MATHEMATICS INFERENCE AND MACHINE LEARNING

PINGCHUAN MA

Parameter Estimation

Definition.
Maximum Likelihood Estimation: A widely used approach to finding the desired parameters $\theta^\star$ is maximum likelihood

$$\max p(\mathbf{y}|\mathbf{X},\theta)$$

where $\mathbf{X} := [\mathbf{x_1}|...|\mathbf{x_N}]^T \in \mathbb{R}^{N\times D}$, $\mathbf{y} := [y_1,...,y_N]^T \in \mathbb{R}^{\mathbb{N}}$

Theorem.

$$\begin{aligned}\theta^\star &\in \arg\min_\theta(-logp(\mathbf{y}|\mathbf{X},\theta))\\ &= \arg\min_\theta(-log\prod_{i=1}^{N}p(y_i|\mathbf{x_i},\theta))\\ &= \arg\min_\theta\sum_{i=1}^{N}-logp(y_i|\mathbf{x_i},\theta)\end{aligned}$$

In the linear regression model, the likelihood is Gaussian

$$\begin{aligned}-logp(y_i|\mathbf{x_i},\theta) &= \frac{1}{2\sigma^2}(y_i-\mathbf{x_i}^T\theta)^2+const\\ L(\theta) &= -logp(\mathbf{y}|\mathbf{X},\theta)\\ &= \frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\theta)^T(\mathbf{y}-\mathbf{X}\theta)\\ &= \frac{1}{2\sigma^2}||\mathbf{y}-\mathbf{X}\theta||^2\end{aligned}$$

Taking the derivative w.r.t the parameter θ and set the gradient to 0, obtian

$$\begin{aligned}\frac{\partial L}{\partial\theta} &= \frac{1}{\sigma^2}(-\mathbf{y}^T\mathbf{X}+\theta\mathbf{X}^T\mathbf{X})\\ \theta^\star &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}\end{aligned}$$

Remark. We consider the linear regression model:

$$y = \phi^T(\mathbf{x}+\epsilon)$$

where $\phi:\mathbb{R}^D\to\mathbb{R}^K$ is a nonlinear transformation of the input $\mathbf{x}$, we can simply replace $\mathbf{X}$ by Φ, and obtain

$$\theta^\star = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{y}$$

Remark. In machine learning, the negative log likelihood function is also called an error function.

Definition.
The property of the polynomials fitting the noise structure is called overfitting.

Definition.
In regularization, a term is added to the log-likelihood that penalizes the amplitude of the parameter θ. Example of a loss function of the form

$$-logp(\mathbf{y}|\mathbf{X},\theta)+\lambda||\theta||_2^2$$

Definition.
Maximum A-Posterior Estimation: The parameter vector θ_{MAP} that maximizes the posterior.

Date: April 2017.

Theorem.
To find the MAP estimate. We start with the log-transform

$$logp(\theta|\mathbf{X},\mathbf{y}) = log(\mathbf{y}|\mathbf{X},\theta) + logp(\theta) + const$$

Remark. Choosing a Gaussian parameter prior $p(\theta) = N(0, b^2\mathbf{I}), b^2 = \frac{1}{2\lambda}$, the negative log-prior will be

$$-logp(\theta) = \lambda\theta^T\theta + const$$

That means for a quadratic regularizatoin, the regularization parameter λ in corresponds to twice the precision of the Gaussian prior $p(\theta)$. The log-prior plays the role of a regularizer that penalizes implausible values.

Theorem.
For the linear regression problem where

$$y = \phi^T(\mathbf{x})\theta + \epsilon, \quad \epsilon \in N(0, \sigma^2)$$

The negative log-posterior for this model

$$-logp(\theta|\mathbf{X},\mathbf{y}) = \frac{1}{2\sigma^2}(\mathbf{y} - \Phi\theta)^T(\mathbf{y} - \Phi\theta) + \frac{1}{2b^2}\theta^T\theta + const$$

To find θ_{MAP}, we take the derivative w.r.t the parameter θ and set the gradient to 0:

$$\frac{1}{\sigma^2}(\theta^T\Phi^T\Phi - \mathbf{y}^T\Phi) + \frac{1}{b^2}\theta^T = 0$$

$$\theta_{MAP} = (\Phi^T\Phi + \frac{\sigma^2}{b^2}\mathbf{I})^{-1}\Phi^T\mathbf{y}$$

Definition.
To find the local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient of the funciton at the current point.

