ISML-II: Machine Learning **Spring 2014**

Lecture 19- Dimensionality Reduction

Lecturer: Lorenzo Rosasco

In many practical applications it is of interest to reduce the dimensionality of the data. In particular, this is useful for data visualization, or for investigating the "effective" dimensionality of the data. This problem is often referred to as dimensionality reduction and can be seen as the problem of defining a map

$$M : X = \mathbb{R}^D \to \mathbb{R}^k, \quad k \ll D,$$

according to some suitable criterion.

19.1 PCA & Reconstruction

PCA is arguably the most popular dimensionality reduction procedure. It is a data driven procedure that given an (unsupervised) sample $S = (x_1, \ldots, x_n)$ derive a dimensionality reduction defined by a linear map M. PCA can be derived from several prospective and here we give a geometric/analytical derivation.

We begin by considering the case where $k = 1$. We are interested into finding the single most relevant dimension according to some suitable criterion. Recall that, if $w \in \mathbb{R}^D$ with $\|w\| = 1$, then the (orthogonal) projection of a point x on w is given by $(w^T x)w$. Consider the problem of finding the direction p which allows the best possible average reconstruction of the training set, that is the solution of the problem

$$\min_{w \in \mathbb{S}^{D-1}} \frac{1}{n} \sum_{i=1}^{n} \|x_i - (w^T x_i)w\|^2, \tag{19.1}$$

where $\mathbb{S}^{D-1} = \{w \in \mathbb{R}^D \mid \|w\| = 1\}$ is the sphere in D dimensions. The norm $\|x_i - (w^T x_i)w\|^2$ measures how much we lose by projecting x along the direction w, and the solution p to problem (19.1) is called the first principal component of the data. A direct computation shows that $\|x_i - (w^T x_i)w\|^2 = \|x_i\| - (w^T x_i)^2$, so that problem (19.1) is equivalent to

$$\max_{w \in \mathbb{S}^{D-1}} \frac{1}{n} \sum_{i=1}^{n} (w^T x_i)^2. \tag{19.2}$$

This latter observation is useful for two different reasons that the we discuss in the following.

19.2 PCA and Maximum Variance

If the data are centered, that is $\bar{x} = \frac{1}{n} x_i = 0$, problem (19.2) has the following interpretation: we a look for the direction along which the data have (on average) maximum variance.

Indeed, we can interpret the term $(w^T x)^2$ has the variance of x in the direction w. If the data are not centered, to keep this interpretation we should replace problem (19.2) with

$$\max_{w \in \mathbb{S}^{D-1}} \frac{1}{n} \sum_{i=1}^{n} (w^T (x_i - \bar{x}))^2, \tag{19.3}$$

which corresponds to the original problem on the centered data $x^c = x - \bar{x}$. In the terms of problem (19.1) it is easy to see that this corresponds to considering

$$\min_{w,b \in \mathbb{S}^{D-1}} \frac{1}{n} \sum_{i=1}^{n} \|x_i - ((w^T (x_i - b))w + b)\|^2. \tag{19.4}$$

where $((w^T (x_i - b))w + b$ is an affine transformation (rather than an orthogonal projection).

19.3 PCA and Associated Eigenproblem

A simple further manipulation allows to write problem (19.2) as an eigenvalue problem. Indeed, using the symmetry of the inner product we have

$$\frac{1}{n} \sum_{i=1}^{n} (w^T x_i)^2 = \frac{1}{n} \sum_{i=1}^{n} w^T x_i w^T x_i = \frac{1}{n} \sum_{i=1}^{n} w^T x_i x_i^T w = w^T (\frac{1}{n} \sum_{i=1}^{n} x_i x_i^T) w$$

so that problem (19.2) can be written as

$$\max_{w \in \mathbb{S}^{D-1}} w^T C_n w, \quad C_n = \frac{1}{n} \sum_{i=1}^{n} x_i x_i^T. \tag{19.5}$$

We need two observations. First, in matrix notation $C_n = \frac{1}{n} \sum_{i=1}^{n} X_n^T X_n$ and it is easy to see that C_n is symmetric and positive semi-definite. If the data are centered the matrix C_n is the so called covariance matrix. Clearly the objective function in (19.5) can be written as

$$\frac{w^T C_n w}{w^T w}$$

where the latter quantity is the so called Rayleigh quotient. Note that, if $C_n u = \lambda u$ then $\frac{u^T C_n u}{u^T u} = \lambda$, since the eigenvector u normalized. In fact, it is possible to show that the Rayleigh quotient achieves its maximum at a vector which corresponds to the maximum eigenvalue of C_n (the proof of this latter fact uses basic results in linear programming). Then computing the first principal component of the data reduced to computing the biggest eigenvalue of the covariance and the corresponding eigenvector.

