Introduction to Machine Learning.
CSCI-UA 9473, Lecture 9.

Augustin Cosse

Ecole Normale Supérieure, DMA & NYU
Fondation Sciences Mathématiques de Paris.

2018

Statistical intuition and Latent Variable models

- Many models discussed so far can in fact be considered as particular instances of latent variable models.
- The general factorization for a latent variable model of the form $\boldsymbol{z}_i \rightarrow \boldsymbol{x}_i$ is $p(\boldsymbol{x}_i|\boldsymbol{z}_i)p(\boldsymbol{z}_i)$. Recall that here $p(\boldsymbol{x}_i|\boldsymbol{z}_i)$ denotes the likelihood and the prior $p(\boldsymbol{z}_i)$ indicates the distribution of the latent state

$p(\boldsymbol{x}_i\|\boldsymbol{z}_i)$	$p(\boldsymbol{z}_i)$	Model
prod. Gauss.	prod. Gauss.	Factor Analysis/proba. PCA
MVN	cat.	Mixture of Gaussians
prod. cat	cat.	Mixture of Multinomials
prod. Gauss.	prod. Laplace	proba. ICA/sparse coding
prod. cat.	prod. Gauss.	multinomial PCA
prod. cat.	Dirichlet	Latent Dirichlet alloc.

Statistical intuition and Latent Variable models

- The simplest latent variable models assume that the data was generated from a distribution governed by a single latent state z_i. In a Bayesian approach, we let $p(z_i)$ to denote the prior for this latent state.
- We further let $p(\boldsymbol{x}_i|z_i = k)$ to denote the (assumed) distribution of the prototypes given the latent state $z_i = k$.
- A popular choice for the prior is the categorical distribution.

Statistical intuition and Latent Variable models

- The categorical distribution is usually used when we represent measurements as belonging to 1 of K possible exclusive classes.
- Clusters are usually represented through dummy encodings.
- The general form of a dummy encoding is $\boldsymbol{x} = [\mathbb{I}(x = 1), \ldots, \mathbb{I}(x = K)]$ where $\mathbb{I}(x = \alpha) = 1$ if x is in the cluster C_α.
- Then if we assume that the class probabilities are independent, and label those probabilities as θ_j, the categorical distribution reads as a particular case of the multinomial

$$\text{Cat}(\boldsymbol{x}|1, \theta) = \prod_{j=1}^{K} \theta_j^{\mathbb{I}(x_j=1)}$$

Statistical intuition and Latent Variable models

- More Generally, we will use the notation π_k to denote the class probabilities (probabilities to belong to the class C_k).
- With this notation, we thus have $p(z_i) = \text{Cat}(\boldsymbol{\pi})$ and in particular, and $p(z_i = k) = \pi_k$
- Given those probabilities, the probability to observe a prototype $\boldsymbol{x}_i$ from the dataset is given by

$$p(\boldsymbol{x}_i|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p_k(\boldsymbol{x}_i|\boldsymbol{\theta})$$

Here $p_k(\boldsymbol{x}_i|\theta)$ denote the k^{th} base distribution (i.e distribution to observe the prototype $\boldsymbol{x}_i$ knowing it belongs to the k^{th} cluster.). The general model above is known as a Mixture Model.

Statistical intuition and Latent Variable models

- The two most popular models are the Mixture of Gaussians

$$p(\boldsymbol{x}_i|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- and the mixture of Multinoullis,

$$p(\boldsymbol{x}_i, z_i = k|\boldsymbol{\theta}) = \prod_{j=1}^{D} \text{Ber}(x_{ij}|\mu_{jk}) = \prod_{j=1}^{D} \mu_{jk}^{x_{ij}} (1 - \mu_{jk})^{1-x_{ij}}$$

- The Mixture of Multinoullis is useful when using dummy encodings, i.e. when $\boldsymbol{x}_i = (0, 1, 0, \ldots, 0)$ and we use μ_{jk} to denote the probability that the j^{th} bit is 1 in sequences from cluster k.

