MACHINE LEARNING II, LECTURE NOTES

José Manuel de Frutos Porras
Universidad Carlos III de Madrid
jofrutos@ing.uc3m.es

1 Probabilistic models for discrete data

Check [1, Sections 2.3.1, 2.3.2, 2.4.5, 2.5.4, 3.1-3.4], [2, Sections 2.1,2.2, 2.4]

1.1 Likelihood and Prior

In Bayesian inference, the likelihood and prior are two fundamental components used to update our beliefs about a hypothesis h given observed data D. From this point onward, we will assume that our **data are independently and identically distributed** (i.i.d.).

Likelihood

The **likelihood**, $p(D|h)$, represents the probability of observing the data D given that a particular hypothesis h is true. It is derived from the generative process that produces the data under the assumption of the hypothesis. Formally, if $D = \{x_1, x_2, \ldots, x_N\}$ are the observed data points, and θ represents the parameters of the hypothesis h, then the likelihood function can be expressed as:

$$p(D|\theta) = \prod_{i=1}^{N} p(x_i|\theta)$$

Prior

The **prior**, $p(h)$, reflects our initial belief about the probability of the hypothesis h before observing any data. It encodes any pre-existing knowledge or assumptions we might have about the hypothesis. The prior can be informed by domain knowledge, previous experiments, or subjective judgment.

The prior plays a crucial role in the Bayesian framework, especially in cases where data is sparse. It helps in regularizing the inference process, preventing the model from overfitting to the observed data.

Bayes' theorem

Bayes' theorem combines the prior and the likelihood to update our belief about the hypothesis after observing the data. The result is the **posterior distribution**, which represents the updated belief:

$$p(h|D) = \frac{p(D|h)p(h)}{p(D)}$$

In this equation, $p(D)$ is the marginal likelihood, also known as evidence, which ensures that the posterior distribution is properly normalized. Since $p(D)$ is constant, it can be disregarded in maximization/minimization scenarios, such as when computing the Maximum A Posteriori (MAP) estimate, where we use:

$$p(h|\mathcal{D}) \propto p(\mathcal{D}|h)p(h)$$

Maximum a posteriori (MAP) estimation

The **MAP estimate** is the value of the parameter θ that maximizes the posterior distribution $p(\theta|D)$. It is given by:

$$\theta_{\text{MAP}} = \arg\max_{\theta} p(\theta|D) = \arg\max_{\theta} [p(D|\theta)p(\theta)]$$

The MAP estimate incorporates both the prior information and the likelihood of the observed data. It is particularly useful when we have prior knowledge about the parameter θ.

Maximum likelihood (ML) estimation

The **ML estimate** is the value of the parameter θ that maximizes the likelihood function $p(D|\theta)$. It is given by:

$$\theta_{\text{ML}} = \arg\max_{\theta} p(D|\theta)$$

The ML estimate does not consider the prior distribution and relies solely on the observed data. It is a common approach when no prior information is available or when we want to estimate the parameter based purely on the data.

Note that the MAP estimate can be written as

$$\hat{h}_{MAP} = \arg\max_{h} p(D|h)p(h) = \arg\max_{h} [\log p(D|h) + \log p(h)]$$

In other words, if we have enough data, we see that the data overwhelms the prior. In this case, the MAP estimate converges towards the MLE.

Posterior Predictive Distribution

The **posterior predictive distribution** is used to make predictions about future data points based on the posterior distribution of the hypothesis. It represents our updated belief about what data we are likely to observe after having seen the initial dataset D.

Definition. *Given a new data point $\tilde{x}$, the posterior predictive distribution is given by:*

$$p(\tilde{x}|D) = \int p(\tilde{x}|\theta)p(\theta|D)d\theta$$

Here, $p(\tilde{x}|\theta)$ is the likelihood of the new data point given the parameter θ, and $p(\theta|D)$ is the posterior distribution over the parameters θ after observing the data D. This integral averages the predictions for $\tilde{x}$ over all possible hypotheses weighted by their posterior probabilities.

Sufficient statistic

In statistical inference, a **sufficient statistic** is a function of the data that provides as much information about the parameter of interest as the entire dataset itself. Formally, consider a random sample $X_1, X_2, \ldots, X_n$ drawn from a probability distribution that depends on a parameter θ. Let the joint probability density function (pdf) be denoted by $p(X_1, X_2, \ldots, X_n|\theta)$.

A statistic $S(X_1, X_2, \ldots, X_n) = S(\mathbf{X})$ is said to be a **sufficient statistic** for the parameter θ if the conditional distribution of the sample $\mathbf{X}$ given the statistic $S(\mathbf{X})$ does not depend on θ. Mathematically, $S(\mathbf{X})$ is sufficient for θ if:

$$p(\mathbf{X}|\theta, T(\mathbf{X})) = p(\mathbf{X}|S(\mathbf{X}))$$

This condition implies that once the value of the sufficient statistic $T(\mathbf{X})$ is known, the sample $\mathbf{X}$ provides no additional information about the parameter θ.

The **Neyman-Fisher factorization theorem** provides a practical method to check whether a statistic is sufficient. See [3, Theorem 2.2].

1.2 Binary data - The Beta-Binomial Model

The Beta-Binomial model is a fundamental concept in Bayesian statistics, especially useful for modeling binary outcomes, such as coin flips, where the underlying probability of success (e.g., the probability of getting heads) is unknown.

Likelihood

Consider a scenario where we have a series of N Bernoulli trials (e.g., coin flips), and we want to infer the probability θ of success (e.g., getting heads). Let $X_i \sim \text{Ber}(\theta)$ be the outcome of each trial, where $X_i = 1$ represents a success and $X_i = 0$ represents a failure.

Given a dataset $D = \{X_1, X_2, \ldots, X_N\}$, where N_1 is the number of successes and N_0 is the number of failures, the likelihood function is given by:

$$p(D|\theta) = \theta^{N_1}(1-\theta)^{N_0}$$

Alternatively, if we consider the data as the count of successes in N trials, then $N_1 \sim \text{Bin}(N, \theta)$, and the likelihood is:

$$p(D|\theta) = \binom{N}{N_1}\theta^{N_1}(1-\theta)^{N_0}$$

Since the binomial coefficient is a constant independent of θ. Therefore, any inferences we make about θ will be the same whether we observe the counts, $D = (N_1, N)$, or a sequence of trials, $D = \{x_1, \ldots, x_N\}$.

Prior

When the prior and the posterior have the same form, we say that the prior is a **conjugate prior** for the corresponding likelihood. Conjugate priors are widely used because they simplify computation, and are easy to interpret,

The conjugate prior for the Bernoulli likelihood is the Beta distribution, defined as:

$$\theta \sim \text{Beta}(a, b)$$

where a and b are the hyperparameters of the prior, representing our prior beliefs about the number of successes and failures, respectively. The Beta distribution is given by:

$$\text{Beta}(\theta|a, b) \propto \theta^{a-1}(1-\theta)^{b-1}$$

Posterior

After observing the data D, we update our belief about θ using Bayes' theorem, which combines the likelihood and the prior to obtain the posterior distribution:

$$p(\theta|D) \propto p(D|\theta)p(\theta)$$

For the Beta-Binomial model, the posterior distribution is also a Beta distribution, given by:

$$\theta|D \sim \text{Beta}(a + N_1, b + N_0)$$

Here, the posterior is simply the Beta distribution with updated parameters:

$$a_{\text{post}} = a + N_1, \quad b_{\text{post}} = b + N_0$$

Posterior Mean and Mode

The posterior mean, which is often used as a point estimate for θ, is given by:

$$\mathbb{E}[\theta|D] = \frac{a + N_1}{a + b + N}$$

The posterior mode (MAP estimate) is:

$$\theta_{\text{MAP}} = \frac{a + N_1 - 1}{a + b + N - 2}$$

When using a uniform prior (i.e., $a = 1$, $b = 1$), the MAP estimate reduces to the maximum likelihood estimate (MLE):

$$\theta_{\text{MLE}} = \frac{N_1}{N}$$

Posterior Predictive Distribution

The posterior predictive distribution is used to make predictions about future observations based on the current posterior. For a future Bernoulli trial, the probability of success is given by:

$$p(\tilde{X} = 1|D) = \mathbb{E}[\theta|D] = \frac{a + N_1}{a + b + N}$$

This can be generalized to predict the number of successes x in M future trials, which follows a Beta-Binomial distribution:

$$p(x|D, M) = \binom{M}{x}\frac{B(x + a_{\text{post}}, M - x + b_{\text{post}})}{B(a_{\text{post}}, b_{\text{post}})}$$

where $B(\cdot, \cdot)$ is the Beta function.