$$\mathbf{x_{i+1}} = \mathbf{x_i} - \gamma_i(\nabla\mathbf{f})(\mathbf{x_i})$$

Definition.
Cross-validation: K-fold cross-validation effectively partitions the data into K chunks, K-1 of which form the training set $\tilde{D}$, and the last chunk serves as the validation set $\mathcal{V}$. The expected generalization error can be computed as follows.

$$\mathbb{E}_{\mathcal{V}}[G(\mathcal{V}|M)] \approx \frac{1}{K}\sum_{k=1}^{K} G(\mathcal{V}^{(k)}|M)$$

where $G(\mathcal{V}|M)$ is the generalization error on the validation set $\mathcal{V}$ for model M.

Definition.
Occam's Razor: finding the simplest model that explains the data reasonably well.

Feature Extraction

Definition.
Eigen-decomposition: Assume square matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ with n linearly independent eigenvectors $\boldsymbol{q}_i, i = 1, ..., n$ and n eigenvalues $\lambda_1, ..., \lambda_n$. Then $\boldsymbol{A}$ can be factorised as

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{-1}$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix whose diagonal elements are the corresponding eigenvalues, i.e., $\boldsymbol{\Lambda} = \text{diag}\{\lambda_1, ..., \lambda_n\}$.

Theorem.
if $\boldsymbol{A}$ is symmetric,

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T$$

Definition.
If $\mathbf{A} \in \mathbb{R}^{m\times n}$, then there exist orthogonal matrices

$$\mathbf{U} \in \mathbb{R}^{m\times m} \text{ and } \mathbf{V} \in \mathbb{R}^{n\times n}$$

such that

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^T$$

where $\boldsymbol{\Lambda}$ is a diagnosis matrices $\boldsymbol{\Lambda} \in \mathbb{R}^{m\times n}$

Theorem.
The 2-norm and Frobenius norm we have the following.

$$||\mathbf{A}||_F^2 = \sigma_1^2 + ... + \sigma_p^2, p = min\{m, n\}$$
$$||\mathbf{A}||_2 = \sigma_1$$

Theorem.
Let the SVD of $\mathbf{A} = \mathbf{U}\sum\mathbf{V}^T$. If $k < r = rank(\mathbf{A})$ and

$$\mathbf{A}_k = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$$

then

$$\min_{rank(\mathbf{B})=k} ||\mathbf{A} - \mathbf{B}||_2 = ||\mathbf{A} - \mathbf{A}_k||_2 = \sigma_{k+1}$$

Proof. Since, $\mathbf{U}^T\mathbf{A}_k\mathbf{V} = \text{diag}(\sigma_1, ..., \sigma_k, 0, ..., 0)$ then rank($\mathbf{A}_k$)=k and $\mathbf{U}(\mathbf{A} - \mathbf{A}_k)\mathbf{V} = \text{diag}(0, .., 0, \sigma_{k+1}, ..., \sigma_p)$. Hence, $||\mathbf{A} - \mathbf{A}_k||_2 = \sigma_{k+1}$. Now suppose rank($\mathbf{B}$)=$k$ for a $\mathbf{B} \in \mathbb{R}^{m\times n}$. It follows that

$$\text{span}\{\mathbf{q}_1, ..., \mathbf{q}_{n-k}\} \cap \text{span}\{\mathbf{v}_1, ..., \mathbf{v}_{k+1}\} \neq \{0\}$$

we have that

$$||\mathbf{A} - \mathbf{B}||_2^2 \geq ||\mathbf{A} - \mathbf{B}\mathbf{z}||_2^2 = ||\mathbf{A}\mathbf{z}||_2^2 \geq \sigma_{k+1} \sum_{i=1}^{k+1} \alpha_i^2 = \sigma_{k+1}$$

□

Remark. Assume we have a collection of n data samples $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$, we stack the samples as columns of matrix $=[\boldsymbol{x}_1, ..., \boldsymbol{x}_n]$. The k low-rank representation of the data is

$$\boldsymbol{X}_k = \boldsymbol{U}_k \boldsymbol{\Lambda}_k \boldsymbol{V}_k^T$$

Component Analysis

Theorem.
Principal component analysis performs a linear mapping of the data to a lower-dimensional space in such a way that the variance of the data in low-dimensional representation is maximized.

