Linear Models

Max Welling

Gatsby Computational Neuroscience Unit
London
welling@gatsby.ucl.ac.uk

1 Introduction

In this class we will study models for which the observed random variables are continuous, and linearly related to the latent variables, denoted by $\mathbf{y}$. We will study Factor Analysis (FA) and Probabilistic Principal Component Analysis (PPCA). Principal Component Analysis (PCA), which is not a probabilistic model, will follow as limit case of PPCA.

PCA is a powerfull method in data analysis to meaningfully reduce the dimensionality of the data by searching for direction in data-space which have highest variance, and subsequently projecting the data onto it. However, PCA is not a probabilistic model, restricting its applicability in problems like MAP estimation in pattern recognition or data synthesis.

PPCA is precisely the extension that builds this probabilistic model around PCA. FA is much like PPCA, but allows a more complicated noise model.

Consider the following example. A class of 30 students is tested on 4 different subjects: math, biology, geography and psychology. Our data therefore consists of 30 grades across 4 disciplines. Let's plot this data in a 4 dimensional space. The question is whether we can uncover "correlations" in this data, for example, if we find a high grade in math, can we predict whether the grade in biology will also be high. In other words, can we find common factors, like intelligence, which can explain the data. Factor analysis is a method to uncover these hidden factors.

2 Factor Analysis (FA)

Consider the $k-\textit{dimenional}$ random variables $\mathbf{y}$ which are distributed according to an isotropic Gaussian $\mathcal{G}_{\mathbf{y}}[0, \mathbf{I}]$, i.e. they have zero mean, unit variance and are independent. These are hidden random variables, describing the sources. If we generate data from this distribution we would see a k-dimenional ball of points, getting more dense towards the centrum. Then, we will linearly combine these hidden variables into the observed random variables, describing the sensors,

$$\mathbf{x} = \mathbf{A}\mathbf{y} + \boldsymbol{\mu} + \boldsymbol{\nu} \tag{1}$$

where $\boldsymbol{\nu}$ is a noise random variable, distributed as, $\mathcal{G}_{\boldsymbol{\nu}}[0, \mathbf{\Sigma}]$. First, ignore the noise term. What happens to the cloud of latent data when we subject them to this linear transform? The constant term $\boldsymbol{\mu}$ simply shifts the center of the cloud, while the matrix $\mathbf{A}$ rotates, scales and skews the cloud into some ellips. This ellips must then be matched to the observed data by adjusting the parameters $\boldsymbol{\mu}$ and $\mathbf{A}$. Since we like to explain (or summarize) the data by a few factors, $\mathbf{A}$ is rectangular, rather than square in general. This implies that we have more observed dimensions than source dimensions, i.e. $d > k$. Next, add the noise term to our considerations. We will assume that every dimension has independent Gaussian noise but with arbitrary variance, i.e.

$$\boldsymbol{\nu} \sim \mathcal{G}_{\boldsymbol{\nu}}[0, \mathbf{\Sigma}], \qquad \mathbf{\Sigma} = \mathrm{diag}[\sigma_1^2, ..., \sigma_d^2]. \tag{2}$$

More precisely, given the factors, the data are independent, i.e. the correlations between the data are only described by the matrix $\mathbf{A}$. We will also assume that the noise is independent of the factors (sources),

$$\mathbf{E}[\mathbf{y}\boldsymbol{\nu}^T] = \mathbf{E}[\mathbf{y}]\mathbf{E}[\boldsymbol{\nu}^T] = 0 \tag{3}$$

Because both the factors and the noise are assumed Gaussian, the data are also normally distributed with the following two moments,

$$\begin{aligned} \mathbf{E}[\mathbf{x}] &= \boldsymbol{\mu} && (4) \\ \mathbf{E}[\mathbf{x}\mathbf{x}^T] &= \mathbf{E}[(\mathbf{A}\mathbf{y} + \boldsymbol{\mu} + \boldsymbol{\nu})(\mathbf{A}\mathbf{y} + \boldsymbol{\mu} + \boldsymbol{\nu})^T] = \mathbf{A}\mathbf{A}^T + \mathbf{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T && (5) \end{aligned}$$

Two important observations. Firstly, we see that the parameter $\boldsymbol{\mu}$ equals the mean of the data. For a Gaussian, the maximum likelihood estimator for the mean is simply the sample mean, so we may use,

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n \tag{6}$$

This is equivalent to subtracting this sample mean from the data in a preprocessing step and using $\boldsymbol{\mu} = 0$ subsequently. In the following, we will keep track of $\boldsymbol{\mu}$, since we will need it for the "mixture of FA" model. Secondly, note that if we rotate $\mathbf{A} \rightarrow \mathbf{AR}$, than the covariance of the data does not change, i.e. there is a redundancy in the parametrization. The result is that we can hope to find the matrix $\mathbf{A}$ only up to this rotation, which rotates the sources in the latent variable space.