1.3 Dirichlet distribution

The Dirichlet distribution is a family of continuous multivariate probability distributions parameterized by a vector of positive reals. It is the conjugate prior of the categorical and multinomial distributions, making it particularly useful in Bayesian inference for discrete probability distributions.

Definition

The Dirichlet distribution is defined over a probability simplex, which means it is a distribution over K-dimensional vectors $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$, where each $\theta_i \geq 0$ and $\sum_{i=1}^{K} \theta_i = 1$. The probability density function (pdf) of the Dirichlet distribution is given by:

$$\text{Dir}(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{i=1}^{K} \theta_i^{\alpha_i - 1}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_K)$ is a vector of positive concentration parameters, and $B(\boldsymbol{\alpha})$ is the multivariate Beta function, defined as:

$$B(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}$$

Here, $\Gamma(\cdot)$ denotes the Gamma function.

Interpretation of parameters

Each α_i parameter can be interpreted as a prior count associated with the outcome i. The larger the value of α_i, the more the distribution is skewed towards θ_i being larger. When all $\alpha_i = 1$, the Dirichlet distribution is uniform over the simplex. When $\alpha_i > 1$, the distribution is biased towards the corresponding component θ_i. When $0 < \alpha_i < 1$, the distribution favors sparsity, meaning that θ_i is more likely to be close to zero.

The sum of the parameters, $\alpha_0 = \sum_{i=1}^{K} \alpha_i$, is referred to as the concentration parameter. A larger α_0 indicates a stronger belief in the prior distribution.

1.4 Categorical data

We generalize the previous results to infer the probability that a dice with K sides comes up as face k.

Likelihood

Suppose we observe N dice rolls, $D = \{x_1, \ldots, x_N\}$, where $x_i \in \{1, \ldots, K\}$. If we assume the data is independent and identically distributed (iid), the likelihood has the form

$$p(D|\boldsymbol{\theta}) = \prod_{k=1}^{K} \theta_k^{N_k},$$

where $N_k = \sum_{i=1}^{N} I(x_i = k)$ is the number of times event k occurred (these are the sufficient statistics for this model). The likelihood function $p(D|\boldsymbol{\theta})$ represents the probability of observing the data D given the parameter vector $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$, where θ_k is the probability of observing outcome k in a single dice roll, so $\sum_{i=1}^{K} \theta_i = 1$.

Prior

Since the parameter vector lives in the K-dimensional probability simplex (i.e. $\sum_{i=1}^{K} \theta_i = 1$), we need a prior that has support over this simplex. Ideally, it would also be conjugate. Fortunately, the Dirichlet distribution satisfies both criteria. So we will use the following prior:

$$\text{Dir}(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1} I(x \in S_K)$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$ is the vector of concentration parameters, $B(\boldsymbol{\alpha})$ is the multivariate Beta function, and S_K represents the K-dimensional probability simplex.

Posterior

Multiplying the likelihood by the prior, we find that the posterior is also Dirichlet:

$$\begin{aligned} p(\boldsymbol{\theta}|D) &\propto p(D|\boldsymbol{\theta})p(\boldsymbol{\theta}) \\ &\propto \prod_{k=1}^{K} \theta_k^{N_k} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1} \\ &= \prod_{k=1}^{K} \theta_k^{\alpha_k + N_k - 1} \\ &= \text{Dir}(\boldsymbol{\theta}|\alpha_1 + N_1, \ldots, \alpha_K + N_K) \end{aligned}$$

Thus, the posterior distribution is also a Dirichlet distribution with updated parameters $\alpha_k + N_k$ for each k.

Properties

- Maximum A Posteriori (MAP):

$$\hat{\theta}_k^{\text{MAP}} = \frac{N_k + \alpha_k - 1}{N + \alpha_0 - K} \tag{1}$$

- Maximum Likelihood (ML):

$$\hat{\theta}_{\text{ML}} = \underset{\theta}{\text{argmax}}\, p(\mathcal{D}|\theta) = \frac{N_k}{N} \tag{2}$$

- The posterior predictive that gives the probability of song x from genre k in a new top list is:

$$p(x = k|\mathcal{D}) = \frac{N_k + \alpha_k}{N + \alpha_0}$$

Proof.

We use Lagrange multiplier with the added condition that $\sum_k \theta_k = 1$. See [1, Section 3.2.1] □

2 Probabilistic models for continuous Data

[1, Sections 4.3, 4.4, 4.5]

2.1 Gaussian pdf

The probability density function (pdf) for a multivariate Gaussian (MVN) in D dimensions is defined as:

$$N(x|\mu, \Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

where μ is the mean vector, and Σ is the covariance matrix. The term inside the exponent is known as the Mahalanobis distance, which measures the distance of a data point x from the mean vector μ, scaled by the covariance structure.

To interpret the Mahalanobis distance, we can decompose Σ as $\Sigma = U\Lambda U^T$, where U contains the eigenvectors and Λ is a diagonal matrix of eigenvalues. This allows us to express the inverse covariance as:

$$\Sigma^{-1} = U\Lambda^{-1}U^T = \sum_{i=1}^{D} \frac{1}{\lambda_i} u_i u_i^T$$

where u_i and λ_i are the eigenvectors and eigenvalues of Σ. Thus, the Mahalanobis distance can be written as a weighted sum of squared projections:

$$(x-\mu)^T\Sigma^{-1}(x-\mu) = \sum_{i=1}^{D} \frac{y_i^2}{\lambda_i}$$

where $y_i = u_i^T(x - \mu)$. In two dimensions, this forms the equation of an ellipse:

$$\frac{y_1^2}{\lambda_1} + \frac{y_2^2}{\lambda_2} = 1$$

In the MVN, the eigenvectors define the orientation of these elliptical contours, and the eigenvalues determine their elongation. The Mahalanobis distance thus represents Euclidean distance in a transformed space, centered by μ and rotated by U, aligned with the main directions of data variance.

2.2 Jointly Gaussian Distributions

Inference in jointly Gaussian distributions involves computing the marginals and conditionals from a joint Gaussian distribution, $p(x_1, x_2)$.

Theorem (Marginals and Conditionals of an MVN)**.** *Suppose $x = (x_1, x_2)$ is jointly Gaussian with parameters:*

$$\mu = \begin{bmatrix}\mu_1 \\ \mu_2\end{bmatrix}, \quad \Sigma = \begin{bmatrix}\Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22}\end{bmatrix}, \quad \Lambda = \Sigma^{-1} = \begin{bmatrix}\Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22}\end{bmatrix}$$

Then the marginals are given by:

$$\begin{cases} p(x_1) = N(x_1|\mu_1, \Sigma_{11}), \\ p(x_2) = N(x_2|\mu_2, \Sigma_{22}). \end{cases}$$

and the posterior conditional is given by:

$$p(x_1|x_2) = N(x_1|\mu_{1|2}, \Sigma_{1|2})$$

where

$$\mu_{1|2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2) = \mu_1 - \Lambda_{11}^{-1}\Lambda_{12}(x_2 - \mu_2) = \Sigma_{1|2}(\Lambda_{11}\mu_1 - \Lambda_{12}(x_2 - \mu_2))$$

and

$$\Sigma_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} = \Lambda_{11}^{-1}$$

Proof. The proof relies on algebraic manipulations and parameter recognition. See [2, Section 2.3.2]. □

Both the marginal and conditional distributions are Gaussian. For the marginals, we simply extract the rows and columns associated with x_1 or x_2. For the conditional distribution, additional steps are required: the conditional mean is a linear function of x_2, while the conditional covariance is a constant matrix, independent of x_2. We provide three equivalent expressions for the posterior mean and two for the posterior covariance, each suited to different applications.

