Solutions to “Pattern Recognition and Machine Learning” by Bishop

tommyod @ github

Finished May 2, 2019.
Last updated June 27, 2019.

Abstract

This document contains solutions to selected exercises from the book “Pattern Recognition and Machine Learning” by Christopher M. Bishop.

Written in 2006, PRML is one of the most popular books in the field of machine learning. It’s clearly written, never boring and exposes the reader to details without being terse or dry. At the time of writing, the book has close to 36 000 citations according to Google.

While short chapter summaries are included in this document, they are not intended to substitute the book in any way. The summaries will largely be meaningless without the book, which I recommend buying if you’re interested in the subject. The solutions and notes were typeset in LaTeX to facilitate my own learning process.

I hope you find my solutions helpful if you are stuck. Remember to make an attempt at solving the problems yourself before peeking. More likely than not, the solutions can be improved by a reader such as yourself. If you would like to contribute, please submit a pull request at `https://github.com/tommyod/lml/`.

Several similar projects exist: there’s an official solution manual, a repository with many solutions at `https://github.com/GoldenCheese/PRML-Solution-Manual` and a detailed errata located at `https://github.com/yousuketakada/prml_errata`.

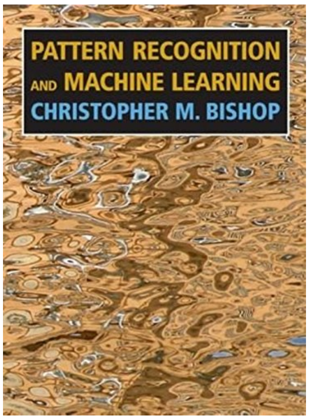


Figure 1: The front cover of [Bishop, 2006].

Contents

1 Chapter summaries

Notation

Scalar data is given by $\mathsf{x} = (x_1, \ldots, x_N)^T$, where N is the number of samples. Vector data is given by $\boldsymbol{X}$, which has dimensions $N \times M$, where N is the number of data points (rows) and M is the dimensionality of the feature space (columns).

Mathematics

Some useful mathematics is summarized here, also see the book appendix.

- The *gamma function* $\Gamma(x)$ satisfies $\Gamma(x) = (x-1)\Gamma(x-1)$, and is given by

  $$\Gamma(x) = \int_0^\infty u^{x-1} e^{-u} \, du.$$

  It's a "continuous factorial," which is proved by integration by parts and induction.
- The *Jensen inequality* states that, for convex functions

  $$f\left(\sum_j \lambda_j x_j\right) \leq \sum_j \lambda_j f(x_j),$$

  where $\sum_j \lambda_j = 1$ and $\lambda_j \geq 0$ for every j.

1.1 Introduction

Probability

The joint probability is given by $p(x, y)$, which is short notation for $p(X = x_i \cap Y = y_j)$.

- The sum rule is

  $$p(x) = \sum_y p(x, y) = \int p(x, y) \, dy.$$

  - Applying the sum rule as above is called "marginalizing out y."
- The product rule is

  $$p(x, y) = p(x|y)p(y).$$

  - Computing $p(x|y)$ is called "conditioning on y."
- Let $\boldsymbol{w}$ be parameters and $\mathcal{D}$ be data. Bayes theorem is given by

  $$p(\boldsymbol{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{w})p(\boldsymbol{w})}{p(\mathcal{D})} \quad \Leftrightarrow \quad \text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}.$$

  - Frequentist: data $\mathcal{D}$ generated from a fixed $\boldsymbol{w}$.
  - Bayesian: data $\mathcal{D}$ fixed, find best $\boldsymbol{w}$ given this data.

- Frequentists generally quantify the properties of data driven quantities in light of the fixed model parameters, while Bayesians generally quantify the properties of unknown model parameters in light of observed data. See [VanderPlas, 2014].

Expectation and covariance

Let x be distributed with density $p(x)$, then

- The expectation of a function $f(x)$ defined over x with probability density $p(x)$ is

$$\mathbb{E}[f] = \sum_j f(x_j)p(x_j) = \int f(x)p(x)\,dx$$

- The variance of $f(x)$ is

$$\mathrm{var}[f] = \mathbb{E}\left[(f - \mathbb{E}[f])^2\right] = \mathbb{E}[f^2] - \mathbb{E}[f]^2$$

- The covariance of x and y given by

$$\mathrm{cov}[x, y] = \mathbb{E}_{x,y}\left[(x - \mathbb{E}[x])(y - \mathbb{E}[y])\right]$$

- The covariance matrix $\boldsymbol{\Sigma}$ has entries σ_{ij} corresponding to the covariance of variables i and j. Thus $\boldsymbol{\Sigma} = \boldsymbol{I}$ means no covariance. (Note that real data may have no covariance and still be dependent, i.e. have predictive power, $x_j = f(x_k)$ where f is non-linear. See “Anscombe’s quartet” on Wikipedia.)

Polynomial fitting

Let $y(x, \boldsymbol{w}) = \sum_{j=1}^{M} w_j x^j$ be a polynomial. We wish to fit this polynomial to values $\mathsf{x} = (x_1, \ldots, x_N)$ and $\mathsf{t} = (t_1, \ldots, t_N)$ i.e. a degree M polynomial fitting N data points.

- The maximum likelihood solution is to minimize

$$E(\boldsymbol{w}, \mathsf{x}) \propto \sum_{n=1}^{N} \left[y(x_n, \boldsymbol{w}) - t_n\right]^2.$$

- Regularization adds a weight-dependent error so that $\widetilde{E}(\boldsymbol{w}, \mathsf{x}) = E(\boldsymbol{w}, \mathsf{x}) + E(\boldsymbol{w})$. For instance, Ridge minimizes the 2-norm:

$$\widetilde{E}(\boldsymbol{w}, \mathsf{x}) \propto \sum_{n=1}^{N} \left[y(x_n, \boldsymbol{w}) - t_n\right]^2 + \lambda \left\|\boldsymbol{w}\right\|_2^2$$

While LASSO (Least Absolute Shrinkage and Selection Operator) minimizes and error with the 1-norm. Both are examples of *Tikhonov regularization.*

Gaussians

The multivariate Gaussian is given by

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

where $\boldsymbol{\mu}$ is the mean and $\boldsymbol{\Sigma}$ is the covariance. Working with the precision $\boldsymbol{\Lambda} := \boldsymbol{\Sigma}^{-1}$ is sometimes easier.

Parameter estimation

Let $\mathsf{x} = \{x_1, \ldots, x_N\}$ be a data set which is identically and independently distributed (i.i.d). The likelihood function for the Gaussian is

$$p\left(\mathsf{x}|\mu,\sigma^2\right) = \prod_{j=1}^{N}\mathcal{N}\left(x_j|\mu,\sigma^2\right).$$

Estimates for the parameters $\boldsymbol{\theta} = (\mu, \sigma^2)$ can be obtained by maximizing the likelihood, which is equivalent to maximizing the log-likelihood $\ln p\left(\mathsf{x}|\mu,\sigma^2\right)$.

- Maximizing the likelihood $p(\mathcal{D}|\boldsymbol{w})$ is equivalent to minimizing $E = \boldsymbol{e}^T\boldsymbol{e}$ in polynomial fitting.
- Maximizing the posterior $p(\boldsymbol{w}|\mathcal{D})$ (MAP) is equivalent to minimizing regularized sum of squares $E = \boldsymbol{e}^T\boldsymbol{e} + \lambda\boldsymbol{w}^T\boldsymbol{w}$.

Model selection

- Both model and model hyperparameters must be determined.
  - Minimize the error over the test set. Balance bias and variance. High bias can lead to underfitting, high variance can lead to overfitting.
- Split data into training, testing and validation.
- If data is scarce, using K-fold cross validation is an option.

Decision theory

- Assign $\boldsymbol{x}$ to a region $\mathcal{R}_j \subseteq \mathbb{R}^M$ corresponding to a class $\mathcal{C}_j$, which might or might not be the true class.
- Minimizing misclassification is done by assigning $\boldsymbol{x}$ to the $\mathcal{C}_j$ which maximizes the posterior $p(\mathcal{C}_j|\boldsymbol{x})$. This is equivalent to maximizing chance of begin correct.
- Loss function $L_{k,j}$ may be used, the loss function assigns a penalty when the true class and the predicted class differ. $L_{k,j} \neq L_{j,k}$ in general. Pick the $\mathcal{C}_j$ which minimizes expected loss, i.e. pick the class $\mathcal{C}_j$ which minimizes

$$\sum_k L_{k,j}p(\boldsymbol{x},\mathcal{C}_j).$$

Three general decision approaches in decreasing order of complexity: (1) inference with class conditional probabilities $p(\boldsymbol{x}|\mathcal{C}_j)$, (2) inference with posterior class probabilities $p(\mathcal{C}_j|\boldsymbol{x})$ and (3) discriminant function.

$$\begin{tikzcd}
\boldsymbol{x} \arrow[r, "\text{inference}"] \arrow[rr, bend right, "\text{discriminant function}"'] & p(\mathcal{C}_k, \boldsymbol{x}) \arrow[r, "\text{decision}"] & \mathcal{C}_k
\end{tikzcd}$$

Information theory

- $h(x) = -\ln p(x)$ measures the degree of surprise.
- The entropy is the expected surprised, defined as

$$\mathrm{H}[p] = \mathbb{E}[h] = -\sum_j p(x_j)\ln p(x_j) = -\int p(x)\ln p(x)\,dx$$

  and measures how many nats are needed to encode the optimal transmission of values drawn from $p(x)$.
  - Discrete entropy is maximized by the uniform distribution.
  - Continuous (or differential) entropy is maximized by the Gaussian.
- Conditional entropy is given by

$$\mathrm{H}[x|y] = -\sum_{i,j} p(x_i, y_j)\ln p(x_i|y_j) = -\iint p(x,y)\ln p(x_i|y_j)\,dx\,dy,$$

  and we have that $\mathrm{H}[x,y] = \mathrm{H}[y|x] + \mathrm{H}[x]$.
- The Kullback-Leibner divergence is given by

$$\mathrm{KL}(p\|q) = -\int p(x)\ln\left(\frac{q(x)}{p(x)}\right)\,dx$$

  and is interpreted as the additional information needed if using $q(x)$ to encode values instead of the correct $p(x)$.

1.2 Probability Distributions

Conjugate priors

- Since we know that

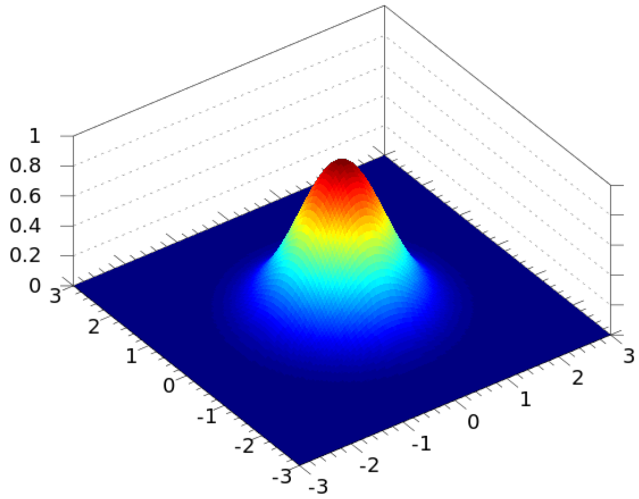
$$p(\boldsymbol{w}|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{w}) \times p(\boldsymbol{w})$$
$$\text{posterior} \propto \text{likelihood} \times \text{prior}$$

  we seek probability density functions such that the left hand side and the right hand side is of the same functional form. In other words, the likelihood $p(\mathcal{D}|\boldsymbol{w})$ is fixed, and we seek priors $p(\boldsymbol{w})$ such that posterior $p(\boldsymbol{w}|\mathcal{D})$ is of the same functional form. The idea is similar to eigenfunctions.

- Example: Since the binomial distribution is proportional to $p^k(1-p)^{n-k}$, the Beta distribution, proportional to $p^{\alpha-1}(1-p)^{\beta-1}$, is a conjugate prior. The product of these distributions then ensures that the posterior is of the same functional form as the prior.

Parameter	Conjugate prior
μ in Gaussian	Gaussian
p in Binomial	Beta-distribution
$\boldsymbol{p}$ in Multinomial	Dirichlet-distribution

The multidimensional Gaussian



- Gaussians arise naturally in sums $x_1 + \cdots + x_N$ and averages, since when $N \to \infty$ the sum is normally distributed by the *central limit theorem.*
- The multidimensional Gaussian can be diagonalized by diagonalizing the precision matrix $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$, then $\exp(\boldsymbol{x}^T\boldsymbol{\Lambda}\boldsymbol{x}) \cong \exp(\boldsymbol{y}^T\boldsymbol{D}\boldsymbol{y})$, where $\boldsymbol{D} = \mathrm{diag}(d_1, \ldots, d_D)$.
- One limitation is the unimodal nature of the Gaussian, i.e. it has a single peak.
- **Partitioned Gaussians.** Let $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$ and

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_a \\ \boldsymbol{x}_b \end{pmatrix} \quad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{pmatrix} \quad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb}. \end{pmatrix}.$$

  - **Conditional distribution**

$$\begin{aligned} p(\boldsymbol{x}_a|\boldsymbol{x}_b) &= \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_{a|b}, \boldsymbol{\Lambda}_{aa}^{-1}) \\ \boldsymbol{\mu}_{a|b} &= \boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{aa}^{-1}\boldsymbol{\Lambda}_{ab}(\boldsymbol{x}_b - \boldsymbol{\mu}_b) \end{aligned}$$

  - **Marginal distribution**

$$p(\boldsymbol{x}_a) = \mathcal{N}(\boldsymbol{x}_a|\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_{aa})$$

    - These results are proved using inverse of 2×2 block matrices and examining the quadratic and linear terms in the exponential.
- There also exist closed-form expressions for Bayes theorem when the prior and likelihood are Gaussians with linear relationship.

Bayesian inference

- Gaussian variables.
    - To estimate μ_N (σ^2 is assumed known), use Gaussian prior.
    - To estimate $\lambda = 1/\sigma^2$, use Gamma function as prior, i.e.

    $$\text{Gam}(\lambda|a,b) = \frac{b^a \lambda^{a-1}}{\Gamma(a)} \exp(-b\lambda)$$

    since it has the same functional form as the likelihood.
- The Student-t distribution may be motivated by:
    - Adding an infinite number of Gaussians with various precisions.
    - It's the distribution of the sample mean $(\bar{X} - \mu)/(S/\sqrt{n})$ when $x_1, \ldots, x_N$ are i.d.d. from a Gaussian.
    - As the degrees of freedom df → ∞, the Student-t distribution converges to a Gaussian. An important property of the Student-t distribution is it's *robustness* to outliers.

Periodic variables

- The mean can be measured as $\bar{\theta}$, where we think of the data as lying in a circle.
- The *von-Mises distribution* is a Gaussian on a periodic domain. It is given by

$$p(x|\theta_0, m) = \frac{1}{2\pi I_0(m)} \exp\left[m \cos(\theta - \theta_0)\right].$$

The exponential family

- The exponential family is given by

$$p(\boldsymbol{x}|\boldsymbol{\eta}) = g(\boldsymbol{\eta}) h(\boldsymbol{x}) \exp\left(\boldsymbol{\eta}^T u(\boldsymbol{x})\right)$$

  and many probability functions are members of this family. The entries of the vector $\boldsymbol{\eta}$ are called *natural parameters*, and $g(\boldsymbol{\eta})$ is a normalization constant.
- Maximum likelihood depends only on the *sufficient statistics* $\sum_n u(\boldsymbol{x}_n)$.
- Non-informative priors make few assumptions, letting the data speak for itself.

Nonparametric methods

- The general equation for density estimation is
$$p(\boldsymbol{x}) \simeq \frac{K}{NV}$$
where K is the number of points in a neighborhood of volume V and N is the total number of points.
- Kernel functions (or Parzen windows) estimate a neighborhood giving decreasing weight to samples further away, e.g. a Gaussian kernel. The volume V is fixed, the data (and kernel function) determines K.
- Nearest neighborhood fixes K, letting V be a function of the data.

1.3 Linear Models for Regression

Linear basis function models

- We assume the dependent data $\boldsymbol{y}$ may be written as
$$y(\boldsymbol{x}, \boldsymbol{w}) = \sum_{j=0}^{M-1} \boldsymbol{w}_j \phi_j(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}).$$
  - The function $y(\boldsymbol{x}, \boldsymbol{w})$ is linear in $\boldsymbol{w}$, but not necessarily in $\boldsymbol{x}$ since $\phi_j(\cdot)$ might be a non-linear function. It's called a *linear model* because it's linear in $\boldsymbol{w}$.
  - Choices for the functions $\{\phi_j\}$ include identity, powers of x, Gaussians, sigmoids, Fourier basis, Wavelet basis, and arbitrary non-linear functions.
- Assuming a noise term $\epsilon \sim \mathcal{N}(0, \beta^{-1})$, the maximum-likelihood solution is
$$\boldsymbol{w}_{\text{ML}} = \left(\boldsymbol{\Phi}^T \boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^T \boldsymbol{t} = \boldsymbol{\Phi}^\dagger \boldsymbol{t},$$
where $\boldsymbol{\Phi}^\dagger$ is the *Moore-Penrose pseudoinverse* and the *design matrix* $\boldsymbol{\Phi}$ has entries $\boldsymbol{\Phi}_{ij} = \phi_j(\boldsymbol{x}_i)$. The ML solution is equivalent to minimizing the sum of squares.
- Sequential learning is possible with e.g. the gradient descent algorithm, which is used to compute $\boldsymbol{w}^{(\tau+1)} = \boldsymbol{w}^\tau - \eta \nabla E_n$. This facilitates on-line learning.
- If there are multiple outputs which are linear in the same set of basis functions, the solution is $\boldsymbol{w}_k = \boldsymbol{\Phi}^\dagger \boldsymbol{t}_k$ for every output k, and the system decouples.
- Regularizing the error $E(\boldsymbol{w})$ with a quadratic term $\alpha \boldsymbol{w}^T \boldsymbol{w}/2$ has ML solution
$$\left(\alpha \boldsymbol{I} + \boldsymbol{\Phi}^T \boldsymbol{\Phi}\right) \boldsymbol{w} = \boldsymbol{\Phi}^T \boldsymbol{t}.$$
The solution above is equivalent to a prior $p(\boldsymbol{w} \mid \alpha) = \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{0}, \alpha^{-1}\boldsymbol{I})$.

The Bias-Variance decomposition

- The bias-variance decomposition is

$$\text{expected loss} = (\text{bais})^2 + \text{variance} + \text{noise}.$$

- Imagine drawing many data sets $\mathcal{D}$ from a distribution $p(t, \boldsymbol{x})$.
    - The **bias** is the distance from the average prediction to the conditional expectation $f(\boldsymbol{x}) = \mathbb{E}[t|\boldsymbol{x}]$. In other words:

$$\operatorname{Bias}\big[\hat{f}(\boldsymbol{x})\big] = \mathbb{E}\left[\hat{f}(\boldsymbol{x}) - f(\boldsymbol{x})\right]$$

    - The **variance** is the variability of $y(\boldsymbol{x}; \mathcal{D})$ around it's average.

$$\operatorname{Var}\big[\hat{f}(\boldsymbol{x})\big] = \mathbb{E}[\hat{f}(\boldsymbol{x})^2] - \mathrm{E}[\hat{f}(\boldsymbol{x})]^2$$

- Flexible models have high variance, while rigid models have high bias.