If we had not constrained $\boldsymbol{\Sigma}$ to be diagonal, then an even bigger redundancy would result: $\mathbf{A} = 0$ and $\boldsymbol{\Sigma}$ equal to the data covariance would be a viable solution; obviously one that we don't want since we explain everything with noise. The diagonality puts the burden of explaining the correlations among the data on $\mathbf{A}$. The noise is then added individually to the sensors.

Let us proceed with the parameter estimation using EM. Again the crux is to compute the posterior density,

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}) &= \frac{p(\mathbf{x}|\mathbf{y})\ p(\mathbf{y})}{p(\mathbf{x})} \\ &= \frac{\mathcal{G}_{\mathbf{x}}[\boldsymbol{\mu} + \mathbf{Ay}, \boldsymbol{\Sigma}]\ \mathcal{G}_{\mathbf{y}}[0, \mathbf{I}]}{p(\mathbf{x})} \end{aligned} \tag{7}$$

We will use the following strategy to compute this posterior. We first notice that since we are dealing with Gaussians in both numerator and denominator, the end-result must be a Gaussian as well. Since it has to be a probability density in $\mathbf{y}$, the overall normalization constant can be determined easily. This is the reason that we do not need to bother about $p(\mathbf{x})$. The trick is now to notice that,

$$\mathcal{G}_{\mathbf{x}}[\boldsymbol{\mu} + \mathbf{Ay}, \boldsymbol{\Sigma}] = k(\mathbf{x})\ \mathcal{G}_{\mathbf{y}}[(\mathbf{A}^T\boldsymbol{\Sigma}^{-1}\mathbf{A})^{-1}\mathbf{A}^T\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}),\ \ (\mathbf{A}^T\boldsymbol{\Sigma}^{-1}\mathbf{A})^{-1}], \tag{8}$$

where $k(\mathbf{x})$ denotes a constant not depending on $\mathbf{y}$. This can be proved by looking at the quadratic form in the exponent and rearanging terms such that it becomes a Gaussian in $\mathbf{y}$. Next, we need to combine this result with $\mathcal{G}_{\mathbf{y}}[0, \mathbf{I}]$ using the following lemma,

$$\mathcal{G}_{\mathbf{y}}[\mathbf{a}, \mathbf{A}]\ \mathcal{G}_{\mathbf{y}}[\mathbf{b}, \mathbf{B}] = \mathcal{G}_{\mathbf{y}}[(\mathbf{A}^{-1} + \mathbf{B}^{-1})^{-1}(\mathbf{A}^{-1}\mathbf{a} + \mathbf{B}^{-1}\mathbf{b}), (\mathbf{A}^{-1} + \mathbf{B}^{-1})^{-1}]\ \mathcal{G}_{\mathbf{a}}[\mathbf{b}, \mathbf{A} + \mathbf{B}]. \tag{9}$$

We are only interested in the first factor since the second one can be absorbed in the normalization constant. After using that, we can still simplify by using the following matrix identities,

$$(\mathbf{P}^{-1} + \mathbf{B}^T\mathbf{R}^{-1}\mathbf{B})^{-1} = \mathbf{P} - \mathbf{PB}^T(\mathbf{BPB}^T + \mathbf{R})^{-1}\mathbf{BP} \tag{10}$$

$$(\mathbf{P}^{-1} + \mathbf{B}^T\mathbf{R}^{-1}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{R}^{-1} = \mathbf{PB}^T(\mathbf{BPB}^T + \mathbf{R})^{-1} \tag{11}$$