2.3 Linear Gaussian systems

Suppose we have two variables, x and y, where $x \in \mathbb{R}^{D_x}$ is a hidden variable and $y \in \mathbb{R}^{D_y}$ is a noisy observation of x. The prior and likelihood are defined as:

$$p(x) = N(x|\mu_x, \Sigma_x), \quad p(y|x) = N(y|Ax + b, \Sigma_y) \tag{3}$$

where A is a $D_y \times D_x$ matrix. This setup represents a linear Gaussian system, schematically shown as $x \to y$, meaning x generates y. We will show how to infer x from y.

Theorem (Bayes Rule for Linear Gaussian Systems)**.** *Given a linear Gaussian system, as in Equation 3, the posterior $p(x|y)$ is given by the following:*

$$p(x|y) = N(x|\mu_{x|y}, \Sigma_{x|y})$$

where

$$\Sigma_{x|y}^{-1} = \Sigma_x^{-1} + A^T\Sigma_y^{-1}A \quad \textit{and} \quad \mu_{x|y} = \Sigma_{x|y}\left(A^T\Sigma_y^{-1}(y - b) + \Sigma_x^{-1}\mu_x\right).$$

Proof. The proof relies on algebraic manipulations and parameter recognition.See [2, Section 2.3.3] □

Inferring an unknown vector from noisy measurements

Suppose we have N vector-valued observations $y_i \sim N(x, \Sigma_y)$ and a Gaussian prior $x \sim N(\mu_0, \Sigma_0)$. Setting $A = I$, $b = 0$, and defining $\bar{y}$ as the effective observation with precision $N\Sigma_y^{-1}$, we get:

$$p(x|y_1, \ldots, y_N) = N(x|\mu_N, \Sigma_N),$$

where

$$\begin{cases} \Sigma_N^{-1} = \Sigma_0^{-1} + N\Sigma_y^{-1}, \\ \mu_N = \Sigma_N \left(\Sigma_y^{-1}(N\bar{y}) + \Sigma_0^{-1}\mu_0\right). \end{cases}$$

This setup can model scenarios where x is an unknown true location (e.g., an object in 2D space), and y_i are noisy observations, like radar blips. As the number of observations increases, the estimate of x becomes more precise.

2.4 The Wishart Distribution

The Wishart distribution is a generalization of the Gamma distribution to positive definite matrices. It is often used to model uncertainty in covariance matrices Σ or their inverses, $\Lambda = \Sigma^{-1}$. The pdf of the Wishart distribution is defined as:

$$\text{Wi}(\Lambda|S, \nu) = \frac{1}{Z_{\text{Wi}}}|\Lambda|^{(\nu-D-1)/2} \exp\left(-\frac{1}{2}\text{tr}(\Lambda S^{-1})\right)$$

where ν is the "degrees of freedom" and S is the "scale matrix". Z_{Wi} is the normalization constant. The mean and mode of the Wishart distribution $\text{Wi}(S, \nu)$ are:

$$\text{mean} = \nu S, \quad \text{mode} = (\nu - D - 1)S$$

The mode exists if $\nu > D + 1$. For $D = 1$, the Wishart distribution reduces to the Gamma distribution:

$$\text{Wi}(\lambda|s^{-1}, \nu) = \text{Ga}(\lambda|\nu/2, 2s)$$

Inverse Wishart Distribution

The inverse Wishart distribution (IW) is the multidimensional generalization of the inverse Gamma distribution. The pdf of the inverse Wishart distribution is defined for $\nu > D - 1$ and $S \succ 0$ as:

$$\text{IW}(\Sigma|S, \nu) = \frac{1}{Z_{\text{IW}}}|\Sigma|^{-(\nu+D+1)/2} \exp\left(-\frac{1}{2}\text{tr}(S^{-1}\Sigma^{-1})\right),$$

where Z_{IW} is the normalization constant.

The mean and mode of the inverse Wishart distribution are:

$$\text{mean} = \frac{S^{-1}}{\nu - D - 1}, \quad \text{mode} = \frac{S^{-1}}{\nu + D + 1}$$

Remark. *Wishart distribution and the Gaussian are connected the following way: if $x_i \sim N(0, \Sigma)$, then the scatter matrix $S = \sum_{i=1}^N x_i x_i^T$ has a Wishart distribution, $S \sim$ Wi(Σ, N), with $E[S] = N\Sigma$.*

Similarly, if $\Sigma^{-1} \sim$ Wi(S, ν), then $\Sigma \sim$ IW$(S^{-1}, \nu + D + 1)$.

2.5 Inferring the Parameters of an MVN

So far, we have discussed inference in a Gaussian assuming known parameters $\theta = (\mu, \Sigma)$. Now we consider how to infer these parameters. Assuming fully observed data $x_i \sim N(\mu, \Sigma)$ for $i = 1, \ldots, N$, we derive the posterior distributions for μ and Σ.

Posterior Distribution of μ

The likelihood for μ is:

$$p(D|\mu) = N\left(\bar{x}|\mu, \frac{1}{N}\Sigma\right)$$

Using a conjugate Gaussian prior $p(\mu) = N(\mu|m_0, V_0)$, we get the posterior:

$$p(\mu|D, \Sigma) = N(\mu|m_N, V_N)$$

where

$$V_N^{-1} = V_0^{-1} + N\Sigma^{-1}, \quad m_N = V_N\left(\Sigma^{-1}N\bar{x} + V_0^{-1}m_0\right)$$

With an uninformative prior ($V_0 = \infty I$), we have $p(\mu|D, \Sigma) = N(\bar{x}, \frac{1}{N}\Sigma)$, so the posterior mean equals the MLE, and the posterior variance decreases as $1/N$.

Posterior Distribution of Σ

For Σ, the likelihood is:

$$p(D|\mu, \Sigma) \propto |\Sigma|^{-N/2} \exp\left(-\frac{1}{2}\text{tr}(S_\mu \Sigma^{-1})\right)$$

where $S_\mu = \sum_{i=1}^{N}(x_i - \mu)(x_i - \mu)^T$. Using an inverse Wishart prior $\text{IW}(\Sigma|S_0^{-1}, \nu_0)$, the posterior is also inverse Wishart:

$$p(\Sigma|D, \mu) = \text{IW}(\Sigma|S_N, \nu_N)$$

where

$$\nu_N = \nu_0 + N, \quad S_N = S_0 + S_\mu$$

Thus, the posterior "strength" ν_N is the sum of the prior strength ν_0 and the sample size N, and the posterior scatter matrix S_N combines the prior and data scatter matrices.

Posterior Distribution of μ and Σ

To compute $p(\mu, \Sigma|D)$, we start with the likelihood and then discuss the choice of prior.

