(s) in the gallery as positive samples and a preselected face image dataset as negative samples. During the query stage, the system evaluates the query image against all models and assign it to the one that gives the highest classification score.

We aligned, scaled and rotated all images according to the location of the eyes. Each face image has the size 140×154 pixels. The LGBP features were generated using 40 Gabor filters (5 scales and 8 orientations), uniform local binary patterns codes(LBP_8^{u2}) and 7×7 blocks. The histograms of the LBP codes were L_2 normalized. The negative image dataset contains 1522 frontal face images that are collected from ATT [18], CMU PIE [19], MOBIO [20], FEI [21], SUMS [22] face databases. To ensure the generality, we used MOBIO face database for feature selection training.

4.2.2. FERET Face Database Result

We used the FERET face database [23], to evaluate the performance of the cardinal SPLS algorithm.

The raw LGBP feature vector has dimension over 115K. It would be intractable to use this long feature vector in a model based classification system. It is also difficult to apply any embedded feature selection method directly. We executed feature selection in two stages: screening stage and moderate stage. In the screening stage, we used the naive variance feature selection method to select 20K features that has the greatest variance. In the moderate stage, we applied the cardinal SPLS feature selection algorithm to further reduce the feature dimension to 10K.

We compared cardinal SPLS feature selection algorithm with naive variance, SPLS-η, and ANOVA method. The performance scores are plotted in Figure 1. The score reported in the figure is the recognition rate, which is the ratio of the number of correct recognized images to the total number of the query images.

The Figure 1 shows that the system using cardinal SPLS algorithm saturates faster than the naive variance and ANOVA algorithms as the feature dimension increases. The cardinal SPLS has slightly better performance than the SPLS-η algorithm. But comparing to SPLS-η, the cardinal SPLS algorithm is much faster.

We also compared our results to the state-of-the-art results of the FERET database in Table 3. Fa, Fb, Dup1 and Dup2 are the most used datasets from the Feret database. Each dataset addresses a different challenge in face recognition. The last column shows the feature vector length used in the comparing system.

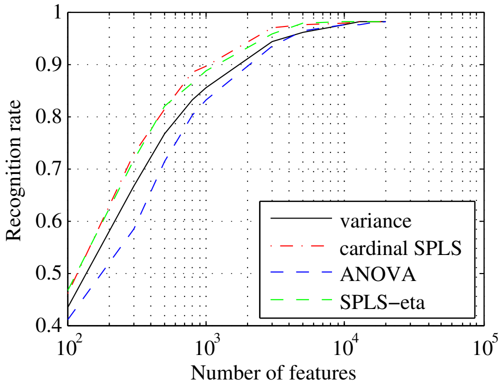

Fig. 1. Comparison of Different Feature Selection Algorithms on FERET fa Face dataset

Table 3. FERET Face Database Recognition Result

METHOD	Fa	Fb	Dup1	Dup2	Features
[17] 2005	0.98	0.97	0.74	0.71	519K*
[24] 2007	0.975	**0.995**	0.795	0.778	2.9M
[25] 2012	**0.99**	0.93	0.76	0.78	39K*
[6] 2012	0.972	0.985	**0.853**	**0.855**	75K
Our system	0.981	0.99	0.784	0.782	**10K**

* Estimated from the paper descriptions.

5. CONCLUSIONS

We have proposed the cardinal SPLS algorithm that selects the most informative and discriminant features given the target cardinality. The proposed method is more suitable for computer vision problems, in which the raw features are highly redundant. The evaluation on the simulated data and a face recognition system proves the efficiency and superiority of the algorithm.

From our experiments, we noticed the high computational complexity of the cardinal SPLS algorithm if the raw feature vectors are extremely long (over 100K). In our face recognition system, we incorporated a screening stage to select a moderate number of features before applying the cardinal SPLS algorithm. In such a case, an improper screening algorithm may lower the system performance. Future work will concentrate on fast algorithms that is capable of handling ultra high dimensional data.

