COS 424: Interacting with Data

Lecturer: David Blei
Scribe: Joseph Perla

Lecture 4/12/2007 #19

---

1 Summary of Last Lecture

Let's begin by restating the problem from the previous lecture.

We have data $\{x_1, \cdots, x_n\}$ with each vector x_i being p-dimensional. We want to transform that into $\{\lambda_1, \cdots, \lambda_n\}$ with each vector λ_i q-dimensional. $q < p$.

We reduce the dimensionality of the data.

2 PCA

Now let's talk about PCA and it's probabilistic interpretation. Say we are trying to reduce two-dimensional data to one dimension. Each 2D data point can be projected onto a subspace which is a line. How do we find the subspace onto which the two-dimensional data will be projected? Well, we can find the sum of squared distance (i.e. reconstruction error) from the data to its projection on the subspace.

So,

$$f(\lambda) = \mu + V_q\lambda$$

Now, μ is a p-dimensional vector which represents the offset in p-space. V_q is a pxq matrix, with q orthogonal unit vectors. Its q vectors are the principal components of the data. λ is the q vector. For example, if you are projecting onto a line, then a λ will be scalar and a high lambda will be further along the line.

$V_q\lambda$ is the projection onto q-space.

Given this, we try to minimize ssd (N is the number of data points):

$$min_{\mu,\lambda,V_q} \sum_{i=1}^{N} ||x_i - (\mu + V_q\lambda_i)||^2$$

And we can predict, from $i = 1, \cdots, N$ that

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

and

$$\hat{\lambda_i} = V_q^T(x_i - \hat{\mu}).$$

This is intuitive if you think about it geometrically.

To make the maths a bit easier, let us first center the data:

$$x_n \hat{=} x_n - \hat{\mu}$$

we substract the mean of the x's from each x_n.

After centering the data,

$$min_{V_q} \sum_{i=1}^{N} ||x_i - (V_q \lambda_i)||^2$$

and

$$min_{V_q} \sum_{i=1}^{N} ||x_i - (V_q V_q^T x_i)||^2$$

since, when optimized, $\lambda_i = V_q^T x_i$ and $\mu = 0$.

3 SVD

V^q is the solution plane. We can solve this optimization using SVD (singular value decomposition).

$$X = UDV^T$$

X = data matrix, $q \times p$
$U = n \times p$ orthogonal matrix (orthogonal meaning $U^T U = I$)
$D = p \times p$, a diagonal matrix where $d_1 \geq d_2 \geq d_3 \geq \cdots \geq d_p \geq 0$
$V^T = p \times p$ orthogonal matrix
Basically, make the following equations linearly independent. In signal processing this is known as "whitening":

$$\vec{x_1} = u_{11} d_1 \vec{v_1} + u_{12} d_2 \vec{v_2} + \cdots + u_{1p} d_p \vec{v_p}$$

$\vec{x_2} = u_{21} d_1 \vec{v_1} + u_{22} d_2 \vec{v_2} + \cdots + u_{2p} d_p \vec{v_p}$

$\vdots$

$\vec{x_N} = u_{N1} d_1 \vec{v_1} + u_{N2} d_2 \vec{v_2} + \cdots + u_{Np} d_p \vec{v_p}$
So, we end up with

$$\vec{\lambda_1} = u_{11} d_1 \vec{v_1} + u_{12} d_2 \vec{v_2} + \cdots + u_{1q} d_q \vec{v_q}$$

$\vec{\lambda_2} = u_{21} d_1 \vec{v_1} + u_{22} d_2 \vec{v_2} + \cdots + u_{2q} d_q \vec{v_q}$

$\vdots$

$\vec{\lambda_N} = u_{N1} d_1 \vec{v_1} + u_{N2} d_2 \vec{v_2} + \cdots + u_{Nq} d_q \vec{v_q}$
Since the rows are independent, we can throw away some of U's columns, D's rows and columns, and V's rows and columns when we find the λ's. Moreover, D contains the variance at each V^T, and remember that the values in the diagonals of D are in descending order. So, cutting off the lowest ones leaves only the dimensions which contribute the most to the observed data.

How do you choose q, the number of dimensions to reduce to? This is a hard problem. You would probably use techniques similar to ones used in K-Means.

4 Probabilistic PCA

From a high-level, we're going to generate some data from many low-dimensional gaussian distributions. Then, you project that into your high-dimensional space. Notice how this is a generative process.

So, let Z $N(0, 1)$. Take a simple two-Gaussian distribution:

$$x_1 \ N(V_1 Z, \sigma^2)$$

x_2 $N(V_2 Z, \sigma^2)$

The variance σ^2 is determining how far away you fall - it's related to the reconstruction error.

Finding V_1 and V_2 is the same as normal PCA. PCA $\equiv$ MLE of V. This draws the connection between PCA and Factor analysis.

If you want to reinterpret standard PCA in probabilistic terms, need to use a gaussian distribution. On the other hand, you can consider extensions to this model that relax the gaussian assumption.

5 Multivariate Gaussian Distribution

Consider the previous notes about multivariate gaussians.

We have the parameters:

mean μ which is a p-dimensional vector where each is the $E[X_i]$ where $X = (p \times 1)$ random vector some distribution with the covariance matrix below.

covariance matrix $\Sigma \succeq 0$,p $\times p$ matrix that is positive definite, i.e. positive and invertible.

Now, each σ_{ij} in Σ is the covariance bewteen the ith and jth components. i.e.

$$\sigma_{ij} = E[x_i x_j] - E[x_i]E[x_j]$$

and, logically, the diagonal values of Σ are just the variances of each dimension (where $i == j$):

$$\sigma_{ij} = E[x_i^2] - E[x_i]^2$$

Thus, very importantly, the probability of each datapoint can be calculated, using knowledge of the guassian equation, according to the following:

$$p(\vec{x}|\vec{u}, \Sigma) = (2\pi)^{n/2}|\Sigma|^{1/2} e^{-1/2(\bar{x}-\hat{u})^T \Sigma^{-1}(\vec{x}-\vec{u})}$$

Basically, you try to maximize the probability. How do we do this? MLE of course.

Data are $\{\vec{x_1}, \cdots, \vec{x_N}\}$, N p-dimensional vectors. MLE of

$$\hat{\mu} = 1/N \sum_{n=1}^{N} \vec{x_n}$$

$\hat{\Sigma} = 1/N \sum_{n=1}^{N} (x_n - \hat{\mu})(x_n - \hat{\mu})^T$

The MLE of μ is intuitive, just the average of the vectors. The MLE of Σ is just the eigenvectors of Σ, the principal components of the multinomial covariance. Ths is what PPCA finds.

If we graph some 2D data, then we can see what Σ might contain. If the graph cloud of points is circular, or is only stretched vertically or horizontally, then clearly the covariance between dimensions is small or non-existant. Only the diagonals of Σ are filled.

Otherwise, the cloud is stretched diagonally in some way. The eigenvectors of Σ would follow these major stretches. In addition, not just the diagonal of Σ will be filled with non-zero values. This would show some covariance between dimensions.