Finally we find that the posterior is given by a Gaussian with mean and covariance,

$$\mathbf{E}[\mathbf{y}|\mathbf{x}_n] = \mathbf{A}^T(\mathbf{AA}^T + \boldsymbol{\Sigma})^{-1}(\mathbf{x}_n - \boldsymbol{\mu}) \tag{12}$$

$$\mathbf{E}[\mathbf{yy}^T|\mathbf{x}_n] - \mathbf{E}[\mathbf{y}|\mathbf{x}_n]\mathbf{E}[\mathbf{y}|\mathbf{x}_n]^T = \mathbf{I} - \mathbf{A}^T(\mathbf{AA}^T + \boldsymbol{\Sigma})^{-1}\mathbf{A} \tag{13}$$

To learn the parameters $\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{A}$ of the model using EM, we need to optimize,

$$\begin{aligned} Q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}) = \sum_{n=1}^{N} \int d\mathbf{y}\ p(\mathbf{y}|\mathbf{x}_n, \boldsymbol{\theta}_{t-1})\ \ \log[p(\mathbf{x}_n|\mathbf{y}, \boldsymbol{\theta}_t)\ p(\mathbf{y})] \\ = \frac{1}{2}\sum_{n=1}^{N} \int d\mathbf{y}\ p(\mathbf{y}|\mathbf{x}_n, \boldsymbol{\theta}_{t-1})\ \{\log[\det(\boldsymbol{\Sigma}^{-1})] - (\mathbf{x}_n - \mathbf{Ay} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \mathbf{Ay} - \boldsymbol{\mu}) + \mathbf{yy}^T\} \end{aligned} \tag{14}$$

The term $\mathbf{yy}^T$ does not depend on the parameters and may be dropped. For the second term it is convenient to define $\mathbf{A}' = [\mathbf{A}, \boldsymbol{\mu}]$ and $\mathbf{y}' = [\mathbf{y}^T, 1]^T$. If we use centered the data, we simply set $\mathbf{A} = \mathbf{A}'$, $\mathbf{y}' = \mathbf{y}$ and $\boldsymbol{\mu} = 0$. We can thus rewrite Q as,

$$Q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}) = \frac{1}{2}\sum_{n=1}^{N} \int d\mathbf{y}\ p(\mathbf{y}|\mathbf{x}_n, \boldsymbol{\theta}_{t-1})\ \{\log[\det(\boldsymbol{\Sigma}^{-1})] - (\mathbf{x}_n - \mathbf{A}'\mathbf{y}')^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \mathbf{A}'\mathbf{y}')\} \tag{15}$$

For the M-step we need to take derivatives with respect to the parameters. First, taking derivatives with respect to $\mathbf{A}'$ we get,

$$\frac{\partial Q}{\partial \mathbf{A}'} = \sum_{n=1}^{N} \boldsymbol{\Sigma}^{-1}\left(\mathbf{x}_n\mathbf{E}[\mathbf{y}'|\mathbf{x}_n]^T - \mathbf{A}'\mathbf{E}[\mathbf{y}'\mathbf{y}'^T|\mathbf{x}_n]\right) \Rightarrow$$

$$\mathbf{A}'^{\mathbf{new}} = \left(\sum_{n=1}^{N} \mathbf{x}_n \mathbf{E}[\mathbf{y}'|\mathbf{x}_n]^T\right)\left(\sum_{n=1}^{N} \mathbf{E}[\mathbf{y}'\mathbf{y}'^T|\mathbf{x}_n]\right)^{-1} \tag{16}$$

Finally, taking derivatives with respect to $\frac{1}{\sigma_k^2}$ gives,

$$\frac{\partial Q}{\partial \mathbf{\Sigma}^{-1}} \frac{\partial \mathbf{\Sigma}^{-1}}{\partial (\sigma_k^2)^{-1}} = \sum_{i,j} \frac{\partial Q}{\partial \mathbf{\Sigma}_{ij}^{-1}} \delta_{i,k}\delta_{j,k}$$

$$= \frac{1}{2}\sum_{n=1}^{N} \left(\mathbf{\Sigma} - \mathbf{x}_n\mathbf{x}_n^T - \mathbf{x}_n\mathbf{E}[\mathbf{y}'|\mathbf{x}_n]^T\mathbf{A}'^T - \mathbf{A}'\mathbf{E}[\mathbf{y}'|\mathbf{x}_n]\mathbf{x}_n^T + \mathbf{A}'\mathbf{E}[\mathbf{y}'\mathbf{y}'^T|\mathbf{x}_n]\mathbf{A}'^T\right)_{ij} \delta_{i,k}\delta_{j,k} \Rightarrow$$

