E0 270 Machine Learning **Lecture 10 (Feb 7, 2013)**

Canonical Correlation Analysis

Lecturer: Chiranjib Bhattacharyya *Scribe: Debarghya Ghoshdastidar*

1 Introduction

Principal component analysis (PCA) tries to find the orthogonal directions along which spread of a random vector X is maximum, *i.e.*, we try to find a representation of X (say Y) such that the variances of the components of Y are in decreasing order. As seen in the reconstruction example, if we had to represent each data with a scalar, then PCA returns a unit vector w such that the projection each data along w leads to minimum reconstruction error.

Now, consider the following problem. There are two correlated random vectors, X and $\widetilde{X}$ (which may lie in different spaces). We want to represent each instance of $(X, \widetilde{X})$ by a pair of scalars $(Y, \widetilde{Y})$ so that these random scalars preserve the correlation between them as much as possible. This problem may be solved in a way similar to PCA, as discussed below.

2 The formulation

Suppose $X \in \mathbb{R}^{d_1}$ and $\widetilde{X} \in \mathbb{R}^{d_2}$. Let $u \in \mathbb{R}^{d_1}$ and $v \in \mathbb{R}^{d_2}$ be fixed vectors, and consider the scalars to be of the form $Y = u^T X$ and $\widetilde{Y} = v^T \widetilde{X}$, *i.e.*, they are projections of X and $\widetilde{X}$ onto u and v, respectively. In this case, goal may reformulated as determining the directions u and v such that the correlation coefficient of Y and $\widetilde{Y}$ is maximized, *i.e.*,

$$\begin{aligned} &\max_{u,v} \rho_{Y\widetilde{Y}} \\ \text{or,} \quad &\max_{u,v} \frac{\mathrm{Cov}(Y,\widetilde{Y})}{\sqrt{\mathrm{Var}(Y)}\sqrt{\mathrm{Var}(\widetilde{Y})}} \end{aligned} \tag{1}$$

Let $\mathbb{E}[X] = \mu$, $\mathbb{E}[\widetilde{X}] = \widetilde{\mu}$, $\mathrm{Var}(X) = \Sigma_{XX}$, $\mathrm{Var}(\widetilde{X}) = \Sigma_{\widetilde{X}\widetilde{X}}$ and $\mathrm{Cov}(X, \widetilde{X}) = \Sigma_{X\widetilde{X}}$. Computing each of the terms in (1),

$$\begin{aligned} \mathrm{Cov}(Y,\widetilde{Y}) &= \mathbb{E}\left[(Y - \mathbb{E}[Y])(\widetilde{Y} - \mathbb{E}[\widetilde{Y}])\right] \\ &= \mathbb{E}\left[(u^T X - \mathbb{E}[u^T X])(v^T \widetilde{X} - \mathbb{E}[v^T \widetilde{X}])\right] \\ &= \mathbb{E}\left[(u^T X - u^T\mu)(v^T \widetilde{X} - v^T\widetilde{\mu})\right] \\ &= \mathbb{E}\left[u^T (X - \mu)(\widetilde{X} - \widetilde{\mu})^T v\right] \\ &= u^T \mathbb{E}\left[(X - \mu)(\widetilde{X} - \widetilde{\mu})^T\right] v = u^T \Sigma_{X\widetilde{X}} v\ . \end{aligned}$$

Similarly, we can compute $\mathrm{Var}(Y) = u^T \Sigma_{XX} u$ and $\mathrm{Var}(\widetilde{Y}) = v^T \Sigma_{\widetilde{X}\widetilde{X}} v$. Plugging these terms back in (1) gives

$$\max_{u,v} \frac{u^T \Sigma_{X\widetilde{X}} v}{\sqrt{u^T \Sigma_{XX} u}\sqrt{v^T \Sigma_{\widetilde{X}\widetilde{X}} v}}$$

