Constrained Subspace Classifier for High Dimensional Datasets

Panos M. Pardalos

Joint work with Orestis P. Panagopoulos & Petros Xanthopoulos

Work of Panos M. Pardalos was conducted at National Research University Higher School of Economics and supported by RSF Grant 14-41-00039
Information Gathered

- Sample/Data point
  - $X \in \mathbb{R}^D$
  - $D = \# \text{ of features}$
- Data Space
  - $S$ with dimension equal to number of features
Binary Classification

Classifying the samples of set $S$ into two groups according to a classification rule
High Dimensional Datasets

- What are they?
  - High number of features
  - Relative small number of samples

- Examples of high dimensional datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Relationship Management data</td>
<td>(Tseng and Huang, 2007)</td>
</tr>
<tr>
<td>Covariation information of stocks</td>
<td>(Campbell and Lo, 1997)</td>
</tr>
<tr>
<td>Text datasets for classification</td>
<td>(Hassell and Arpinar, 2006)</td>
</tr>
<tr>
<td>Data collected from Surveys</td>
<td>(Belloni and Hansen, 2014)</td>
</tr>
<tr>
<td>Netflix dataset</td>
<td>(Bennett and Lanning, 2007)</td>
</tr>
<tr>
<td>MRI data</td>
<td>(Kampa et al., 2014)</td>
</tr>
<tr>
<td>Mass Spectroscopy data</td>
<td>(Fenn and Pappu, 2012)</td>
</tr>
</tbody>
</table>
Difficulties with High Dimensional Datasets

- **Curse of dimensionality**
  - Can cause model overfitting and estimation instability
  - Common classifiers fail

Difficulties with High Dimensional Datasets

- Volume increases exponentially as dimensionality increases
  - Points tend to become equidistant
  - Metric functions fail

Difficulties with High Dimensional Datasets

- Estimation of class covariance matrix unreliable
  - Most statistical classifiers require knowing class covariances \textit{apriori}
  - Statistical classifiers fail
- \textit{How do we deal with the aforementioned issues?}
  - Reducing the dimensionality of the dataset prior to classification
    - \textit{Feature Selection}
    - \textit{Feature Extraction}
Feature Selection

- Select only a subset of relevant features to use for classification
- Good for removing irrelevant data, increasing learning accuracy, and improving result comprehensibility

Feature Selection

- Categories of feature selection techniques:
  - Filter methods
  - Wrapper methods
  - Embedded methods

Feature Selection

- **Filter methods**
  - Access features during a separate process prior to classification
  - Variables are given a score according to a filtering function and are ordered accordingly
  - Features with the lowest scores are discarded while the rest are used from the classifier

- **Hypothesis testing, t-test**
Feature Selection

- **Wrapper methods**
  - Use the classifier structure itself to evaluate the importance of features
  - Based on the idea that the classifier can provide a better estimate of accuracy than a separate independent process
  - Increased computational power is often required - the classification process has to be repeated for each feature set considered

- **Metaheuristics**
Feature Selection

- **Embedded methods**
  - Perform feature selection in a way so that the classification algorithm is executed while variables are evaluated and selected

- **Recursive feature elimination in SVMs**

- **Random forests for feature evaluation**
Feature Extraction

- Feature extraction techniques transform the input data into a set of meta-features that extract the relevant information from the input data for classification

- *Principal Component Analysis (PCA)*

PCA

- Removes redundancy by transforming the data from a higher dimensional space into an **orthogonal lower dimensional space**
- First principal component captures as much variation in the data as possible - each succeeding component accounts for a decreasing amount of variance
- Number of retained principal components is less than or equal to the number of original variables
  - Criteria: *eigenvalue-one criterion, scree test, proportion of variance accounted for*
Local Subspace Classifier

- Local Subspace Classifier (LSC) utilizes PCA to perform classification

  **Training phase**
  - A lower dimensional subspace is found for each class that approximates the data

  **Testing phase**
  - A new data point is classified by calculating the distance of the point to each subspace and choosing the class with minimal distance

Laaksonen, Jorma *Local subspace classifier* - Artificial Neural Networks ICANN (1997)
Consider a binary classification problem

Let the matrices \( X_1 \in \mathbb{R}^{p \times m} \) and \( X_2 \in \mathbb{R}^{p \times l} \) be given, whose columns represent the training examples of two classes \( C_1 \) and \( C_2 \) respectively.

LSC attempts to find two subspaces separately, one for each class that best approximates the data.

