Case Studies: Nonlinear Optimization

Fin, Stan, Stefan, Jakob, Roland

29. April 2015

Inhaltsverzeichnis

Notation
Math environment in aligned mode:

$$\begin{aligned} f(x) &= x \\ &= e^{\log(x)} \end{aligned}$$

The numbered versions of this is

$$\begin{aligned} f(x) &= x \\ &= e^{\log(x)} \end{aligned} \tag{0.1}$$

Citations work this way: (0.1)

1 Optimal Feature Transform

We want to consider feature transforms Φ which operate on the digital signal f. The resulting feature vector $c = \Phi f$ is later used for classification.
One of the aims in this section is to find an optimal feature transform. This can be a very challenging task, however. Therefore we first consider feature transforms which reduce the dimension of the feature space by projection on one or more axes, which is also called Linear Discriminant Analysis.

1.1 Model

Suppose you are given a set of K vectors $X_i \in \mathrm{R}^n$, $i \in \{1, \dots K\}$ with zero mean $\bar{X} = \frac{1}{K}\sum_{i=1}^{K} X_i = 0$. The goal in PCA is to find a orthogonal normalized linear transformation $\Phi : \mathrm{R}^n \to \mathrm{R}^m$, $m \leq n$, such that the spread of the features ΦX_i is maximized.

$$\max_{\Phi \in \mathrm{R}^{m \times n}} \sum_{i=1}^{K} \|\Phi X_i\|_2^2 \quad s.t. \, \|\phi_k\|_2^2 \leq 1 \;\; (k = 1, \dots, m)$$

where the $\phi_k \in \mathrm{R}^n$ are the columns of $\Phi = [\phi_1, \phi_2, \dots \phi_m]$. Using the Lagrange Formulation of this optimization problem we get

$$\max_{\Phi \in \mathrm{R}^{m \times n}} \left\{ \sum_{i=1}^{K} \|\Phi X_i\|_2^2 - \sum_{k=1}^{m} \lambda_k (\|\phi_k\|_2^2 - 1) \right\} \tag{1.1}$$

for $\lambda_k \in \mathrm{R}$. The first part represents the spread of the transformed features, the second part poses a boundary condition on the length of the matrix vectors.

1.2 Reformulation

We can rearrange the first part in the objective function (**??**):

$$\begin{aligned} \sum_{i=1}^{K} \|\Phi X_i\|^2 = \sum_{i=1}^{K} X_i^T \Phi^T \Phi X_i &= \sum_{i=1}^{K} X_i^T \left(\sum_{k=1}^{m} \phi_k \phi_k^T \right) X_i \\ &= \sum_{k=1}^{m} \phi_k^T \left(\sum_{i=1}^{K} X_i X_i^T \right) \phi_k =: \sum_{k=1}^{m} \phi_k^T \Sigma \phi_k \end{aligned}$$

with $\Sigma = \sum_{i=1}^{K} X_i X_i^T$, which is K-times the covariance matrix. At the maximum point we have that the first derivitive $D\phi$ of Φ is zero, and in particular for all $k = 1, \dots M$ the gradients the columns of $D\Phi$ are zero:

$$\begin{aligned} &\nabla \phi_k^T \Sigma \phi_k - \nabla \lambda_k (\phi_k^T \phi_k - 1) = 0 \\ &\Leftrightarrow 2\Sigma\phi_k - 2\lambda_k \phi_k = 0 \\ &\Leftrightarrow \Sigma\phi_k = \lambda_k \phi_k \quad (k = 1, \dots m) \end{aligned}$$

i.e. The solution to the PCA problem is the solution of the eigenvalue problem $\Sigma\phi_k = \lambda_k \phi_k$.

1.3 Solution of the eigenvalue problem

In order to solve the eigenvalue problem, we make use of the Principal Axis Theorem. Since the matrix Σ is symmetric and real, there exists a orthogonal matrix $U \in \mathrm{R}^{n \times n}$ such that

$\Sigma = UDU^T$, with diagnonal matrix D containing the eigenvalues of Σ.

Moreover, the orthogonal normal matrix $U^{-1} = U^T$ transformes any vector $c \in \mathrm{R}^N$ into the basis of the eigenvectors of Σ.
We can use the Singular Value Decomposition (SVD) in order to compute the Hauptachsentransfromation of Σ.
, we get a basis of eigenvectors such that the eigenvalues are a set of ordered non-negative real numbers.

$$\begin{aligned} \Sigma = UDU^T &= \left(UD^{\frac{1}{2}}\right)\left(UD^{\frac{1}{2}}\right)^T, \text{ and} \\ \Sigma^{-1} = \left(UD^{\frac{1}{2}}\right)^{-T}\left(UD^{\frac{1}{2}}\right)^{-1} &= \left(D^{-\frac{1}{2}}U^T\right)^T\left(D^{-\frac{1}{2}}U^T\right) \end{aligned}$$

In other words: By multiplying each sample $\{X_i \in \mathrm{R}^N\}_{i=1}^M$ with the matrix $B := D^{-\frac{1}{2}}U^T$, we get a normalized set of vectors with zero mean and uniform covariance.

1.4 Solution to the PCA problem

Using the steps above we have now found a soulution to (**??**) above in the special case where m equals n. If we want to have $m < n$, according to the eigenvalue problem, we have to limit the number of eigenvectors we want to use. For this we order the eigenvalues and the column vectors of the matrix U in descending order of the diagnonal entries of D.
Then, we apply the transform $B = D^{-\frac{1}{2}}U^T$ to all the features X_i and we only keep the top m entries of the transformed vectors BX_i.

1.5 Classification

For classification using PCA transformed feature vectors, it is necessary to transform the whole Dataset into the new basis of eigenvectors. Features which we want to classify will be transformed by subtracting the mean $\bar{X}$ from them, applying the transformation matrix B and discard all entries with index larger than m.

1.6 Algorithm

1. Normalize the data to zero mean.
2. Compute the covariance matrix Σ of $\{X_i\}_{i=1}^{K}$.
3. Determine the eigenvalues and eigenvectors of Σ using SVD
4. Order the eigenvalues and eigenvectors in descending order.
5. Save the transformation matrix $B = D^{-\frac{1}{2}}U^T$.
6. For classification subtract $\bar{X}$ and transform using B.
7. Use only the top m entries of the transformed vectors.
8. Use any classification algorithm for learning.

A Appendix