Theorem.
The optimal features $\{y_1^o, ..., y_n^o\} = \arg\max \frac{1}{2}\sigma_y^2$

$$\begin{aligned}
\mathbf{w}_o &= \arg\max_{\mathbf{w}} \frac{1}{2n} \sum_{i=1}^{n} (\mathbf{w}^T \mathbf{x}_i)^2 \\
&= \arg\max_{\mathbf{w}} \frac{1}{2n} \sum_{i=1}^{n} \mathbf{w}^T \mathbf{x}_i \mathbf{x}_i^T \mathbf{w} \\
&= \arg\max_{\mathbf{w}} \frac{1}{2} \mathbf{w}^T \mathbf{S}_t \mathbf{w}
\end{aligned}$$

Theorem.
In order to avoid the trivial solution we incorporate the constraint $||\mathbf{w}||_2 = 1$. The optimisation problem can be reformulated as

$$\begin{aligned}
\mathbf{w}_o &= \arg\max_{\mathbf{w}} \frac{1}{2} \mathbf{w}^T \mathbf{S}_t \mathbf{w} \\
\textbf{s.t.} \quad & \mathbf{w}^T \mathbf{w} = 1
\end{aligned}$$

This Lagrangian of the above optimisation problem is

$$L(\mathbf{w}, \lambda) = \frac{1}{2} \mathbf{w}^T \mathbf{S}_t \mathbf{w} - \lambda(\mathbf{w}^T \mathbf{w} - 1)$$

Forcing $\nabla_{\mathbf{w}} L(\mathbf{w}) = \mathbf{0}$, we end up to

$$\mathbf{S}_t \mathbf{w} = \lambda \mathbf{w}$$

Replacing it to the optimisation problem, we obtain

$$\boldsymbol{\lambda}_o = \arg\max_{\lambda} \lambda$$

Theorem.
Assume $\boldsymbol{y}_i \in \mathbb{R}^d$

$$\begin{aligned}\boldsymbol{W}_o &= \arg\max_{\boldsymbol{W}} \frac{1}{2}\sum_{k=1}^{d} \boldsymbol{w}_k^T \boldsymbol{S}_t \boldsymbol{w}_k \\ &= \arg\max_{\boldsymbol{W}} tr(\boldsymbol{W}^T \boldsymbol{S}_t \boldsymbol{W}) \\ &= \arg\max_{\boldsymbol{W}} tr(\boldsymbol{\Lambda}_d) \\ &= \sum_{k=1}^{d} \lambda_k\end{aligned}$$

where $\boldsymbol{S}_t = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$, and $\boldsymbol{W}^T\boldsymbol{S}_t\boldsymbol{W} = \boldsymbol{W}^T\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\boldsymbol{W}$.

Theorem.
Assume $\boldsymbol{B} = \boldsymbol{X}\boldsymbol{X}^T$ and $\boldsymbol{C} = \boldsymbol{X}^T\boldsymbol{X}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ have the same positive eigenvalues $\boldsymbol{\Lambda}$. Assume $N < F$, the eigenvectors of U and V can be represented as

$$\boldsymbol{U} = \boldsymbol{X}\boldsymbol{V}\boldsymbol{\Lambda}^{-1/2}$$

Proof. Assume $\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$

$$\begin{aligned}\boldsymbol{S}_t &= \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T \\ \boldsymbol{\Lambda} &= \boldsymbol{U}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{U} \\ \boldsymbol{\Lambda} &= \boldsymbol{\Lambda}^{-1/2}\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{\Lambda}^{-1/2}\end{aligned}$$

In which we can obtain $\boldsymbol{U} = \boldsymbol{X}\boldsymbol{V}\boldsymbol{\Lambda}^{-1/2}$. □

- Compute $\mathbf{X^T X} = [(\mathbf{x_i} - \mu)^T(\mathbf{x_j} - \mu)]$
- Perform eigenanalysis of $\mathbf{X^T X} = \mathbf{V\Lambda V^T}$
- Compute eigenvectors $\mathbf{U} = \mathbf{XV\Lambda^{-1/2}}$
- Compute d features $\mathbf{Y} = \mathbf{U^T X}$

Definition.
Considering whitening,

$$\boldsymbol{Y} = \boldsymbol{U}^T\boldsymbol{X}$$

we want to $\boldsymbol{Y}\boldsymbol{Y}^T = \boldsymbol{U}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{U} = \boldsymbol{\Lambda}$, therefore, we obtain

$$\boldsymbol{W} = \boldsymbol{U}\boldsymbol{\Lambda}^{-1/2}$$

Definition.
Linear Discriminant Analysis is a method to find a linear combination of features that characterizes or seperates two or more classes of objects or event. It can be formulated as follows.