Laaksonen, Jorma *Local subspace classifier* - Artificial Neural Networks ICANN (1997)
Local Subspace Classifier

Let

\[ U_1 = [u_1^{(1)}, u_2^{(1)}, \ldots, u_k^{(1)}]_{p \times k} \] (1)

and

\[ U_2 = [u_1^{(2)}, u_2^{(2)}, \ldots, u_k^{(2)}]_{p \times k} \] (2)

represent orthonormal bases of two \( k \)-dimensional linear subspaces \( S_1 \) and \( S_2 \) that approximate classes \( C_1 \) and \( C_2 \) respectively.

- We assume the dimensionality of subspaces \( S_1 \) and \( S_2 \) to be the same and equal to \( k \)
Training Phase

- $S_1$ and $S_2$ attempt to capture *maximal* variance in classes $C_1$ and $C_2$ respectively by solving the following optimization problems:

\[
\text{maximize } \quad \text{tr}(U_1^T X_1 X_1^T U_1) \\
\text{subject to } \quad U_1^T U_1 = I_k
\]  

The solution to (3) is given by the eigenvectors corresponding to $k$ largest eigenvalues of matrix $X_1 X_1^T$

\[
\text{maximize } \quad \text{tr}(U_2^T X_2 X_2^T U_2) \\
\text{subject to } \quad U_2^T U_2 = I_k
\]  

Similarly the orthonormal basis $U_2$ is obtained by choosing eigenvectors corresponding to $k$ largest eigenvalues of matrix $X_2 X_2^T$
Testing Phase

A new point $x$ is classified by computing the distance from subspaces $S_1$ and $S_2$:

$$\text{dist}(x, S_i) = \text{tr}(U_i^T x x^T U_i)$$  \hspace{1cm} (5)

and the class of $x$ is determined as:

$$\text{class}(x) = \arg\min_{i \in \{1, 2\}} \{\text{dist}(x, S_i)\}$$  \hspace{1cm} (6)
Motivating Example

Motivation

- Though the subspaces $S_1$ and $S_2$ approximate the classes well, these projections may not be ideal for classification tasks as each of them are obtained without the knowledge of another class/subspace.

In order to account for the presence of another subspace, we consider the relative orientation of the subspaces.
Datasets are generated from two bivariate normal distributions \( \mathcal{N}_1(\mu_1, \Sigma_1) \) and \( \mathcal{N}_2(\mu_2, \Sigma_2) \) representing classes \( C_1 \) and \( C_2 \). Each class consists of 100 randomly generated points from \( \mathcal{N}_1 \) and \( \mathcal{N}_2 \) respectively.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>( \mathcal{N}_1 )</th>
<th>( \mathcal{N}_2 )</th>
<th>LSC</th>
<th>CSC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \mu_1 )</td>
<td>( \Sigma_1 )</td>
<td>( \mu_2 )</td>
<td>( \Sigma_2 )</td>
</tr>
<tr>
<td>Example 1</td>
<td>9</td>
<td>4</td>
<td>1.1</td>
<td>2</td>
</tr>
<tr>
<td>Example 2</td>
<td>3</td>
<td>4</td>
<td>-2</td>
<td>10</td>
</tr>
</tbody>
</table>
LSC and CSC are trained on the data with $k = 1$ and the classification accuracies are obtained via 10-fold cross validation.

Example 1:
- LSC
**Motivating Example**

- **Example 1:**
  - CSC
Motivating Example

Example 2:

- LSC
Example 2:
- CSC

These examples show that relative orientation of the subspaces should also be considered in addition to capturing maximal variance in data.
Constrained Subspace Classifier (CSC) finds two subspaces simultaneously, one for each class, such that each subspace accounts for maximal variance in the data in the presence of the other class/subspace.
Relative orientation in terms of principal angles

Definition 1: Let $U_1 \in \mathbb{R}^{p \times k}$ and $U_2 \in \mathbb{R}^{p \times k}$ be two orthonormal matrices spanning subspaces $S_1$ and $S_2$. The principal angles $0 \leq \theta_1 \leq \theta_2 \leq \theta_3 \leq \cdots \leq \theta_k \leq \pi/2$ between subspaces $S_1$ and $S_2$, are defined recursively by:

$$\cos \theta_i = \max_{x_m \in S_1} \max_{y_n \in S_2} x_m^T y_n$$

subject to

$$x_m^T x_n = 1, \quad y_m^T y_n = 1, \quad \text{for } m = n$$

$$x_m^T x_n = 0, \quad y_m^T y_n = 0, \quad \text{for } m \neq n$$

$$\forall m, n = 1, 2, \ldots, k$$

**Theorem 1**: Let $U_1 \in \mathbb{R}^{p \times k}$ and $U_2 \in \mathbb{R}^{p \times k}$ be rectangular matrices whose column vectors span the subspaces $S_1 \in \mathbb{R}^k$ and $S_2 \in \mathbb{R}^k$ respectively. Let $M = U_1^\top U_2 \in \mathbb{R}^{k \times k}$, using SVD we can express $M$ by:

$$M = YCZ^\top$$

(8)

where $Y^\top Y = I_k$, $Z^\top Z = I_k$ and $C = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k)$