The EM algorithm

- When we want to fit a probabilistic model to the data, we usually minimize the negative log-likelihood (we look for the parameters that make it maximally likely to observe the given data)
- In the case of a GMM, the log likelihood reads as

$$\log\ p(\boldsymbol{x}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

- Minimizing this expression directly is hard because the log cannot be pushed inside the sum

The EM algorithm

- The general form of a probability distribution from the exponential family reads as

$$p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp(\boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{\theta}))$$

- And the parameters we want to find also appear in the normalizing constant. I.e for a distribution from the exponential family, the observed log-likelihood reads as

$$\begin{aligned} \ell(\theta) &= \sum_i \log \sum_{\boldsymbol{z}_i} p(\boldsymbol{x}_i, \boldsymbol{z}_i|\boldsymbol{\theta}) \\ &= \sum_i \log \left[\sum_{\boldsymbol{z}_i} e^{\boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{z}_i, \boldsymbol{x}_i)} \right] - N \log Z(\boldsymbol{\theta}) \end{aligned}$$

- The log sum exp is convex and the normalizing constant $Z(\theta)$ is convex as well. However, the difference of two convex function

The EM algorithm

- An easier approach would be to optimize the complete data log-likelihood

$$\ell_c(\boldsymbol{\theta}) = \sum_i \log p(\boldsymbol{x}_i, \boldsymbol{z}_i|\boldsymbol{\theta}) = \boldsymbol{\theta}^T \left(\sum_i \phi(\boldsymbol{x}_i, \boldsymbol{z}_i) \right) - NZ(\boldsymbol{\theta})$$

- In the exponential family the normalizing constant is convex so that the whole function is concave and can be optimized efficiently

The EM algorithm

- For the reasons listed above, we would clearly prefer to work with the complete data log-likelihood

$$\ell_c(\theta) = \sum_{i=1}^{N} \log p(\boldsymbol{x}_i, \boldsymbol{z}_i | \boldsymbol{\theta})$$

- The problem is that we don't have access to the joint probability $p(\boldsymbol{x}_i, \boldsymbol{z}_i | \boldsymbol{\theta})$

- To get round the difficulty, and to estimate the parameters of the mixture together with the latent states, the EM algorithm works on the expected complete data log-likelihood

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1}) = \mathbb{E}\left\{\ell_c(\boldsymbol{\theta}) | \mathcal{D}, \boldsymbol{\theta}^{t-1}\right\}$$

Here $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1})$ is called the auxilliary function.

The EM algorithm

- For the reasons listed above, we would clearly prefer to work with the complete data log-likelihood

$$\ell_c(\theta) = \sum_{i=1}^{N} \log p(\boldsymbol{x}_i, \boldsymbol{z}_i | \boldsymbol{\theta})$$

- To get round the difficulty, and to estimate the parameters of the mixture together with the latent states, the EM algorithm works on the expected complete data log-likelihood

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1}) = \mathbb{E}\left\{\ell_c(\boldsymbol{\theta}) | \mathcal{D}, \boldsymbol{\theta}^{t-1}\right\}$$

  Here $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1})$ is called the auxilliary function.

- The E-step computes the expression of Q (or the terms needed to express Q). The M-step optimizes Q with respect to $\boldsymbol{\theta}$.

The Auxilliary function

- The auxilliary function reads as

$$
\begin{aligned}
Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1}) &= \mathbb{E}\left\{ \sum_i \log\ p(\boldsymbol{x}_i, z_i|\boldsymbol{\theta}) \right\} \\
&= \sum_i \mathbb{E}\left\{ \log \left[\prod_{k=1}^{K} (\pi_k p(\boldsymbol{x}_i|\boldsymbol{\theta}_k))^{\mathbb{I}(z_i=k)} \right] \right\} \\
&= \sum_i \sum_k \mathbb{E}\left\{ \mathbb{I}(z_i = k) \right\} \log(\pi_k p(\boldsymbol{x}_i|\boldsymbol{\theta}_k)) \\
&= \sum_i \sum_k p(z_i = k|\boldsymbol{x}_i, \boldsymbol{\theta}^{t-1}) \log(\pi_k p(\boldsymbol{x}_i|\boldsymbol{\theta}_k)) \\
&= \sum_i \sum_k r_{ik} \log \pi_k + \sum_i \sum_k r_{ik} \log p(\boldsymbol{x}_i|\boldsymbol{\theta}_k)
\end{aligned}
$$