Likelihood

The likelihood for N observations is:

$$p(D|\mu, \Sigma) = (2\pi)^{-ND/2}|\Sigma|^{-N/2} \exp\left(-\frac{1}{2}\sum_{i=1}^{N}(x_i - \mu)^T\Sigma^{-1}(x_i - \mu)\right)$$

This can be rewritten in terms of the sample mean $\bar{x}$ and scatter matrix $S_x = \sum_{i=1}^{N}(x_i - \bar{x})(x_i - \bar{x})^T$:

$$p(D|\mu, \Sigma) = (2\pi)^{-ND/2}|\Sigma|^{-N/2} \exp\left(-\frac{N}{2}(\mu - \bar{x})^T\Sigma^{-1}(\mu - \bar{x}) - \frac{N}{2}\text{tr}(\Sigma^{-1}S_x)\right) \tag{4}$$

Prior

To model $p(\mu, \Sigma)$, we use the **Normal-Inverse-Wishart** (NIW) distribution, which provides a fully conjugate prior:

$$p(\mu, \Sigma) = \mathrm{N}(\mu|m_0, V_0)\,\mathrm{IW}(\Sigma|S_0, \nu_0)$$

This prior is not fully conjugate to the likelihood because μ and Σ are intertwined in the likelihood expression and thus remain coupled in the posterior. This type of prior is sometimes called **semi-conjugate** or **conditionally conjugate**, as the conditionals $p(\mu|\Sigma)$ and $p(\Sigma|\mu)$ are each conjugate individually. To achieve a fully conjugate prior, we need a form where μ and Σ are explicitly dependent on each other. This can be done by using a joint distribution structured as:

$$p(\mu, \Sigma) = p(\mu|\Sigma)p(\Sigma)$$

Looking at the form of the likelihood equation 4, we see that a natural conjugate prior has the form of a Normal-inverse-Wishart (NIW) distribution, defined as follows:

$$\mathrm{NIW}(\mu, \Sigma|m_0, \kappa_0, \nu_0, S_0) = \mathrm{N}(\mu|m_0, \kappa_0^{-1}\Sigma) \times \mathrm{IW}(\Sigma|S_0, \nu_0)$$

In this setup:

- m_0 is the prior mean for μ,
- κ_0 is the strength of belief in m_0,
- S_0 is proportional to the prior mean of Σ,
- ν_0 controls the confidence in S_0.

The NIW distribution ensures that both $p(\mu|\Sigma)$ and $p(\Sigma|\mu)$ remain conjugate, making it suitable for Bayesian inference on μ and Σ jointly.

Posterior

The posterior distribution can be shown to follow a Normal-inverse-Wishart (NIW) distribution with updated parameters:

$$\begin{cases} p(\mu, \Sigma|D) = \mathrm{NIW}(\mu, \Sigma|m_N, \kappa_N, \nu_N, S_N), \\ \kappa_N = \kappa_0 + N, \\ m_N = \frac{\kappa_0 m_0 + N\bar{x}}{\kappa_N}, \\ \nu_N = \nu_0 + N \\ S_N = S_0 + S_x + \kappa_0 m_0 m_0^T - \kappa_N m_N m_N^T. \end{cases}$$

where $S = \sum_{i=1}^{N} x_i x_i^T$ is the uncentered sum-of-squares matrix.

This result is intuitive: the posterior mean m_N is a weighted combination of the prior mean m_0 and the sample mean $\bar{x}$, with total "strength" $\kappa_0 + N$. The posterior scatter matrix S_N combines the prior scatter S_0, the empirical scatter S_x, and an additional term reflecting the uncertainty in the mean, which introduces virtual scatter.

Posterior mode

The mode of the joint distribution is given by

$$\arg\max p(\mu, \Sigma|D) = \left(m_N, \frac{S_N}{\nu_N + D + 2}\right).$$

If we set $\kappa_0 = 0$, this reduces to:

$$\arg\max p(\mu, \Sigma|D) = \left(x, \frac{S_0 + S_x}{\nu_0 + N + D + 2}\right).$$

Posterior predictive

The posterior predictive is given by

$$p(x|D) = \frac{p(x, D)}{p(D)},$$

which can be evaluated in terms of a ratio of marginal likelihoods. This ratio has the form of a multivariate Student-T distribution:

$$p(x|D) = \int N(x|\mu, \Sigma)\, \text{NIW}(\mu, \Sigma|m_N, \kappa_N, \nu_N, S_N)\, d\mu\, d\Sigma = \mathcal{T}\left(x \middle| m_N, \frac{\kappa_N + 1}{\kappa_N(\nu_N - D + 1)} S_N, \nu_N - D + 1\right).$$

The Student-T distribution has wider tails than a Gaussian, accounting for the uncertainty in Σ.

2.6 Posterior for scalar data

We now specialize the above results to the case where x_i is 1-dimensional. These results are widely used in the statistics literature. It is conventional not to use the Wishart distribution but instead to use the normal-inverse-chi-squared (NIX) distribution, defined by

$$\begin{aligned}\text{NI}\chi^2(\mu, \sigma^2|m_0, \kappa_0, \nu_0, \sigma_0^2) &\propto N(\mu|m_0, \sigma^2/\kappa_0)\, \chi^{-2}(\sigma^2|\nu_0, \sigma_0^2) \\ &= \frac{1}{\sigma^2}^{(\nu_0+3)/2} \exp\left[-\frac{\nu_0\sigma_0^2 + \kappa_0(\mu - m_0)^2}{2\sigma^2}\right].\end{aligned}$$

Along the μ axis, the distribution is shaped like a Gaussian, and along the σ^2 axis, the distribution is shaped like a χ^{-2}; the contours of the joint density have a "squashed egg" appearance. The contours for μ are more peaked for small values of σ^2, which makes sense, since if the data has low variance, we can estimate its mean more reliably.

One can show that the posterior is given by

$$p(\mu, \sigma^2|\mathcal{D}) = \text{NI}\chi^2(\mu, \sigma^2|m_N, \kappa_N, \nu_N, \sigma_N^2),$$

where

$$\begin{cases} m_N = \frac{\kappa_0 m_0 + N\bar{x}}{\kappa_0 + N}, \\ \kappa_N = \kappa_0 + N, \\ \nu_N = \nu_0 + N, \\ \nu_N\sigma_N^2 = \nu_0\sigma_0^2 + \sum(x_i - x)^2 + \frac{\kappa_0 N}{\kappa_0 + N}(m_0 - x)^2. \end{cases}$$

3 Gaussian processes

In supervised learning, we observe inputs x_i and outputs y_i, assuming $y_i = f(x_i)$ for some unknown function f, possibly with noise. The goal is to infer a distribution over functions, $p(f|X, y)$, and use it to make predictions for new inputs, by computing:

$$p(y^*|x^*, X, y) = \int p(y^*|f, x^*)p(f|X, y)\, df$$

Previously, we used parametric representations of f, inferring $p(\theta|D)$ instead of $p(f|D)$. Here, we use Gaussian processes (GPs), which define a prior over functions. Once data is observed, the GP prior converts into a posterior over functions. A GP assumes that $p(f(x_1), \ldots, f(x_N))$ is jointly Gaussian with mean $\mu(x)$ and covariance $\Sigma(x)$, where $\Sigma_{ij} = \kappa(x_i, x_j)$, and κ is a kernel function.

In regression, these computations can be done in closed form in $O(N^3)$ time. For classification, approximations like the Gaussian approximation are needed, as the posterior is not exactly Gaussian.

3.1 Gaussian processes for regression

In Gaussian processes (GPs) for regression, we place a GP prior on the regression function $f(x)$, written as:

$$f(x) \sim \text{GP}(m(x), \kappa(x, x'))$$

where $m(x)$ is the mean function, defined as $m(x) = \mathbb{E}[f(x)]$, and $\kappa(x, x')$ is the covariance (or kernel) function, defined as:

$$\kappa(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))^T]$$

The kernel κ must be positive definite. For a finite set of points, this GP prior results in a joint Gaussian distribution:

$$p(f|X) = N(f|\mu, K)$$

where $K_{ij} = \kappa(x_i, x_j)$ and $\mu = (m(x_1), \ldots, m(x_N))$. Commonly, $m(x) = 0$ is chosen, as the GP is sufficiently flexible to model the mean.

3.2 Predictions using noise-free observations

In the case of noise-free observations, suppose we have a training set $D = \{(x_i, f_i), i = 1, \ldots, N\}$, where $f_i = f(x_i)$ represents noise-free function values at x_i. For a test set X^* of size $N^* \times D$, we want to predict the function outputs f^*.

In this setting, if the GP is queried with a training point x, it should return $f(x)$ without uncertainty, behaving as an interpolator. This behavior only holds with noiseless observations.

The joint distribution over training outputs f and test outputs f^* is:

$$\begin{bmatrix} f \\ f^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu \\ \mu^* \end{bmatrix}, \begin{bmatrix} K & K_* \\ K_*^T & K_{**} \end{bmatrix} \right)$$

where $K = \kappa(X, X)$ is $N \times N$, $K_* = \kappa(X, X^*)$ is $N \times N^*$, and $K_{**} = \kappa(X^*, X^*)$ is $N^* \times N^*$.