Bayesian linear regression

- We introduce a parameter distribution over $\boldsymbol{w}$, for instance an isotropic Gaussian distribution with covariance matrix $\boldsymbol{S}_0 = \alpha^{-1}\boldsymbol{I}$.

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{m}_0, \boldsymbol{S}_0)$$

Although the prior $p(\boldsymbol{w})$ is isotropic, the posterior $p(\boldsymbol{w} \mid \mathsf{t})$ need not be.

$$\begin{aligned} p(\boldsymbol{w}) &= \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{m}_0, \boldsymbol{S}_0) && \text{(prior)} \\ p(\mathsf{t} \mid \boldsymbol{w}) &= \prod_{n=1}^{N} p(t_n \mid \boldsymbol{w}) && \text{(likelihood)} \\ p(\boldsymbol{w} \mid \mathsf{t}) &= \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{m}_N, \boldsymbol{S}_N) && \text{(posterior)} \end{aligned}$$

- Analytical calculations are possible, leading to refinement of the posterior distribution of the parameters $\boldsymbol{w}$ as more data is seen.
- A predictive distribution $p(t|\mathsf{t}, \alpha, \beta)$ can be found. The predictive distribution accounts for uncertainty of the parameters α and β.
- The model may be expressed via an *equivalent kernel* $k(x, x_n)$ as

$$y(\boldsymbol{x}, \boldsymbol{w}) = y(\boldsymbol{x}, \boldsymbol{m}_N) = \beta\boldsymbol{\phi}(\boldsymbol{x})\boldsymbol{S}_N\boldsymbol{\Phi}^T\mathsf{t} = \sum_{n=1}^{N} \underbrace{\beta\boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{S}_N\boldsymbol{\phi}(\boldsymbol{x}_n)}_{k(x,x_n)} t_n$$

In this context, a kernel is a “similarity function,” a dot product in some space. This can reduce to lower dimensions and make computations faster.

Bayestian model selection and limitations

- Bayestian model selection uses Bayes theorem with models $\{\mathcal{M}_i\}$ and data $\mathcal{D}$ as

$$p(\mathcal{M}_i \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \mathcal{M}_i)p(\mathcal{M}_i)}{p(\mathcal{D})}$$

  where $\mathcal{M}_i$ is a model (distribution). It's possible to choose the best model given the data by evaluating the *model evidence* (or *marginal likelihood*) $p(\mathcal{D} \mid \mathcal{M}_i)$ via marginalization over $\boldsymbol{w}$.
- Some disadvantages of the simple linear model includes the fact that the functions $\phi_j(\boldsymbol{x})$ are fixed before data is observed, and the number of functions often grow exponentially with the number of inputs. These shortcomings are often alleviated in other models by the fact that data typically lies on a lower-dimensional manifold.

1.4 Linear Models for Classification

Least squares and Fisher's linear discriminant

- The following are non-probabilistic models, where the output is not a probability.
- *Least squares classification* minimizes a quadratic error function, and is analytically tractable. The results are not probabilities, the method is very sensitive to outliers, and does not necessarily give good results even when the data is linearly separable.
- *Fisher's linear discriminant* seeks to find $\boldsymbol{w}$ to minimize

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{S}_W \boldsymbol{w}} = \frac{\text{between class variance}}{\text{within class variance}}.$$

  The method projects data to a (hopefully) desirable subspace, where generative or discriminative methods may be used. Solved by $\boldsymbol{w} \propto \boldsymbol{S}_W^{-1}(\boldsymbol{m}_2 - \boldsymbol{m}_1)$.
- The *perceptron algorithm* find a separating plane if the data is linearly separable, but does terminate otherwise. Historically important.

Generalized linear models and probabilistic generative models

- A *generalized linear model* (GLM) is of the form $y(\boldsymbol{x}) = f\left(\boldsymbol{w}^T \boldsymbol{x} + w_0\right)$, where the *activation function* $f(\cdot)$ may be non-linear.
- A probabilistic model first models $p(\boldsymbol{x} \mid \mathcal{C}_k)$ and $p(\mathcal{C}_k)$, and then uses Bayes theorem to model $p(\mathcal{C}_k \mid \boldsymbol{x})$. From Bayes theorem, we have

$$p(\mathcal{C}_1 \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid \mathcal{C}_1)p(\mathcal{C}_1)}{p(\boldsymbol{x} \mid \mathcal{C}_1)p(\mathcal{C}_1) + p(\boldsymbol{x} \mid \mathcal{C}_2)p(\mathcal{C}_2)} = \underbrace{\frac{1}{1 + \exp(-a)}}_{\text{the sigmoid } \sigma(a)},$$

  where $a = \ln\left(p(\boldsymbol{x} \mid \mathcal{C}_1)p(\mathcal{C}_1)\right) - \ln\left(p(\boldsymbol{x} \mid \mathcal{C}_2)p(\mathcal{C}_2)\right)$. If we assume normal distributions with shared covariance matrices, then $a(\boldsymbol{x})$ is linear in $\boldsymbol{x}$.

Probabilistic discriminative models

- A probabilistic model finds $p(\mathcal{C}_k \mid \boldsymbol{x})$ directly, without modeling the class-conditional distribution of $p(\boldsymbol{x} \mid \mathcal{C}_k)$. In other words, we can determine $\boldsymbol{w}$ in the GLM $y(\boldsymbol{x}) = f\left(\boldsymbol{w}^T\boldsymbol{x} + w_0\right)$ directly. This entails fitting $\mathcal{O}(M)$ coefficients instead of $\mathcal{O}(M^2)$.
- To find $\boldsymbol{w}$ in $y(\boldsymbol{x}) = \sigma\left(\boldsymbol{w}^T\boldsymbol{x} + w_0\right)$, we minimize the *cross entropy*, given by the negative log-likelihood

$$E(\boldsymbol{w}) = -\ln \underbrace{p(\mathsf{t} \mid \boldsymbol{w})}_{\text{likelihood}} = -\sum_{n=1}^{M} \underbrace{t_n \ln(y_n) + (1-t_n)\ln(1-y_n)}_{\text{cross entropy}}.$$

- Using the *Newton-Raphson* method

$$\boldsymbol{w}^{n+1} = \boldsymbol{w}^n - \boldsymbol{H}^{-1}(\boldsymbol{w}^n)\nabla E(\boldsymbol{w}^n)$$

  on the error function $E(\boldsymbol{w})$ is an instance of the *iterative reweighed least squares* (IRLS) method. Every step involves a weighted least squares problem, and $E(\boldsymbol{w})$ has a unique global minimum.

The Laplace approximation

- The idea behind the *Laplace approximation* is to place a normal distribution on a mode $\boldsymbol{z}_0$ of the function $f(\boldsymbol{z})$

$$f(\boldsymbol{z}) \simeq f(\boldsymbol{z}_0)\exp\left[-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{z}_0)^T\boldsymbol{A}(\boldsymbol{z}-\boldsymbol{z}_0)\right] \qquad \boldsymbol{A} = -\nabla\nabla \ln f(\boldsymbol{z})\big|_{\boldsymbol{z}=\boldsymbol{z}_0}$$

- One application of Laplace approximation is Bayesian logistic regression, which is generally intractable. Using the Laplace approximation, an approximation to exact Bayesian inference is possible.

1.5 Neural networks

The basic idea of neural networks

- A neural network has a fixed number of adaptable basis functions. Unlike the algorithms considered so far, neural networks let the *activation functions* themselves be learned, not just the weights of their linear combinations. The notation is:

$$\begin{tikzcd} x_i,\ i=1,\dots,D \arrow[r, "w_{ji}^{(1)}"] \arrow[rr, bend left, "w_{ki}^{(3)}"] & z_j,\ j=1,\dots,M \arrow[r, "w_{kj}^{(2)}"] & y_k,\ k=1,\dots,K \end{tikzcd}$$

- In a two-layer network, the equation for the output is

$$y_k = \sigma\left(\sum_{j=0}^{M} w_{kj}^{(2)} h \underbrace{\left(\sum_{i=0}^{M} w_{ji}^{(1)} x_i \right)}_{z_j} \right),$$

  where $\sigma(\cdot)$ and $h(\cdot)$ are the activation functions in the final (3rd) and hidden (2nd) layer, respectively. They may differ in general.
  - In regression problems, the final activation function $\sigma(\cdot)$ is often the identity, and the error $E(\boldsymbol{w})$ is the sum-of-squares error.
  - In regression problems, the final activation function $\sigma(\cdot)$ is often the softmax, and the error $E(\boldsymbol{w})$ is the cross entropy.
- To see how the first layer learns activation functions, consider a D-D-2 network with softmax activation functions trained using cross entropy error. The first part of the network learns non-linear functions $\phi_j(\boldsymbol{x})$, which are used for logistic regression in the second part. This is perhaps best seen if we write

$$y_k = \sigma\left(\sum_{j=0}^{D} w_{kj}^{(2)} \phi_j(\boldsymbol{x}) \right), \qquad \phi_j(\boldsymbol{x}) = h\left(\sum_{i=0}^{D} w_{ji}^{(1)} x_i \right).$$

Network training

- Network training involves finding weights $\boldsymbol{w}$ to minimize $E(\boldsymbol{w})$, this is a non-convex optimization problem, typically solved iteratively, i.e.

$$\boldsymbol{w}^{k+1} = \boldsymbol{w}^k + \Delta \boldsymbol{w}^k$$

  where $\Delta \boldsymbol{w}^k$ is some update rule. For instance $\Delta \boldsymbol{w}^k = -\eta \nabla E\left(\boldsymbol{w}^k\right)$ for gradient descent, which requires gradient information.
- Computing the gradient $\nabla_{\boldsymbol{w}} E\left(\boldsymbol{w}^k\right)$ is done using back-propagation, which is application of the chain rule of calculus to the network. First information is propagated forward in the network, then an error is computed and the $\partial_{w_{jk}} E$ are found by propagating information backward in the network.
- Second order derivatives, i.e. the Hessian $\boldsymbol{H} = \nabla\nabla E$, may be approximated or computed. Approximation schemes include diagonal approximation, outer product approximation, inverse from outer product (using the Woodbury matrix identity) and finite differences. Exact evaluation is also possible, and computing fast multiplication is possible by considering the operator $\mathcal{R}\{\cdot\}$ in $\boldsymbol{v}^T\boldsymbol{H} = \boldsymbol{v}^T\nabla\nabla E = \mathcal{R}\{\nabla E\}$.

Neural network regularization

- Naively adding a regularization term such as $\widetilde{E}(\boldsymbol{w} = E(\boldsymbol{w}) + \alpha \boldsymbol{w}^T\boldsymbol{w}/2$ will be inconsistent with scaling properties. A better regularization is to use

$$\frac{\alpha_1}{2} \sum_{w \in \mathcal{W}_1} w^2 + \frac{\alpha_2}{2} \sum_{w \in \mathcal{W}_2} w^2$$

where $\mathcal{W}_1$ denotes (non-bias) weights in layer 1.

  - The regularization above is called *weight-decay*, since it corresponds to decaying weights while training by multiplying with a factor between 0 and 1. It can be shown that

$$\text{weight decay} \cong \text{early stopping.}$$

- Four ways to learn invariance is
  - **Augmenting the training data while learning**, using some function $\boldsymbol{s}(\boldsymbol{x}, \xi)$, where $\boldsymbol{s}(\boldsymbol{x}, 0) = \boldsymbol{x}$. As an example, the function might rotate an image slightly.
  - Use **tangent propagation regularization** to penalize the Jacobian of the neural network, with no penalization along the tangent of the transformation.
  - **Pre-process** the data and extract invariant features by hand.
  - **Build invariance into the structure** of the neural net, e.g. CNNs.
  - It can be shown that

$$\text{augmenting data while learning} \cong \text{tangent propagation regularizer.}$$

Soft weight sharing and mixture density networks

- Using *soft weight sharing*, we assume a Gausian mixture distribution as a prior over the weights in the network.
- *Mixture density networks* work well for *inverse problems*, where $p(t \mid \boldsymbol{x})$ might be multimodal. The approach is to use a neural network to learn π_k, $\boldsymbol{\mu}_k$ and σ_k^2 in the mixture density

$$p(t \mid \boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}\left(t \mid \boldsymbol{\mu}_k(\boldsymbol{x}), \boldsymbol{I}\sigma_k^2(\boldsymbol{x})\right).$$

  The $\{\pi_k\}$ have softmax activations in the output to enforce the summation constraint, while the $\{\sigma_k^2(\boldsymbol{x})\}$ have exponential activations to enforce positivity.
- A full Bayesian treatment of neural networks is not analytically intractable. However, using the Laplace approximation for the posterior parameter distribution, and alternative re-estimation of α and β it is possible to use approximate evidence approximation.

1.6 Kernel methods

Introduction to kernel methods

- The *dual representation* expresses a prediction $y(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x})$ entirely in terms of the N seen data points by use of the *kernel* $k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\phi}(\boldsymbol{x})$. This is in contrast to learning weights $\boldsymbol{w}$. The dual formulation of Ride regression is

$$y(\boldsymbol{x}) = \boldsymbol{k}(\boldsymbol{x})^T \left(\boldsymbol{K} + \lambda \boldsymbol{I}\right)^{-1} \mathsf{t}, \tag{1}$$

where $\boldsymbol{k}(\boldsymbol{x})^T = (k(\boldsymbol{x}_1, \boldsymbol{x}), k(\boldsymbol{x}_2, \boldsymbol{x}), \ldots, k(\boldsymbol{x}_N, \boldsymbol{x}))$ and $\boldsymbol{K}_{nm} = k(\boldsymbol{x}_n, \boldsymbol{x}_m)$.

  - Typically the number of data points N is much greater than the dimensionality M of the features $\boldsymbol{\phi}(\boldsymbol{x})$. Since (1) needs to invert $\boldsymbol{K} \in \mathbb{R}^{N \times N}$, it's not obviously useful. One advantage is that infinite dimensional feature mappings need not be computed explicitly. A simple example is $\boldsymbol{\phi}(x) = (1, x, x^2, \ldots)$, which is infinite dimensional, but $k(x, x')$ becomes $(1 - x^2)^{-1}$.
  - A kernel is valid if it corresponds to an inner product a feature space. A kernel is valid if $\boldsymbol{K}$ is positive definite for every possible $\boldsymbol{x}$. Valid kernels may be constructed from other valid kernels, for instance

$$k = k_1 + k_2, \qquad k = k_1 k_2, \qquad k = \exp(k_1), \qquad \ldots$$

    This is called *kernel engineering.*

- *Kernel regression* (the *Nadaraya-Watson* model) models $p(\boldsymbol{x}, t)$ as

$$p(\boldsymbol{x}, t) = \frac{1}{N} \sum_{n=1}^{N} f(\boldsymbol{x} - \boldsymbol{x}_n, t - t_n).$$

Gaussian processes

- A prior over weights $\boldsymbol{w} \sim p(\boldsymbol{w})$, implicitly defines a distribution over functions $y(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x})$. Predictions on a finite set of values is given by $\mathsf{y} = \boldsymbol{\Phi} \boldsymbol{w}$. Formulated in term of kernels, a *Gaussian process* is specified by

$$\mathbb{E}\left[y(\boldsymbol{x}_n) y(\boldsymbol{x}_m)\right] = k(\boldsymbol{x}_n, \boldsymbol{x}_m) = \operatorname{cov}\left[\mathsf{y}\right].$$

- The kernel may for instance be given by

$$k(\boldsymbol{x}_n, \boldsymbol{x}_m \mid \boldsymbol{\theta}) = \phi_0 \exp\left(-\frac{\theta_1}{2} \left\| \boldsymbol{x}_n - \boldsymbol{x}_m \right\|\right) + \theta_2 + \theta_3 \boldsymbol{x}_n^T \boldsymbol{x}_m.$$

  If $\boldsymbol{x}_n \approx \boldsymbol{x}_m$, then the kernel will be comparatively large and the covariance will be larger. In other words; points that are close are more highly correlated.

- In *Gaussian process regression*, the predicted mean and variance is given by

$$\mu(X_{N+1}) = \boldsymbol{k}^T \boldsymbol{C}_N^{-1} \mathsf{t}$$
$$\sigma^2(X_{N+1}) = c - \boldsymbol{k}^T \boldsymbol{C}_N^{-1} \boldsymbol{k},$$

  where $\boldsymbol{C}_N = k(\boldsymbol{x}_n, \boldsymbol{x}_m) + \beta^{-1} \delta_{nm}$ is the covariance matrix after observing N points.

Hyperparameters and extensions

- Hyperparameters $\boldsymbol{\theta}$ in $k(\boldsymbol{x}_n, \boldsymbol{x}_m \mid \boldsymbol{\theta})$ can be optimized by maximizing the log likelihood $\ln p(\mathsf{t} \mid \boldsymbol{\theta})$, which is in general non-convex. This is type 2 maximum likelihood.
- Gaussian processes can be used for classification, by composing the output with a sigmoid so that $y = \sigma(\boldsymbol{a}(\boldsymbol{x}))$. Not analytically tractable, but approximate methods exist: (1) variational inference, (2) expectation maximization and (3) Laplace approximation.

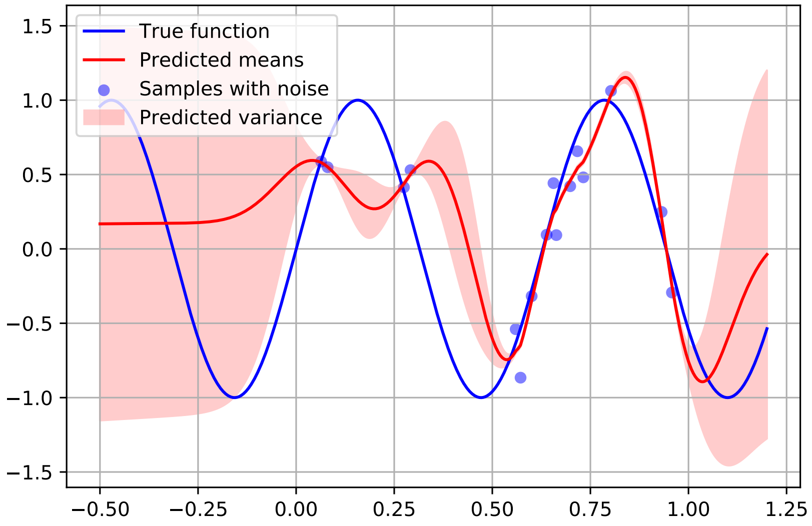


Figure 2: Gaussian process regression on data from $y_i = \sin(x_i) + \epsilon$.