19.4 Beyond the First Principal Component

Next, we discuss how the above reasoning can be generalized to $k > 1$, that is more than one principle component. The idea is simply to iterate the above reasoning to describe the

input data beyond what is allowed by the first principal component. Towards this end, we consider the one dimensional projection which can best reconstruct the residuals

$$r_i = x_i - (p^T x_i)p_i,$$

that is we replace problem (19.1) by

$$\min_{w \in \mathbb{S}^{D-1}, w \perp p} \frac{1}{n} \sum_{i=1}^{n} \|r_i - (w^T r_i)w\|^2. \tag{19.6}$$

Note that for all $i = 1, \ldots, n$,

$$\|r_i - (w^T r_i)w\|^2 = \|r_i\|^2 - (w^T r_i)^2 = \|r_i\|^2 - (w^T x_i)^2$$

since $w \perp p$. Then, following the reasoning from (19.1) to (19.2), problem (19.6) can equivalently be written as

$$\max_{w \in \mathbb{S}^{D-1}, w \perp p} \frac{1}{n} \sum_{i=1}^{n} (w^T x_i)^2 = w^T C_n w. \tag{19.7}$$

Again, we have to minimize the Rayleigh quotient of the covariance matrix, however, when compared to (19.2), we see that there is the new constraint $w \perp p$. Indeed, it can be proven that the solution of problem (19.7) is given by the second eigenvector of C_n, and the corresponding eigenvalue. The proof of this latter fact follows the same line of the one for the first principal component. Clearly, the above reasoning can be generalized to consider more than two components. The computation of the principal components reduces to the problem of finding the eigenvalues and eigenvectors of C_n. The complexity of this problem is roughly $O(kD^2)$ (note that the complexity of forming C_n is $O(nD^2)$).

The principal components can be stacked as columns of a k by D matrix M, and in fact, because of the orthogonality constraint, the matrix M is orthogonal, $MM^T = I$. The dimensionality reduction induced by PCA is hence linear.

19.5 Singular Value Decomposition

We recall the notion of singular valued decomposition of a matrix which allows in some situations to improve the computations of the principal components, while suggesting a possible way to generalize the algorithm to consider non linear dimensionality reduction.

Consider the data matrix X_n, its singular value decomposition is given by

$$X_n = U \Sigma P^T.$$

where U is a n by d orthogonal matrix, P is a D by d orthogonal matrix, Σ is a diagonal matrix such that $\Sigma_{i,i} = \sqrt{\lambda_i}$, $i = 1, \ldots, d$ and $d \leq \min\{n, D\}$. The columns of U and the columns of V are called respectively the left and right singular vectors and the diagonal entries of Σ the singular values. The singular value decomposition can be equivalently described by the following equations, for $j = 1, \ldots, d$,

$$\begin{aligned} C_n p_j &= \lambda_j p_j, \quad \frac{1}{n} K_n u_j = \lambda_j u_j, \\ X_n p_j &= \sqrt{\lambda_j} u_j, \quad \frac{1}{n} X_n^T u_j = \sqrt{\lambda_j} p_j, \end{aligned} \tag{19.8}$$

where $C_n = \frac{1}{n} X_n^T X_n$ and $\frac{1}{n} K_n = \frac{1}{n} X_n X_n^T$.

If $n \ll p$ the above equations can be used to speed up the computation of the principal components. Indeed we can consider the following procedure:

- form the matrix K_n, which is $O(Dn^2)$,
- find the first k eigenvectors of K_n, which is $O(kn^2)$,
- find the principal components using (19.8), i.e.

$$p_j = \frac{1}{\sqrt{\lambda_j}} X_n^T u_j = \frac{1}{\sqrt{\lambda_j}} \sum_{i=1}^{n} x_i u_j^i, \quad j = 1, \dots, d \tag{19.9}$$

  where $u = (u^1, \dots, u^n)$, which is again $O(knd)$ if we consider k principal components.

19.6 Kernel PCA

The latter reasoning suggests how to generalize the intuition behind PCA beyond linear dimensionality reduction by using kernels (or feature maps). Indeed, from equation (19.9) we can see that the projection of a point x on a principal component p can be written as

$$(M(x))^j = x^T p_j = \frac{1}{\sqrt{\lambda_j}} x^T X_n^T u_j = \frac{1}{\sqrt{\lambda_j}} \sum_{i=1}^{n} x^T x_i u_j^i, \tag{19.10}$$

for $j = 1, \dots, d$.

What if we were to map the data using a possibly non linear feature map $\Phi : X \to F$, before performing PCA? If the feature map is finite dimensional, e.g. $F = \mathbb{R}^p$ we could simply replace $x \mapsto \Phi(x)$ and follow exactly reasoning in the previous sections. Note that in particular that equation (19.10) becomes

$$(M(x))^j = \Phi(x)^T p_j = \frac{1}{\sqrt{\lambda_j}} \sum_{i=1}^{n} \Phi(x)^T \Phi(x_i) u_j^i, \tag{19.11}$$

for $j = 1, \dots, d$. More generally one could consider a positive definite kernel $K : X \times X \to \mathbb{R}$, in which case (19.10) becomes

$$(M(x))^j = \frac{1}{\sqrt{\lambda_j}} \sum_{i=1}^{n} K(x, x_i) u_j^i, \tag{19.12}$$

for $j = 1, \dots, d$. Note that in this latter case, while it is not clear how to form C_n, we can still form and diagonalize K_n, which is in fact the kernel matrix.