Using Gaussian conditioning, the posterior distribution for f^* is:

$$p(f^*|X^*, X, f) = \mathcal{N}(f^*|\mu_*, \Sigma_*),$$

where

$$\begin{cases} \mu_* = \mu(X^*) + K_*^T K^{-1}(f - \mu(X)), \\ \Sigma_* = K_{**} - K_*^T K^{-1} K_* \end{cases}$$

This provides the GP prediction for the test outputs, accounting for correlations with the training data.

3.3 Predictions using noisy observations

When we observe a noisy version of the function, $y = f(x) + \epsilon$, with $\epsilon \sim \mathcal{N}(0, \sigma_y^2)$, the model no longer interpolates exactly but instead approximates the observed data. The covariance of the noisy observations y_p and y_q is:

$$\text{cov}[y_p, y_q] = \kappa(x_p, x_q) + \sigma_y^2 \delta_{pq}$$

where $\delta_{pq} = 1$ if $p = q$ and 0 otherwise. Thus, we can write:

$$\text{cov}[y|X] = K + \sigma_y^2 I_N \equiv K_y$$

The noise term adds a diagonal component to the covariance matrix, as the noise is independent for each observation.

The joint distribution of the observed noisy outputs y and the latent (noise-free) function outputs f^* at test points is:

$$\begin{bmatrix} y \\ f^* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K_y & K_* \\ K_*^T & K_{**} \end{bmatrix} \right)$$

where, for simplicity, we assume a zero mean. The posterior predictive distribution is then:

$$p(f^*|X^*, X, y) = \mathcal{N}(f^*|\mu_*, \Sigma_*),$$

with

$$\begin{cases}\mu_* = K_*^T K_y^{-1} y, \\ \Sigma_* = K_{**} - K_*^T K_y^{-1} K_*. \end{cases}$$

This setup allows us to predict the underlying function f^* while accounting for noise in the observations.

4 Gaussian mixture models

[1, Sections 11]

4.1 Latent Variable Models (LVMs)

Graphical models define high-dimensional joint probability distributions by modeling dependencies between variables using graph edges. An alternative approach assumes that observed variables are correlated due to shared hidden causes, represented by latent variables. Such models, known as **latent variable models** (LVMs), are more challenging to fit but offer two key advantages:

- Parameter Efficiency: LVMs often use fewer parameters compared to direct correlation models.
- Data Compression: Latent variables act as a bottleneck, providing compressed representations of data, fundamental to unsupervised learning.

By adjusting the likelihood $p(x_i|z_i)$ and prior $p(z_i)$, LVMs can model diverse structures.

4.2 Mixture Models

A **mixture model** is the simplest type of latent variable model (LVM), where the latent variable $z_i \in \{1, \dots, K\}$ represents a discrete state. The prior $p(z_i)$ follows a categorical distribution, Cat(π), and the likelihood $p(x_i|z_i = k) = p_k(x_i)$ uses p_k, the k-th base distribution for observations.

The overall model combines K base distributions into a weighted sum:

$$p(x_i) = \sum_{k=1}^{K} \pi_k p_k(x_i),$$

where the weights π_k satisfy $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$, ensuring a convex combination.

4.3 Mixtures of Gaussians

The **Gaussian Mixture Model** (GMM) is the most widely used mixture model. Each base distribution in the mixture is a multivariate Gaussian with mean μ_k and covariance matrix Σ_k. The model is expressed as:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k),$$

where π_k are the mixing weights.

4.4 Mixtures of Bernoulli distributions

In the Bernoulli Mixture Model (BMM) each component of the mixture is a Bernoulli distribution, making it suitable for data where each feature is binary. The model is expressed as:

$$p(x) = \sum_{k=1}^{K} \pi_k p_k(\mathbf{x}|\boldsymbol{\theta}_k) = \sum_{k=1}^{K} \pi_k \prod_{i=1}^{D} \theta_{ki}^{x_i} \left(1 - \theta_{ki}\right)^{1-x_i},$$

where:

- $\mathbf{x} = (x_1, x_2, \ldots, x_D)$ is a binary data vector with D dimensions.
- $\boldsymbol{\theta}_k = (\theta_{k1}, \theta_{k2}, \ldots, \theta_{kD})$ are the parameter vectors for the k-th Bernoulli distribution, where each θ_{ki} represents the probability that the i-th binary variable is 1.

4.5 The EM algorithm

Estimating Maximum Likelihood (ML) or Maximum A Posteriori (MAP) parameters is straightforward with complete data but becomes challenging with missing data or latent variables. While gradient-based optimizers can minimize the negative log-likelihood (NLL), they often require enforcing constraints (e.g., positive-definite covariance matrices, normalized mixing weights), which can be cumbersome.

The Expectation-Maximization (EM) algorithm simplifies this process. EM is an iterative method that alternates between:

- E-Step: Inferring missing values based on current parameters.
- M-Step: Optimizing parameters using the inferred data.

EM often provides closed-form updates and automatically enforces constraints, making it an effective tool for parameter estimation in incomplete data scenarios.

4.5.1 Basic Idea

Let x_i represent observed variables and z_i the hidden or missing variables. The goal is to maximize the log-likelihood of the observed data:

$$l(\theta) = \sum_{i=1}^{N} \log \sum_{z_i} p(x_i, z_i|\theta).$$

Direct optimization is difficult because the logarithm cannot be pushed inside the summation. The EM algorithm addresses this by introducing the complete data log-likelihood:

Definition (Complete data log-likelihood)**.**

$$l_c(\theta) = \sum_{i=1}^{N} \log p(x_i, z_i|\theta),$$

The complete data log-likelihood becomes more manageable if z_i were known, as it eliminates the need to marginalize over z_i. However, since z_i is unobserved, the EM algorithm calculates the **expected complete data log-likelihood**, also referred to as the auxiliary function Q.

Definition (Expected complete data log-likelihood)**.**

$$Q(\theta, \theta^{(t-1)}) = \mathbb{E}_{p(z|x,\theta^{(t-1)})}[l_c(\theta)],$$

where θ^{t-1} represents parameters from the previous iteration.

Remark. *Estimating the Maximum A Posteriori (MAP) or Maximum Likelihood (ML) for latent variable models (LVMs) is challenging due to the non-convex nature of the log-likelihood function. This complexity arises because the logarithm cannot be moved inside the summation, complicating algebraic simplifications. While distributions in the exponential family offer a concave complete-data log-likelihood (and hence a unique maximum), the presence of missing data introduces a log-sum-exp term that makes the objective function non-convex. As a result, the optimization problem has multiple local optima, requiring techniques like random restarts, careful initialization, or EM to handle the non-convexity effectively. For more details check [1, Section 11.3.2].*

The EM-algorithm alternates between:

1. E-Step: Compute $Q(\theta, \theta^{t-1})$, or the expected sufficient statistics (ESS), given the observed data and current parameters.
2. M-Step: Maximize $Q(\theta, \theta^{t-1})$ with respect to θ:

$$\theta^t = \arg\max_{\theta} Q(\theta, \theta^{t-1}).$$

For MAP estimation, the M-step includes a prior $p(\theta)$, modifying the objective to:

$$\theta^t = \arg\max_\theta Q(\theta, \theta^{t-1}) + \log p(\theta).$$

The E-step remains unchanged. This iterative process ensures efficient parameter updates, leveraging the complete data structure for optimization. The EM algorithm guarantees that the log-likelihood of the observed data (or the log posterior in the case of MAP estimation) either increases or remains constant at each iteration. This ensures a monotonic improvement in the objective function throughout the optimization process.

4.5.2 EM for GMMs

For mixture models, the expected complete data log-likelihood simplifies as follows:

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

where the posterior probabilities $r_{ik} = p(z_i = k|x_i, \theta^{(t-1)})$ is called the responsibility that cluster k takes for data point i.