- It is fully determined from the responsibilities r_{ik} of the cluster k in the realization of the prototype $\boldsymbol{x}_i$ as well as the likelihoods $p(\boldsymbol{x}_i|\boldsymbol{\theta}_k)$ which follow from gaussianity and θ

The EM algorithm for GMMs

- Given the likelihoods, the E-step updates the responsibilities as

$$r_{ik} = \frac{\pi_k p(\boldsymbol{x}_i|\boldsymbol{\theta}^{t-1})}{\sum_{k'} \pi_{k'} p(\boldsymbol{x}_i|\boldsymbol{\theta}_{k'}^{t-1})}$$

- The M-step then optimizes Q with respect to the class probabilities π_k and the parameters $\boldsymbol{\theta}_k$
- The estimates for π_k are given by

$$\pi_k = \frac{1}{N} \sum_i r_{ik} = \frac{r_k}{N}$$

- The estimates for $\theta = (\mu_k, \sigma_k)$ are obtained by substituting the Normal distribution for $p(\boldsymbol{x}_i, \theta_k)$ in the log likelihood and minimizing

The EM algorithm for GMMs (E-step)

- Substituting the normal distributions for the $p(\boldsymbol{x}_i|\theta_k)$, we get

$$\begin{aligned}\ell(\boldsymbol{m}_k, \boldsymbol{\Sigma}_k) &= \sum_k \sum_i r_{ik} \log\ p(\boldsymbol{x}_i|\boldsymbol{\theta}_k) \\ &= -\frac{1}{2} \sum_i r_{ik} \left[\log|\boldsymbol{\Sigma}_k| + (\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)\right]\end{aligned}$$

- Setting the derivatives to zero, we get

$$\boldsymbol{\mu}_k = \frac{\sum_i r_{ik} \boldsymbol{x}_i}{r_k}$$

$$\boldsymbol{\sigma}_k = \frac{\sum_i r_{ik} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T}{r_k} = \frac{\sum_i r_{ik} \boldsymbol{x}_i \boldsymbol{x}_i^T}{r_k} - \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T$$

Relation to K-means

- Remember the K means algorithm? K means can be considered as a particular instance of the EM algorithm. If we assume that $\boldsymbol{\Sigma}_k = \sigma^2 \boldsymbol{I}_D$ and $\pi_K = \frac{1}{K}$ is fixed, we only update the centers of the the clusters
- Instead of the previous responsibilities, one can assume that the probability of a prototype belonging to a cluster is either 1 or 0.
- We can choose the only possible cluster for a prototype to be the one that maximizes the likelihood

$$z_i^* = \underset{k}{\operatorname{argmax}}\, p(z_i = k | \boldsymbol{x}_i, \boldsymbol{\theta})$$

- We then set the probability that $\boldsymbol{x}_i$ belongs to this cluster to 1. We can do this because we assumed $\pi_k = 1/K$ fixed and $\boldsymbol{\Sigma}_k = \sigma^2 \boldsymbol{I}$

- Under the earlier hypotheses, maximizing the posterior to find the most likely assignement reduces to the minimization

$$z_i^* = \underset{k}{\operatorname{argmin}} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|$$

- And the M-step updates the centers as

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i|z_i=k} \boldsymbol{x}_i$$

K-means vs EM

- K means
  - + Better running time
  - + More interesting for high dimensional data
  - + Interpretation is easier
  - − Assumes clusters are spherical (see Mouse dataset). So does not work well with complex shape.
  - − The "Hard assignement" approach might lead to misclassification
- EM Clustering
  - + Works usually better when there is some uncertainty regarding the assignment
  - + Does not assume any predefined geometry for the clusters
  - − Uses more information than K-means so more difficult to implement in high dimension.
  - − More difficult to interpret

Statistical intuition for FA, PCA and ICA

- Now that we have introduced GMMs and the notion of latent variable model, we are ready to discuss the statistical intuition for Factor analysis, PCA and ICA models.
- Gaussian mixture models are very general in that every observation is assumed to have been generated from one of k independent clusters with their respective mean and covariance.
- An alternative is to view the distribution of prototypes as something smoother and to replace the hard assignement (i.e each of the prototype belongs (exlusively) to one of the k clusters) by the assumption that the prototypes are organized according to a set of gaussian distributions concentrated around a single point.