If we assume that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k$ then the principal angles are given by $\cos \theta_k = \sigma_k(M) \quad \forall i = 1, 2, \ldots, k$

Seeking a Distance Metric

We consider the metric that defines the relative orientation between the two subspaces $S_1$ and $S_2$ spanned by $U_1$ and $U_2$ respectively to be the projection $F$-norm defined by:

$$d_{pF}(U_1, U_2) = \frac{1}{\sqrt{2}} \| U_1 U_1^T - U_2 U_2^T \|_F$$  \hspace{1cm} (9)

In terms of principal angles

\[ \| U_1 U_1^T - U_2 U_2^T \|^2_F \]

\[ = \text{tr}((U_1 U_1^T - U_2 U_2^T)^T (U_1 U_1^T - U_2 U_2^T)) \]
\[ = \text{tr}(U_1 U_1^T U_1 U_1^T - U_1 U_1^T U_2 U_2^T - U_2 U_2^T U_1 U_1^T + U_2 U_2^T U_2 U_2^T) \]
\[ = \text{tr}(U_1 U_1^T) + \text{tr}(U_2 U_2^T) - 2\text{tr}(U_1 U_1^T U_2 U_2^T) \]
\[ = \text{tr}(U_1^T U_1) + \text{tr}(U_2^T U_2) - 2\text{tr}(U_2^T U_1 U_1^T U_2) \]
\[ = \| U_1 \|^2_F + \| U_2 \|^2_F - 2\| U_2^T U_1 \|^2_F \]

According to Theorem 1:

\[ \| U_2^T U_1 \|^2_F = \sum_{i=1}^{k} \sigma_i^2 = \sum_{i=1}^{k} \cos^2 \theta_i \]
In terms of principal angles

Using (11) on (10) becomes:

\[
\sum_{i=1}^{k} \lambda_i + \sum_{i=1}^{k} \lambda_i - 2 \sum_{i=1}^{k} \cos^2 \theta_i
\]

\[
= k + k - 2 \sum_{i=1}^{k} \cos^2 \theta_i
\]

\[
= 2 \left[ k - \sum_{i=1}^{k} \cos^2 \theta_i \right]
\]

\[
= 2 \left[ (1 - \cos^2 \theta_1) + (1 - \cos^2 \theta_2) + \cdots + (1 - \cos^2 \theta_k) \right]
\]

\[
= 2 \sum_{i=1}^{k} \sin^2 \theta_i
\]
In terms of principal angles

Hence the projection F-norm becomes:

\[
d_{pF}(\mathbf{U}_1, \mathbf{U}_2) = \frac{1}{\sqrt{2}} \| \mathbf{U}_1 \mathbf{U}_1^\top - \mathbf{U}_2 \mathbf{U}_2^\top \|_F = \sqrt{\sum_{i=1}^{k} \sin^2 \theta_i} \quad (13)
\]
The projection metric is utilized to incorporate the relative orientation between subspaces in LSC.

The formulation of LSC is modified as shown below to obtain the *Constrained Subspace Classifier (CSC)*:

\[
\begin{align*}
\text{maximize} & \quad \text{tr}(U_1^T X_1 X_1^T U_1) + \text{tr}(U_2^T X_2 X_2^T U_2) - C \| U_1 U_1^T - U_2 U_2^T \|_F^2 \\
\text{subject to} & \quad U_1^T U_1 = I_k, \quad U_2^T U_2 = I_k \\
\end{align*}
\]

where the parameter $C$ controls the tradeoff between the relative orientation of the subspaces and the approximation of the data.
Using (11) and (12):
\[ \| U_1 U_1^T - U_2 U_2^T \|_F^2 = 2k - 2 \text{tr}(U_1^T U_2 U_2^T U_1) \]

Hence the optimization problem becomes:

\[
\begin{align*}
\text{maximize} & \quad \text{tr}(U_1^T X_1 X_1^T U_1) + \text{tr}(U_2^T X_2 X_2^T U_2) + C \text{tr}(U_1^T U_2 U_2^T U_1) \\
\text{subject to} & \quad U_1^T U_1 = I_k, \quad U_2^T U_2 = I_k
\end{align*}
\]

(15)
We introduce an alternating optimization algorithm to solve (15).