$$\mathbf{\Sigma}^{\mathbf{new}} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{diag}\left[\mathbf{x}_n\mathbf{x}_n^T - \mathbf{A}'\mathbf{E}[\mathbf{y}'|\mathbf{x}_n]\mathbf{x}_n^T\right] \tag{17}$$

where we used the update rule for $\mathbf{A}'$, and "**diag**" means taking only the values from the diagonal and setting the off-diagonal values zero. Note again that we must use the "new" parameter $\mathbf{A}'$ (i.e. the one updated in the same iteration). Iterating E-steps and M-steps will again converge to a ML estimate of the parameters.

The log-likelihood is also easy to compute since it is determined by a Gaussian with mean $\boldsymbol{\mu}$ and covariance $\mathbf{A}\mathbf{A}^T + \mathbf{\Sigma}$ (5),

$$L = \sum_{n=1}^{N} \log\left[p(\mathbf{x}_n)\right] \propto -\frac{1}{2}N\log\left[\det(\mathbf{A}\mathbf{A}^T + \mathbf{\Sigma})\right] - \frac{1}{2}\sum_{n=1}^{N}(\mathbf{x}_n - \boldsymbol{\mu})^T(\mathbf{A}\mathbf{A}^T + \mathbf{\Sigma})^{-1}(\mathbf{x}_n - \boldsymbol{\mu}) \tag{18}$$

3 Probabilistic Principle Component Analysis (PPCA)

In the following we will consider a special case of the above by setting, $\mathbf{\Sigma} = \sigma^2\mathbf{I}$, i.e. isotropic Gaussian noise. This simplification thus implies that all sensors must have the same noise variance.

In this case the E-step simplifies to,
E-step:

$$\mathbf{E}[\mathbf{y}|\mathbf{x}_n] = (\mathbf{A}^T\mathbf{A} + \sigma^2\mathbf{I})^{-1}\mathbf{A}^T(\mathbf{x}_n - \boldsymbol{\mu}) \tag{19}$$

$$\mathbf{E}[\mathbf{y}\mathbf{y}^T|\mathbf{x}_n] - \mathbf{E}[\mathbf{y}|\mathbf{x}_n]\mathbf{E}[\mathbf{y}|\mathbf{x}_n]^T = \sigma^2(\mathbf{A}^T\mathbf{A} + \sigma^2\mathbf{I})^{-1} \tag{20}$$

where we used (11) from right to left. This has the advantage that we only have to do one inverse of a $k \times k$ matrix instead of one inverse of a $d \times d$ matrix. The M-steps remain basically the same, (16). One can simplify (17) to,

$$\sigma^{\mathbf{new}\,2} = \frac{1}{d}\frac{1}{N}\sum_{n=1}^{N} \mathbf{trace}\left[\mathbf{x}_n^T\mathbf{x}_n - \mathbf{x}_n^T\mathbf{A}'\mathbf{E}[\mathbf{y}'|\mathbf{x}_n]\right] \tag{21}$$

This model with isotropic Gaussian noise is known as probabilistic principal component analysis, and it can be shown that the columns of $\mathbf{A}$ will span the principal subspace of the data(Tipping and Bishop, 1997). To make this more precise will first define what a PCA would deliver. The idea is to find a subspace such that if we project the data onto that subspace, we have found the maximal variance projection. As we will see, this also implies that we have minimized the reconstruction error of the data. To find the maximal variance subspace, we perform an eigen-value decomposition of the covariance matrix,

$$\mathbf{C} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \boldsymbol{\mu})(\mathbf{x}_n - \boldsymbol{\mu})^T = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \tag{22}$$

where $\mathbf{U}$ is an orthogonal matrix whose columns are the orthonormal eigenvectors of the sample covariance matrix, and $\mathbf{\Lambda}$ is the diagonal matrix containing the eigenvalues of $\mathbf{C}$[1]. Notice, that if we transform to the

[1] Alternatively, and more efficiently, we could do a singular value decomposition on the matrix $\frac{1}{\sqrt{N}}\mathbf{X}_{i,n}$ which would give,

$$\frac{1}{\sqrt{N}}\mathbf{X} = \mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{V}^T \tag{23}$$

revealing both $\mathbf{U}$ and $\mathbf{\Lambda}$, together with an "uninteresting" $d \times N$ matrix $\mathbf{V}$.