Statistical intuition for FA, PCA and ICA

- Such model then relies on a first continuous prior for the latent variables z_i,

$$p(z_i) = \mathcal{N}(z_i|\mu_0, \Sigma_0)$$

An then model each of the prototypes distributions as gaussian distributions centered around a mean defined from this continously varying latent variable z_i

$$p(x_i|z_i, \theta) = \mathcal{N}(Wz_i + \mu, \Psi)$$

In this case, $\theta = (W, \mu, \Psi)$

- Factor analysis can thus be understood as a GMM with constraints on the mean and covariance but continuous latent variables

Statistical intuition for FA, PCA and ICA

$$p(\boldsymbol{x}_i|\boldsymbol{z}_i, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{W}\boldsymbol{z}_i + \boldsymbol{\mu}, \boldsymbol{\Psi})$$

- Here $\boldsymbol{W}$ is known as the factor loading matrix and $\boldsymbol{\Psi}$ is the covariance matrix.
- In practice, the covariance matrix $\boldsymbol{\Psi}$ is usually taken to be diagonal and we therefore turn to the $\boldsymbol{z}_i$ and their connection through the latent distribution $\mathcal{N}(\boldsymbol{z}_i|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$
- When the covariance is taken to be spherical, i.e. $\boldsymbol{\Psi} = \sigma^2\boldsymbol{I}$, we get the probabilistic PCA model as we will see.

Low rank covariance

- The model $p(\boldsymbol{x}_i|\boldsymbol{\theta})$ given by the combination of the prior $p(\boldsymbol{z}_i)$ and the likelihood $p(\boldsymbol{x}_i|\boldsymbol{z}_i, \boldsymbol{\theta})$ is known as a linear gaussian system.

- For a general Gaussian system

$$\begin{cases} p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x) \\ p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}, \boldsymbol{\Sigma}_y) \end{cases}$$

  The distribution of $\boldsymbol{y}$ (a.k.a normalizing constant) is defined as

$$p(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{A}\boldsymbol{\mu}_x + \boldsymbol{b}, \boldsymbol{\Sigma}_y + \boldsymbol{A}\boldsymbol{\Sigma}_x\boldsymbol{A}^T)$$

- Using this to derive the distribution of prototypes in FA, we get the distribution

$$\begin{aligned} p(\boldsymbol{x}_i|\boldsymbol{\theta}) &= \int \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{W}\boldsymbol{z}_i + \boldsymbol{\mu}, \boldsymbol{\Psi})\mathcal{N}(\boldsymbol{z}_i|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)\, d\boldsymbol{z}_i \\ &= \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{W}\boldsymbol{\mu}_0 + \boldsymbol{\mu}, \boldsymbol{\Psi} + \boldsymbol{W}\boldsymbol{\Sigma}_0\boldsymbol{W}^T) \end{aligned}$$

FA as a low rank model for the covariance

- The mean and covariance of the factor analysis model thus read as

$$\mathbb{E}\{\boldsymbol{x}_i\} = \boldsymbol{W}\boldsymbol{\mu}_0 + \boldsymbol{\mu}, \quad \text{cov}\{\boldsymbol{x}_i\} = \boldsymbol{W}\mathbb{E}\left\{\boldsymbol{z}\boldsymbol{z}^T\right\}\boldsymbol{W}^T + \boldsymbol{\Psi}$$
$$= \boldsymbol{W}\boldsymbol{\Sigma}_0\boldsymbol{W}^T + \boldsymbol{\Psi}$$

- An additional interpretation of the factor analysis model can be obtained by noting that one can always write $\boldsymbol{\mu}' = \boldsymbol{\mu} + \boldsymbol{W}\boldsymbol{\mu}_0$ and take $\boldsymbol{\mu}'_0 = 0$. We can also always take $\boldsymbol{\Sigma}_0 = \boldsymbol{I}$ as we can always write the model by introducing the factor $\tilde{\boldsymbol{W}} = \boldsymbol{W}\boldsymbol{\Sigma}_0^{-1/2}$,

$$\text{cov}(\boldsymbol{x}_i) = \boldsymbol{W}\boldsymbol{\Sigma}_0\boldsymbol{W}^T + \boldsymbol{\Psi} = \tilde{\boldsymbol{W}}\tilde{\boldsymbol{W}}^T + \boldsymbol{\Psi} \qquad (1)$$