For a fixed $U_2$, (15) reduces to:

$$\text{maximize} \quad \operatorname{tr}(U_1^T(X_1X_1^T + C U_2 U_2^T)U_1)$$

subject to $U_1^T U_1 = I_k$

The solution to (16) is obtained by choosing eigenvectors corresponding to $k$ largest eigenvalues of symmetric matrix $X_1X_1^T + C U_2 U_2^T$. 

Panos M. Pardalos

Constrained Subspace Classifier for HDD
Similarly, for a fixed $\mathbf{U}_1$, (15) reduces to:

$$
\text{maximize } \mathbf{U}_2 \in \mathbb{R}^{p \times k} \quad \text{tr}(\mathbf{U}_2^T (\mathbf{x}_2 \mathbf{x}_2^T + C \mathbf{U}_1 \mathbf{U}_1^T) \mathbf{U}_2)
$$

subject to $\mathbf{U}_2^T \mathbf{U}_2 = \mathbf{I}_k$

where the solution to (17) is again obtained by choosing eigenvectors corresponding to $k$ largest eigenvalues of symmetric matrix $\mathbf{x}_2 \mathbf{x}_2^T + C \mathbf{U}_1 \mathbf{U}_1^T$
Termination Rules

We define the following three termination rules:

- Maximum limit $Z$ on the number of iterations,
- Relative change in $U_1$ and $U_2$ at iteration $m$ and $m+1$,

\[ \text{tol}_U^m = \frac{\| U_1^{(m+1)} - U_1^{(m)} \|_F}{\sqrt{q}} \quad \text{tol}_U^m = \frac{\| U_2^{(m+1)} - U_2^{(m)} \|_F}{\sqrt{q}} \]  

where $q = pk$

- Relative change in objective function value of (14) at iteration $m$ and $m+1$,

\[ \text{tol}_f^m = \frac{F^{(m+1)} - F^{(m)}}{|F^{(m)}| + 1} \]  

(18)
CSC Algorithm

The algorithm for CSC can be summarized as follows:

**Algorithm 1** CSC ($\mathcal{X}_1$, $\mathcal{X}_2$, $k$, $C$)

1. Initialize $\mathbf{U}_1$ and $\mathbf{U}_2$ such that $\mathbf{U}_1^T \mathbf{U}_1 = I_k$, $\mathbf{U}_2^T \mathbf{U}_2 = I_k$
2. Find eigenvectors corresponding to the $k$ largest eigenvalues of symmetric matrix $\mathcal{X}_1 \mathcal{X}_1^T + C \mathbf{U}_2 \mathbf{U}_2^T$
3. Find eigenvectors corresponding to the $k$ largest eigenvalues of symmetric matrix $\mathcal{X}_2 \mathcal{X}_2^T + C \mathbf{U}_1 \mathbf{U}_1^T$
4. Alternate between 2 and 3 until one of the termination rules is satisfied

Algorithm 1 converges. For proof of convergence see:

The performance of CSC is evaluated on four high dimensional publicly available datasets.

CSC is also tested on two lower dimensional datasets.

The performance of CSC is evaluated for different values of $C$, and compared to that of LSC.

The values of $C$ are chosen in such a way that they vary uniformly.

The classification performance is evaluated using leave-one-out cross validation (LOOCV) technique.

The value of $k$ is chosen as $\{1, 3, 10\}$.

Experiments are performed with a 2.60GHz Intel Core i5 CPU running OS X with 8.0 GB of main memory.
Diffuse large B-cell lymphoma DLBCL, the most common lymphoid malignancy in adults, is curable in less than 50% of patients. The DLBCL dataset consists of 77 samples with 5469 features. CSC was used to identify cured versus fatal or refractory disease.
Breast Cancer

Breast Cancer dataset consists of 77 samples of breast tumors. 4869 features describe each one of those tumors. CSC classified the tumors as recurring or non-recurring.
Colon

40 tumor and 22 normal colon tissue samples make up Colon dataset. 2000 features describe each one of those samples. CSC classified the samples as tumorous or not
DBWorld dataset consists of 64 e-mails (samples) divided in two classes. The first one consists of only subject lines, while the second consists of only bodies. 4702 features describe each one of those samples. CSC classified the samples as subjects or bodies.

