Kernel Canonical Correlation Analysis

Max Welling
Department of Computer Science
University of Toronto
10 King's College Road
Toronto, M5S 3G5 Canada
welling@cs.toronto.edu

Abstract

This is a note to explain kCCA.

1 Canonical Correlation Analysis

Imagine you are given 2 copies of a corpus of documents, one written in English, the other written in German. You may consider an arbitrary representation of the documents, but for definiteness we will use the "vector space" representation where there is an entry for every possible word in the vocabulary and a document is represented by count values for every word, i.e. if the word "the appeared 12 times and the first word in the vocabulary we have $X_1(doc) = 12$ etc.

Let's say we are interested in extracting low dimensional representations for each document. If we had only one language, we could consider running PCA to extract directions in word space that carry most of the variance. This has the ability to infer semantic relations between the words such as synonymy, because if words tend to co-occur often in documents, i.e. they are highly correlated, they tend to be combined into a single dimension in the new space. These spaces can often be interpreted as topic spaces.

If we have two translations, we can try to find projections of each representation separately such that the projections are maximally correlated. Hopefully, this implies that they represent the same topic in two different languages. In this way we can extract language independent topics.

Let $\mathbf{x}$ be a document in English and $\mathbf{y}$ a document in German. Consider the projections: $u = \mathbf{a}^T\mathbf{x}$ and $v = \mathbf{b}^T\mathbf{y}$. Also assume that the data have zero mean. We now consider the following objective,

$$\rho = \frac{\mathbf{E}[uv]}{\sqrt{\mathbf{E}[u^2]\mathbf{E}[v^2]}} \tag{1}$$

We want to maximize this objective, because this would maximize the correlation between the univariates u and v. Note that we divided by the standard deviation of the projections to remove scale dependence.

This exposition is very similar to the Fisher discriminant analysis story and I encourage you to reread that. For instance, there you can find how to generalize to cases where the data is not centered. We also introduced the following "trick". Since we can rescale $\mathbf{a}$ and

$\mathbf{b}$ without changing the problem, we can constrain them to be equal to 1. This then allows us to write the problem as,

$$\begin{aligned} \text{maximize}_{\mathbf{a},\mathbf{b}} \quad & \rho = \mathbf{E}[uv] \\ \text{subject to} \quad & \mathbf{E}[u^2] = 1 \\ & \mathbf{E}[v^2] = 1 \end{aligned} \tag{2}$$

Or, if we construct a Lagrangian and write out the expectations we find,

$$\min_{\mathbf{a},\mathbf{b}} \max_{\lambda_1,\lambda_2} \sum_i \mathbf{a}^T \mathbf{x}_i \mathbf{y}_i^T \mathbf{b} - \frac{1}{2}\lambda_1 (\sum_i \mathbf{a}^T \mathbf{x}_i \mathbf{x}_i^T \mathbf{a} - N) - \frac{1}{2}\lambda_2 (\sum_i \mathbf{b}^T \mathbf{y}_i \mathbf{y}_i^T \mathbf{b} - N) \tag{3}$$

where we have multiplied by N. Let's take derivatives wrt to $\mathbf{a}$ and $\mathbf{b}$ to see what the KKT equations tell us,

$$\sum_i \mathbf{x}_i \mathbf{y}_i^T \mathbf{b} - \lambda_1 \sum_i \mathbf{x}_i \mathbf{x}_i^T \mathbf{a} = 0 \tag{4}$$

$$\sum_i \mathbf{y}_i \mathbf{x}_i^T \mathbf{a} - \lambda_2 \sum_i \mathbf{y}_i \mathbf{y}_i^T \mathbf{b} = 0 \tag{5}$$

First notice that if we multiply the first equation with $\mathbf{a}^T$ and the second with $\mathbf{b}^T$ and subtract the two, while using the constraints, we arrive at $\lambda_1 = \lambda_2 = \lambda$. Next, rename $S_{xy} = \sum_i \mathbf{x}_i \mathbf{y}_i^T$, $S_x = \sum_i \mathbf{x}_i \mathbf{x}_i^T$ and $S_y = \sum_i \mathbf{y}_i \mathbf{y}_i^T$. We define the following larger matrices: S_D is the block diagonal matrix with S_x and S_y on the diagonal and zeros on the off-diagonal blocks. Also, we define S_O to be the off-diagonal matrix with S_{xy} on the off diagonal. Finally we define $\mathbf{c} = [\mathbf{a}, \mathbf{b}]$. The two equations can then we written jointly as,

$$S_O \mathbf{c} = \lambda S_D \mathbf{c} \;\; \Rightarrow \;\; S_D^{-1} S_O \mathbf{c} = \lambda \mathbf{c} \;\; \Rightarrow \;\; S_O^{\frac{1}{2}} S_D^{-1} S_O^{\frac{1}{2}} (S_O^{\frac{1}{2}} \mathbf{c}) = \lambda (S_O^{\frac{1}{2}} \mathbf{c}) \tag{6}$$

which is again an regular eigenvalue equation for $\mathbf{c}' = S_O^{\frac{1}{2}} \mathbf{c}$