basis defined by the columns of $\mathbf{U}$, then the sample covariance becomes diagonal,

$$\mathbf{C}' = \mathbf{U}^T \mathbf{C} \mathbf{U} = \mathbf{U}^T \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T \mathbf{U} = \mathbf{\Lambda} \tag{24}$$

This means that in this basis the correlations dissappear (we have decorrelated the data) and the eigenvalues of $\mathbf{\Lambda}$ can be interpreted as variances along the direction of the new basisvectors,

$$\sigma_i^2 = \mathbf{\Lambda}_{ii} \tag{25}$$

The best rank-k approximation (in L_2 norm) to this covariance matrix, is given by choosing the first k columns of $\mathbf{U}$ and the first (largest) k eigenvalues from the diagonal of $\mathbf{\Lambda}$,

$$\mathbf{C}_k = \mathbf{U}_k \mathbf{\Lambda}_k \mathbf{U}_k^T \tag{26}$$

The projection onto this subspace is then given by,

$$\mathbf{x}'_n = \mathbf{U}_k \; \mathbf{U}_k^T \mathbf{x}_n \qquad \forall \;\; n \tag{27}$$

The claim that PPCA finds the same principal subspace can now be made more precise by saying that,

$$\mathbf{A} = \mathbf{U}_k (\mathbf{\Lambda}_k - \sigma^2 \mathbf{I}_k)^{\frac{1}{2}} \mathbf{R} \tag{28}$$

The matrix $\mathbf{R}$ is the arbitrary rotation matrix which we already mentioned after (5) and which rotates the basis inside the principal plane. Since the log-likelihood remains invariant if we choose another rotation $\mathbf{R}$, this implies that we cannot expect to find this $\mathbf{R}$ from the data. The redundancy in $\mathbf{R}$ may be lifted by pointing one basisvector (columns of $\mathbf{A}$) in the direction of the largest variance, choosing the second eigenvector in the plane of highest variance and orthogonal to the first one, etc. In that case, we may compute $\mathbf{R}$ by noticing that,

$$\mathbf{A}^T \mathbf{A} = \mathbf{R}^T (\mathbf{\Lambda}_k - \sigma^2 \mathbf{I}) \mathbf{R} \tag{29}$$

Therefore a simple eigenvector decomposition of the matrix $\mathbf{A}^T \mathbf{A}$ (which is a small $k \times k$ matrix) will determine $\mathbf{R}$ and $\mathbf{\Lambda}_k$ (but note that we need not be interested in this $\mathbf{R}$, since it just represent an arbitrary choice of basis in our pricipal subspace). Furthermore, we could derive the following result,

$$\sigma^2 = \frac{1}{d-k} \sum_{i=k+1}^{d} \mathbf{\Lambda}_{ii} \tag{30}$$

which states that σ^2 is the variance lost in projecting the data, averaged over the ignored dimensions.

The important contribution of PPCA is thus that it is guaranteed to find the principal subspace of the covariance matrix, like PCA, but the model is still fully probabilistic, unlike PCA. This may have a number of advantages in for instance data synthesis, pattern recognition, statistical testing, model comparison, mixture modeling. An additional advantage of PPCA is that it is nested. This implies that a higher dimensional subspace will always include a lower dimensional subspace. This property is not necessarily true for FA.

4 Principal component Analysis (PCA)

Now we will take the limit $\sigma^2 \to 0$. The posterior becomes a Gaussian with smaller and smaller variance. In the limit it approaches a delta function,

$$p(\mathbf{y}|\mathbf{x}_n) \to \delta(\mathbf{y} - (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T (\mathbf{x}_n - \boldsymbol{\mu})) \tag{31}$$

In this limit we may therefore alternate the following steps,

$$\mathbf{E}[\mathbf{y}|\mathbf{x}_n] = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T (\mathbf{x}_n - \boldsymbol{\mu}) \tag{32}$$

and (16), which now reduces to,

$$\mathbf{A}'^{\mathbf{new}} = \left(\sum_{n=1}^{N} \mathbf{x}_n \mathbf{E}[\mathbf{y}'|\mathbf{x}_n]^T \right) \left(\sum_{n=1}^{N} \mathbf{E}[\mathbf{y}'|\mathbf{x}_n] \mathbf{E}[\mathbf{y}'|\mathbf{x}_n]^T \right)^{-1} \tag{33}$$

Notice, that in this limit the correlations between the latent variables $\mathbf{y}$ vanish.