E Step

The E step computes the responsibilities r_{ik}, which are defined as:

$$r_{ik} = \frac{\pi_k p(x_i|\theta_k)}{\sum_{k'} \pi_{k'} p(x_i|\theta_{k'})}$$

These responsibilities quantify the probability that each data point i belongs to cluster k, given the current parameters $\theta^{(t-1)}$.

M Step

In the M step, we optimize Q with respect to π_k and $\theta_k = (\mu_k, \Sigma_k)$. For π_k, we have:

$$\pi_k = \frac{\sum_i r_{ik}}{N},$$

where r_{ik} is the responsibility that cluster k takes for data point i.

For μ_k, the mean of cluster k, we compute:

$$\mu_k = \frac{\sum_i r_{ik} x_i}{\sum_i r_{ik}}.$$

For Σ_k, the covariance of cluster k, we compute:

$$\Sigma_k = \frac{\sum_i r_{ik}(x_i - \mu_k)(x_i - \mu_k)^T}{\sum_i r_{ik}} = \frac{\sum_i r_{ik} x_i x_i^T}{\sum_i r_{ik}} - \mu_k \mu_k^T.$$

These equations intuitively make sense:

- The mean μ_k is the weighted average of all points assigned to cluster k.
- The covariance Σ_k is proportional to the weighted empirical scatter matrix of the points in cluster k.

After computing the new estimates, we update the parameters as $\theta^t = (\pi_k, \mu_k, \Sigma_k)$ for $k = 1, \ldots, K$, and proceed to the next E step.

4.6 EM for Mixture of Bernoullis

To apply the EM algorithm, introduce latent variables $\mathbf{Z} = \{z_n\}$, where each z_n indicates the component membership of data point $\mathbf{x}_n$.

The complete-data log likelihood is expressed as:

$$\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}, \boldsymbol{\pi}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \left(\ln \pi_k + \sum_{i=1}^{D} [x_{ni} \ln \theta_{ki} + (1 - x_{ni}) \ln(1 - \theta_{ki})] \right) \tag{5}$$

E-Step (Expectation)

Objective: Compute the posterior probabilities (responsibilities) $\gamma(z_{nk})$ that each component k is responsible for each data point $\mathbf{x}_n$.

Calculation:

$$p(\mathbf{z}_{nk}|\mathbf{x}_n, \boldsymbol{\theta}, \pi) = \gamma(\mathbf{z}_{nk}) = \frac{\pi_k \prod_{i=1}^{D} \theta_{ki}^{x_{ni}} (1 - \theta_{ki})^{1 - x_{ni}}}{\sum_{j=1}^{K} \pi_j \prod_{i=1}^{D} \theta_{ji}^{x_{ni}} (1 - \theta_{ji})^{1 - x_{ni}}} \tag{6}$$

M-Step (Maximization)

Objective: Update the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$ to maximize the expected complete-data log likelihood computed in the E-Step.

Update Mixing Coefficients (π_k):

$$\pi_k = \frac{N_k}{N} \quad \text{where} \quad N_k = \sum_{n=1}^{N} \gamma(z_{nk}) \tag{7}$$

Update Probability Parameters (θ_{ki}):

$$\theta_{ki} = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) x_{ni}}{N_k} \tag{8}$$

This sets θ_{ki} to the weighted average of the data points, with weights given by the responsibilities $\gamma(z_{nk})$.

4.7 Using Mixture Models for Clustering

Mixture models have two primary applications:

1. **Black-Box Density Modeling**
   - **Purpose**: Serve as flexible density estimators $p(\mathbf{x}_i)$.
   - **Uses**: Data compression, outlier detection, and creating generative classifiers by modeling class-conditional densities $p(\mathbf{x} \mid y = c)$ with mixture distributions.
2. **Clustering**
   - **Process**:
     (a) **Model Fitting**: Fit the mixture model to the data.

(b) **Responsibility Calculation**: Compute the posterior probability

$$r_{ik} = p(z_i = k \mid \mathbf{x}_i, \theta)$$

which indicates the probability that data point i belongs to cluster k. This is done using Bayes' rule:

$$r_{ik} = \frac{p(z_i = k \mid \theta)\, p(\mathbf{x}_i \mid z_i = k, \theta)}{\sum_{k'=1}^{K} p(z_i = k' \mid \theta)\, p(\mathbf{x}_i \mid z_i = k', \theta)}$$

- **Soft Clustering**: Assigns probabilities to cluster memberships, reflecting uncertainty in assignments. This approach is similar to generative classifiers, with the key difference being that mixture models do not observe cluster assignments z_i during training, whereas generative classifiers do observe class labels y_i.
- **Hard Clustering**: When uncertainty is low (i.e., $1 - \max_k r_{ik}$ is small), a hard assignment can be made using the Maximum A Posteriori (MAP) estimate:

$$z_i^* = \arg\max_k r_{ik} = \arg\max_k \left(\log p(\mathbf{x}_i \mid z_i = k, \theta) + \log p(z_i = k \mid \theta)\right)$$

4.8 EM Monotonically Increases the Observed Data Log-Likelihood

At iteration t, let $\theta^{(t)}$ denote the current parameter estimates, and $\theta^{(t+1)}$ the updated estimates after the EM step. The EM algorithm satisfies the following inequality:

$$l(\theta^{(t+1)}) \geq Q(\theta^{(t+1)}, \theta^{(t)}) \geq Q(\theta^{(t)}, \theta^{(t)}) = l(\theta^{(t)}), \tag{9}$$

where:

- $Q(\theta, \theta^{(t)})$ is the *Q-function*, acting as a lower bound for $l(\theta)$.
- The first inequality holds because $Q(\theta, \theta^{(t)}) \leq l(\theta)$.
- The second inequality follows from the maximization step, where $\theta^{(t+1)}$ maximizes $Q(\theta, \theta^{(t)})$.
- The equality $Q(\theta^{(t)}, \theta^{(t)}) = l(\theta^{(t)})$ is based on the definition of the Q-function.

For details of these inequalities see [1, Section 11.4.7.1].

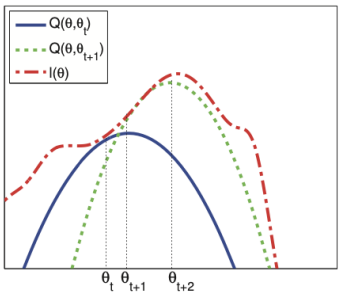

Figure 1: Illustration of EM as a bound optimization algorithm.

Implications

- **Convergence Guarantee**: The observed data log-likelihood $l(\theta)$ increases monotonically with each EM iteration, ensuring convergence to at least a local optimum.
- Error Detection: If $l(\theta)$ does not increase monotonically during implementation, it indicates potential errors in the mathematical formulation or coding of the algorithm. For Maximum A Posteriori (MAP) estimation, ensure that the log prior is included in the objective function.

4.9 BIC

5 Principal component analysis (PCA)

[1, Section 12.2]

Definition (Matrix with Orthonormal Columns)**.** *Let $Q \in \mathbb{R}^{m \times n}$, where $m \geq n$. The matrix Q is said to have **orthonormal columns** if its columns are orthogonal unit vectors, which is equivalent to the condition:*

$$Q^\top Q = I_n$$

Remark (Key Properties)**.**

- *Each column of Q is a unit vector:*

$$\|\mathbf{q}_i\| = 1, \quad \text{for } i = 1, 2, \ldots, n,$$

  where $\mathbf{q}_i$ is the i-th column of Q.

- *Distinct columns of Q are orthogonal:*

$$\mathbf{q}_i^\top \mathbf{q}_j = 0, \quad \text{for } i \neq j.$$

- ***Preservation of Norms**: For any vector $\mathbf{x} \in \mathbb{R}^n$,*

$$\|Q\mathbf{x}\| = \|\mathbf{x}\|.$$

- ***Linear Independence**: The columns of Q form a linearly independent set.*

The **reconstruction error** in PCA measures the discrepancy between the original data and its reconstruction from the reduced-dimensional representation. It quantifies the information loss due to dimensionality reduction.