<table>
<thead>
<tr>
<th>Classification Accuracy (%)</th>
<th>K=1</th>
<th>K=3</th>
<th>K=10</th>
</tr>
</thead>
<tbody>
<tr>
<td>60.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>68.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>72.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>76.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>84.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>88.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>92.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>96.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100.00</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Classification Accuracy Graph](#)
Mushroom dataset describes characteristics of gilled mushrooms. It consists of 8124 samples with 126 features. CSC classified the samples onto two categories, edible and non-edible.
Spambase dataset consists of 4601 samples (emails) with 57 features. CSC was used to find whether an email is spam or not.
For DLBCL and Colon datasets, classification accuracy is improved by reducing the relative angle between subspaces for $k = 3$, $k = 10$ and $k = 1$, $k = 3$ respectively.

In the case of Breast dataset, increasing the relative angle for $k = 1$ considerably improves the classification accuracy.

The classification accuracy of CSC was almost identical to that of LSC for the DBWorld dataset.

With respect to the lower dimensional datasets, CSC performed at least as good as LSC.

In the case of Spambase dataset, CSC was able to slightly increase the accuracy of classification for positive values of $C$. 
## Comparative computational results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SVM</th>
<th>PCA/SVM</th>
<th>Naive Bayes</th>
<th>CSC</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLBCL</td>
<td>94.8</td>
<td>97.5</td>
<td>75</td>
<td>97.4</td>
</tr>
<tr>
<td>Breast</td>
<td>68</td>
<td>68</td>
<td>62.5</td>
<td>63.6</td>
</tr>
<tr>
<td>Colon</td>
<td>75.9</td>
<td>92.1</td>
<td>71.4</td>
<td>90.3</td>
</tr>
<tr>
<td>DBWorld</td>
<td>88</td>
<td>88</td>
<td>57.1</td>
<td>89</td>
</tr>
<tr>
<td>Mushroom</td>
<td>100</td>
<td>100</td>
<td>88.1</td>
<td>98.9</td>
</tr>
<tr>
<td>Spambase</td>
<td>91</td>
<td>66</td>
<td>56.3</td>
<td>87.9</td>
</tr>
</tbody>
</table>

- CSC demonstrates competitive behavior with respect to dataset dimensionality
- CSC remains robust
Conclusion

- A new classification algorithm, CSC, was proposed and designed for high dimensional datasets
- CSC *improves* upon local subspace classifier
- The *improvement in classification accuracy* shows the importance of considering the relative angle between subspaces while approximating the classes
- The robust nature of CSC reveals that it can serve as a *one-step method* for preprocessing-free classification
Future Research

- A *cost sensitive* version for *imbalanced classification* problems
- A stream mining version that will incrementally retrain as new training data samples arrive in the form of a data stream
- A robust optimization version for handling datasets that are inexact or uncertain
Panagopoulos, O. P., Pappu, V., Xanthopoulos, P., Pardalos, P. M.  
*Constrained subspace classifier for high dimensional datasets* - Omega  
(Volume 59, Part A, March 2016, Pages 40–46)  
THANK YOU!


Köppen, M., The curse of dimensionality, 5th Online World Conference on Soft Computing in Industrial Applications (WSC5), 4–8, 2000


Laaksonen, Jorma, Local subspace classifier, Artificial Neural Networks ICANN’97, 637–642, 1997

Golub, Gene H and Van Loan, Charles F, Matrix computations, 2012

Shipp, Margaret A and Ross, Ken N and Tamayo, Pablo and Weng, Andrew P and Kutok, Jeffery L and Aguiar, Ricardo CT and Gaasenbeek, Michelle and Angelo, Michael and Reich, Michael and Pinkus, Geraldine S and others, Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning, 68–74, 2002

Petros Xanthopoulos, Panos M. Pardalos, Theodore B. Trafalis, Robust Data Mining, 2010

Zhang, L. and Lin, X., Some considerations of classification for high dimension low-sample size data, Statistical Methods in Medical Research, 2011


Fenn, M.B. and Pappu, V., Data Mining for Cancer Biomarkers with Raman Spectroscopy, Data Mining for Biomarker Discovery, pages 143–168, 2012


Lee, Sang Min, Dong Seong Kim, Ji Ho Kim, and Jong Sou Park, Spam detection using feature selection and parameters optimization, Intelligent and Software Intensive Systems (CISIS), 883–888, 2010
References

Jiang, Sheng-Yi, and Xia Li, A hybrid clustering algorithm, Fuzzy Systems and Knowledge Discovery, 2009


Yu, Lei, and Huan Liu, Feature selection for high-dimensional data: A fast correlation-based filter solution, 856–863, 2003