1.7 Sparse Kernel Machines

Support vector machines

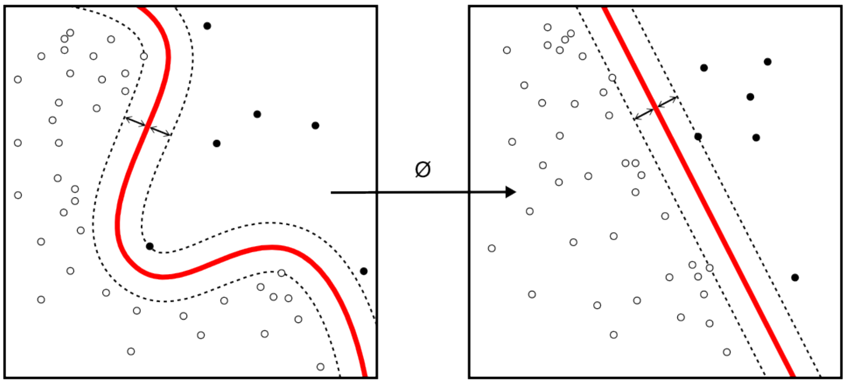


Figure 3: Data which is linearly separable in the feature vector space $\boldsymbol{\phi}(\boldsymbol{x})$, but not in $\boldsymbol{x}$-space. The maximum margin hyperplane is non-linear in $\boldsymbol{x}$-space. Source: Wikipedia.

- The main idea (when data is linearly separable) is to optimize $\boldsymbol{w}$ and b in the equation $y(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}) + b$ so that the separating hyperplane has a maximal margin. The points closest to the margin are called *support vectors*, and the hyperplane depends only on these points.
- Lagrange multipliers and the Karush-Kuhn-Tucker (KKT) conditions are needed to solve the problem. For the problem below, the Lagrangian is $L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) - \lambda g(\boldsymbol{x})$.

Optimization problem		KKT-conditions
minimize	$f(\boldsymbol{x})$	$g(\boldsymbol{x}) \geq 0, \lambda \geq 0$
subject to	$g(\boldsymbol{x}) \geq 0$	$\lambda g(\boldsymbol{x}) = 0$

The KKT conditions must be true at the optimum. They state that each constraint is either *active* or *inactive*, and generalize Lagrange multipliers to deal with inequality constraints.

- The **linearly separable classification problem** has Lagrange function

$$L(\boldsymbol{w}, b, \boldsymbol{a}) = \frac{1}{2} \|\boldsymbol{w}\|^2 - \sum_{n=1}^{N} a_n \left(t_n y_n - 1\right).$$

  Differentiating with respect to $\boldsymbol{w}$ and b, and substituting back into the Lagrange function yields the *dual form* of the optimization problem

$$\widehat{L}(\boldsymbol{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(\boldsymbol{x}_n, \boldsymbol{x}_m) \tag{2}$$

  which is expressed entirely in terms of the kernel $k(\boldsymbol{x}_n, \boldsymbol{x}_m)$ and the lagrange multipliers. The constraints are $a_n \geq 0$ and $\sum_n a_n t_n = 0$, and predictions are given by $y(\boldsymbol{x}) = \sum_n a_n t_n k(\boldsymbol{x}_n, \boldsymbol{x}) + b$. Since $a_n = 0$ when point n is not a support vector (the constraint is *inactive*), prediction relies only on the support vectors.

- The **linearly inseparable classification problem** minimizes

$$C \sum_{n=1}^{N} \xi_n + \frac{1}{2} \|\boldsymbol{w}\|^2,$$

  where ξ_n are *slack variables* and the constraint is $t_n y_n \geq 1 - \xi_n$ instead of $t_n y_n \geq 1$. The dual Lagrangian function $\widehat{L}(\boldsymbol{a})$ is exactly equal to Equation (2), but the constraints are now $0 \leq a_n \leq C$ (*box constraints*) and $\sum_n a_n t_n = 0$.

- The optimization problems above are quadratic programs (QP), and specialized methods for solving QP for SVMs exist: chunking, decomposition methods and sequential minimal optimization. The problems are often large, since $k(\boldsymbol{x}_n, \boldsymbol{x}_m)$ must be evaluated for every pair of points.

- The **regression problem** introduces a more robust error function

$$\frac{1}{2} \sum_{n} (y_n - t_n)^2 + \frac{\lambda}{2} \|\boldsymbol{w}\|^2 \quad \leftrightarrows \quad C \sum_{n} E_\epsilon \left(y_n - t_n\right) + \frac{1}{2} \|\boldsymbol{w}\|^2,$$

  where $E_\epsilon(\cdot)$ increases linearly outside of a margin of width 2ϵ. Two slack variables $\xi_n \geq 0$ and $\widehat{\xi}_n \geq 0$ are introduced, and minimizing the above is equivalent to

$$\underset{\boldsymbol{\xi}, \widehat{\boldsymbol{\xi}}, \boldsymbol{w}}{\text{minimize}} \quad C \sum_{n} (\xi_n + \widehat{\xi}_n) + \frac{1}{2} \|\boldsymbol{w}\|^2,$$

  and $\widehat{L}(\boldsymbol{a}, \widehat{\boldsymbol{a}})$ is again quadratic. The resulting optimization problem is a QP.

- The ν-SVM is mathematically equivalent to the above, but uses a parameter ν which controls the fraction of the data points that become support vectors. This is a more intuitive parameterization.

Relevance vector machines

- For regression, the model and prior are formulated as

$$p(t \mid \boldsymbol{x}, \boldsymbol{w}, \beta) = \mathcal{N}(t \mid y(\boldsymbol{x}, \boldsymbol{w}), \beta^{-1}), \quad \text{where } y(\boldsymbol{x}, \boldsymbol{w}) = \sum_n w_n k(\boldsymbol{x}_n, \boldsymbol{x}) + b$$

$$p(\boldsymbol{w} \mid \boldsymbol{\alpha}) = \prod_i^M \mathcal{N}(w_i \mid 0, \alpha_i^{-1}).$$

  The name *relevance vector machine* comes from the fact that automatic relevance determination is used to infer the important data points in an SVM-like function $y(\boldsymbol{x}, \boldsymbol{w})$, and these *relevance vectors* are analogous to support vectors.
- The hyperparameters $\{\alpha_i\}$ and β are efficiently determined through re-estimation equations. The expectation maximization algorithm is also an option.
- The model is typically very sparse, even more sparse than SVM, as many of the $\{\alpha_i\}$ are driven to infinity. The downside is that the optimization problem is non-convex, and in general takes $\mathcal{O}(M^3)$ time, where $M = N + 1$ if a SVM-like kernel is used. For classification, Laplace approximation may be used to derive results.

1.8 Graphical Models

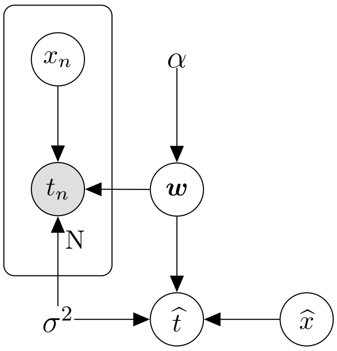
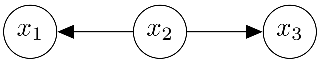


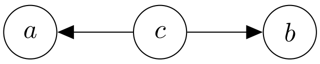
Figure 4: A graphical model for polynomial regression. Model parameters are shown without circles (e.g. σ^2), random variables are encircled (e.g. x_n), observed random variables are shaded (e.g. t_n) and *plate notation* is used to repeat the enclosed nodes.

Directed graphs

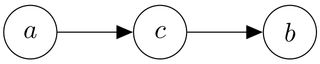
- A *graph* is associated with a factorization of the joint probability density function $p(\boldsymbol{x})$. For instance, the graph below means that $p(\boldsymbol{x})$ can be factored as $p(x_1 \mid x_2)p(x_2)p(x_3 \mid x_2)$. It imposes structure on $p(\boldsymbol{x})$ by it's *lack* of edges.



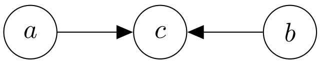
- Approaches to reducing the number of model parameters include:
  - Removing edges: induces factorization properties on $p(\boldsymbol{x})$.
  - Sharing parameters: merging parents together into a common node.
  - Restricting the functional form: for instance assuming $p(y = 1 \mid \boldsymbol{x}) = \sigma\left(\boldsymbol{w}^T \boldsymbol{x}\right)$.
- The three examples below demonstrate *conditional independence properties* in simple directed graphs. Observing c blocks the *tail-to-tail* and *head-to-tail* paths, but observing c (or any descendant of c) unblocks the *head-to-head* path.



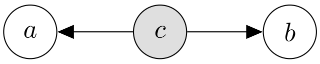
$a \not\perp\!\!\!\perp b \mid \emptyset$



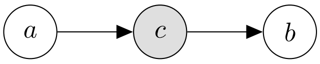
$a \not\perp\!\!\!\perp b \mid \emptyset$



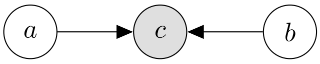
$a \perp\!\!\!\perp b \mid \emptyset$



$a \perp\!\!\!\perp b \mid c$



$a \perp\!\!\!\perp b \mid c$

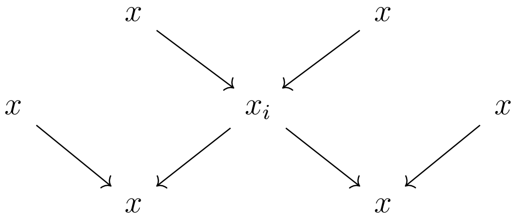


$a \not\perp\!\!\!\perp b \mid c$

- *D-separation.* Consider disjoint subsets A, B and C of edges in a graph. Let C be observed. A path from A to B is said to be blocked if

  - There is a tail-to-tail or head-to-tail path with a node in C in the path.
  - There is a head-to-head path, where the middle not is not in C, nor any of it's descendants.

  If *every* path from A to B is blocked, then $A \perp\!\!\!\perp B \mid C$.
- A graph can be thought of as a *filter*. The filter inputs are pdfs $p(\boldsymbol{x})$, and those factoring according to the given graph structure pass through the filter. For instance, $p(\boldsymbol{x}) = \prod_i p(x_i)$ would pass any such filter. The set $\mathcal{DF}$, for *directed factorization*, is the set of probability density functions passing through a specific filter.
- The *Markov blanket* of a graph is the minimal set of nodes that isolates x_i from the rest of the graph. In other words, $p(x_i \mid x_{\{i \neq i\}})$ is only functionally dependent on the nodes in the Markov blanket, which consists of parents, children and co-parents when the graph is directed.



  The Markov blanket in an undirected graph has a similar, but simpler structure; it only consists of the parents and children.

Markov random fields

- *Markov random fields* are undirected graphs whose nodes are random variables.

- Sets of nodes A and B are conditionally independent given C, denoted $A \perp\!\!\!\perp B \mid C$, if removing the nodes in C leaves no path from A to B. This is simpler than in the directed case, where d-separation and path blocking is more nuanced.
- The factors in the factorization of $p(\boldsymbol{x})$ are functions of the maximal *cliques* in the graph. Cliques are subsets of nodes which are fully connected. Let $\boldsymbol{x}_C$ be nodes associated with a clique C, then

$$\underbrace{\psi_C(\boldsymbol{x}_C)}_{\text{potential function}} = \underbrace{\exp\big(- \underbrace{E(\boldsymbol{x}_C)}_{\text{energy function}} \big)}_{\text{Boltzmann distribution}}$$

  is the *potential function* associated with a clique. The joint distribution is given by

$$p(\boldsymbol{x}) = \frac{1}{Z} \prod_C \psi_C(\boldsymbol{x}_C),$$

  where Z is a normalization constant called the *partition function.*
- A directed graph can be related to an undirected graph by *moralization.* This involves "marrying the parents" and converting directed edges to undirected edges. It represents a loss of structure.

Inference

- Inference on a *chain* is accomplished by sending *messages*; one forward and one backward. This let's us evaluate marginals $p(x_n)$ efficiently. If each variable has K possible states, the algorithm is $\mathcal{O}(NK^2)$ instead of the naive $\mathcal{O}(K^N)$.

$$\begin{tikzcd} x_1 \arrow[r] & x_2 \arrow[r] & \ldots \arrow[r] & x_N \end{tikzcd}$$

- *Factor graphs* comprise factor nodes and variable nodes. They can be constructed from undirected trees, directed trees and directed polytrees. A factor graph is bipartite, and they are used in the sum product algorithm.
- The *sum-product algorithm* facilitates efficient computation of marginals $p(\boldsymbol{x}_s)$ in factor graphs. It works by sending messages $\mu_{f \to x}(x)$ and $\mu_{x \to f}(x)$ from leaves to an arbitrary root note, then back to the leaves. Marginals are then computed as

$$p(x) = \prod_{s \in \operatorname{ne}(x)} \mu_{f_s \to x}(x).$$

- The *max-sum* algorithm finds the state $\boldsymbol{x}^{\max}$ maximizing the joint probability function, as well as the value $p(\boldsymbol{x}^{\max})$. The algorithm involves first sending messages from leaves to root, then *backtracking* to find the state $\boldsymbol{x}^{\max}$.

1.9 Mixture Models and EM

K-means and mixtures of Gaussians

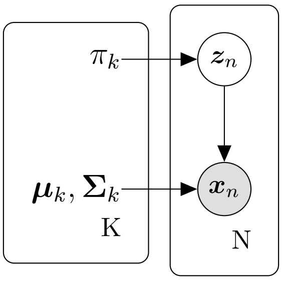
- In K-means classification, the EM algorithm minimizes the objective function

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \left\| \boldsymbol{x}_n - \boldsymbol{\mu}_k \right\|^2 .$$

  - The **expectation step** re-assigns points $\boldsymbol{x}_n$ to clusters via $r_{nk} \in \{0,1\}^K$.
  - The **maximization step** re-computes the k prototypes $\{\boldsymbol{\mu}_k\}$.
- *Mixtures of Gaussians* are given by the equation

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

  where π_k are the *mixing coefficients.* If $p(z_k = 1) = \pi_k$, we obtain the model below. For every $\boldsymbol{x}_n$, there is a latent (unobserved) $\boldsymbol{z}_n$ corresponding to which mixture $\boldsymbol{x}_n$ is from. For instance, if $\boldsymbol{z}_n = (0, 1, \ldots)$, then $\boldsymbol{x}_n$ is drawn from mixture number 2.



  - The EM algorithm for Gaussian mixtures is the following.
    - * The **expectation step** evaluates the *responsibilities* $\gamma(z_{nk})$, defined as the posterior probabilities $\gamma(z_{nk}) := p(z_{nk} = 1 \mid \boldsymbol{x}_n)$.
    - * The **maximization step** re-computes the $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ (functions of $\gamma(z_{nk})$).

The expectation-maximization (EM) algorithm

- The goal is to maximize the log likelihood $\ln p(\boldsymbol{X} \mid \boldsymbol{\theta})$. If this is difficult, but maximizing $\ln p(\boldsymbol{X}, \boldsymbol{Z} \mid \boldsymbol{\theta})$ is easier, then the EM algorithm is applicable.
  - The **expectation step** evaluates the posterior probability of the latent variables $\boldsymbol{Z}$ given $\boldsymbol{X}$ and $\boldsymbol{\theta}^{\text{old}}$. These are the responsibilities denoted by γ above.

$$p(\boldsymbol{Z} \mid \boldsymbol{X}, \boldsymbol{\theta}^{\text{old}}) = \frac{p(\boldsymbol{Z}, \boldsymbol{X} \mid \boldsymbol{\theta}^{\text{old}})}{\sum_{\boldsymbol{Z}} p(\boldsymbol{Z}, \boldsymbol{X} \mid \boldsymbol{\theta}^{\text{old}})}$$

  - The **maximization step** maximizes the expectation of the complete data log-likelihood over the posterior probability of the latent variables.

$$\begin{aligned}\boldsymbol{\theta}^{\text{new}} &= \arg\max_{\boldsymbol{\theta}} \quad Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) \\ Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) &= \mathbb{E}_{\boldsymbol{Z}} \underbrace{\left[\ln p(\boldsymbol{Z}, \boldsymbol{X} \mid \boldsymbol{\theta}^{\text{old}})\right]}_{\text{complete data log-likelihood}} = \sum_{\boldsymbol{Z}} p(\boldsymbol{Z} \mid \boldsymbol{X}, \boldsymbol{\theta}^{\text{old}}) \ln p(\boldsymbol{Z}, \boldsymbol{X} \mid \boldsymbol{\theta}^{\text{old}})\end{aligned}$$

    The maximization can often be accomplished by setting derivatives to zero.

- The general EM algorithm maximizes $\ln p(\boldsymbol{X} \mid \boldsymbol{\theta})$, which can be decomposed as

$$\ln p(\boldsymbol{X} \mid \boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta}) + \mathrm{KL}(q||p).$$

  - The **expectation step** maximizes the functional $\mathcal{L}(q, \boldsymbol{\theta})$ with respect to the distribution $q(\boldsymbol{Z})$ while keeping $\boldsymbol{\theta}$ constant. This amounts to minimizing $\mathrm{KL}(q||p)$, since $\ln p(\boldsymbol{X} \mid \boldsymbol{\theta})$ is not a function of $q(\boldsymbol{Z})$. The Kullback–Leibler divergence $\mathrm{KL}(q||p)$ is minimized when $q(\boldsymbol{Z}) = p(\boldsymbol{Z} \mid \boldsymbol{X}, \boldsymbol{\theta})$.
  - The **maximization step** maximizes the functional $\mathcal{L}(q, \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ while $q(\boldsymbol{Z})$ is held constant. This causes $\ln p(\boldsymbol{X} \mid \boldsymbol{\theta})$ to increase.
- Examples of EM include K-means, mixtures of Gaussian distributions, mixtures of Bernoulli distributions (latent class analysis) and EM for Bayesian linear regression.

1.10 Approximate Inference

The goal of approximate inference is to find the posterior distribution over the latent variables, i.e. $p(\boldsymbol{Z} \mid \boldsymbol{X})$, and take expectations with respect to this distribution.

The two approaches are either:

- Stochastic: converge eventually, but often slow in practice, e.g. sampling.
- Deterministic: approximate $p(\boldsymbol{Z} \mid \boldsymbol{X})$ analytically, .e.g. variational inference. This approach is “exactly wrong,” but often tractable. Uses a set of re-estimation equations.

Variational inteference and factorized distributions

- The log model evidence (marginal probability) $\ln p(\boldsymbol{X})$ can be decomposed as

$$\ln p(\boldsymbol{X}) = \mathcal{L}(q) + \mathrm{KL}(q||p),$$

  where $\mathrm{KL}(q||p)$ is the Kullback-Leibler divergence of $q(\boldsymbol{Z})$ with respect to $p(\boldsymbol{Z} \mid \boldsymbol{X})$. Recall that the distribution $q(\boldsymbol{Z})$ is arbitrary—the decomposition holds for any choice of $q(\boldsymbol{Z})$. Variational inference restricts the functional form of $q(\boldsymbol{Z})$ and maximizes $\mathcal{L}(q)$, which is equivalent to by minimizing the KL divergence $\mathrm{KL}(q||p)$.