Let us define $\rho(u,v) = \dfrac{u^T \Sigma_{X\widetilde{X}} v}{\sqrt{u^T \Sigma_{XX} u}\sqrt{v^T \Sigma_{\widetilde{X}\widetilde{X}} v}}$. Now, it can be observed that if s_1, s_2 be any two positive scalars, then $\rho(s_1 u, s_2 v) = \rho(u,v)$. Hence, the optimization problem is not affected by multiplication with positive scalars, *i.e.*, it does not change under scaling, and hence, has infinitely many solutions. To fix the scaling we further constrain the problem by introducing constraints

$$u^T \Sigma_{XX} u = v^T \Sigma_{\widetilde{X}\widetilde{X}} v = 1.$$

This leads to a constrained optimization of the following form.

$$\begin{aligned} \max_{u,v}\ & u^T \Sigma_{X\widetilde{X}} v \\ \text{s.t. } & u^T \Sigma_{XX} u = 1 \\ & v^T \Sigma_{\widetilde{X}\widetilde{X}} v = 1 \end{aligned} \tag{2}$$

3 Generalized Eigenvalue problem

The Lagrangian of the above problem can be written as

$$L(u,v,\lambda_1,\lambda_2) = u^T \Sigma_{X\widetilde{X}} v - \lambda_1 \left(u^T \Sigma_{XX} u - 1\right) - \lambda_2 \left(v^T \Sigma_{\widetilde{X}\widetilde{X}} v - 1\right). \tag{3}$$

By the KKT conditions if $(u^*, v^*, \lambda_1^*, \lambda_2^*)$ be a maxima of (3), then we have

$$\left.\frac{\partial L}{\partial u}\right|_{(u^*,v^*,\lambda_1^*,\lambda_2^*)} = \Sigma_{X\widetilde{X}} v^* - \lambda_1^* \Sigma_{XX} u^* = 0 \tag{4}$$

$$\text{and} \quad \left.\frac{\partial L}{\partial v}\right|_{(u^*,v^*,\lambda_1^*,\lambda_2^*)} = \Sigma_{X\widetilde{X}}^T u^* - \lambda_2^* \Sigma_{\widetilde{X}\widetilde{X}} v^* = 0 \tag{5}$$

From (4), we have

$$\begin{aligned} \Sigma_{X\widetilde{X}} v^* &= \lambda_1^* \Sigma_{XX} u^* \\ \text{or, } {u^*}^T \Sigma_{X\widetilde{X}} v^* &= \lambda_1^* {u^*}^T \Sigma_{XX} u^* = \lambda_1^*, \end{aligned}$$

since ${u^*}^T \Sigma_{XX} u^* = 1$. Similarly, (5) leads to

$${v^*}^T \Sigma_{X\widetilde{X}}^T u^* = \lambda_2^* {v^*}^T \Sigma_{\widetilde{X}\widetilde{X}} v^* = \lambda_2^*.$$

Thus, we have the maximum value of the objective function $= {u^*}^T \Sigma_{X\widetilde{X}} v^* = \lambda_1^* = \lambda_2^* = \lambda$ (say). Then (4) and (5) can be together written as

$$\begin{pmatrix} 0 & \Sigma_{X\widetilde{X}} \\ \Sigma_{X\widetilde{X}}^T & 0 \end{pmatrix} \begin{pmatrix} u^* \\ v^* \end{pmatrix} = \lambda \begin{pmatrix} \Sigma_{X\widetilde{X}} & 0 \\ 0 & \Sigma_{\widetilde{X}\widetilde{X}}^T \end{pmatrix} \begin{pmatrix} u^* \\ v^* \end{pmatrix}. \tag{6}$$

Such a set of linear equations is commonly termed as a generalized eigenvalue problem, which has the generic form

$$Ax = \lambda Bx\ . \tag{7}$$

The solutions (λ, x) of (7) are called generalized eigenvalues and generalized eigenvectors, respectively. It is easy to see that (7) simplifies to an eigenvalue problem when B is the identity matrix. Such a problem can be solved using the MATLAB command `eig(A,B)`, which returns a set of generalized eigenvalues and corresponding eigenvectors.

But considering the optimization problem in (2), we need to choose the solution corresponding to the maximum generalized eigenvalue, which will be same as λ^*. The corresponding generalized eigenvector, when scaled appropriately, corresponds to the optimal vectors (u^*, v^*).

4 Going further

In the previous discussion we found a pair (u, v) which are the cannonical components. As in the PCA case we can generalize it to k pairs. Again it is easy to see how CCA can be *kernelized*, by following the steps in *kernel* PCA.