- Factor analysis can thus be understood as a Gaussian model with a low rank covariance matrix !

$$p(\boldsymbol{x}_i|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}, \boldsymbol{W}\boldsymbol{W}^T + \boldsymbol{\Psi})$$

Unidentifiability

- Learning a factor analysis model is ill posed in general (i.e we say that the parameters of the FA model are unidentifiable)
- Replacing the factor $\boldsymbol{W}$ by any other matrix of the form $\boldsymbol{WR}$ where $\boldsymbol{R}$ is an orthogonal matrix $\boldsymbol{RR}^T = \boldsymbol{I}$, we get

$$\text{cov}[\boldsymbol{x}] = \boldsymbol{WRR}^T\boldsymbol{W}^T + \boldsymbol{\Psi} = \boldsymbol{WW}^T + \Psi$$

- There exists a couple of approaches to reduce the number of dof
  - Force $\boldsymbol{W}$ to be orthonormal (this is the approach followed by PCA)
  - Force $\boldsymbol{W}$ to be lower triangular together with $W_{ii} > 0$ for all i
  - Sparsity prior on the $\boldsymbol{W}$ in the form of ℓ_1 regularization (this is known as sparse factor analysis)
  - Select the rotation matrix $\boldsymbol{R}$ that leads to easier interpretations (e.g. enforce sparsity)
  - Use non gaussian priors on the latent variables $\boldsymbol{z}_i$ (ICA)

Mixture of Factor Analyzers

- So far we assumed that the means $\boldsymbol{\mu} + \boldsymbol{W}\boldsymbol{z}_i$ all live in the same affine subspace.

- An alternative if we want to capture the low dimensional nature of the data locally (and keep a small number of latent variables) is to introduce multiple subspaces, $\{\boldsymbol{\mu}_k, \boldsymbol{W}_k\}_{k=1}^K$

- The model, which is known as mixture of factor analyzers (MFA) then read as

$$
\begin{aligned}
p(\boldsymbol{x}_i|\boldsymbol{z}_i, q_i = k, \boldsymbol{\theta}) &= \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k + \boldsymbol{W}_k\boldsymbol{z}_i, \boldsymbol{\Psi}) \\
p(\boldsymbol{z}_i|\boldsymbol{\theta}) &= \mathcal{N}(\boldsymbol{z}_i|\boldsymbol{0}, \boldsymbol{I}) \\
p(q_i|\boldsymbol{\theta}) &= \text{Cat}(q_i|\boldsymbol{\theta})
\end{aligned}
$$

  where we introduced the latent variables q_i which indicates the local subspace to be used and use the Categorical distribution to encode the corresponding distributions of those variables.

Fitting Mixtures of FA and the EM algorithm

- When learning a FA model, we learn the parameters of the posterior $p(\boldsymbol{x}_i|\boldsymbol{z}_i, \boldsymbol{\theta})$ and of the prior, or latent distribution $p(\boldsymbol{z}_i)$. Once we have those parameters, we usually want to see whether we can discover something meaningful on the data based the latent variables $\boldsymbol{z}_i$
- One can then analyze the shape of $p(\boldsymbol{z}_i|\boldsymbol{m}_i, \boldsymbol{\Sigma}_i)$ (as we deal with Gaussian distribution it is possible to compute a closed form expression for this distribution)

$$
\begin{aligned}
p(\boldsymbol{z}_i|\boldsymbol{x}_i, \boldsymbol{\theta}) &= \mathcal{N}(\boldsymbol{z}_i|\boldsymbol{m}_i, \boldsymbol{\Sigma}_i) \\
\boldsymbol{\Sigma}_i &= (\boldsymbol{\Sigma}_0^{-1} + \boldsymbol{W}^T \boldsymbol{\Psi}^{-1} \boldsymbol{W})^{-1} \\
\boldsymbol{m}_i &= \boldsymbol{\Sigma}_i (\boldsymbol{W}^T \boldsymbol{\Psi}^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}) + \boldsymbol{\Sigma}_0^{-1} \boldsymbol{\mu}_0)
\end{aligned}
$$