$$\begin{aligned}\min \quad & \sigma_y^2(c_1) + \sigma_y^2(c_2) \\ \text{s.t.} \quad & (\mu_y(c_1) - \mu_y(c_2))^2\end{aligned}$$

Definition.
The formulation is given as follows.

$$\begin{aligned}\max \quad & \mathbf{w}^T\mathbf{S}_b\mathbf{w} \\ \text{s.t.} \quad & \mathbf{w}^T\mathbf{S}_w\mathbf{w} = 1\end{aligned}$$

Using lagrangian:

$$\begin{aligned}L(\mathbf{w}, \lambda) &= \mathbf{w}^T\mathbf{S}_b\mathbf{w} - \lambda(\mathbf{w}^T\mathbf{S}_w\mathbf{w} - 1) \\ \frac{\partial L}{\partial \mathbf{w}} &= 0 \\ \lambda\mathbf{S}_w\mathbf{w} &= \mathbf{S}_b\mathbf{w}\end{aligned}$$

Remark. PCA is an unsupervised approach for compression of data. Whitened PCA is a technique to make the input less redundant. Compared with PCA, whitened PCA can reduce the correlation among features. After whitening, each feature has the same variance. LDA is a supervised approach for compression of data. As all data is labelled, this method takes classification into account. It is different from PCA because PCA doesn't take labels into considerations. The maximal number of dimension is $C - 1$, where C is the number of labels.

Support Vector Machines

Definition.
The SVM techniques tries to find the separating hyperplane with the largest margin between two classes, measured along a line perpendicular to the hyperplane.

Remark. This means we find a line with parameters $\mathbf{w}$ and b such that the distance between $\boldsymbol{w}^T\boldsymbol{x} + b = \pm 1$

Theorem.
Assume a point $\tilde{x}$ on $\mathbf{w}^T\mathbf{x} + b = -1$, and we assume point $\tilde{x} + t\mathbf{w}$ touches the line $\mathbf{w}^T\mathbf{x} + b = 1$. Then, $t\mathbf{w}^T\mathbf{w} = 2$. So the length of $t\mathbf{w}$ is $||t\mathbf{w}||_2 = \frac{2}{||\mathbf{w}||_2}$. As maximizing $\frac{2}{||\mathbf{w}||_2}$ is equivalent to minimizing $\frac{\mathbf{w}^T\mathbf{w}}{2}$. So the equivalent problem to SVM is as follows.

$$\begin{aligned} \min_{w,b} \quad & \frac{\mathbf{w}^T\mathbf{w}}{2} \\ \text{s.t.} \quad & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \\ & i = 1, .., l \end{aligned}$$

Definition.
Constrained optimization problem is of the form

$$\begin{aligned} \min_{w} \quad & f(\mathbf{w}) \\ \text{s.t.} \quad & g(\mathbf{w}) \leq 0 \end{aligned}$$

We define the Lagrangian to be the original objective function added to a weighted combination of the constraints.

$$L(\mathbf{w}, a) = f(\mathbf{w}) + ag(\mathbf{w})$$

Theorem.
The original minimization problem can be written as

$$\min_{\mathbf{w}} \max_{\mathbf{a} \geq 0} L(\mathbf{w}, a)$$

Proof. $g(\mathbf{w}) \leq 0$, we maximize $L(\mathbf{w}, a)$ by setting $a = 0$. When $g(\mathbf{w}) \geq 0$, we can get $\max_{a \geq 0} L(\mathbf{w}, a) = \infty$. Minimizing the outer loop, we obtain the minimum value of $f(\mathbf{w})$ such that $g(\mathbf{w}) \leq 0$ holds. □

Definition.
The primal solution to the problem is given by

$$\mathbf{p}^\star = \min_{\mathbf{w}} \max_{\mathbf{a} \geq 0} L(\mathbf{w}, a)$$

The dual solution to the problem is given by

$$\mathbf{d}^\star = \max_{\mathbf{a} \geq 0} \min_{\mathbf{w}} L(\mathbf{w}, a)$$