- The technique of *factorized distributions* assumes that $q(\boldsymbol{Z})$ factors as $\prod_{i=1}^{M} q_i(\boldsymbol{Z}_i)$, and the minimizer for a single factor is given by

$$\ln q_j^{\star}(\boldsymbol{Z}_j) = \mathop{\mathbb{E}}_{i \neq j}\left[\ln p(\boldsymbol{Z}, \boldsymbol{X})\right] + \text{const} = \int \ln p(\boldsymbol{Z}, \boldsymbol{X}) \prod_{i \neq j}^{M} q(\boldsymbol{Z}_i)\, d\boldsymbol{Z}_i + \text{const},$$

  where the expectation $\mathop{\mathbb{E}}_{i \neq j}[\cdot]$ is taken with respect to all the M groups of variables in the factorized distribution $q(\boldsymbol{Z}) = \prod_{i=1}^{M} q_i(\boldsymbol{Z}_i)$ except the jth.
  - This leads to coupled *re-estimation* equations, since the optimal $q_j(\boldsymbol{Z}_j)$ is dependent on the other factors. The equations are solved by cycling through the groups of variables and solving each in turn.
  - Additional *induced factorizations* may naturally arise from interactions between the assumed factorizations $\prod_i q_i(\boldsymbol{Z}_i)$ and the conditionally independent properties of the true joint distribution $p(\boldsymbol{Z}, \boldsymbol{X})$.
- If we minimize the reverse KL divergence $\mathrm{KL}(p||q)$ instead of $\mathrm{KL}(q||p)$, we get the simple analytical solution

$$q_j^{\star}(\boldsymbol{Z}_j) = \int p(\boldsymbol{Z}) \prod_{i \neq j} d\boldsymbol{Z}_j = p(\boldsymbol{Z}_j).$$

  However, minimizing the reversed KL leads to averaging over several modes of the posterior $p(\boldsymbol{Z} \mid \boldsymbol{X})$, which in the context of mixtures yields bad solutions.

Examples of factorized distributions

- Some examples using factorized distributions given in the text are (1) univariate Gaussians (for tutorial purposes), (2) mixtures of Gaussians, (3) general model comparison and (3) linear regression.
  - Approximating the predictive distribution $p(\widehat{\boldsymbol{x}} \mid \boldsymbol{X})$ of possible unobserved values is possible, and evaluating the variational lower bound $\mathcal{L}(q)$ is also possible (this bound should never decrease, and provides a practical check that the math and convergence is correct).
- The assumed factorization in mixtures of Gaussians is

$$q(\boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = \underbrace{q(\boldsymbol{Z})}_{\text{variables}}\ \underbrace{q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})}_{\text{parameters}},$$

  and this approach of grouping variables and parameters separately leads to good results when the distributions are members of the exponential family. Variables (*extensive*) scale with observed data, while parameters (*intensive*) do not.
- There are several advantages of variational inference compared to EM. For mixture models of Gaussians singularities vanish when appropriate priors are used, and the model learns the optimal value of K automatically. Since there are $K!$ symmetries, proper regularization must be used however.

Local variational methods and expectation propagation

- *Local variational methods* bound (typically univariate) functions by means of simpler functions. An example is bounding convex functions $y(x)$ using a tangent linear function. When η is the slope parameterizing the tangent and $-g(\eta)$ is the intercept, convex duality states that

$$g(\eta) = \max_x \{\eta x - f(x)\} \qquad f(x) = \max_\eta \{\eta x - g(\eta)\}.$$

  If $y(x)$ is not convex, suitable transformations may be used.
- To see how local variational methods can be applied, consider the integral

$$I = \underbrace{\int \sigma(a)p(a)\, da}_{\text{intractable}} \geq \underbrace{\int f(a,\xi)p(a)\, da}_{\text{tractable}} = F(\xi).$$

  We can optimize $F(\xi)$ for it's maximizer $\xi^\star$, but note that this does not optimize $f(a,\xi)p(a)$ for every value of a, so it will in general not be an exact bound.
- *Expectation propagation* minimizes the reverse Kullback Leibler divergence. It's an iterate algorithm which cycles through factors, optimizing them one-by-one.

1.11 Sampling Methods

Basics

- The purpose of sampling methods is often to evaluate expectations such as

$$\mathbb{E}[f] = \int f(\boldsymbol{z})p(\boldsymbol{z})\, d\boldsymbol{z} \simeq \frac{1}{L}\sum_{\ell=1}^{L} f\left(\boldsymbol{z}^\ell\right)$$

  where $\{\boldsymbol{z}^1, \boldsymbol{z}^2, \ldots, \boldsymbol{z}^L\}$ are samples drawn from the probability density function $p(\boldsymbol{z})$. We assume that no analytical expression for the integral $\int f(\boldsymbol{z})p(\boldsymbol{z})\, d\boldsymbol{z}$ exists.
- If the indefinite integral of the desired probability distribution can be inverted, then a uniform sampler $p_y(y)$ can be used to find $q_z(z)$ by transformation. A software implementation of uniform sampling from $p_y(y)$ is almost always available.

$$\begin{tikzcd} \underset{\text{uniform}}{p_y(y)} \arrow[r, bend left, "f"] & \underset{\text{desired}}{q_z(z)} \arrow[l, bend left, "h"] \end{tikzcd}$$

- *Rejection sampling* uses a proposal distribution $kq(z)$ to sample from $\widetilde{p}(z)$. The proposal function is not normalized, and $kq(z) \geq \widetilde{p}(z)$ for every value of z.

  (a) A value z_0 is drawn from the proposal function $q(z)$.

  (b) A value u_0 is uniformly drawn from the interval $[0, kq(z_0)]$.

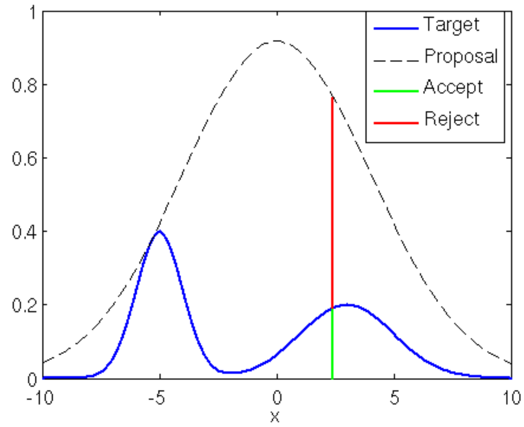


Figure 5: Rejection sampling visualized with $\widetilde{p}(z)$ in blue and $kq(z)$ in black. Source: `https://theclevermachine.files.wordpress.com`

(c) If $u_0 \leq \widetilde{p}(z)$, the sample z_0 is kept. If not, it's discarded.

Adaptive rejection sampling constructs the proposal function on the fly (typically piecewise linear in $\ln p(z)$). Sampled points which are not accepted are used to refine the proposal function.

- *Importance sampling* computes expectations without sampling from $\widetilde{p}(z)$. A proposal function $\widetilde{q}(z)$ is used, and the success of the algorithm depends crucially on how close $\widetilde{q}(z)$ is to $\widetilde{p}(z)$.

$$\mathbb{E}[f] = \int f(\boldsymbol{z})p(\boldsymbol{z})\,d\boldsymbol{z} \simeq \frac{1}{L}\sum_{\ell=1}^{L}\frac{p(\boldsymbol{z}^\ell)}{q(\boldsymbol{z}^\ell)}f(\boldsymbol{z}^\ell) = \frac{1}{L}\sum_{\ell=1}^{L} w_\ell f(\boldsymbol{z}^\ell)$$

The weights w_ℓ measure how close the distributions are at $\boldsymbol{z}^\ell$. Notice that in general the distributions are not required to be normalized.

- *Sampling importance-resampling* uses a sampling distribution $q(\boldsymbol{z})$. The use of a sampling distribution is similar to rejection sampling, but we need not determine the constant k required for $kq(z) \geq \widetilde{p}(z)$ in rejection sampling.

Initially L samples are drawn from $q(\boldsymbol{z})$, and then for each $\boldsymbol{z}^\ell$ a weight $w_\ell = p(\boldsymbol{z})/q(\boldsymbol{z})$ is assigned. New samples are then drawn (bootstrapped) from the distribution given by the samples $\boldsymbol{z}^1, \ldots, \boldsymbol{z}^L$ and corresponding probabilities $w_1, \ldots, w_L$.

MCMC, Gibbs sampling and Hybrid Monte Carlo

- The *Markov Chain Monte Carlo* (MCMC) sampler moves around on the p.d.f. $p(\boldsymbol{z})$ by means of a symmetric proposal distribution $q(\boldsymbol{z} \mid \boldsymbol{z}^\tau)$. A proposed point $\boldsymbol{z}^\star$ is accepted with acceptance probability $A(\boldsymbol{z}^\star, \boldsymbol{z}^\tau)$ given by

$$A(\boldsymbol{z}^\star, \boldsymbol{z}^\tau) = \min\left(1, \frac{\widetilde{p}(\boldsymbol{z}^\star)}{\widetilde{p}(\boldsymbol{z}^\tau)}\right).$$

If a point is not accepted, the previous point is again; this leads to duplicates of the same point. The effective sample size can be much lower than the apparent sample size due to the potentially high correlation of the samples, especially if $p(\boldsymbol{z})$ is multimodal, highly correlated or complicated in other ways.

  - The *Metropolis Hastings* algorithm does not require the proposal distribution to be symmetric. The acceptance probability for a possible transition k is

$$A_k(\boldsymbol{z}^\star, \boldsymbol{z}^\tau) = \min\left(1, \frac{\widetilde{p}(\boldsymbol{z}^\star) q_k(\boldsymbol{z}^\tau \mid \boldsymbol{z}^\star)}{\widetilde{p}(\boldsymbol{z}^\tau) q_k(\boldsymbol{z}^\star \mid \boldsymbol{z}^\tau)}\right).$$

  - *Gibbs sampling* cycles through the conditional probabilities $p(z_i \mid \boldsymbol{z}_{\setminus i})$ in turn and samples new z_is. It's a special case of the Metropolis Hastings algorithm, and the pratical applicability depends on being able to sample from $p(z_i \mid \boldsymbol{z}_{\setminus i})$. If this is not analytically feasible, simpler sampling algorithms can be used as sub-routines.

- *Hybrid Monte Carlo* moves over the probability density function by viewing it as a dynamical system. This reduces random walk behavior. The system's *Hamiltonian* function is given by

$$H(\boldsymbol{z}, \boldsymbol{r}) = \underbrace{E(\boldsymbol{z})}_{\text{potential}} + \underbrace{K(\boldsymbol{r})}_{\text{kinetic}},$$

where $\boldsymbol{r}$ is the momentum (velocity times mass in physics) and $\boldsymbol{z}$ the state.

  - *Leapfrog integration* integrates along a path where the Hamiltonian is constant, i.e. $\partial_t H = 0$, and the new proposed state is accepted with probability

$$\min\left(1, \exp\left[H(\boldsymbol{z}, \boldsymbol{r}) - H(\boldsymbol{z}^\star, \boldsymbol{r}^\star)\right]\right).$$

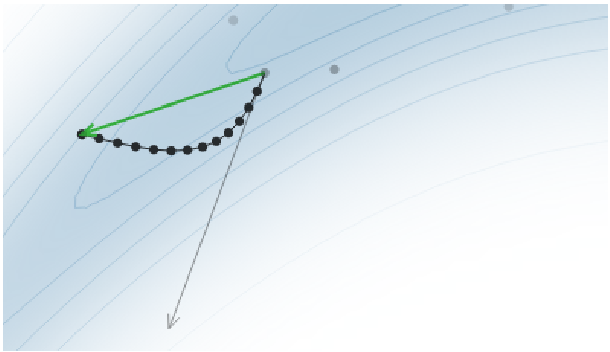


Figure 6: A Hybrid Monte Carlo step: the black dotted path shows the leapfrog integration path. Source: `https://chi-feng.github.io/mcmc-demo/app.html`

1.12 Continuous Latent Variables

Principal Component Analysis (PCA)

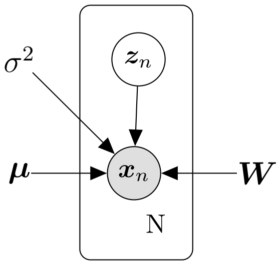
- The goal of PCA is to project D-dimensional data to $M \leq D$ dimensions. This is achieved by solving either one of the following equivalent optimization problems, whose analytical solution is an eigenvalue decomposition of the data covariance matrix.

  - Maximize the variance of the projected data.
  - Minimize the squared distance from the data $\boldsymbol{x}_n$ to it's projection $\tilde{\boldsymbol{x}}_n$, i.e.

$$J = \frac{1}{N} \sum_{n=1}^{N} \left\| \boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n \right\|^2 .$$

- Applications of PCA include
  - Visualization of data when $M = 2$.
  - Compression of data when $M < D$.
  - Whitening the data (diagonalizing the covariance matrix) when $M = D$
- Various algorithms exist for solving the eigenvalue problem, which has computational cost proportional to either $\mathcal{O}(D^3)$ or $\mathcal{O}(N^3)$. The SVD can be used for greater numerical stability, iterate algorithms exist, and sparse algorithms also exist.

Probabilistic Principal Component Analysis (PPCA)



- PPCA is a probabilistic model with a continuous M-dimensional latent variable z. The prior over the latent variable $\boldsymbol{z}$ and the conditional probability over the observed variable $\boldsymbol{x}$ respectively given by

$$p(\boldsymbol{z}) = \mathcal{N}\left(\boldsymbol{z} \mid \boldsymbol{0}, \boldsymbol{I}\right) \qquad p(\boldsymbol{x} \mid \boldsymbol{z}) = \mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu}, \sigma^2 \boldsymbol{I}\right) . \tag{3}$$

  The model is generative, and we can think of $\boldsymbol{x}$ as being generated by

$$\boldsymbol{x} = \boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}, \quad \text{where} \quad \boldsymbol{\epsilon} = \mathcal{N}\left(\boldsymbol{\epsilon} \mid \boldsymbol{0}, \sigma^2 \boldsymbol{I}\right) .$$

  This is a particular instance of the *linear-Gaussian* framework. Both the marginal distribution $p(\boldsymbol{x})$ and the posterior $p(\boldsymbol{z} \mid \boldsymbol{x})$ can be found analytically, and they are both governed by Gaussian distributions.
- If we let $\sigma^2 \to 0$, we recover the standard PCA model.
- Some advantages of the probabilistic formulation include: an iterative EM-algorithm that is sometimes computationally preferable, automatic determination of M from the data via automatic relevance determination, existence of a likelihood function and being able to run the model generatively.

- *Factor analysis* is a generalization of PPCA. The constant diagonal covariance $\sigma^2 \boldsymbol{I}$ in Equation (3) for PPCA is replaced by an arbitrary diagonal covariance $\boldsymbol{\Psi}$. The posterior then becomes becomes

$$p(\boldsymbol{x} \mid \boldsymbol{z}) = \mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu}, \boldsymbol{\Psi}\right).$$

The *independent variance* of $\boldsymbol{x}$ given $\boldsymbol{z}$ is explained by $\boldsymbol{\Psi}$, and the *covariance between variables* is explained by $\boldsymbol{W}$. No analytical solution is available, but the iterative EM-algorithm can be used to compute a maximum likelihood solution for the parameters $\boldsymbol{W}$ and $\boldsymbol{\Psi}$.

Kernel PCA and non-linear latent variable analysis

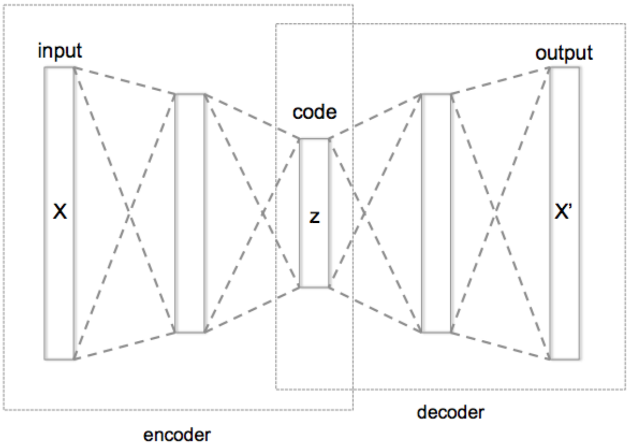


Figure 7: A dense autoencoder network with 4 layers of weights. Source: Wikipedia.

- *Kernel PCA* performs PCA in an M-dimensional feature space mapped to by a feature map $\boldsymbol{\phi} : \mathbb{R}^D \to \mathbb{R}^M$. However, the dot product $\boldsymbol{\phi}(\boldsymbol{x}_n)^T \boldsymbol{\phi}(\boldsymbol{x}_m)$ is never explicitly computed, instead the kernel function $k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\phi}(\boldsymbol{x}')$ is used. The computation involves the eigendecomposition of a $N \times N$ matrix, where N is the number of observations—so it can be computationally expensive.
- *Independent component analysis* (ICA) decomposes a multivariate signal into independent non-Gaussian signals. Non-linearity and non-Gaussianity are related, since a general density can be obtained by a non-linear transformation of a Gaussian.
- *Autoencoders* (or auto-associative neural networks) perform non-linear PCA when the number of layers with weights are 4. With 2 layers of weights, the optimal solution to the objective function (minimizing the sum-of-squares) is the PCA—even when the activation functions are non-linear.
- Some other methods for modeling non-linear manifolds are: principal curves, multidimensional scaling, local linear embedding, isometric feature mappings and self-organizing maps.

1.13 Sequential Data

MC, state space models and Hidden Markov Models (HMM)

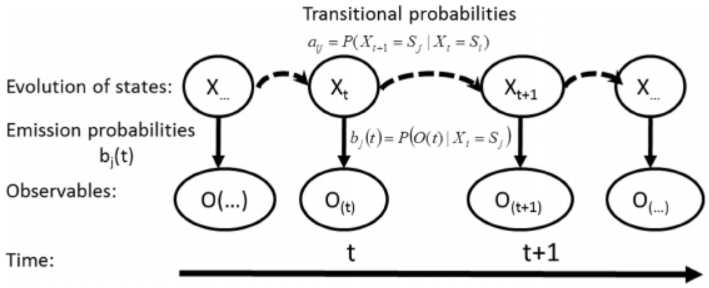


Figure 8: Hidden Markov Model with latent states and observed variables. The source is `https://www.researchgate.net/publication/278639262_micromachines_Reciprocal_Estimation_of_Pedestrian_Location_and_Motion_State_toward_a_Smartphone_Geo-Context_Computing_Solution`.

- An Mth order *Markov Chain* assumes that the probability of $\boldsymbol{x}_n$ is conditional on the previous M observations $\boldsymbol{x}_{n-1}, \boldsymbol{x}_{n-2}, \ldots, \boldsymbol{x}_{n-M}$.
- A *state space model* introduces a set of latent variables $\boldsymbol{z}_n$ in addition to the observed variables $\boldsymbol{x}_n$. It's governed by probability density functions

  $$p(\boldsymbol{z}_n \mid \boldsymbol{z}_{n-1}) \quad \text{and} \quad p(\boldsymbol{x}_n \mid \boldsymbol{z}_n).$$

  The key property is that $\boldsymbol{z}_{n+1} \perp\!\!\!\perp \boldsymbol{z}_{n-1} \mid \boldsymbol{z}_n$, i.e. when $\boldsymbol{z}_n$ is observed, observing $\boldsymbol{z}_{n\pm1}$ gives no additional information about $\boldsymbol{z}_{n\mp1}$.
- A *Hidden Markov Model* (HMM) is a state space model with discrete latent variables $\boldsymbol{z}_n$. At each time step, the HMM can be thought of as a mixture model.
  - Structuring the transitions probabilities in a matrix $\boldsymbol{A}$, we have

    $$p(\boldsymbol{z}_n \mid \boldsymbol{z}_{n-1}, \boldsymbol{A}) = \prod_{k=1}^{K} \prod_{k=1}^{K} A_{jk}^{z_{n-1,j} z_{nk}}.$$

- To perform maximum likelihood in a HMM, the iterative EM algorithm is used. In the E-step forward and backward messages are sent using a forward-backward algorithm, and in the M step these are used to maximize the model parameters.
- The most likely sequence of states is found using the Viterbi algorithm, which is the general max-sum algorithm in the context of HMMs.
- Extensions include: (1) sampling $p(T \mid k)$, which is the duration T of a state k, upon entering that state, (2) autoregressive HMMs, (3) input-output HMMs and (4) factorial HMMs, which have several latent sequences.