2 Kernel CCA

As usual, the starting point to map the data-cases to feature vectors $\Phi(\mathbf{x}_i)$ and $\Psi(\mathbf{y}_i)$.When the dimensionality of the space is larger than the number of data-cases in the training-set, then the solution must lie in the span of data-cases, i.e.

$$\mathbf{a} = \sum_i \alpha_i \Phi(\mathbf{x}_i) \quad \mathbf{b} = \sum_i \beta_i \Psi(\mathbf{y}_i) \tag{7}$$

Using this equation in the Lagrangian we get,

$$\mathcal{L} = \boldsymbol{\alpha}^T K_x K_y \boldsymbol{\beta} - \frac{1}{2}\lambda (\boldsymbol{\alpha}^T K_x^2 \boldsymbol{\alpha} - N) - \frac{1}{2}\lambda (\boldsymbol{\beta}^T K_y^2 \boldsymbol{\beta} - N) \tag{8}$$

where $\boldsymbol{\alpha}$ is a vector in a different N-dimensional space than e.g. $\mathbf{a}$ which lives in a D-dimensional space, and $K_x = \sum_i \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_i)$ and similarly for K_y.

Taking derivatives w.r.t. $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ we find,

$$K_x K_y \boldsymbol{\beta} = \lambda K_x^2 \boldsymbol{\alpha} \tag{9}$$

$$K_y K_x \boldsymbol{\alpha} = \lambda K_y^2 \boldsymbol{\beta} \tag{10}$$

Let's try to solve these equations by assuming that K_x is full rank (which is typically the case). We get, $\boldsymbol{\alpha} = \lambda^{-1} K_x^{-1} K_y \boldsymbol{\beta}$ and hence, $K_y^2 \boldsymbol{\beta} = \lambda^2 K_y^2 \boldsymbol{\beta}$ which always has a solution for $\lambda = 1$. By recalling that,

$$\rho = \frac{1}{N} \sum_i \mathbf{a}^T S_{xy} \mathbf{b} = \frac{1}{N} \sum_i \lambda \mathbf{a}^T S_x \mathbf{a} = \lambda \tag{11}$$

we observe that this represents the solution with maximal correlation and hence the preferred one. This is a typical case of over-fitting emphasizes again the need to regularize in kernel methods. This can be done by adding a diagonal term to the constraints in the Lagrangian (or equivalently to the denominator of the original objective), leading to the Lagrangian,

$$\mathcal{L} = \boldsymbol{\alpha}^T K_x K_y \boldsymbol{\beta} - \frac{1}{2}\lambda(\boldsymbol{\alpha}^T K_x^2 \boldsymbol{\alpha} + \eta||\boldsymbol{\alpha}||^2 - N) - \frac{1}{2}\lambda(\boldsymbol{\beta}^T K_y^2 \boldsymbol{\beta} + \eta||\boldsymbol{\beta}||^2 - N) \quad (12)$$

One can see that this acts as a quadratic penalty on the norm of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. The resulting equations are,

$$K_x K_y \boldsymbol{\beta} = \lambda(K_x^2 + \eta I)\boldsymbol{\alpha} \quad (13)$$

$$K_y K_x \boldsymbol{\alpha} = \lambda(K_y^2 + \eta I)\boldsymbol{\beta} \quad (14)$$

Analogues to the primal problem, we will define big matrices, K_D which contains $(K_x^2 + \eta I)$ and $(K_y^2 + \eta I)$ as blocks on the diagonal and zeros at the blocks off the diagonal, and the matrix K_O which has the matrices $K_x K_y$ on the right-upper off diagonal block and $K_y K_x$ at the left-lower off-diagonal block. Also, we define $\boldsymbol{\gamma} = [\boldsymbol{\alpha}, \boldsymbol{\beta}]$. This leads to the equation,

$$K_O\boldsymbol{\gamma} = \lambda K_D\boldsymbol{\gamma} \;\Rightarrow\; K_D^{-1}K_O\boldsymbol{\gamma} = \lambda\boldsymbol{\gamma} \;\Rightarrow\; K_O^{\frac{1}{2}}K_D^{-1}K_O^{\frac{1}{2}}(K_O^{\frac{1}{2}}\boldsymbol{\gamma}) = \lambda(K_O^{\frac{1}{2}}\boldsymbol{\gamma}) \quad (15)$$

which is again a regular eigenvalue equation. Note that the regularization also moved the smallest eigenvalue away from zero, and hence made the inverse more numerically stable. The value for η needs to be chosen using cross-validation or some other measure. Solving the equations using this larger eigen-value problem is actually not quite necessary, and more efficient methods exist (see book).

The solutions are not expected to be sparse, because eigen-vectors are not expected to be sparse. One would have to replace L_2 norm penalties with L_1 norm penalties to obtain sparsity.