Definition (Reconstruction error)**.** *The reconstruction error J is defined as the squared Frobenius norm of the difference between the mean-centered data $\tilde{X}$ and its projection onto the k-dimensional principal subspace spanned by W:*

$$J(W, Z) = \|X - XWW^\top\|_F^2 = \frac{1}{N} \sum_{i=1}^{n} \|\mathbf{x}_i - W\mathbf{z}_i\|_2^2$$

where $\mathbf{x}_i \in \mathbb{R}^p$ is the i-th row of X, $\|\cdot\|_F$ denotes the Frobenius norm, and $\|\cdot\|_2$ denotes the Euclidean norm.

The synthesis view of classical PCA is summarized in the following theorem.

Theorem (Classical PCA)**.** *Suppose we want to find an orthogonal set of L linear basis vectors $\mathbf{w}_j \in \mathbb{R}^D$, and the corresponding scores $\mathbf{z}_i \in \mathbb{R}^L$, such that we minimize the average reconstruction error*

$$J(W, Z) = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{x}_i - W\mathbf{z}_i\|^2,$$

where W is an orthonormal matrix containing the basis vectors $\mathbf{w}_j$, and Z is the matrix containing the scores $\mathbf{z}_i$.

Furthermore, the optimal solution is obtained by setting $\hat{W} = V_L$, where V_L contains the L eigenvectors with the largest eigenvalues of the empirical covariance matrix

$$\hat{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^\top.$$

(We assume the $\mathbf{x}_i$ have zero mean for notational simplicity.) Additionally, the optimal low-dimensional encoding of the data is given by

$$\hat{\mathbf{z}}_i = W^\top \mathbf{x}_i,$$

which is an orthogonal projection of the data onto the column space spanned by the eigenvectors.

Proof. See [1, Section 12.2.2] □

Algorithm 1 Principal Component Analysis (PCA)

Require: Data matrix $X \in \mathbb{R}^{n \times p}$ where n is the number of samples and p is the number of features.
Ensure: Principal components and projected data matrix Z.

Standardize the Data:
Compute the mean of each feature:

$$\mu_j = \frac{1}{n} \sum_{i=1}^{n} X_{ij}, \quad \text{for } j = 1, 2, \ldots, p$$

Center the data by subtracting the mean:

$$\tilde{X}_{ij} = X_{ij} - \mu_j, \quad \text{for } i = 1, 2, \ldots, n; \quad j = 1, 2, \ldots, p$$

Compute the Covariance Matrix:

$$C = \frac{1}{n-1} \tilde{X}^\top \tilde{X} \in \mathbb{R}^{p \times p}$$

Compute Eigenvalues and Eigenvectors:
Solve the eigenvalue problem:

$$C\mathbf{v}_k = \lambda_k \mathbf{v}_k, \quad \text{for } k = 1, 2, \ldots, p$$

Collect eigenvalues λ_k and corresponding eigenvectors $\mathbf{v}_k$.
Sort Eigenvalues and Eigenvectors:
Sort the eigenvalues in decreasing order:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$$

Rearrange the eigenvectors accordingly.
Select Top k Components:
Choose the top k eigenvectors to form the projection matrix:

$$W = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k] \in \mathbb{R}^{p \times k}$$

Project the Data onto the New Subspace:
Compute the projected data matrix:

$$Z = \tilde{X} W \in \mathbb{R}^{n \times k}$$

6 Markovian models

6.1 Markov models

A Markov chain assumes X_t captures all relevant information for predicting the future, making it a sufficient statistic. For discrete time steps, the joint distribution can be written as follows:

$$P(X_1, \ldots, X_T) = P(X_1) \prod_{t=2}^{T} P(X_t \mid X_{t-1}).$$

If the transition function $P(X_t \mid X_{t-1})$ is time-invariant, the chain is called *homogeneous*, *stationary*, or *time-invariant*. This parameter-tying assumption enables modeling an arbitrary number of variables with fixed parameters, forming a *stochastic process*.

For discrete states, where $X_t \in \{1, \ldots, K\}$, the chain is a *finite-state Markov chain*.

When X_t is discrete, $X_t \in \{1, \ldots, K\}$, the conditional distribution $P(X_t \mid X_{t-1})$ can be represented as a $K \times K$ *transition matrix* A, where $A_{ij} = P(X_t = j \mid X_{t-1} = i)$ denotes the probability of transitioning from state i to state j. Each row of A sums to one, $\sum_j A_{ij} = 1$, making it a *stochastic matrix*.

Theorem (Chapman-Kolmogorov Equations)**.** *Let $P_{ij}^{(n)}$ denote the probability of transitioning from state i to state j in n steps in a Markov chain. Then, for any non-negative integers m and n, the following holds:*

$$P_{ij}^{(m+n)} = \sum_k P_{ik}^{(m)} P_{kj}^{(n)},$$

where the summation is over all states k. This can be equivalently expressed in matrix form as:

$$A^{(m+n)} = A^{(m)} A^{(n)},$$

where $A^{(n)}$ is the n-step transition matrix. Consequently, the n-step transition matrix satisfies:

$$A^{(n)} = A^n.$$

Thus, the probabilities of transitions over multiple steps can be computed by raising the transition matrix to the appropriate power.

6.2 Hidden Markov models

Definition (Hidden Markov Model (HMM))**.** *A Hidden Markov Model (HMM) consists of:*

1. *A discrete-time, discrete-state Markov chain with hidden states $z_t \in \{1, \ldots, K\}$.*
2. *An observation model $P(x_t \mid z_t)$, where observations x_t may be discrete or continuous.*

The joint distribution of the hidden states $z_{1:T}$ and observations $x_{1:T}$ is:

$$P(z_{1:T}, x_{1:T}) = P(z_{1:T}) P(x_{1:T} \mid z_{1:T}) = P(z_1) \prod_{t=2}^{T} P(z_t \mid z_{t-1}) \prod_{t=1}^{T} P(x_t \mid z_t).$$

Observation Models:

- *For discrete observations, the observation model is typically represented by an* observation matrix *B, where:*
$$P(x_t = l \mid z_t = k, \theta) = B(k, l).$$
- *For continuous observations, the observation model is often a conditional Gaussian:*
$$P(x_t \mid z_t = k, \theta) = \mathcal{N}(x_t \mid \mu_k, \Sigma_k).$$

HMMs generalize Gaussian Mixture Models (GMMs) by incorporating Markovian dynamics into the cluster membership. For example, with $K = 3$ states emitting different Gaussians, clusters exhibit temporal dependencies, resulting in sequences of observations within the same region followed by abrupt transitions to new clusters.

6.2.1 Inference in HMMs

1. **Filtering:** Compute the belief state $P(z_t \mid x_{1:t})$ *online* or recursively as data streams in. Filtering is named for its ability to reduce noise by leveraging all evidence up to time t, rather than relying solely on the current observation $P(z_t \mid x_t)$. Sequential application of Bayes' rule enables efficient computation of the filtered belief state.
2. **Smoothing:** Compute $P(z_t \mid x_{1:T})$ *offline*, using all observations $x_{1:T}$. By conditioning on both past and future evidence, smoothing significantly reduces uncertainty.

6.2.2 The forwards algorithm

Definition (Recursive Computation of Filtered Marginals in HMMs)**.** *The filtered marginals $P(z_t \mid x_{1:t})$ in a Hidden Markov Model (HMM) can be computed recursively using the* predict-update cycle*, which consists of the following steps:*

1. ***Prediction Step:** Compute the one-step-ahead predictive density, which serves as the prior for time t:*
$$P(z_t = j \mid x_{1:t-1}) = \sum_i P(z_t = j \mid z_{t-1} = i) P(z_{t-1} = i \mid x_{1:t-1}),$$
where $P(z_t = j \mid z_{t-1} = i)$ is the transition probability.