Definition.
$d^\star \leq p^\star$. Let $\mathbf{w}^\star$ be the $\mathbf{w}$ value that corresponds to the optimal primal solution $p^\star$. We can write for all $a \geq 0$

$$\max_{\tilde{a} \geq 0} L(\mathbf{w}^\star, \tilde{a}) \geq L(\mathbf{w}^\star, a) \geq \min_{\mathbf{w}} L(\mathbf{w}, a)$$

Theorem.
If certain conditions are met, namely

- $f(\mathbf{w})$ is convex
- $g(\mathbf{w})$ is affine

then $d^\star = p^\star$

Definition.
Support vector regression sloves the following optimization problem:

$$\begin{aligned} \min_{w,b,\xi,\xi^\star} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}(\xi_i + \xi_i^\star) \\ \text{s.t.} \quad & (\mathbf{w}^T\phi(\mathbf{x}_i) + b) - y_i \leq \epsilon + \xi_i \\ & y_i - (\mathbf{w}^T\phi(\mathbf{x}_i) + b) \leq \epsilon + \xi_i^\star \\ & \xi_i, \xi_i^\star \geq 0, \quad i = 1, ..., l \end{aligned}$$

Slack variables ξ_i is the upper training error ($\xi_i^\star$ is the lower) subject to the ϵ-intensitive tube $|y_i-(\mathbf{w}^T\phi(\mathbf{x}_i))+b| \leq \epsilon$, $\boldsymbol{w}^T\boldsymbol{w}$ is added to smooth the function.

Remark. Linear regression finds a linear function $\boldsymbol{w}^T\boldsymbol{x}+b$ so that ($\boldsymbol{w}$, b) is an optimal solution of

$$\min_{w,b} \sum_{i=1}^{n} (y_i - (\boldsymbol{w}^T\boldsymbol{x}_i + b))^2$$

If the data is nonlinearly distributed, a linear function is not enough, and we also want to map data to higher dimensional space by a function $\phi(\boldsymbol{x})$. Therefore $F \leq$ dimensionality of $\phi(\boldsymbol{x})$ so again overfitting happens. Support Vector Regression can remedy the overfitting problem in these two strategies.

- A threshold ϵ is given, if the point($\boldsymbol{x}_i$, y_i) is located in the threshold, then $\xi_i, \xi_i^\star = 0$.
- To reduce the model complexity and smooth the function, an additional term $\boldsymbol{w}^T\boldsymbol{w}$ is added to the objective function.

The difference between linear regression and SVM:

- loss function is different
  LR:
  $$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} log h_\theta(x^{(i)}) + (1-y^{(i)}) log(1-h_\theta(x^{(i)}))]$$
  SVM:
  $$\mathcal{L}(w,b,\alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i (y_i(w^T x_i + b) - 1)$$
  LR:
  $$\min_w \lambda||w||_2 + \sum_i log(1 + exp^{1-y_i w^T x_i})$$
  SVM:
  $$\min_w \lambda||w||_2 + \sum_i \max\{0, 1 - y_i w^T x_i\}$$
- SVM considers the point around the optimal hyperplane while LR considers all points.
- As for solving nonlinear problems, SVM adapt kernel functions while LR don't because of the high complexity.
- SVM has a regularizer.

Kernel PCA

Theorem.
In kernel Principal Component Analysis, $\phi(\cdot)$ may not be explicitly known or is expensive to compute and store. What we explicitly known is the dot product in H.

$$\phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j) = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

Remark. All positive semi-definite functions can be used as kernels.

- Gaussian Radial Basis Function (RBF) kernel:
  $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2/r^2}$
- Polynomial kernel:
  $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i^T\boldsymbol{x}_j + b)^n$
- Hyperbolic Tangent kernel:
  $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = tanh(\boldsymbol{x}_i^T\boldsymbol{x}_j + b)$

Clustering

Definition.
k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean.

Theorem.
Given a set of observations $(\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n})$, where each observation is a d-dimensional real vector. The objective is to find:

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} ||\mathbf{x} - \mu_i||^2 = \arg\min_{S} \sum_{i=1}^{k} |S_i| Var S_i$$

The process is as follows.
Assignment step:

$$S_i^{(t)} = \{x_p : ||x_p - m_i^{(t)}||^2 \leq ||x_p - m_j^{(t)}||^2, 1 \leq j \leq k\}$$

Update step:

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

Definition.
An expectation-maximization (EM) algorithm is an iterative method to find maximum likelihood or maximum a posterior estimates of parameters, where the model depends on latent variables.