Fitting Mixtures of FA and the EM algorithm

- The simplest way to fit an FA model is to use the EM algorithm
- Applying the exact same steps as for the GMM, we first estimate the responsibilities of each pair (cluster,prototype), (c, i) by using Bayes rule (E-step)

$$r_{i,c} = p(q_i = c|\boldsymbol{x}_i, \boldsymbol{\theta}) \propto \pi_c \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{W}_c \boldsymbol{W}_c^T + \boldsymbol{\Psi})$$

- In the M-Step, we then update the parameters using the parametrization of the posteriors $p(\boldsymbol{z}_i|\boldsymbol{x}_i, q_i = c, \boldsymbol{\theta})$ (we assume $\boldsymbol{\mu}_0 = 0$ and $\boldsymbol{\Sigma}_0 = \boldsymbol{I}$),

$$p(\boldsymbol{z}_i|\boldsymbol{x}_i, q_i = c, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{z}_i|\boldsymbol{m}_{ic}, \boldsymbol{\Sigma}_{ic})$$
$$\boldsymbol{\Sigma}_{ic} = (\boldsymbol{I}_L + \boldsymbol{W}_c^T \boldsymbol{\Psi}_c^{-1} \boldsymbol{W}_c)^{-1}$$
$$\boldsymbol{m}_{ic} = \boldsymbol{\Sigma}_{ic}(\boldsymbol{W}_c^T \boldsymbol{\Psi}_c^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_c))$$

- The M-step is then completed by estimating the parameters $\hat{\boldsymbol{W}}$, $\hat{\boldsymbol{\Psi}}$ and $\hat{\pi}_c$ as

$$\hat{\boldsymbol{W}}'_c = \left[\sum_i r_{ic} \boldsymbol{x}_i \boldsymbol{b}_{ic}^T\right] \left[\sum_i r_{ic} \boldsymbol{C}_{ic}\right]^{-1}$$

$$\hat{\boldsymbol{\Psi}} = \frac{1}{N} \text{diag} \left\{\sum_{ic} r_{ic} \left(\boldsymbol{x}_i - \hat{\boldsymbol{W}}'_c \boldsymbol{b}_{ic}\right) \boldsymbol{x}_i^T\right\}$$

$$\hat{\pi}_c = \frac{1}{N} \sum_{i=1}^{N} r_{ic}$$

where we defined $\boldsymbol{W}'_c$, $\boldsymbol{b}_{ic}$ and $\boldsymbol{C}_{ic}$ as $\boldsymbol{W}'_c = [\boldsymbol{W}_c, \boldsymbol{\mu}_c]$

$$\boldsymbol{b}_{ic} = \mathbb{E}\left\{\boldsymbol{z}' | \boldsymbol{x}_i, q_i = c\right\} = [\boldsymbol{m}_{ic}; 1], \quad \boldsymbol{z}' = (\boldsymbol{z}, 1),$$

$$\boldsymbol{m}_{ic} = \mathbb{E}\left\{\boldsymbol{z}'(\boldsymbol{z}')^T | \boldsymbol{x}_i, q_i = c\right\},$$

$$= \begin{pmatrix} \mathbb{E}\left\{\boldsymbol{z}\boldsymbol{z}^T | \boldsymbol{x}_i, q_i = c\right\} & \mathbb{E}\left\{\boldsymbol{z} | \boldsymbol{x}_i, q_i = c\right\} \\ \mathbb{E}\left\{\boldsymbol{z} | \boldsymbol{x}_i, q_i = c\right\} & 1 \end{pmatrix}$$

Probabilistic PCA and classical PCA

- Constraint the covariance matrix of a FA model by requiring $\boldsymbol{\Psi} = \sigma^2 \boldsymbol{I}$ and $\boldsymbol{W}$ to be orthonormal leads to the classical PCA model when $\sigma^2 \to \infty$.
- When we only require $\sigma^2 > 0$, the model is known as probabilistic PCA.
- The connections between PCA and probabilitic PCA as well as their respective (statistical) interpretation is given by writing down the data log likelihood $\log p(\boldsymbol{X} | \boldsymbol{W}, \sigma^2)$

From Probabilistic to classical PCA

Probabilistic PCA (see Tipping, Bishop '99)