Linear dynamical systems

- A *linear dynamical system* is a linear-Gaussian model of the form:

$$\begin{aligned} p(\boldsymbol{z}_n \mid \boldsymbol{z}_{n-1}) &= \mathcal{N}\left(\boldsymbol{z}_n \mid \boldsymbol{A}\boldsymbol{z}_{n-1}, \boldsymbol{\Gamma}\right) && \text{(transitions)} \\ p(\boldsymbol{x}_n \mid \boldsymbol{z}_n) &= \mathcal{N}\left(\boldsymbol{x}_n \mid \boldsymbol{C}\boldsymbol{z}_{n-1}, \boldsymbol{\Sigma}\right) && \text{(emissions)} \\ p(\boldsymbol{z}_1) &= \mathcal{N}\left(\boldsymbol{z}_1 \mid \boldsymbol{\mu}_0, \boldsymbol{P}_0\right) && \text{(initial value)} \end{aligned}$$

- The sum-product algorithm (Kalman filter and Kalman smoother equations) can be used to find the marginal distribution $p(\boldsymbol{z})$ of the latent variables conditional on the observation sequence.
- Inference of the model parameters is possible using the EM algorithm.
- Extensions include allowing the $\boldsymbol{z}_n$ to be governed by mixtures of Gaussians as well as combining hidden Markov models with LDS.

1.14 Combining Models

Bagging and boosting

- In *bayesian model averaging* we assume that the dataset $\boldsymbol{X}$ is generated by one model h, and we try to determine which one. As more data is seen, $p(h \mid \boldsymbol{X})$ peaks for some model h.
  In contrast, *model combination* allows for data in different regions of the input space to potentially be generated by different models.
- *Bagging* (committees) reduces the variance of predictions by bootstrapping the data set, fitting several models, and averaging predictions.
- *Boosting* trains a sequence of models y_ℓ. A common boosting method is AdaBoost, which minimizes the exponential error function

$$E = \sum_{n=1}^{N} \exp\left(-t_n f_m(\boldsymbol{x}_n)\right), \quad \text{where} \quad f_m(\boldsymbol{x}) = \frac{1}{2}\sum_{\ell=1}^{M} \alpha_\ell \underbrace{y_\ell(\boldsymbol{x})}_{\text{weak learner}} .$$

Conditional mixture models

- A mixture of linear regression models can be defined by

$$p(t \mid \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(t \mid \boldsymbol{w}_k^T \boldsymbol{\phi}, \beta^{-1}).$$

  Optimal values for the model parameters can be found using EM and weighted least squares. The model $p(t \mid \boldsymbol{\theta})$ can have k modes, and will potentially assign probability to regions with no data points.
- In a *mixture of experts* the model weights $\pi_k(\boldsymbol{x})$ are functions of the input $\boldsymbol{x}$.

2 Exercises

2.1 Introduction

Exercise 1.2

We start with Equation (1.4) from the book and differentiate it with respect to w_i:

$$\begin{aligned}
\partial_{w_i}\tilde{E}(\boldsymbol{w}) &= \sum_{n=1}^{N}\left[y(x_n,\boldsymbol{w})-t_n\right]\partial_{w_i}y(x_n,\boldsymbol{w})+\lambda w_i \\
&= \sum_{n=1}^{N}\left[y(x_n,\boldsymbol{w})-t_n\right](x_n)^i+\lambda w_i \\
&= \sum_{n=1}^{N}\left[\sum_{j=0}^{M}w_j(x_n)^j-t_n\right](x_n)^i+\lambda w_i=0
\end{aligned}$$

Multiplying through the factor $(x_n)^i$ and rearranging the terms yields

$$\sum_{j=0}^{M}w_j\underbrace{\sum_{n=1}^{N}(x_n)^{i+j}}_{A_{ij}}+\lambda w_i=\underbrace{\sum_{n=1}^{N}(x_n)^i t_n}_{T_{ij}},$$

where the definitions of A_{ij} and T_{ij} are identical to those given in Exercise 1.1. Finally we employ the Kronecker delta symbol δ_{ij} to pull the w_i into the sum, since $\lambda w_i = \sum_{j=0}^{M}\lambda\delta_{ij}w_j$ we have

$$\sum_{j=0}^{M}\left(A_{ij}+\delta_{ij}\lambda\right)w_j=T_{ij}.$$

This solves the problem. Notice that in vector notation this system can be written as

$$\left(\boldsymbol{\Phi}^T\boldsymbol{\Phi}+\boldsymbol{I}\lambda\right)\boldsymbol{w}=\boldsymbol{\Phi}^T\boldsymbol{t},$$

where $\boldsymbol{\Phi}_{ij}=(x_i)^j$. Solving the system solves the regularized polynomial fitting problem.

Exercise 1.8

We first show that $\mathbb{E}[x]=\mu$. We define $K(\sigma)=1/\sqrt{2\pi}\sigma$ and evaluate the integral

$$\begin{aligned}
\mathbb{E}[x] &= \int K(\sigma)\exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)x\,dx \\
&= \int K(\sigma)\exp\left(-\frac{1}{2\sigma^2}z^2\right)(z+\mu)\,dz \qquad \text{(change of variables)} \\
&= \int K(\sigma)\exp\left(-\frac{1}{2\sigma^2}z^2\right)z\,dz+\mu\int K(\sigma)\exp\left(-\frac{1}{2\sigma^2}z^2\right)dz \\
&= 0+\mu,
\end{aligned}$$

where the first integral in the second to last line is zero because it's an odd function integrated over the real line, and the second integral evaluates to μ since the integrand is unity (it's a centered normal distribution, which has integral 1).

The second part of the problem asks us to verify that $\mathbb{E}\left[x^2\right] = \mu^2 + \sigma^2$. We factor the normal distribution as $\mathcal{N}\left(x \mid \mu, \sigma^2\right) = K(\sigma^2)E(\sigma^2)$, where

$$\begin{aligned} K(\sigma^2) &= \left(2\pi\sigma^2\right)^{-1/2} & \frac{\partial K}{\partial \sigma^2} &= -\left(2\pi\sigma^2\right)^{-3/2}\pi = -K(\sigma^2)^3\pi \\ E(\sigma^2) &= \exp\left(-\frac{1}{2\sigma^2}\left(x-\mu\right)^2\right) & \frac{\partial E}{\partial \sigma^2} &= \frac{1}{2\sigma^4}\left(x-\mu\right)^2 E(\sigma^2). \end{aligned}$$

We expedite notation by writing these functions as K and E, and their derivatives with respect to σ^2 as K' and E'. Using the product rule of calculus, we have

$$\begin{gathered} \frac{\partial}{\partial \sigma^2}\left(\int K(\sigma^2)E(\sigma^2)\,dx = 1\right) \\ \int K'E + KE'\,dx = 0 \\ \int KE\left(-\pi K^2 + \frac{1}{2\sigma^4}(x-\mu)^2\right)\,dx = 0. \end{gathered}$$

Substituting $-\pi K^2 = -1/(2\sigma^2)$, expanding the square term, multiplying out the KE term and performing the integrals, we obtain

$$-\frac{1}{2\sigma^2} + \frac{1}{2\sigma^4}\int KEx^2\,dx + 0 - \frac{\mu^2}{2\sigma^4} = 0,$$

and solving this for the unknown integral yields $\mathrm{E}\left[x^2\right] = \int KEx^2\,dx = \mu^2 + \sigma^2$ as required.

To show that Equation (1.51) from the book holds, notice that

$$\begin{aligned} \operatorname{var}\left[x\right] &= \mathbb{E}\left[(x - \mathbb{E}\left[x\right])^2\right] = \mathbb{E}\left[x^2 - 2x\mu + \mu^2\right] \\ &= \mathbb{E}\left[x^2\right] - 2\mu^2 + \mu^2 = \mathbb{E}\left[x^2\right] - \mathbb{E}\left[x\right]^2, \end{aligned}$$

where we have used μ interchangeably with $\mathbb{E}\left[x\right]$.

Exercise 1.10

Recall that the definition of the expected value is $\mathbb{E}\left[x\right] = \int p(x)x\,dx$, and that $\int p(x)\,dx = 1$ Statistical independence means that $p(x, y)$ factors as $p(x)p(y)$, so we have

$$\begin{aligned} \mathbb{E}\left[x+y\right] &= \iint p(x,y)(x+y)\,dx\,dy = \iint p(x)p(y)(x+y)\,dx\,dy && \text{(independence)} \\ &= \iint p(x)p(y)x\,dx\,dy + \iint p(x)p(y)y\,dx\,dy \\ &= \int p(y)\left(\int p(x)x\,dx\right)\,dy + \int p(y)y\left(\int p(x)\,dx\right)\,dy \\ &= \int p(y)\left(\mathbb{E}\left[x\right]\right)\,dy + \int p(y)y\,dy = \mathbb{E}\left[x\right] + \mathbb{E}\left[y\right] && \text{(by definition)} \end{aligned}$$

To show that $\mathrm{var}[x+y] = \mathrm{var}[x] + \mathrm{var}[y]$, we use the preceding result along with the definition from Equation (1.38) in [Bishop, 2006] $\mathrm{var}[x] = \mathbb{E}[(x - \mathbb{E}[x])^2]$ to write

$$\begin{aligned}
\mathrm{var}[x+y] &= \mathbb{E}\left[((x+y) - \mathbb{E}[x+y])^2\right] \\
&= \mathbb{E}\left[((x - \mathbb{E}[x]) + (y - \mathbb{E}[y]))^2\right] && \text{(rearranging)} \\
&= \mathbb{E}\left[(x - \mathbb{E}[x])^2 + 2(x - \mathbb{E}[x])(y - \mathbb{E}[y]) + (y - \mathbb{E}[y])^2\right] \\
&= \mathbb{E}\left[(x - \mathbb{E}[x])^2\right] + \underbrace{\mathbb{E}\left[2(x - \mathbb{E}[x])(y - \mathbb{E}[y])\right]}_{0} + \mathbb{E}\left[(y - \mathbb{E}[y])^2\right] \\
&= \mathrm{var}[x] + \mathrm{var}[y].
\end{aligned}$$

The cross term vanishes since x and y are independent. We will not show this in detail, but it can be shown by first noticing that $\mathbb{E}[xy] = \mathbb{E}[x]\,\mathbb{E}[y]$ when x and y are independent, and then showing that $\mathbb{E}[(x - \mathbb{E}[x])] = \mathbb{E}[x] - \mathbb{E}[x] = 0$.

Exercise 1.15

Due to the size of this problem, we split the solution into parts.

a) The redundancy is present due to the fact fact multiplication is commutative, so the weights may be factored out. For instance, when $M = 2$, we see that

$$w_{ij}x_ix_j + w_{ji}x_jx_i = (w_{ij} + w_{ji})x_ix_j = \widetilde{w}_{ij}x_ix_j.$$

We remove redundancy by ordering the products in a common term with $i_1 \geq i_2 \geq \cdots \geq i_M$. This ordering corresponds to Equation (1.134).

For instance, instead of summing over terms with $x_1x_2x_3$, $x_1x_3x_2$, $x_2x_1x_3$ and so forth, we make use of a common weight for the $x_3x_2x_1$-term.

b) The total number of terms equals the number of terms in the nested sum

$$n(D, M) = \sum_{i_1=1}^{D} \sum_{i_2=1}^{i_1} \cdots \sum_{i_M=1}^{i_{M-1}} 1,$$

which contains M sums. To prove the recursive formula, we expand the outer sum and notice that the result is D nested sums over $M - 1$ sums each. We have

$$\begin{aligned}
n(D, M) &= \sum_{i_1=1}^{D} \sum_{i_2=1}^{i_1} \cdots \sum_{i_M=1}^{i_{M-1}} 1 \\
&= \left(\sum_{i_2=1}^{1} \cdots \sum_{i_M=1}^{i_{M-1}} 1 \right) + \left(\sum_{i_2=1}^{2} \cdots \sum_{i_M=1}^{i_{M-1}} 1 \right) + \cdots + \left(\sum_{i_2=1}^{D} \cdots \sum_{i_M=1}^{i_{M-1}} 1 \right) \\
&= n(D = 1, M - 1) + n(D = 2, M - 1) + \cdots + n(D = D, M - 1) \\
&= \sum_{i=1}^{D} n(i, M - 1).
\end{aligned}$$

c) We skip the base case, which is easily verified. Assuming the result holds for D, we we show that it holds for $D+1$ by writing

$$\sum_{i=1}^{D+1} \frac{(i+M-2)!}{(i-1)!\,(M-1)!} = \sum_{i=1}^{D} \frac{(i+M-2)!}{(i-1)!\,(M-1)!} + \frac{(D+M-1)!}{D!\,(M-1)!}.$$

The sum on the right hand side is the given result for D, which we assume is true. Substituting this fact, we write

$$\begin{aligned}\sum_{i=1}^{D+1} \frac{(i+M-2)!}{(i-1)!\,(M-1)!} &= \frac{(D+M-1)!}{(D-1)!\,M!} + \frac{(D+M-1)!}{D!\,(M-1)!} \\ &= \frac{(D+M-1)!\,D}{D!\,M!} + \frac{(D+M-1)!\,M}{D!\,M!} \\ &= \frac{(D+M-1)!\,(D+M)}{D!\,M!} = \frac{(D+M)!}{D!\,M!} \\ &= \frac{((D+1)+M-1)!}{((D+1)-1)!\,M!},\end{aligned}$$

which shows that the result holds for $D+1$ when it holds for D.

d) We skip the base case of the inductive argument, as it should be easy to carry out.

The inductive step is performed as follows. Below, the first equality comes from Equation (1.135) in the book, the second comes from assuming the result holds for $M-1$, and the third comes from Equation (1.136).

$$n(D,M) = \sum_{i=1}^{D} n(i, M-1) = \sum_{i=1}^{D} \frac{(i+M-2)!}{(i-1)!\,(M-1)!} = \frac{(D+M-1)!}{(D-1)!\,M!}.$$

Comparing the first and final expression, we observe that if we assume the relation holds for $M-1$, it does indeed hold for M too.

Exercise 1.21

Starting with the inequality $a \leq b$, we multiply both sides by $a > 0$ to obtain $a^2 \leq ab$. We can take the square root of both sides and preserve the inequality, since the square root is monotonically increasing. Doing so, we obtain the desired inequality.

To prove the integral inequality, we apply the inequality on each term in Equation (1.78), then replace the integral over $\mathcal{R}_1 \cup \mathcal{R}_2$ with the real line:

$$\begin{aligned}p(\text{mistake}) &= \int_{\mathcal{R}_1} p(\boldsymbol{x}, \mathcal{C}_2)\, d\boldsymbol{x} + \int_{\mathcal{R}_2} p(\boldsymbol{x}, \mathcal{C}_1)\, d\boldsymbol{x} \\ &\leq \int_{\mathcal{R}_1} \left\{p(\boldsymbol{x}, \mathcal{C}_1)\, p(\boldsymbol{x}, \mathcal{C}_2)\right\}^{1/2} d\boldsymbol{x} + \int_{\mathcal{R}_2} \left\{p(\boldsymbol{x}, \mathcal{C}_2)\, p(\boldsymbol{x}, \mathcal{C}_1)\right\}^{1/2} d\boldsymbol{x} \\ &= \int \left\{p(\boldsymbol{x}, \mathcal{C}_2)\, p(\boldsymbol{x}, \mathcal{C}_1)\right\}^{1/2} d\boldsymbol{x}\end{aligned}$$

Exercise 1.25

This closely follows the derivation in Section 1.5.5. If $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{t} \in \mathbb{R}^m$, we view $\mathbb{E}\left[L(\boldsymbol{t}, \boldsymbol{y}(\boldsymbol{x}))\right]$ as a functional from the set of functions $\{f \mid f : \mathbb{R}^n \to \mathbb{R}^m\}$ to $\mathbb{R}$. Comparing the ordinary derivative and the functional derivative, we see that

$$\begin{aligned} f(\boldsymbol{x} + \boldsymbol{\epsilon}) &= f(\boldsymbol{x}) + \boldsymbol{\epsilon}^T \nabla f(\boldsymbol{x}), \quad \text{and} \\ F(\boldsymbol{y}(\boldsymbol{x}) + \epsilon \boldsymbol{\eta}(\boldsymbol{x})) &= F(\boldsymbol{y}(\boldsymbol{x})) + \epsilon \int \int \boldsymbol{\eta}^T \delta_{\boldsymbol{y}_j(\boldsymbol{x})} F(\boldsymbol{y}(\boldsymbol{x}))\, d\boldsymbol{t}\, d\boldsymbol{x} \\ &= F(\boldsymbol{y}(\boldsymbol{x})) + \epsilon \sum_{j=1}^{n} \int \underbrace{\left(\int \frac{\delta F(\boldsymbol{y}(\boldsymbol{x}))}{\delta \boldsymbol{y}_j(\boldsymbol{x})}\, d\boldsymbol{t} \right)}_{\text{must be 0}} \eta_j(\boldsymbol{x})\, d\boldsymbol{x}. \end{aligned}$$

The above condition implies that

$$\left(\int \frac{\delta F(\boldsymbol{y}(\boldsymbol{x}))}{\delta \boldsymbol{y}_j(\boldsymbol{x})}\, d\boldsymbol{t} \right) = 2 \int (\boldsymbol{y}_j(\boldsymbol{x}) - \boldsymbol{t}_j) p(\boldsymbol{x}, \boldsymbol{t})\, d\boldsymbol{t} = 0,$$

which, following the derivation leading to Equation (1.89), leads to

$$\boldsymbol{y}_j(\boldsymbol{x}) = \int t_j p(\boldsymbol{t} \mid \boldsymbol{x})\, d\boldsymbol{t} = \mathbb{E}_{\boldsymbol{t}}\left[t_j \mid \boldsymbol{x}\right].$$

This applies to any component j of $\boldsymbol{y}(\boldsymbol{x})$, so $\boldsymbol{y}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{t}}\left[\boldsymbol{t} \mid \boldsymbol{x}\right]$. To show that this reduces to Equation (1.89) in the case of a single target variable, simply define $\boldsymbol{t} := (t)$, i.e. a vector with one component.