2. ***Update Step:*** *Absorb the observed data at time t using Bayes' rule:*
$$P(z_t = j \mid x_{1:t}) \propto P(x_t \mid z_t = j)P(z_t = j \mid x_{1:t-1}),$$
where the normalization constant is given by:
$$Z_t = \sum_j P(z_t = j \mid x_{1:t-1})P(x_t \mid z_t = j).$$

The filtered belief state at time t, $P(z_t \mid x_{1:t})$, is often denoted as α_t and can be expressed in matrix-vector notation:

$$\alpha_t \propto \psi_t \odot (\Psi^T \alpha_{t-1}),$$

where:

- $\psi_t(j) = P(x_t \mid z_t = j)$ *represents the local evidence at time t.*
- $\Psi(i, j) = P(z_t = j \mid z_{t-1} = i)$ *is the transition matrix.*
- $\odot$ *denotes the Hadamard product (elementwise multiplication).*

This process, known as the forwards algorithm*, computes the filtered belief state α_t at each time step t.*

6.2.3 The forwards-backwards algorithm

Theorem (Forwards-Backwards Algorithm for Smoothed Marginals)**.** *To compute the smoothed marginals $P(z_t = j \mid x_{1:T})$ in a Hidden Markov Model (HMM) using offline inference, we decompose the chain into past and future components conditioned on z_t:*

$$P(z_t = j \mid x_{1:T}) \propto P(z_t = j \mid x_{1:t})P(x_{t+1:T} \mid z_t = j).$$

Define the following terms:

- $\alpha_t(j) = P(z_t = j \mid x_{1:t})$*: the* filtered belief state*, computed recursively using the forwards algorithm.*
- $\beta_t(j) = P(x_{t+1:T} \mid z_t = j)$*: the* conditional likelihood of future evidence*, computed recursively in a backwards fashion.*
- $\gamma_t(j) = P(z_t = j \mid x_{1:T})$*: the* smoothed posterior marginal*, given by:*
$$\gamma_t(j) \propto \alpha_t(j)\beta_t(j).$$

Backwards Recursion: *The β terms are computed recursively as:*
$$\beta_{t-1}(i) = \sum_j \Psi(i, j)\psi_t(j)\beta_t(j),$$

where:

- $\Psi(i, j) = P(z_t = j \mid z_{t-1} = i)$ *is the transition matrix.*
- $\psi_t(j) = P(x_t \mid z_t = j)$ *is the local evidence.*
- $\odot$ *denotes the Hadamard product (elementwise multiplication).*

In matrix-vector form:
$$\beta_{t-1} = \Psi(\psi_t \odot \beta_t).$$

The base case is:
$$\beta_T(i) = 1,$$
corresponding to the likelihood of a non-event beyond the final observation.

Forwards-Backwards Algorithm: *The smoothed posterior marginal is computed by combining α_t and β_t at each time step:*
$$\gamma_t(j) \propto \alpha_t(j)\beta_t(j).$$

This algorithm involves passing messages from left-to-right (for α_t) and right-to-left (for β_t), combining them at each node. It generalizes to belief propagation in more complex models.

6.2.4 The Viterbi algorithm

Definition (Viterbi Algorithm)**.** *The Viterbi algorithm computes the most probable sequence of states $z^*_{1:T}$ in a chain-structured graphical model, given the observations $x_{1:T}$. Formally, it solves the optimization problem:*

$$z^*_{1:T} = \arg\max_{z_{1:T}} P(z_{1:T} \mid x_{1:T}).$$

6.3 Learning for HMMs

To estimate the parameters $\theta = (\pi, A, B)$ of a Hidden Markov Model (HMM), where:

- $\pi(i) = P(z_1 = i)$: the initial state distribution,
- $A(i, j) = P(z_t = j \mid z_{t-1} = i)$: the transition matrix,
- B: parameters of the class-conditional densities $P(x_t \mid z_t = j)$,

we consider two scenarios:

1. **Observed States:** When $z_{1:T}$ is observed in the training set, parameter estimation is straightforward using maximum likelihood estimation.
2. **Hidden States:** When $z_{1:T}$ is hidden, estimation becomes more challenging and requires techniques like the Expectation-Maximization (EM) algorithm.

6.3.1 EM for HMMs (the Baum-Welch algorithm)

When the hidden states z_t are unobserved, parameter estimation for Hidden Markov Models (HMMs) is analogous to fitting a mixture model. The most common approach for this task is the **Expectation-Maximization (EM)** algorithm, which finds the Maximum Likelihood Estimate (MLE) or Maximum A Posteriori (MAP) parameters. When applied to HMMs, the EM algorithm is referred to as the **Baum-Welch algorithm**. The process is outlined below.

6.4 E Step

The expected complete data log likelihood is given by:

$$Q(\theta, \theta^{\text{old}}) = \mathbb{E}\left[\sum_{k=1}^{K} N_{k1} \log \pi_k + \sum_{j=1}^{K}\sum_{k=1}^{K} N_{jk} \log A_{jk} + \sum_{i=1}^{N}\sum_{t=1}^{T_i}\sum_{k=1}^{K} p(z_t = k \mid x_i, \theta^{\text{old}}) \log p(x_{i,t} \mid \phi_k)\right].$$

where the expected counts are given by:

$$\mathbb{E}[N_{jk}] = \sum_{i=1}^{N}\sum_{t=2}^{T_i} p(z_{i,t-1} = j, z_{i,t} = k \mid x_i, \theta^{\text{old}}),$$

$$\mathbb{E}[N_{k1}] = \sum_{i=1}^{N} p(z_{i1} = k \mid x_i, \theta^{\text{old}}).$$

These expected sufficient statistics can be computed by running the forwards-backwards algorithm on each sequence. In particular, this algorithm computes the following smoothed node and edge marginals:

$$\gamma_{i,t}(j) = p(z_t = j \mid x_{i,1:T_i}, \theta),$$

$$\xi_{i,t}(j, k) = p(z_{t-1} = j, z_t = k \mid x_{i,1:T_i}, \theta).$$

6.5 M Step

Using the expected sufficient statistics from the E step, the M step updates the parameters:

1. **Transition Matrix A:**
$$\hat{A}_{jk} = \frac{\mathbb{E}[N_{jk}]}{\sum_{k'} \mathbb{E}[N_{jk'}]},$$
where $\mathbb{E}[N_{jk}]$ is the expected number of transitions from j to k.
2. **Initial State Distribution π:**
$$\hat{\pi}_k = \frac{\mathbb{E}[N_{k1}]}{\sum_{k'} \mathbb{E}[N_{k'1}]}.$$
3. **Observation Model:**
   - *Multinoulli Observations:*
   $$\hat{B}_{jl} = \frac{\sum_{i=1}^{N} \sum_{t=1}^{T_i} \gamma_{i,t}(j) \mathbb{I}(x_{i,t} = l)}{\sum_{i=1}^{N} \sum_{t=1}^{T_i} \gamma_{i,t}(j)},$$
   where $\mathbb{I}(x_{i,t} = l)$ is an indicator function for the observation l.
   - *Gaussian Observations:* Compute the expected sufficient statistics:
   $$\mathbb{E}[x_k] = \sum_{i=1}^{N} \sum_{t=1}^{T_i} \gamma_{i,t}(k) x_{i,t},$$
   $$\mathbb{E}[xx^\top]_k = \sum_{i=1}^{N} \sum_{t=1}^{T_i} \gamma_{i,t}(k) x_{i,t} x_{i,t}^\top,$$
   and update the parameters:
   $$\hat{\mu}_k = \frac{\mathbb{E}[x_k]}{\mathbb{E}[N_k]},$$
   $$\hat{\Sigma}_k = \frac{\mathbb{E}[xx^\top]_k}{\mathbb{E}[N_k]} - \hat{\mu}_k \hat{\mu}_k^\top.$$

References

[1] KP Murphy. *Machine Learning–A probabilistic Perspective*. The MIT Press, 2012.

[2] Christopher M Bishop. Pattern recognition and machine learning. *Springer google schola*, 2:1122–1128, 2006.

[3] Ryan Martin. Stat 511–lecture notes ii exponential families, sufficiency & information. 2014.

[4] Christopher M Bishop and Hugh Bishop. *Deep learning: Foundations and concepts*. Springer Nature, 2023.

[5] Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media, 2013.