We consider a factor analysis model with $\Psi = \sigma^2 \boldsymbol{I}$. The data (or observed) log-likelihood is given by

$$\log p(\boldsymbol{X}|\boldsymbol{W}, \sigma^2) = -\frac{N}{2} \ln |\boldsymbol{C}| - \frac{1}{2} \sum_{i=1}^{N} \boldsymbol{x}_i^T \boldsymbol{C}^{-1} \boldsymbol{x}_i$$

where $\boldsymbol{C} = \boldsymbol{W}\boldsymbol{W}^T + \sigma^2 \boldsymbol{I}$ and $\boldsymbol{S} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_i \boldsymbol{x}_i^T = (\frac{1}{N}) \boldsymbol{X}\boldsymbol{X}^T$ (again we assumed that the $\boldsymbol{x}_i$ have been centered). The maxima of the log-likelihood are defined as

$$\hat{\boldsymbol{W}} = \boldsymbol{V}(\boldsymbol{\Lambda} - \sigma^2 \boldsymbol{I})^{1/2} \boldsymbol{R}$$

Where $\boldsymbol{R}$ is an arbitrary $L \times L$ orthogonal matrix, $\boldsymbol{V}$ is the $D \times L$ matrix whose columns are the first L eigenvectors of $\boldsymbol{S}$ and $\boldsymbol{\Lambda}$ is the corresponding diagonal matrix of eigenvalues. Without loss of generality we can set $\boldsymbol{R} = \boldsymbol{I}$.

Independent Component Analysis

- Just as PCA, ICA can be expressed a special instance of a Factor Analysis model. Recall that in FA we were expression the parameters as a linear function in the latent variables

$$\boldsymbol{x}_t = \boldsymbol{W}\boldsymbol{z}_t + \varepsilon_t$$

- $\boldsymbol{W}$ is thus called the mixing matrix and ε_t is viewed as some Gaussian noise $\varepsilon_t \sim \mathcal{N}(0, \boldsymbol{\Psi})$

- In PCA we assumed that the source were independent and distributed following a Gaussian distribution,

$$p(\boldsymbol{z}_t) = \prod_{j=1}^{L} \mathcal{N}(z_{t_j}|0, 1)$$

- In ICA, we relax the Gaussian assumption and let the source distributions be any non Gaussian distribution

$$p(\boldsymbol{z}_i) = \prod_{j=1}^{L} p_j(\boldsymbol{z}_{t_j})$$

Independent Component Analysis as MLE

- Just as before, we can write the log-likelihood for ICA. Here we assume that the data has been centered and whitened (which can be done by a first application of PCA)
- The covariance reads as $\mathbb{E}\left\{\boldsymbol{x}\boldsymbol{x}^T\right\} = \boldsymbol{W}\mathbb{E}\left\{\boldsymbol{z}\boldsymbol{z}^T\right\}\boldsymbol{W}^T$
- Using the whitening assumption, $\mathbb{E}\left\{\boldsymbol{z}\boldsymbol{z}^T\right\} = \boldsymbol{I}$ as well as the fact that the data is centered, $\mathbb{E}\left\{\boldsymbol{x}\boldsymbol{x}^T\right\}$, we see that the matrix $\boldsymbol{W}$ must be orthogonal
- Now using a change of variables, we can write the sample posterior $p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{W})$ as

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{W}, \boldsymbol{z}) &= p_x(\boldsymbol{W}\boldsymbol{z}) \\ &= p_z(\boldsymbol{z})|\det(\boldsymbol{W}^{-1})| \\ &= p_z(\boldsymbol{W}^{-1}\boldsymbol{x})|\det(\boldsymbol{W}^{-1}| \end{aligned}$$

Independent Component Analysis as MLE

- From the posterior $p(\boldsymbol{x}|\boldsymbol{W}, \boldsymbol{z})$, the data log-likelihood for a set of T samples follows as

$$\frac{1}{T}\log p(\mathcal{D}|\boldsymbol{V}) = \log|\det(\boldsymbol{V})| + \frac{1}{T}\sum_{j=1}^{L}\sum_{t=1}^{T}\log p_j(\boldsymbol{v}_j^T\boldsymbol{x}_t)$$

- Using orthogonality of the rows of $\boldsymbol{V}$, $\boldsymbol{v}_j$, and replacing the sum over the data with a population average, we get the reduced formulation for the negative LL

$$NLL(\boldsymbol{V}) = \sum_{j=1}^{L}\mathbb{E}\{G_j(z_j)\}$$

where $z_j = \boldsymbol{v}_j^T\boldsymbol{x}$ and $G_j(z) = -\log p_j(z)$.