Exercise 1.33

A hint as to why this is true is given on page 54, which states that $\mathrm{H}\left[y \mid x\right]$ is "the average additional information needed to specify y, given x." If the conditional entropy is zero, then y must be completely specified by x, i.e. a function of x. If $\mathrm{H}\left[y \mid x\right] = 0$, then

$$\sum_i \sum_j p(x_i, y_j) \ln p(y_j \mid x_i) = 0.$$

Using $p(x_i, y_j) = p(y_j \mid x_i) p(x_i)$ and rearranging the sums, we obtain

$$\sum_i p(x_i) \left[\sum_j p(y_j \mid x_i) \ln p(y_j \mid x_i) \right] = 0.$$

We know that $0 \leq p(x_i) \leq 1$ for every x_i. Assuming now that $p(x_i) > 0$, the bracketed term must be zero for every i, i.e.

$$\sum_j p(y_j \mid x_i) \ln p(y_j \mid x_i) = 0 \quad \text{for every } i.$$

Consider now the term $p(y_j \mid x_i) \ln p(y_j \mid x_i)$. The functional form of this term is $z \ln z$, and this function is negative except when $z = 0$ and when $z = 1$, where it is zero. Since we are adding terms that are either negative or zero, and the sum must evaluate to zero, every term in the sum must be zero. Therefore, $p(y_j \mid x_i)$ must be 0 or 1 for each value of y_j. However, since $\sum_j p(y_j \mid x_i) = 1$, every value of $p(y_j \mid x_i)$ must be zero except for one. In other words, x_i completely determines y_j.

Exercise 1.38

We wish to prove Equation (1.115) based on Equation (1.114). Clearly the case $M = 1$ is true, and we assume the case $M = 2$ is true, since this is given by Equation (1.114). For the inductive step, we assume the identity holds in the base case, and in the $M - 1$ case, and then

$$f\left(\sum_{i=1}^{M} \lambda_i x_i\right) = f\left(\sum_{i=1}^{M-1} \lambda_i x_i + \lambda_M x_M\right) = f\left(\frac{\sum_{k=1}^{M-1} \lambda_k}{\sum_{k=1}^{M-1} \lambda_k} \sum_{i=1}^{M-1} \lambda_i x_i + \lambda_M x_M\right).$$

The sum of $\sum_{k=1}^{M-1} \lambda_k$ and λ_M is unity, and $\sum_{i=1}^{M-1} \lambda_i x_i / \sum_{k=1}^{M-1} \lambda_k$ is a weighted average of x-values. Since it's a weighted average, it may be treated as just another x-value.

Applying the base case for two x values, we obtain

$$f\left(\frac{\sum_{k=1}^{M-1} \lambda_k}{\sum_{k=1}^{M-1} \lambda_k} \sum_{i=1}^{M-1} \lambda_i x_i + \lambda_M x_M\right) \leq \left(\sum_{k=1}^{M-1} \lambda_k\right) f\left(\frac{1}{\sum_{k=1}^{M-1} \lambda_k} \sum_{i=1}^{M-1} \lambda_i x_i\right) + \lambda_M f\left(x_M\right).$$

Now we can appeal to the $M - 1$ case of the inequality, which we assume to be true by the induction hypothesis. We can appeal to it since the $\lambda_i / \sum_{k=1}^{M-1} \lambda_k$ are normalized and sum to unity in the equation above due to the normalizing factor in front of the sum. To ease notation, we define $\alpha = \sum_{k=1}^{M-1} \lambda_k$ and write

$$\alpha f\left(\sum_{i=1}^{M-1} \frac{\lambda_i}{\alpha} x_i\right) + \lambda_M f\left(x_M\right) \leq \alpha \left(\sum_{i=1}^{M-1} \frac{\lambda_i}{\alpha} f(x_i)\right) + \lambda_M f\left(x_M\right) = \sum_{i=1}^{M} \lambda_i f(x_i).$$

This proves the Jensen inequality.

2.2 Probability Distributions

Exercise 2.6

We split this problem into three parts: (a) expected value, (b) variance and (c) mode.

a) The **expected value** is computed from the definition as

$$\begin{aligned}\mathbb{E}[\mu] &= \int \mu p(\mu)\, d\mu = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \int_0^1 \mu\, \mu^{a-1}(1-\mu)^{b-1}\, d\mu \\ &= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \int_0^1 \mu^{(a+1)-1}(1-\mu)^{b-1}\, d\mu \qquad \text{(re-cast as Beta)} \\ &= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \frac{\Gamma(a+1)\Gamma(b)}{\Gamma(a+b+1)} = \frac{a}{a+b},\end{aligned}$$

where we used Equation (2.265) from [Bishop, 2006], along with $\Gamma(x+1) = x\Gamma(x)$.

b) To compute the **variance**, we will employ $\text{var}[\mu] = \mathbb{E}[\mu^2] - \mathbb{E}[\mu]^2$. Similarly to the sub-problem above, we observe that

$$\begin{aligned}\mathbb{E}[\mu^2] &= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \int_0^1 \mu^{(a+2)-1}(1-\mu)^{b-1}\, d\mu \\ &= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \frac{\Gamma(a+2)\Gamma(b)}{\Gamma(a+b+2)} = \frac{(a+1)a}{(a+b+1)(a+b)},\end{aligned}$$

where in the last equality we have used $\Gamma(x+2) = (x+1)x\Gamma(x)$, which is simply repeated application of $\Gamma(x+1) = x\Gamma(x)$ twice.

The remaining computation is

$$\text{var}[\mu] = \mathbb{E}[\mu^2] - \mathbb{E}[\mu]^2 = \frac{(a+1)a}{(a+b+1)(a+b)} - \left(\frac{a}{a+b}\right),$$

which after some manipulation equals the desired result. We omit the details.

c) The **mode** is the maximum of the distribution. We differentiate the p.d.f. using the product rule of calculus to obtain

$$p'(\mu) = (a-1)\mu a - 2(1-\mu)^{b-1} + \mu^{a-1}(b-1)(1-\mu)^{b-2}(-1) = 0.$$

Diving by μ^a and $(1-\mu)^b$ and rearranging, we obtain

$$\frac{(a-1)}{(b-1)} \frac{(1-\mu)}{\mu} = 1.$$

Solving the equation above for μ yields $(a-1)/(a+b-2)$ as required.

Exercise 2.8

The **first part** is to prove Equation (2.270). We apply the definitions of $E_x\left[x \mid y\right]$ and $E_y[f(y)]$ in turn, and then the product rule of probability. Doing so, we observe that

$$\begin{aligned}E_y\left[E_x\left[x \mid y\right]\right] &= E_y\left[\int xp(x \mid y)\, dx\right] = \int \left(\int xp(x \mid y)\, dx\right) p(y)\, dy \\ &= \iint xp(x,y)\, dx\, dy = \mathbb{E}[x].\end{aligned}$$

The **second part**, which consists of proving Equation (2.271), is slightly more involved. It helps to keep track of whether the quantities are constants, functions of x or functions of y. It's also useful to know, from the definition of conditional variance, that

$$\mathrm{var}_x[x \mid y] = \int p(x \mid y)\,(x - E_x\,[x \mid y])^2\,dx = \mathbb{E}_x[x^2 \mid y] - \mathbb{E}_x[x \mid y]^2. \tag{4}$$

The result above does not come as a surprise, since it's merely the familiar result for variance, i.e. $\mathrm{var}_x[x] = \mathbb{E}_x[x^2] - \mathbb{E}_x[x]^2$, conditioned on y in every term.

Let's examine the **first term** in the right hand side of (2.270) first, i.e. $\mathbb{E}_y\,[\mathrm{var}_x[x \mid y]]$. Using Equation (4) above, we see that

$$\begin{aligned}\mathbb{E}_y\,[\mathrm{var}_x[x \mid y]] &= \mathbb{E}_y\left[\mathbb{E}_x[x^2 \mid y] - \mathbb{E}_x[x \mid y]^2\right] = \mathbb{E}_y\left[\mathbb{E}_x[x^2 \mid y]\right] - \mathbb{E}_y\left[\mathbb{E}_x[x \mid y]^2\right] \\ &= \mathbb{E}[x^2] - \mathbb{E}_y\left[\mathbb{E}_x[x \mid y]^2\right], \end{aligned} \tag{5}$$

where we used Equation (2.270) from the book in the last equality.

We now investigate the **second term** in Eqn (2.270), i.e. $\mathrm{var}_y\,[\mathbb{E}_x\,[x \mid y]]$. Using the definition of variance (Equation (1.38) in [Bishop, 2006]), followed by Equation (2.270) from the book, and the linearity of the expected value, we obtain

$$\begin{aligned}\mathrm{var}_y\,[\mathbb{E}_x\,[x \mid y]] &= \mathbb{E}_y\left[\left(\underbrace{\mathbb{E}_x\,[x \mid y]}_{f(y)} - \underbrace{\mathbb{E}_y\,[\mathbb{E}_x\,[x \mid y]]}_{\text{scalar}}\right)^2\right] \\ &= \mathbb{E}_y\left[(\mathbb{E}_x\,[x \mid y] - \mathbb{E}[x])^2\right] && \text{(previous result)} \\ &= \mathbb{E}_y\left[\mathbb{E}_x\,[x \mid y]^2 - 2\,\mathbb{E}_x\,[x \mid y]\,\mathbb{E}[x] + \mathbb{E}[x]^2\right] && \text{(multiply)} \\ &= \mathbb{E}_y\left[\mathbb{E}_x\,[x \mid y]^2\right] - \mathbb{E}_y\,[2\,\mathbb{E}_x\,[x \mid y]\,\mathbb{E}[x]] + \mathbb{E}_y\left[\mathbb{E}[x]^2\right] && \text{(linearity)} \\ &= \mathbb{E}_y\left[\mathbb{E}_x\,[x \mid y]^2\right] - 2\,\mathbb{E}_y\,[\mathbb{E}_x\,[x \mid y]]\,\mathbb{E}[x] + \mathbb{E}[x]^2 && \text{(constants)} \\ &= \mathbb{E}_y\left[\mathbb{E}_x\,[x \mid y]^2\right] - \mathbb{E}[x]^2 && (6)\end{aligned}$$

Finally, adding Equations (5) and (6) produces the result we're after.

Exercise 2.15

The entropy is given by $\mathrm{H}[p(\boldsymbol{x})] = -\int p(\boldsymbol{x}) \ln p(\boldsymbol{x})\,d\boldsymbol{x}$ from it's definition, and we start by computing the logarithm of $p(\boldsymbol{x})$, which is

$$\ln p(\boldsymbol{x}) = -\frac{D}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{\Sigma}| - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}).$$

Defining Δ^2 as $(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})$ (see Equation (2.44) in the book), we have

$$\mathrm{H}[p(\boldsymbol{x})] = -\int p(\boldsymbol{x}) \ln p(\boldsymbol{x})\,d\boldsymbol{x} = \frac{D}{2}\ln(2\pi) + \frac{1}{2}\ln|\boldsymbol{\Sigma}| + \frac{1}{2}\int p(\boldsymbol{x})\Delta^2\,d\boldsymbol{x},$$

since $\int Kp(\boldsymbol{x})\,d\boldsymbol{x} = K$ for constants K such as $D\ln(2\pi)/2$. The only troublesome term is the last one, so what remains to do is show that $\int p(\boldsymbol{x})\Delta^2\,d\boldsymbol{x} = D$. If we can show this, then we've proven the equation given in the problem statement.

We will now show that $\int p(\boldsymbol{x})\Delta^2\,d\boldsymbol{x} = D$. First, however, we'll present a result for the univariate Gaussian, which we will need later on. Here's the univariate result, easily verified by means of partial integration.

$$\frac{1}{\sqrt{2\pi}\sigma}\int \exp\left(-\frac{x^2}{2\sigma^2}\right)\,dx = \frac{1}{\sqrt{2\pi}\sigma}\int \frac{x^2}{\sigma^2}\exp\left(-\frac{x^2}{2\sigma^2}\right)\,dx = 1 \tag{7}$$

We now rotate into $\boldsymbol{y}$-space by $\boldsymbol{y} = \boldsymbol{U}(\boldsymbol{x}-\boldsymbol{\mu})$, which diagonalizes $\boldsymbol{\Sigma}^{-1}$. From Equations (2.50) and (2.56) in [Bishop, 2006], we see that the integral $\int \Delta^2 p(\boldsymbol{x})\,d\boldsymbol{x}$ now becomes

$$\int_{\mathbb{R}^D}\left(\sum_{i=1}^{D}\frac{y_i^2}{\lambda_i}\right)\prod_{j=1}^{D}\frac{1}{(2\pi\lambda_j)^{1/2}}\exp\left(-\frac{y_j^2}{2\lambda_j}\right)\,d\boldsymbol{y}.$$

Multiplying the product into the sum, and integrating each term, we have

$$\sum_{i=1}^{D}\int_{\mathbb{R}^D}\frac{y_i^2}{\lambda_i}\prod_{j=1}^{D}\frac{1}{(2\pi\lambda_j)^{1/2}}\exp\left(-\frac{y_j^2}{2\lambda_j}\right)\,d\boldsymbol{y},$$

where $d\boldsymbol{y} := dy_1\,dy_2\,\cdots\,dy_D$. To clarify the meaning of the equation above, let's focus on term i in the outer sum and write out the products more explicitly, the above integral is

$$\left(\int\frac{1}{(2\pi\lambda_1)^{1/2}}\exp\left(-\frac{y_1^2}{2\lambda_1}\right)\,dy_1\right)\cdots\left(\int\frac{1}{(2\pi\lambda_D)^{1/2}}\exp\left(-\frac{y_D^2}{2\lambda_D}\right)\,dy_D\right), \tag{8}$$

but the ith factor is special, since it has the additional y_i^2/λ_i factor, i.e. it's given by

$$\left(\int\frac{1}{(2\pi\lambda_i)^{1/2}}\frac{y_i^2}{\lambda_i}\exp\left(-\frac{y_i^2}{2\lambda_i}\right)\,dy_i\right). \tag{9}$$

Now comes the crucial observation; every factor in (8) except the ith is a univariate Gaussian, so these integrals evaluate to 1. The special ith term (9) is in same functional form as Equation (7) (with $\lambda_i = \sigma^2$), so it also evaluates to 1. Therefore the product of integrals for every term i evaluates to a product of ones, and the sum is simply

$$\sum_{i=1}^{D}\int_{\mathbb{R}^D}\frac{y_i^2}{\lambda_i}\prod_{j=1}^{D}\frac{1}{(2\pi\lambda_j)^{1/2}}\exp\left(-\frac{y_j^2}{2\lambda_j}\right)\,d\boldsymbol{y} = \sum_{i=1}^{D}1 = D.$$

We've shown that $\int p(\boldsymbol{x})\Delta^2\,d\boldsymbol{x} = D$, and this solves the problem.

Exercise 2.20

We expand $\boldsymbol{a}$ in the eigenvector basis of $\boldsymbol{\Sigma}$, writing $\boldsymbol{a} = \sum_i \alpha_i \boldsymbol{u}_i$. Using the fact that $\boldsymbol{u}_i \boldsymbol{u}_j = \delta_{ij}$, we have

$$\begin{aligned}\boldsymbol{a}^T \boldsymbol{\Sigma} \boldsymbol{a} &= \boldsymbol{a}^T \boldsymbol{\Sigma} \sum_i \alpha_i \boldsymbol{u}_i = \boldsymbol{a}^T \sum_i \alpha_i \boldsymbol{\Sigma} \boldsymbol{u}_i = \boldsymbol{a}^T \sum_i \alpha_i \lambda_i \boldsymbol{u}_i \\ &= \sum_j \alpha_j \boldsymbol{u}_j^T \sum_i \alpha_i \lambda_i \boldsymbol{u}_i = \sum_i \alpha_i^2 \lambda_i.\end{aligned}$$

If $\lambda_i > 0$ for every i, then clearly the sum is positive—so it's sufficient. On the other hand, if the sum is positive, then no λ_i can be negative or zero. If it were, we could choose $\boldsymbol{a}$ in the direction of the corresponding eigenvector and obtain a negative sum, which would be a contradiction. In other words, it's a also necessary condition that $\lambda_i > 0$ for every i.

Exercise 2.27

We split this exercise into two parts.

a) We wish to show that $\mathbb{E}[\boldsymbol{x} + \boldsymbol{z}] = \mathbb{E}[\boldsymbol{x}] + \mathbb{E}[\boldsymbol{z}]$. Using the definition of the expected value, along with independence, we see that

$$\begin{aligned}\mathbb{E}[\boldsymbol{x} + \boldsymbol{z}] &= \iint (\boldsymbol{x} + \boldsymbol{z}) p(\boldsymbol{x}, \boldsymbol{z}) \, d\boldsymbol{x} \, d\boldsymbol{z} = \iint (\boldsymbol{x} + \boldsymbol{z}) p(\boldsymbol{x}) p(\boldsymbol{z}) \, d\boldsymbol{x} \, d\boldsymbol{z} \\ &= \int \boldsymbol{x} p(\boldsymbol{x}) p(\boldsymbol{z}) \, d\boldsymbol{x} \, d\boldsymbol{z} + \int \boldsymbol{z} p(\boldsymbol{x}) p(\boldsymbol{z}) \, d\boldsymbol{x} \, d\boldsymbol{z} = \mathbb{E}[\boldsymbol{x}] + \mathbb{E}[\boldsymbol{z}].\end{aligned}$$

b) As in the previous sub-problem, we use the definition to obtain

$$\begin{aligned}\operatorname{cov}[\boldsymbol{x} + \boldsymbol{z}] &= \mathbb{E}\left[(\boldsymbol{x} + \boldsymbol{z} - \mathbb{E}[\boldsymbol{x} + \boldsymbol{z}])(\boldsymbol{x} + \boldsymbol{z} - \mathbb{E}[\boldsymbol{x} + \boldsymbol{z}])^T\right] \\ &= \mathbb{E}\left[(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}] + \boldsymbol{z} - \mathbb{E}[\boldsymbol{z}])(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}] + \boldsymbol{z} - \mathbb{E}[\boldsymbol{z}])^T\right].\end{aligned}$$

We wish to expand the inner square term. In order to avoid too heavy notation, we introduce $\boldsymbol{a} := \boldsymbol{x} - \mathbb{E}[\boldsymbol{x}]$ and $\boldsymbol{b} := \boldsymbol{z} - \mathbb{E}[\boldsymbol{z}]$. From there, we observe that

$$\mathbb{E}\left[(\boldsymbol{a} + \boldsymbol{b})(\boldsymbol{a} + \boldsymbol{b})^T\right] = \mathbb{E}\left[\boldsymbol{a}\boldsymbol{a}^T + \boldsymbol{a}\boldsymbol{b}^T + \boldsymbol{b}\boldsymbol{a}^T + \boldsymbol{b}\boldsymbol{b}^T\right] = \mathbb{E}\left[\boldsymbol{a}\boldsymbol{a}^T\right] + 2\,\mathbb{E}\left[\boldsymbol{a}\boldsymbol{b}^T\right] + \mathbb{E}\left[\boldsymbol{b}\boldsymbol{b}^T\right].$$

Using the definitions of $\boldsymbol{a}$ and $\boldsymbol{b}$, we see that this is

$$\operatorname{cov}[\boldsymbol{x}] + 2\operatorname{cov}[\boldsymbol{x}, \boldsymbol{z}] + \operatorname{cov}[\boldsymbol{z}],$$

and since the variables are independent, $\operatorname{cov}[\boldsymbol{x}, \boldsymbol{z}] = \boldsymbol{0}$. We have shown that

$$\operatorname{cov}[\boldsymbol{x} + \boldsymbol{z}] = \operatorname{cov}[\boldsymbol{x}] + \operatorname{cov}[\boldsymbol{z}].$$

Exercise 2.31

If $y = x + z$, then $p_y(y) = p_x(x) * p_z(z)$, where $*$ denotes the *convolution* operator. To see why this is true, draw an x-z coordinate system and sketch a line $x + z = a$. The probability that $y = x + z = a$ is $p_y(y = a) = \int p_x(a - t)p_z(a)\, dt$, since this integrates over all possible ways to obtain the sum a.