Definition.
For Gaussian Mixture Models,

$$\begin{aligned} p(\boldsymbol{X}, \boldsymbol{Z}|\theta) &= p(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N, \boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}|\theta) \\ &= \prod_{n=1}^{N} p(\boldsymbol{x}_n|\boldsymbol{z}_n, \theta_x) \prod_{n=1}^{N} p(\boldsymbol{z}_n|\theta_z) \\ &= \prod_{n=1}^{N} \prod_{k=1}^{3} N(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \sum\nolimits_k)^{z_{nk}} \prod_{n=1}^{N} \prod_{k=1}^{3} \pi_k^{z_{nk}} \end{aligned}$$

$$lnp(\boldsymbol{X}, \boldsymbol{Z}|\theta) = \sum_{n=1}^{N} \sum_{k=1}^{3} z_{nk} \{lnN(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \sum\nolimits_k) + ln\pi_k\}$$

Applying operator $E_{p(\boldsymbol{Z}|\boldsymbol{X},\theta)}$

$$\begin{aligned} &E_{p(\boldsymbol{Z}|\boldsymbol{X},\theta)}[lnp(\boldsymbol{X}, \boldsymbol{Z}|\theta)] \\ &= \sum_{n=1}^{N} \sum_{k=1}^{3} E_{p(\boldsymbol{Z}|\boldsymbol{X},\theta)}[z_{nk}] \{lnN(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \sum\nolimits_k) + ln\pi_k\} \\ &G(\theta) = E_{p(\boldsymbol{Z}|\boldsymbol{X},\theta)}[lnp(\boldsymbol{X}, \boldsymbol{Z}|\theta)] \\ &= \sum_{n=1}^{N} \sum_{k=1}^{3} \gamma(z_{nk}) \{-\frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \sum\nolimits_k^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}_k) \\ &\quad - \frac{1}{2}(Fln2\pi + ln|\sum\nolimits_k|) + ln\pi_k\} \end{aligned}$$

Theorem.

$$
\begin{aligned}
G(\theta) &= E_{p(Z|X,\theta)}[\ln p(X,Z|\theta)] \\
&= \sum_{n=1}^{N}\sum_{k} E_{p(Z|X,\theta)}[z_{nk}] ln N(x_n|\mu_k, \sum\nolimits_k) + ln\pi_k \\
&\frac{dG(\theta)}{d\boldsymbol{\mu}_k} = 0 \quad \frac{dG(\theta)}{d\sum_k} = 0 \\
\pi_k &= \frac{\sum_{n=1}^{N}\gamma(z_{nk})}{N} \\
\mu_k &= \frac{\sum_{n=1}^{N}\gamma(z_{nk})x_n}{\sum_{n=1}^{N}\gamma(z_{nk})} \\
\sum\nolimits_k &= \frac{\sum_{n=1}^{N}\gamma(z_{nk})(x_n-\mu_k)(x_n-\mu_k)^T}{\sum_{n=1}^{N}\gamma(z_{nk})}
\end{aligned}
$$

Probabilistic Principal Component Analysis

Markov Chains

Definition.
Given a set of observations $\boldsymbol{D}_l = \{x_1^l, x_2^l, ..., x_T^l\}, l = 1, ..., N$, we need to find the parameters $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{A}\}$ that maximize $p(\boldsymbol{D}_1, \boldsymbol{D}_2, ..., \boldsymbol{D}_N|\boldsymbol{\theta})$

Theorem.

$$
\begin{aligned}
p(\boldsymbol{D}_1, \boldsymbol{D}_2, ..., \boldsymbol{D}_N|\theta) &= \prod_{l=1}^{N} p(\boldsymbol{D}_l|\boldsymbol{\theta}) \\
&= \prod_{l=1}^{N} p(\boldsymbol{x}_1^l) \prod_{t=2}^{T} p(\boldsymbol{x}_t^l|\boldsymbol{x}_{t-1}^l) \\
&= \prod_{l=1}^{N}\prod_{k=1}^{5} \pi_k^{x_{1k}^l} \prod_{t=2}^{T}\prod_{j=1}^{5}\prod_{k=1}^{5} a_{jk}^{x_{t-1j}^l x_{tk}^l} \\
\ln p(\boldsymbol{D}_1, \boldsymbol{D}_2, ..., \boldsymbol{D}_N|\boldsymbol{\theta}) &= \sum_{l=1}^{N}\sum_{k=1}^{5} x_{1k}^l ln\pi_k + \\
&\quad \sum_{l=1}^{N}\sum_{t=2}^{T}\sum_{j=1}^{5}\sum_{k=1}^{5} x_{1k}^l ln a_{jk} \\
&= \sum_{k=1}^{5}\sum_{l=1}^{N} x_{1k}^l ln\pi_k + \\
&\quad \sum_{j=1}^{5}\sum_{k=1}^{5}\sum_{l=1}^{N}\sum_{t=2}^{T} x_{1k}^l ln a_{jk}
\end{aligned}
$$

Hidden Markov Model

Definition.
Hidden Markov model is a statistical Markov model in which the system begin modeled in assumed to be a Markov process with hidden states.