- We then minimize the NLL under the constraints that the rows $\boldsymbol{v}_j$ are orthogonal and have unit norm (which follows from the whitening assumption and $\mathbb{E}\left\{\boldsymbol{v}_j^T\boldsymbol{x}\boldsymbol{x}^T\boldsymbol{v}_j\right\} = \|\boldsymbol{v}_j\|^2 = \mathbb{E}\left\{z_j^2\right\}$)

Fast ICA on the NLL (I)

- Whitening and centering are essentially used to reduce the computational complexity as they reduce the number of parameters from n^2 to $n(n-1)/2$
- Whitening can be obtained with a first application of PCA from where one can then apply any Fast ICA algorithm relying on the orthogonality of the matrix $\boldsymbol{W}$.
- There exists several algorithms to perform ICA. Here we focus on an algorithm that can be used to minimize the NLL.
- Fast ICA on the NLL can be considered a particular instance of a Newton method.

Fast ICA on the NLL (II)

- For the negative log-likelihood derived earlier, if we let $g = \frac{d}{dz} G(z)$, in the constrained framework, the contributions from each independent component to objective function, gradient and Hessian can respectively read as

$$f(\boldsymbol{v}) = \mathbb{E}\left\{G(\boldsymbol{v}^T\boldsymbol{x})\right\} + \lambda(1 - \boldsymbol{v}^T\boldsymbol{v})$$

$$\nabla f(\boldsymbol{v}) = \mathbb{E}\left\{\boldsymbol{x}g(\boldsymbol{v}^T\boldsymbol{x})\right\} - \beta\boldsymbol{v}$$

$$\boldsymbol{H}(\boldsymbol{v}) = \mathbb{E}\left\{\boldsymbol{x}\boldsymbol{x}^T g'(\boldsymbol{v}^T\boldsymbol{x})\right\} - \beta\boldsymbol{I}$$

  $\beta = 2\lambda$ is a Lagrange multiplier.

- If we make the approximation, $\mathbb{E}\left\{\boldsymbol{x}\boldsymbol{x}^T g'(\boldsymbol{v}^T\boldsymbol{x})\right\} \approx \mathbb{E}\left\{\boldsymbol{x}\boldsymbol{x}^T\right\}\mathbb{E}\left\{g'(\boldsymbol{v}^T\boldsymbol{x})\right\} = \mathbb{E}\left\{g'(\boldsymbol{v}^T\boldsymbol{x})\right\}$, The Hessian is easy to invert and we get the Newton step

$$\boldsymbol{v} \leftarrow \boldsymbol{v} - \frac{\mathbb{E}\left[\boldsymbol{x}g(\boldsymbol{v}^T\boldsymbol{x})\right] - \beta\boldsymbol{v}}{\mathbb{E}\left[g'(\boldsymbol{v}^T\boldsymbol{x})\right] - \beta}$$

Fast ICA on the NLL (III)

- After the Newton step has been applied, we simply project the resulting vector $\boldsymbol{v}$ onto the subspace orthogonal to the other independent components and normalize it.
- As the objective is non convex, there are multiple local minimas

Possible distributions

- As we have seen, Gaussian priors won't work well for ICA so what distributions can we use instead?
- There are several possible distributions one can use besides the Gaussian distribubtion:
  - Super-Gaussian distributions (e.g Laplace distribution). Super Gaussian distributions are distributions with a big spike at the mean and heavy tails. Generally speaking we say that a distribution is Super Gaussian when its kurtosis, $\text{kurt}(z) = \mu^4/\sigma^4 - 3$ is positive, $\text{kurt}(z) > 0$. Here $\mu_k = \mathbb{E}\left\{(X - \mathbb{E}(X))^k\right\}$
  - Sub-Gaussian distributions. (e.g. uniform distribution). Subgaussian distributions have negative kurtosis.
  - Skewed distributions (e.g. Gamma distribution). A distribution can be different from the Gaussian distribution by being assymetric. We define the skewness (measure of assymetry) of a distribution as $\text{skew}(z) = \mu^3/\sigma^3$.