In this case, when have $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{z}$, so we evaluate

$$\begin{aligned} p_y(\boldsymbol{y}) = p_x(\boldsymbol{x}) * p_z(\boldsymbol{z}) &= \int p_x(\boldsymbol{y} - \boldsymbol{t})p_z(\boldsymbol{y})\, d\boldsymbol{t} \\ &= \int \underbrace{\mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{\mu}_{\boldsymbol{x}} + \boldsymbol{t}, \boldsymbol{\Sigma}_{\boldsymbol{x}}\right)}_{p(\boldsymbol{y}|\boldsymbol{t})} \underbrace{\mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{\mu}_{\boldsymbol{z}}, \boldsymbol{\Sigma}_{\boldsymbol{z}}\right)}_{p(\boldsymbol{y})}\, d\boldsymbol{t}. \end{aligned}$$

Matching terms with Equations (2.109) and (2.110) we see that

$$p_y(\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{\mu}_{\boldsymbol{z}} + \boldsymbol{\mu}_{\boldsymbol{x}}, \boldsymbol{\Sigma}_{\boldsymbol{z}} + \boldsymbol{\Sigma}_{\boldsymbol{x}}\right).$$

Exercise 2.40

The posterior is proportional to the prior times the likelihood. When the prior is Gaussian, we have

$$p(\boldsymbol{\mu} \mid \boldsymbol{X}) \propto p(\boldsymbol{\mu})p(\boldsymbol{X} \mid \boldsymbol{\mu}) = \mathcal{N}\left(\boldsymbol{\mu} \mid \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0\right) \prod_{n=1}^{N} \mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}\right).$$

Let us ignore normalization constants and only consider the product of the exponential functions, this approach yields the following exponential function (which is not a p.d.f.)

$$\begin{aligned} p(\boldsymbol{\mu} \mid \boldsymbol{X}) &\propto \mathcal{N}\left(\boldsymbol{\mu} \mid \boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N\right) \\ &= \exp\left(-\frac{1}{2}\left((\boldsymbol{\mu} - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\mu} - \boldsymbol{\mu}_0) + \sum_{n=1}^{N}(\boldsymbol{x}_n - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(x_n - \boldsymbol{\mu})\right)\right). \end{aligned} \tag{10}$$

We will multiply out the quadratic forms in Equation (10) above, and compare it to

$$(\boldsymbol{\mu} - \boldsymbol{\mu}_N)^T \boldsymbol{\Sigma}_N^{-1}(\boldsymbol{\mu} - \boldsymbol{\mu}_N) = \boldsymbol{\mu}^T \boldsymbol{\Sigma}_N^{-1} \boldsymbol{\mu} - 2\boldsymbol{\mu}_N^T \boldsymbol{\Sigma}_N^{-1} \boldsymbol{\mu} + \text{const.} \tag{11}$$

Multiplying out the terms in the inner parenthesis of Equation (10) and rearranging, we obtain the following quadratic dependence on $\boldsymbol{\mu}$:

$$\boldsymbol{\mu}^T \left(\boldsymbol{\Sigma}_0^{-1} + N\boldsymbol{\Sigma}^{-1}\right) \boldsymbol{\mu} - 2\left(\boldsymbol{\mu}_0^T \boldsymbol{\Sigma}_0^{-1} + N\boldsymbol{\mu}_{\text{ML}}^T \boldsymbol{\Sigma}^{-1}\right) \boldsymbol{\mu} + \text{const.}$$

Comparing the above with Equation (11), we immediately see that

$$\boldsymbol{\Sigma}_N^{-1} = \boldsymbol{\Sigma}_0^{-1} + N\boldsymbol{\Sigma}^{-1} \tag{12}$$

$$\boldsymbol{\mu}_N^T \boldsymbol{\Sigma}_N^{-1} = \boldsymbol{\mu}_0^T \boldsymbol{\Sigma}_0^{-1} + N\boldsymbol{\mu}_{\text{ML}}^T \boldsymbol{\Sigma}^{-1}. \tag{13}$$

Equation (12) corresponds with (2.141) from [Bishop, 2006], and answers the first part of the exercise. Equation (13) needs a little more refinement, and we will make use of the following matrix identity, which is (C.5) in the appendix.

$$(\boldsymbol{A}^{-1} + \boldsymbol{B}^{-1})^{-1} = \boldsymbol{B}(\boldsymbol{A} + \boldsymbol{B})^{-1}\boldsymbol{A} = \boldsymbol{A}(\boldsymbol{B} + \boldsymbol{A})^{-1}\boldsymbol{B}$$

Using the above on Equation (13), we see that

$$\begin{aligned} \boldsymbol{\Sigma}_N &= \left(\boldsymbol{\Sigma}_N^{-1}\right)^{-1} = \left(\boldsymbol{\Sigma}_0^{-1} + N\boldsymbol{\Sigma}^{-1}\right)^{-1} \\ &= \frac{1}{N}\boldsymbol{\Sigma}\left(\boldsymbol{\Sigma}_0 + \frac{1}{N}\boldsymbol{\Sigma}\right)^{-1}\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_0\left(\frac{1}{N}\boldsymbol{\Sigma} + \boldsymbol{\Sigma}_0\right)^{-1}\frac{1}{N}\boldsymbol{\Sigma}. \end{aligned}$$

Right multiplying Equation (13) with $\boldsymbol{\Sigma}_N$ and using the equation above, we obtain

$$\boldsymbol{\mu}_N^T = \boldsymbol{\mu}_0^T\left(\boldsymbol{\Sigma} + N\boldsymbol{\Sigma}_0\right)^{-1}\boldsymbol{\Sigma} + \boldsymbol{\mu}_{\text{ML}}^T\left(\boldsymbol{\Sigma} + N\boldsymbol{\Sigma}_0\right)^{-1}N\boldsymbol{\Sigma}_0.$$

We have solved the second part of the problem. Observe also how this corresponds with (2.142), which represents the univariate case. It's reassuring that the multivariate solution corresponds elegantly with the univariate case.

Exercise 2.43

We integrate, perform a change of variables, and finally recognize the gamma function:

$$\begin{aligned} \int_{-\infty}^{\infty} \exp\left(-\frac{|x|^q}{2\sigma^2}\right) dx &= 2\int_0^{\infty} \exp\left(-\frac{x^q}{2\sigma^2}\right) dx && \text{(symmetry)} \\ &= 2\int_0^{\infty} \exp\left(-u\right)\frac{2\sigma^2}{q}x^{1-q}\, du && \text{(substitute } u = x^q/2\sigma^2\text{)} \\ &= 2\int_0^{\infty} \exp\left(-u\right)\frac{2\sigma^2}{q}\left[\left(2\sigma^2 u\right)^{1/q}\right]^{1-q} du && \text{(substitute definition)} \\ &= 2\frac{2\sigma^2}{q}(2\sigma^2)^{1/q-1}\int_0^{\infty}\exp\left(-u\right)u^{1/q-1}\, du && \text{(recognize } \Gamma(1/q)\text{)} \\ &= 2\frac{2\sigma^2}{q}(2\sigma^2)^{1/q-1}\,\Gamma(1/q) = \frac{2}{q}(2\sigma^2)^{1/q}\,\Gamma(1/q) \end{aligned}$$

Exercise 2.47

In this problem, we only consider the factor dependent on x. We see that

$$\lim_{\nu\to\infty}\left[\left(1 + \frac{\lambda(x-\mu)^2}{\nu}\right)^{\nu}\right]^{-1/2} = \exp\left(\lambda(x-\mu)^2\right)^{-1/2} = \exp\left(-\frac{\lambda(x-\mu)^2}{2}\right),$$

where we have used the fact that $\exp(x) = \lim_{n\to\infty}\left(1 + x/n\right)^n$.

Exercise 2.49

a) Using the definitions and changing the order of integration, we have

$$\begin{aligned}
\mathbb{E}[\boldsymbol{x}] &= \int_{-\infty}^{\infty} \boldsymbol{x} p(\boldsymbol{x})\, d\boldsymbol{x} \\
&= \int_{-\infty}^{\infty} \boldsymbol{x} \left(\int_0^{\infty} \mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{\mu}, (\eta \boldsymbol{\Lambda})^{-1}\right) \operatorname{Gam}\left(\eta \mid \nu/2, \nu/2\right)\, d\eta \right) d\boldsymbol{x} \\
&= \int_0^{\infty} \operatorname{Gam}\left(\eta \mid \nu/2, \nu/2\right) \left[\int_{-\infty}^{\infty} \boldsymbol{x} \mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{\mu}, (\eta \boldsymbol{\Lambda})^{-1}\right)\, d\boldsymbol{x} \right] d\eta \\
&= \boldsymbol{\mu} \int_0^{\infty} \operatorname{Gam}\left(\eta \mid \nu/2, \nu/2\right)\, d\eta = \boldsymbol{\mu}
\end{aligned}$$

b) We start by introducing a preliminary result, which is the fact that

$$\int_0^{\infty} \operatorname{Gam}(\lambda \mid a, b) \frac{1}{\lambda}\, d\lambda = \frac{b}{a-1} \int_0^{\infty} \operatorname{Gam}(\lambda \mid a, b) \frac{1}{\lambda}\, d\lambda = \frac{b}{a-1}. \tag{14}$$

To solve the problem we will make use of the result given in Equation (14) above, as well as the shortcut formula $\operatorname{cov}[\boldsymbol{x}] = \mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^T\right] - \mathbb{E}[\boldsymbol{x}]\, \mathbb{E}[\boldsymbol{x}]^T$.

The term $\mathbb{E}[\boldsymbol{x}]\, \mathbb{E}[\boldsymbol{x}]^T$ is already known, it's $\boldsymbol{\mu}\boldsymbol{\mu}^T$ as a result of the the previous sub-problem. The term second term, i.e. $\mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^T\right]$, is computed similarly to the previous sub problem, and we make use of Equation (14) to write:

$$\begin{aligned}
\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T] &= \int_{-\infty}^{\infty} \boldsymbol{x}\boldsymbol{x}^T p(\boldsymbol{x})\, d\boldsymbol{x} \\
&= \int_{-\infty}^{\infty} \boldsymbol{x}\boldsymbol{x}^T \left(\int_0^{\infty} \mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{\mu}, (\eta \boldsymbol{\Lambda})^{-1}\right) \operatorname{Gam}\left(\eta \mid \nu/2, \nu/2\right)\, d\eta \right) d\boldsymbol{x} \\
&= \int_0^{\infty} \operatorname{Gam}\left(\eta \mid \nu/2, \nu/2\right) \left[\int_{-\infty}^{\infty} \boldsymbol{x}\boldsymbol{x}^T \mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{\mu}, (\eta \boldsymbol{\Lambda})^{-1}\right)\, d\boldsymbol{x} \right] d\eta \\
&= \left((\eta \boldsymbol{\Lambda})^{-1} + \boldsymbol{\mu}\boldsymbol{\mu}^T \right) \int_0^{\infty} \operatorname{Gam}\left(\eta \mid \nu/2, \nu/2\right)\, d\eta \\
&= \boldsymbol{\Lambda}^{-1} \int_0^{\infty} \operatorname{Gam}\left(\eta \mid \nu/2, \nu/2\right) \frac{1}{\eta}\, d\eta + \boldsymbol{\mu}\boldsymbol{\mu}^T \\
&= \boldsymbol{\Lambda}^{-1} \frac{\nu/2}{\nu/2 - 1} + \boldsymbol{\mu}\boldsymbol{\mu}^T
\end{aligned}$$

Combining the results of $\mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^T\right]$ and $\mathbb{E}[\boldsymbol{x}]\, \mathbb{E}[\boldsymbol{x}]^T$ above solves the problem.

c) We differentiate (2.161) with respect to $\boldsymbol{x}$ to obtain

$$\int_0^{\infty} 2\eta \boldsymbol{\Lambda}(\boldsymbol{x} - \boldsymbol{\mu}) \mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{\mu}, (\eta \boldsymbol{\Lambda})^{-1}\right) \operatorname{Gam}\left(\eta \mid \nu/2, \nu/2\right)\, d\eta = \boldsymbol{0},$$

which is identically zero when $\boldsymbol{x} = \boldsymbol{\mu}$.

Exercise 2.55

Starting with Equation (2.187) in [Bishop, 2006], we have

$$\begin{aligned} A(m_{\mathrm{ML}}) &= \left(\frac{1}{N} \sum_{n=1}^{N} \cos\theta_n \right) \cos\theta_0^{\mathrm{ML}} + \left(\frac{1}{N} \sum_{n=1}^{N} \sin\theta_n \right) \sin\theta_0^{\mathrm{ML}} \\ &= \bar{r}\cos\bar{\theta}\cos\theta_0^{\mathrm{ML}} + \bar{r}\sin\bar{\theta}\sin\theta_0^{\mathrm{ML}}, \end{aligned}$$

where we used (2.268) in the last equality. Next we recognize that $\bar{\theta} = \theta_0^{\mathrm{ML}}$ and apply $\sin^2\bar{\theta} + \cos^2\bar{\theta} = 1$ to finish the result, showing that indeed $A(m_{\mathrm{ML}}) = \bar{r}$.

Exercise 2.57

The multidimensional Gaussian is given by

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

where the quadratic term has been multiplied out. One way to write the term $\boldsymbol{x}^T\boldsymbol{\Sigma}\boldsymbol{x}$ as a linear combination of $\boldsymbol{\eta}^T$ and $u(\boldsymbol{x})$ is to recolonize that

$$\boldsymbol{x}^T\boldsymbol{\Sigma}\boldsymbol{x} = \operatorname{vect}\left(\boldsymbol{\Sigma}\right)^T \operatorname{vect}\left(\boldsymbol{x}\boldsymbol{x}^T\right),$$

where vect $(\cdot)$ maps the (i,j)-th entry of a $D \times D$ matrix to the $D(i-1)+j$th entry of a column vector, i.e. $\operatorname{vect}\left(\begin{smallmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{smallmatrix}\right) = (a_{11}, a_{12}, a_{21}, a_{22})^T$. We then obtain the following.

$$\begin{aligned} h(\boldsymbol{x}) &= (2\pi)^{-D/2} & \boldsymbol{\eta} &= \begin{pmatrix} \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \\ -\frac{1}{2}\operatorname{vect}\left(\boldsymbol{\Sigma}^{-1}\right) \end{pmatrix} \\ u(\boldsymbol{x}) &= \begin{pmatrix} \boldsymbol{x} \\ \operatorname{vect}\left(\boldsymbol{x}\boldsymbol{x}^T\right) \end{pmatrix} & g(\boldsymbol{\eta}) &= \frac{1}{\left|\boldsymbol{\Sigma}\right|^{1/2}} \exp\left(-\frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{\Sigma}\boldsymbol{\mu} \right) \end{aligned}$$

2.3 Linear Models for Regression

Exercise 3.7

The prior and likelihood are respectively given by

$$\begin{aligned} p(\boldsymbol{w}) &= \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{m_0}, \boldsymbol{S_0}) \\ p(t \mid \boldsymbol{w}) &= \prod_{n=1}^{N} \mathcal{N}(t_n \mid \boldsymbol{w}^T\boldsymbol{\phi}(x_n), \beta^{-1}). \end{aligned}$$

Computing the posterior distribution of the weights $\boldsymbol{w}$ gives the product $p(\boldsymbol{w} \mid t) = p(t \mid \boldsymbol{w})p(\boldsymbol{w})$. We only focus on α in the resulting exponential $\exp(\alpha/2)$, which becomes

$$\sum_{n=1}^{N} (t_n - \boldsymbol{w}^T\boldsymbol{\phi}(x_n))^T \beta (t_n - \boldsymbol{w}^T\boldsymbol{\phi}(x_n)) + (\boldsymbol{w} - \boldsymbol{m_0})^T \boldsymbol{S_0}^{-1} (\boldsymbol{w} - \boldsymbol{m_0}). \tag{15}$$

We want to compare this to

$$f(\boldsymbol{w}) = (\boldsymbol{w} - \boldsymbol{m_N})^T \boldsymbol{S}_N^{-1} (\boldsymbol{w} - \boldsymbol{m_N}) = \boldsymbol{w}^T \boldsymbol{S}_N^{-1} \boldsymbol{w} - 2\boldsymbol{m}_{\boldsymbol{N}}^T \boldsymbol{S}_N^{-1} \boldsymbol{w} + \text{const.} \tag{16}$$

The term quadratic in $\boldsymbol{w}$ The quadratic term in $\boldsymbol{w}$ in Equation (15) is

$$\sum_{n=1}^{N} (\boldsymbol{w}^T \boldsymbol{\phi}(x_n))^T \beta \boldsymbol{w}^T \boldsymbol{\phi}(x_n)) + \boldsymbol{w}^T \boldsymbol{S}_{\boldsymbol{0}}^{-1} \boldsymbol{w} = \boldsymbol{w}^T \left[\sum_{n=1}^{N} \boldsymbol{\phi}(x_n) \beta \boldsymbol{\phi}(x_n)^T + \boldsymbol{S}_{\boldsymbol{0}}^{-1} \right] \boldsymbol{w}.$$

Matching with Equation (16), we see that

$$\boldsymbol{S}_N^{-1} = \sum_{n=1}^{N} \boldsymbol{\phi}(x_n) \beta \boldsymbol{\phi}(x_n)^T + \boldsymbol{S}_{\boldsymbol{0}}^{-1} = \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi} + \boldsymbol{S}_{\boldsymbol{0}}^{-1},$$

since $\left(\boldsymbol{\Phi}^T \boldsymbol{\Phi}\right)_{ij} = \sum_k \boldsymbol{\Phi}_{ki} \boldsymbol{\Phi}_{kj} = \sum_k \boldsymbol{\phi}_i(x_k) \boldsymbol{\phi}_j(x_k)$. This solves the first part of the problem.

The term linear in $\boldsymbol{w}$ The linear term in $\boldsymbol{w}$ in Equation (15) is

$$\begin{aligned} 2\sum_{n=1}^{N} (\boldsymbol{w}^T \boldsymbol{\phi}(x_n))^T \beta t_n + 2\boldsymbol{m}_{\boldsymbol{0}}^T \boldsymbol{S}_{\boldsymbol{0}}^{-1} \boldsymbol{w} &= 2 \left[\beta \sum_{n=1}^{N} t_n \boldsymbol{\phi}(x_n)^T + \boldsymbol{m}_{\boldsymbol{0}}^T \boldsymbol{S}_{\boldsymbol{0}}^{-1} \right] \boldsymbol{w} \\ &= 2 \left[\beta \boldsymbol{t}^T \boldsymbol{\Phi} + \boldsymbol{m}_{\boldsymbol{0}}^T \boldsymbol{S}_{\boldsymbol{0}}^{-1} \right] \boldsymbol{w}. \end{aligned}$$

Matching with the second term in the right hand side of Equation (16), we see that

$$\boldsymbol{m}_N^T \boldsymbol{S}_N^{-1} = \beta \boldsymbol{t}^T \boldsymbol{\Phi} + \boldsymbol{m}_{\boldsymbol{0}}^T \boldsymbol{S}_{\boldsymbol{0}}^{-1}.$$

Right multiplying with $\boldsymbol{S}_N$, taking the transpose and using the symmetry of $\boldsymbol{S}_N$ completes the problem.