Definition.
Filtering: $p(\boldsymbol{z}_t|\boldsymbol{x}_1, ..., \boldsymbol{x}_t) = \frac{p(\boldsymbol{x}_1,...,\boldsymbol{x}_t,\boldsymbol{z}_t)}{p(\boldsymbol{x}_1,...,\boldsymbol{x}_t)}$
Smoothing: $p(\boldsymbol{z}_t|\boldsymbol{x}_1, ..., \boldsymbol{x}_T)$

Decoding: $\arg \max_{\boldsymbol{y}_1,...,\boldsymbol{y}_t} p(\boldsymbol{y}_1, ..., \boldsymbol{y}_t | \boldsymbol{x}_1, ..., \boldsymbol{x}_t)$

Evaludation: $p(\boldsymbol{x}_1, ..., \boldsymbol{x}_T)$

Theorem.
The parameters emission probability $p(\boldsymbol{x}|\boldsymbol{z})$, probability for the first timestamp $p(\boldsymbol{x}_1)$ and transition matrix $\boldsymbol{A}$ are obtained by maximizing the probability. After taking the expectation with regard to the posterior

$$\begin{aligned} G(\theta) = &\sum_{l=1}^{N}\sum_{t=1}^{T}\sum_{j=2}^{5}\sum_{k=1}^{K} x_{tj}^l E[z_{tk}^l] ln b_{jk} + \sum_{l=1}^{N}\sum_{k=1}^{K} E[z_{1k}^l] ln \pi_k \\ &+ \sum_{l=1}^{N}\sum_{t=2}^{T}\sum_{j=1}^{5}\sum_{k=1}^{K} E[z_{t-1j}^l z_{tk}^l] ln a_{jk} \end{aligned}$$

$$\begin{aligned} E[z_{1k}^l] &= \sum_{z_{1k}^l} z_{1k}^l p(z_{1k^l=1|\boldsymbol{x}_1^l,...,\boldsymbol{x}_T^l}) = \frac{\alpha(z_{1k}^l)\beta(z_{1k}^l)}{p(\boldsymbol{x}_1^l, ..., \boldsymbol{x}_T^l)} \\ E[z_{tk}^l] &= \sum_{z_{tk}^l} z_{tk}^l p(z_{tk^l=1|\boldsymbol{x}_1^l,...,\boldsymbol{x}_T^l}) = \frac{\alpha(z_{tk}^l)\beta(z_{tk}^l)}{p(\boldsymbol{x}_1^l, ..., \boldsymbol{x}_T^l)} \\ E[z_{t-1j}^l z_{tk}^l] &= \sum_{z_{t-1j}^l}\sum_{z_{tk}^l} z_{t-1j}^l z_{tk}^l p(z_{t-1j}^l = 1, z_{tk}^l = 1 | \boldsymbol{x}_1^l, ..., \boldsymbol{x}_T^l) \\ &= \frac{a(z_{t-1j}^l)\prod_{r=1}^{5} b_{kr}^{x_{tr}^l} a_{jk}\beta(z_{tk}^l)}{p(\boldsymbol{x}_1^l, ..., \boldsymbol{x}_T^l)} \end{aligned}$$

The Lagrangian can be applied.