We have provided a Matlab implementation of the cardinal SPLS algorithm at `http://goo.gl/BA2OHe`.

REFERENCES

[1] R. Bellman, *Adaptive control processes: a guided tour*. Princeton university press Princeton, 1961, vol. 4.

[2] D. L. Donoho, "High-dimensional data analysis: The curses and blessings of dimensionality," *AMS Math Challenges Lecture*, 2000.

[3] P. Hall, J. S. Marron, and A. Neeman, "Geometric representation of high dimension, low sample size data," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 3, 2005.

[4] B. Schölkopf, C. J. Burges, and A. J. Smola, *Advances in kernel methods: support vector learning*. The MIT press, 1999.

[5] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006.

[6] W. Schwartz, H. Guo, J. Choi, and L. Davis, "Face identification using large feature sets," *IEEE Transactions on Image Processing*, vol. 21, no. 4, Apr. 2012.

[7] H. Jégou, M. Douze, C. Schmid *et al.*, "Aggregating local descriptors into a compact image representation," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010.

[8] D. Chen, X. Cao, F. Wen, and J. Sun, "Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification," in *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[9] I. T. Jolliffe, N. T. Trendafilov, and M. Uddin, "A modified principal component technique based on the LASSO," *Journal of Computational and Graphical Statistics*, vol. 12, no. 3, Sep. 2003.

[10] Z. Qiao, L. Zhou, and J. Z. Huang, "Sparse linear discriminant analysis with applications to high dimensional low sample size data," *International Journal of Applied Mathematics*, vol. 39, no. 1, 2009.

[11] H. Chun and S. Keles, "Sparse partial least squares regression for simultaneous dimension reduction and variable selection," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 72, no. 1, 2010.

[12] S. de Jong, "SIMPLS: an alternative approach to partial least squares regression," *Chemometrics and Intelligent Laboratory Systems*, vol. 18, no. 3, Mar. 1993.

[13] S. Wold, M. Sjöström, and L. Eriksson, "PLS-regression: a basic tool of chemometrics," *Chemometrics and Intelligent Laboratory Systems*, vol. 58, no. 2, Oct. 2001.

[14] H. T. Nguyen, K. Franke, and S. Petrovi'c, "On general definition of l1-norm support vector machines for feature selection," *The International Journal of Machine Learning and Computing*, vol. 1, no. 3, 2011.

[15] J. Liu and J. Ye, "Moreau-yosida regularization for grouped tree structure learning," in *Advances in Neural Information Processing Systems*, 2010.

[16] A. d'Aspremont, F. Bach, and L. E. Ghaoui, "Optimal solutions for sparse principal component analysis," *arXiv:0707.0705*, Jul. 2007.

[17] W. Zhang, S. Shan, W. Gao *et al.*, "Local gabor binary pattern histogram sequence (LGBPHS): a novel non-statistical model for face representation and recognition," in *Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005*, vol. 1, Oct. 2005.

[18] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*, 1994.

[19] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression database," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, 2003.

[20] C. McCool, S. Marcel, A. Hadid *et al.*, "Bi-modal person recognition on a mobile phone: Using mobile phone data," in *2012 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2012.

[21] C. E. Thomaz and G. A. Giraldi, "A new ranking method for principal components analysis and its application to face image analysis," *Image and Vision Computing*, vol. 28, no. 6, 2010.

[22] C. Netzer and P. Srinivasan, "Stanford medical student face database," 2001.

[23] P. Phillips, H. Moon, S. Rizvi, and P. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, 2000.

[24] B. Zhang, S. Shan, X. Chen, and W. Gao, "Histogram of gabor phase patterns (hgpp): A novel object representation approach for face recognition," *Image Processing, IEEE Transactions on*, vol. 16, no. 1, 2007.

[25] Z. Lei, D. Yi, and S. Z. Li, "Discriminant image filter learning for face recognition with local binary pattern like representation," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2012.