We will now find an explanation of the above update rules in terms of the "optimal reconstruction" property of PCA. We will do this by deriving the objective function that is optimized if we take the limit $\sigma \to 0$. We will look at $L_\sigma(\mathbf{x}^N) = \sigma^2 L(\mathbf{x}^N)$ and subsequently take the limit. We first recall that the log-likelihood can be decomposed into $L = Q+H$, where H was the entropy of the posterior probability. We have already seen that this posterior will become a single delta function. Such functions are in fact deterministic instead of random (there is no uncertainty) and its entropy therefore vanishes (since the entropy is a measure of uncertainty). Formally one needs to take the limit carefully to arrive at this result where we must use $\lim_{x\to 0} x \log x = 0$. We are therefore left with Q. Again, since the posterior is a deltafunction, we can do the intergral over $\mathbf{y}$ trivially, and arrive at,

$$L_{\sigma\to 0} = -\frac{1}{2}\sum_{n=1}^{N} \parallel (\mathbf{I} - \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T)(\mathbf{x}_n - \boldsymbol{\mu}) \parallel^2 \tag{34}$$

which is exactly the reconstruction error after projecting the data on the columns of $\mathbf{A}$. It is easily checked that

$$\mathbf{P} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T \tag{35}$$

is an orthogonal projection, ($\mathbf{P} = \mathbf{P}^T$, $\mathbf{P}^2 = \mathbf{P}$). Therefore, $(\mathbf{I} - \mathbf{P})(\mathbf{x}_n - \boldsymbol{\mu})$ is the reconstruction error. Minimizing this costfunction over $\mathbf{A}$ therefore boils down to finding a subspace, such that the data can be represented as accurate as possible (on average) after projecting them onto a subspace. Instead of trying to optimize the reconstruction error directly, we can split the problem into two subproblems and iterate them. We write,

$$L_0 = -\frac{1}{2}\sum_{n=1}^{N} \parallel \mathbf{x}_n - \boldsymbol{\mu} - \mathbf{A}\mathbf{y}_n \parallel^2 \tag{36}$$

where $\mathbf{y}_n$ are the data projected down in latent variable space. First, we consider these $\mathbf{y}_n$ fixed and ask what the optimal reconstruction matrix $\mathbf{A}$ is. Taking derivatives with respect to $\mathbf{A}$ and equating to zero gives,

$$\mathbf{A}' \to \left(\sum_{n=1}^{N} \mathbf{x}_n {\mathbf{y}'}_n^T\right)\left(\sum_{n=1}^{N} \mathbf{y}'_n {\mathbf{y}'}_n^T\right)^{-1} \tag{37}$$

Then, we fix the subspace spanned by the columns of $\mathbf{A}$, and optimize over the locations of the latent variables, which gives,

$$\mathbf{y}_n \to (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T(\mathbf{x}_n - \boldsymbol{\mu}) \tag{38}$$

These are of course simply the EM iterations in disguise (set $\mathbf{E}[\mathbf{y}|\mathbf{x}_n] = \mathbf{y}_n$). We conclude that by taking the limit $\sigma \to 0$ we will find the principal subspace of the data, such that the reconstruction error is minimized.

One could visualize the above process as follows using a spring model (Roweis, 1997). Project the data (orthogonally) onto the current estimate of the subplane. Fix the subplane at the origin such that it can still rotate. Attach springs between the projected points and the original samples and relax the system. It will equilibriate at the newly estimated position of the subspace. Project the samples again orthogonally onto the new subspace and relax etc. Iterating these projections (E-step) and relaxing of the plane (M-step) will then minimize the total springlength (L_0).

References

Roweis, S. (1997). Em algorithms for pca and sensible pca. Technical Report CNS-TR-97-02, California Institute of Technology, Computation and Neural Systems. To appear in NIPS 10.

Tipping, M. and Bishop, C. (1997). Probabilistic principal component analysis. Technical Report NCRG/97/010, Aston University, Department of Computer Science and Applied Mathematics, Neural Computing Research Group.