$$\begin{aligned} L(\theta) = &\sum_{l=1}^{N}\sum_{t=1}^{T}\sum_{j=2}^{5}\sum_{k=1}^{K} x_{tj}^l E[z_{tk}^l] ln b_{jk} + \sum_{l=1}^{N}\sum_{k=1}^{K} E[z_{1k}^l] ln \pi_k \\ &+ \sum_{l=1}^{N}\sum_{t=2}^{T}\sum_{j=1}^{5}\sum_{k=1}^{K} E[z_{t-1j}^l z_{tk}^l] ln a_{jk} \\ &+ \lambda(\sum_{j=1}^{5} b_{jk} - 1) + \gamma(\sum_{k=1}^{K} \pi_k - 1) + \delta(\sum_{k=1}^{K} a_{jk} - 1) \end{aligned}$$

which gives us

$$\begin{aligned} b_{jk} &= \frac{\sum_{l=1}^{N}\sum_{t=1}^{T} E[z_{tk}^l] x_{tj}^l}{\sum_{l=1}^{N}\sum_{t=1}^{T} E[z_{tk}^l]} \\ \pi_k &= \frac{\sum_{l=1}^{N} E[z_{1k}^l]}{\sum_{l=1}^{N}\sum_{r=1}^{K} E[z_{1r}^l]} \\ a_{jk} &= \frac{\sum_{l=1}^{N}\sum_{t=2}^{T} E[z_{t-1j}^l z_{tk}^l]}{\sum_{r=1}^{K}\sum_{l=1}^{N}\sum_{t=2}^{T} E[z_{t-1j}^l z_{tr}^l]} \end{aligned}$$

Markov Random Fields

Definition.
Local Markovianity: given its neighborhood, a variable is indepdent on the rest of the variables.

$$p(x_i | x_{V\setminus\{i\}}) = p(x_i | x_{N(i)})$$

Definition.
Global Markovianity: Let A, B, C be three disjoint subset of V. If C seperates A from $B \rightarrow$

$$p(\mathbf{x}_A, \mathbf{x}_B | \mathbf{x}_C) = p(\mathbf{x}_A | \mathbf{x}_C) p(\mathbf{x}_B | \mathbf{x}_C)$$

Theorem.
If the field has the local Markov property, then $p(x)$ can be written as a Gibbs distribution.

$$p(x) = \frac{1}{Z}\prod_c \Psi_c(\mathbf{x}_c) = \frac{1}{Z}e^{-\sum_C V_C(\mathbf{x}_c)}$$

Definition.
A random vector $\mathbf{x} = (x_1, ..., x_N)^T$ is called a GMRF wrt the graph $G = V, E$ with mean μ and a positive semi-definite precision matrix $\mathbf{Q}$ its density has the form

$$p(\mathbf{x}) = \frac{|\mathbf{Q}|}{(2\pi)^{N/2}} exp(-\frac{1}{2}(\mathbf{x}-\mu)^T\mathbf{Q}(\mathbf{x}-\mu))$$

Linear Dynamical Systems

$\boldsymbol{x}_t = \boldsymbol{W}\boldsymbol{y}_t + \boldsymbol{e}_t$ $\quad \boldsymbol{e} \sim N(\boldsymbol{e}|\mathbf{0}, \sum)$
$\boldsymbol{y}_1 = \boldsymbol{\mu}_0 + \boldsymbol{u}$ $\quad \boldsymbol{u} \sim N(\boldsymbol{\mu}|\mathbf{0}, \boldsymbol{P}_0)$
$\boldsymbol{y}_t = \boldsymbol{A}\boldsymbol{y}_{t-1} + \boldsymbol{v}_t$ $\quad \boldsymbol{v} \sim N(\boldsymbol{v}|\mathbf{0}, \boldsymbol{\Gamma})$

where $\boldsymbol{y}_i$ is the latent variable corresponding to $\boldsymbol{x}_i$. The parameters required to be determined is $\theta = \{\boldsymbol{W}, \boldsymbol{A}, \boldsymbol{\mu}_0, \sum, \boldsymbol{\Gamma}, \boldsymbol{P}_0\}$
Assuming all these variables follows Gaussian distribution, the process is described as follows.

$$p(\boldsymbol{y}_1) = N(\boldsymbol{y}_1|\boldsymbol{\mu}_0, \boldsymbol{P}_0)$$
$$p(\boldsymbol{y}_t|\boldsymbol{y}_{t-1}) = N(\boldsymbol{y}_t|\boldsymbol{A}\boldsymbol{y}_{t-1}, \boldsymbol{\Gamma})$$
$$p(\boldsymbol{x}_t|\boldsymbol{y}_t) = N(\boldsymbol{x}_t|\boldsymbol{W}\boldsymbol{y}_t, \sum)$$

The difference between HMM and LDS: a Markov Chain with **discrete** latent variables ($\boldsymbol{\pi}$, $\boldsymbol{A}$, $\boldsymbol{B}$) and a Markov Chain with **continuous** latent variables.