Exercise 3.11

Showing that $\sigma_{N+1}^2(\boldsymbol{x}) \leq \sigma_N^2(\boldsymbol{x})$ entails showing that

$$\boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{S}_{N+1} \boldsymbol{\phi}(\boldsymbol{x}) \leq \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}).$$

We use the Woodbury matrix identity from Appendix C in [Bishop, 2006] to write

$$\boldsymbol{S}_{N+1} = \left[\boldsymbol{S}_N^{-1} + \beta \boldsymbol{\phi}(\boldsymbol{x}) \boldsymbol{\phi}(\boldsymbol{x})^T \right]^{-1} = \boldsymbol{S}_N - \frac{\boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}) \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{S}_N}{1 + \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x})}.$$

Substituting this into the inequality above reveals that we only have to show that

$$\boldsymbol{\phi}(\boldsymbol{x})^T \frac{\boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}) \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{S}_N}{1 + \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x})} \boldsymbol{\phi}(\boldsymbol{x}) \geq 0.$$

This is easy to see, since $\gamma := \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}) \geq 0$ as long as the covariance matrix $\boldsymbol{S}_N$ is positive semi-definite (which we assume it is), and then the expression above becomes

$$\frac{\gamma^2}{1 + \gamma} \geq 0.$$

Exercise 3.16 (Unfinished)

Evaluating the integral is straightforward

$$\begin{aligned} p(\mathsf{t} \mid \alpha, \beta) &= \int p(\mathsf{t} \mid \boldsymbol{w}, \beta) p(\boldsymbol{w} \mid \alpha)\, d\boldsymbol{w} \\ &= \int \mathcal{N}\left(\mathsf{t} \mid \boldsymbol{\Phi w}, \boldsymbol{I}\beta^{-1}\right) \mathcal{N}\left(\boldsymbol{w} \mid \boldsymbol{0}, \boldsymbol{I}\alpha^{-1}\right)\, d\boldsymbol{w} && \text{(definitions)} \\ &= \mathcal{N}\left(\mathsf{t} \mid \boldsymbol{0}, \boldsymbol{I}\beta^{-1} + \boldsymbol{\Phi\Phi}^T\alpha^{-1}\right) && \text{(using Eqn (2.115))} \end{aligned}$$

The logarithm of the above expression, as a function of t, should be proportional to the quadratic form $\mathsf{t}^T\left(\boldsymbol{I}\beta^{-1} + \boldsymbol{\Phi\Phi}^T\alpha^{-1}\right)\mathsf{t}$. We must show that this matches with $E(\boldsymbol{m}_M)$ in Equation (3.86). I failed to show this, below is my attempt.

The below is unfinished work We expand Equation (3.82) to

$$E(\boldsymbol{m}_N) = \frac{1}{2}\left(\beta\mathsf{t}^T\mathsf{t} - 2\beta\mathsf{t}^T\boldsymbol{\Phi m}_N\right) + \frac{1}{2}\boldsymbol{m}_N^T\left(\underbrace{\beta\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \boldsymbol{I}\alpha}_{A}\right)\boldsymbol{m}_N$$

Substituting $\boldsymbol{m}_N = \beta\boldsymbol{A}^{-1}\boldsymbol{\Phi}^T\mathsf{t}$ into the expression above and writing it as a quadratic form, we obtain

$$\frac{1}{2}\mathsf{t}^T\left(\beta\boldsymbol{I} - \beta^2\boldsymbol{\Phi A}^{-1}\boldsymbol{\Phi}^T\right)\mathsf{t}$$

I also know that

$$\boldsymbol{A}^{-1} = \left(\beta\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \boldsymbol{I}\alpha\right)^{-1} = \alpha^{-1}\boldsymbol{\Phi}^T\left(\alpha^{-1}\boldsymbol{\Phi\Phi}^T + \beta{-}1\boldsymbol{I}\right)\beta^{-1}\boldsymbol{\Phi}^{-T},$$

by Equation (C.5) from the Appendix, but I am unable to show that this reduces to the result above.

Exercise 3.20

We verify the steps that are non-trivial in list-form below.

- Let's consider two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, not necessarily the $\boldsymbol{A}$ given in the section of the book. If $\boldsymbol{A} = \alpha\boldsymbol{I} + \boldsymbol{B}$, then $\boldsymbol{B}$ has eigenvalues given by $\boldsymbol{B}u_B = \lambda_B u_B$. The matrix $\boldsymbol{A}$ has eigenvalues given by $\boldsymbol{A}u_A = \lambda_A u_A$. Substituting the definition of $\boldsymbol{A}$ into this, we have

$$\left(\alpha I + \boldsymbol{B}\right)\boldsymbol{u}_A = \lambda_A\boldsymbol{u}_A \quad \Rightarrow \quad \boldsymbol{B}u_A = (\lambda_A - \alpha)\boldsymbol{u}_A.$$

  From this we see that $\boldsymbol{A}$ and $\boldsymbol{B}$ share eigenvectors, and the eigenvalues of $\boldsymbol{A}$ are shifted by α compared to the eigenvalues of $\boldsymbol{B}$, so that $\lambda_A = \lambda_B + \alpha$.

- Taking the logarithm of the determinant is achieved by eigenvalue decomposition

$$\ln|\boldsymbol{A}| = \ln\left|\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T\right| = \ln|\boldsymbol{Q}|\,|\boldsymbol{\Lambda}|\,\left|\boldsymbol{Q}^T\right| = \ln|\boldsymbol{\Lambda}| = \ln\left(\prod_i^M(\lambda_i+\alpha)\right).$$

- The differentiation should be straightforward.
- We pull the M into the sum by writing

$$\gamma = M - \alpha\sum_i^M\frac{1}{\lambda_i+\alpha} = \sum_i^M\frac{\lambda_i+\alpha}{\lambda_i+\alpha} - \sum_i^M\frac{\alpha}{\lambda_i+\alpha} = \sum_i^M\frac{\lambda_i}{\lambda_i+\alpha}.$$

2.4 Linear Models for Classification

Exercise 4.6

This exercise is mostly algebra. Our strategy will be to show the right-hand side first, and then the left-hand side. The right hand side requires less algebra.

To show the **right-hand side**, we split the sum, which yields

$$\sum_{n=1}^N\left(\boldsymbol{w}^T\boldsymbol{x}_n+w_0-t_n\right)\boldsymbol{x}_n = \sum_{n\in\mathcal{C}_1}\left(\boldsymbol{w}^T\boldsymbol{x}_n+w_0-t_n\right)\boldsymbol{x}_n + \sum_{n\in\mathcal{C}_2}\left(\boldsymbol{w}^T\boldsymbol{x}_n+w_0-t_n\right)\boldsymbol{x}_n = \boldsymbol{0}.$$

We now move the $t_n\boldsymbol{x}_n$-terms to the right hand-side, and use the fact that $t_n = N/N_1$ when $n\in\mathcal{C}_1$ and $t_n = -N/N_2$ when $n\in\mathcal{C}_1$. Doing so, we obtain the correct expression

$$\sum_{n\in\mathcal{C}_1}\left(\boldsymbol{w}^T\boldsymbol{x}_n+w_0\right)\boldsymbol{x}_n + \sum_{n\in\mathcal{C}_2}\left(\boldsymbol{w}^T\boldsymbol{x}_n+w_0\right)\boldsymbol{x}_n = N(\boldsymbol{m}_1-\boldsymbol{m}_2), \tag{17}$$

since $\sum_{n\in\mathcal{C}_1}t_n\boldsymbol{x}_n + \sum_{n\in\mathcal{C}_2}t_n\boldsymbol{x}_n = N(\boldsymbol{m}_1-\boldsymbol{m}_2)$ on the right-hand side.

Let us now focus on the **left-hand side** of Eqn (17). Notice first that $\boldsymbol{w}$ may be pulled outside of the sums, by the following algebra (here showed over a single sum, not both)

$$\sum_n\left(\boldsymbol{w}^T\boldsymbol{x}_n+w_0\right)\boldsymbol{x}_n = \sum_n\left(\boldsymbol{w}^T\boldsymbol{x}_n-\boldsymbol{w}^T\boldsymbol{m}\right)\boldsymbol{x}_n = \sum_n\left(\boldsymbol{x}_n\boldsymbol{x}_n^T-\boldsymbol{x}_n\boldsymbol{m}^T\right)\boldsymbol{w}.$$

Using this result on both sums in Eqn (17) leaves us with the expression

$$\left(\sum_{n\in\mathcal{C}_1}\left(\boldsymbol{x}_n\boldsymbol{x}_n^T-\boldsymbol{x}_n\boldsymbol{m}^T\right) + \sum_{n\in\mathcal{C}_2}\left(\boldsymbol{x}_n\boldsymbol{x}_n^T-\boldsymbol{x}_n\boldsymbol{m}^T\right)\right)\boldsymbol{w} = N(\boldsymbol{m}_1-\boldsymbol{m}_2).$$

At this point we're pretty close, but we have to reduce the term in the left parenthesis. Observe that the first term becomes

$$\begin{aligned}\sum_{n\in\mathcal{C}_1}\boldsymbol{x}_n\boldsymbol{x}_n^T-\boldsymbol{x}_n\boldsymbol{m}^T &= \sum_{n\in\mathcal{C}_1}\boldsymbol{x}_n\boldsymbol{x}_n^T-\boldsymbol{x}_n\left(\frac{N_1}{N}\boldsymbol{m}_1+\frac{N_2}{N}\boldsymbol{m}_2\right)^T\\ &= \sum_{n\in\mathcal{C}_1}\boldsymbol{x}_n\boldsymbol{x}_n^T-\frac{N_1N_1}{N}\boldsymbol{m}_1\boldsymbol{m}_1^T-\frac{N_2N_1}{N}\boldsymbol{m}_1\boldsymbol{m}_2^T, \end{aligned}\tag{18}$$

and the second term will be very similar due to symmetry. We complete the square above:

$$\sum_{n \in \mathcal{C}_1} \left(\boldsymbol{x}_n \boldsymbol{x}_n^T - \underbrace{2\boldsymbol{m}_1 \boldsymbol{x}_n^T + \boldsymbol{m}_1 \boldsymbol{m}_1^T}_{\text{added this}} \right) + \underbrace{2N_1 \boldsymbol{m}_1 \boldsymbol{m}_1^T - N_1 \boldsymbol{m}_1 \boldsymbol{m}_1^T}_{\text{and subtracted it}} - \frac{N_1 N_1}{N} \boldsymbol{m}_1 \boldsymbol{m}_1^T - \frac{N_2 N_1}{N} \boldsymbol{m}_1 \boldsymbol{m}_2^T,$$

Now we make use of $N_1 - N_1^2/N = N_1 N_2/N$ to obtain

$$\underbrace{\sum_{n \in \mathcal{C}_1} \left(\boldsymbol{x}_n \boldsymbol{x}_n^T - 2\boldsymbol{m}_1 \boldsymbol{x}_n^T + \boldsymbol{m}_1 \boldsymbol{m}_1^T \right)}_{\text{half of } \boldsymbol{S}_W} + \frac{N_2 N_1}{N} \underbrace{\left(\boldsymbol{m}_1 \boldsymbol{m}_1^T - \boldsymbol{m}_1 \boldsymbol{m}_2^T \right)}_{\text{half of } \boldsymbol{S}_B},$$

and we're finished. If we apply the same algebra as performed from Equation (18) and onwards to $\mathcal{C}_2$ too, and add the results, it will yield $\boldsymbol{S}_W + N_1 N_2 \boldsymbol{S}_B / N$.

Exercise 4.9

The probability of class $\mathcal{C}_j$ is given by $p(\mathcal{C}_j) = \pi_j$, and $p(\boldsymbol{\phi}_n, \mathcal{C}_j) = p(\boldsymbol{\phi}_n \mid \mathcal{C}_j) p(\mathcal{C}_j)$ by the product rule of probability. The probability of a single data point becomes

$$p(\boldsymbol{t}_n, \boldsymbol{\phi}_n \mid \boldsymbol{\pi}) = \prod_{j=1}^{K} \left(\pi_j \, p(\boldsymbol{\phi}_n \mid \mathcal{C}_j) \right)^{t_{nj}}.$$

The notation above is somewhat heavy, so it helps to consider a specific example. Consider the case when the number of classes $K = 3$, and the observation $\boldsymbol{t}_n = (0, 1, 0)^T$. The probability of observing this value, for an arbitrary $\boldsymbol{\phi}_n$, is

$$p(\boldsymbol{t}_n = (0, 1, 0), \boldsymbol{\phi}_n \mid \boldsymbol{\pi}) = \pi_2 \, p(\boldsymbol{\phi}_n \mid \mathcal{C}_2).$$

Assuming i.i.d. data, the log-likelihood function becomes

$$\ln p(\mathcal{D} \mid \boldsymbol{\pi}) = \ln p(\mathsf{t}, \boldsymbol{\Phi} \mid \boldsymbol{\pi}) = \sum_{n=1}^{N} \sum_{j=1}^{K} t_{nj} \left(\ln(\pi_j) + \ln\left(p(\boldsymbol{\phi}_n \mid \mathcal{C}_j) \right) \right).$$

We wish to maximize the log-likelihood with respect to the vector $\boldsymbol{\pi}$, given the condition that the prior class probabilities sum to unity, i.e. $\sum_{j=1}^{K} \pi_j = 1$. Setting up the Lagrange function and differentiating with respect to π_k yields

$$\sum_{n=1}^{N} \frac{t_{nk}}{\pi_k} + \lambda = \frac{N_k}{\pi_k} + \lambda = 0, \tag{19}$$

where λ is a Lagrange multiplier. Solving the above for π_k yields $-N_k/\lambda$, and

$$\sum_{j=1}^{K} \pi_j = -\frac{1}{\lambda} \sum_{j=1}^{K} N_j = 1 \implies \lambda = -N.$$

Substituting the above value for the Lagrange multiplier λ into Equation (19) gives the corrcect answer, which is $\pi_k = N_k/N$.

Exercise 4.10

This problem is similar to the previous one, but now $p(\boldsymbol{\phi}_n \mid \mathcal{C}_j) = \mathcal{N}(\boldsymbol{\phi}_n \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma})$. The likelihood is given by

$$p(\mathsf{t}, \boldsymbol{\Phi} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} \prod_{j=1}^{K} \left[\pi_j \mathcal{N}(\boldsymbol{\phi}_n \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma})\right]^{t_{nj}}.$$

Taking logarithms gives the expression

$$-\frac{1}{2} \sum_{n=1}^{N} \sum_{j=1}^{K} t_{nj} \left[K + \ln|\boldsymbol{\Sigma}| + (\boldsymbol{\phi}_n - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\phi}_n - \boldsymbol{\mu}_j)\right], \tag{20}$$

where K is a constant which is not a function of $\boldsymbol{\mu}_j$ or $\boldsymbol{\Sigma}$.

Differentiating Equation (20) with respect to $\boldsymbol{\mu}_k$ and equating it to $\mathbf{0}$ gives

$$\sum_{n=1}^{N} t_{nk} \boldsymbol{\Sigma}^{-1} (\boldsymbol{\phi}_n - \boldsymbol{\mu}_k) = \mathbf{0}.$$

Solving this for $\boldsymbol{\mu}_k$ reveals the desired answer, where we must use $\sum_{n=1}^{N} t_{nk} = N_k$.

Studying the terms dependent on $\boldsymbol{\Sigma}$, we obtain

$$-\frac{N}{2} \ln|\boldsymbol{\Sigma}| - \frac{1}{2} \sum_{j=1}^{K} \sum_{n \in \mathcal{C}_j} (\boldsymbol{\phi}_n - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\phi}_n - \boldsymbol{\mu}_j).$$

Now the apply the *trace trick*. The trace trick uses the fact that the trace of a scalar is equal to itself, and $\operatorname{tr}(\boldsymbol{a}^T \boldsymbol{b}) = \operatorname{tr}(\boldsymbol{b} \boldsymbol{a}^T)$, to write a quadratic form as

$$\boldsymbol{a}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{a} = \operatorname{tr}\left(\boldsymbol{a}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{a}\right) = \operatorname{tr}\left(\boldsymbol{\Sigma}^{-1} \boldsymbol{a} \boldsymbol{a}^T\right).$$

From this, we can derive the following

$$\begin{aligned}
&-\frac{N}{2} \ln|\boldsymbol{\Sigma}| - \frac{1}{2} \sum_{j=1}^{K} \sum_{n \in \mathcal{C}_j} (\boldsymbol{\phi}_n - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\phi}_n - \boldsymbol{\mu}_j) \\
&= -\frac{N}{2} \ln|\boldsymbol{\Sigma}| - \frac{1}{2} \sum_{j=1}^{K} \sum_{n \in \mathcal{C}_j} \operatorname{tr}\left(\boldsymbol{\Sigma}^{-1} (\boldsymbol{\phi}_n - \boldsymbol{\mu}_j)(\boldsymbol{\phi}_n - \boldsymbol{\mu}_j)^T\right) && \text{(trace trick)} \\
&= -\frac{N}{2} \ln|\boldsymbol{\Sigma}| - \frac{1}{2} \operatorname{tr}\left(\sum_{j=1}^{K} \sum_{n \in \mathcal{C}_j} \boldsymbol{\Sigma}^{-1} (\boldsymbol{\phi}_n - \boldsymbol{\mu}_j)(\boldsymbol{\phi}_n - \boldsymbol{\mu}_j)^T\right) && \text{(linearity of trace)} \\
&= -\frac{N}{2} \ln|\boldsymbol{\Sigma}| - \frac{N}{2} \operatorname{tr}\left(\boldsymbol{\Sigma}^{-1} \underbrace{\frac{1}{N} \sum_{j=1}^{K} \sum_{n \in \mathcal{C}_j} (\boldsymbol{\phi}_n - \boldsymbol{\mu}_j)(\boldsymbol{\phi}_n - \boldsymbol{\mu}_j)^T}_{\boldsymbol{S}}\right). && \text{(definition of } \boldsymbol{S}\text{)}
\end{aligned}$$

We have shown that Equation (4.77) holds for K classes. Differentiating

$$-\frac{N}{2}\ln|\boldsymbol{\Sigma}| - \frac{N}{2}\operatorname{tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{S}\right)$$

with respect to $\boldsymbol{\Sigma}$ shows that $\boldsymbol{\Sigma}^{-1} = \boldsymbol{S}$. We will trust that the result in 4.2.2 is true, and not show this. To show it, one must first study the derivative of $\ln|\boldsymbol{\Sigma}|$ and $\operatorname{tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{S})$, and we will do not that here.

Exercise 4.13

Differentiating $E(\boldsymbol{w})$ with respect to $\boldsymbol{w}$, we must keep in mind that $y_n = \sigma(\boldsymbol{w}^T\boldsymbol{\phi}_n)$ is a function of $\boldsymbol{w}$. We obtain

$$\nabla_{\boldsymbol{w}} E(\boldsymbol{w}) = -\sum_{n=1}^{N} \frac{t_n}{y_n}\boldsymbol{y}_n' + \frac{1-t_n}{1-y_n}(-\boldsymbol{y}_n')$$

where $\boldsymbol{y}_n' = y_n(1-y_n)\boldsymbol{\phi}_n$, substituting this into the equation above and simplifying gives

$$-\sum_{n=1}^{N} \frac{t_n}{y_n} y_n(1-y_n)\boldsymbol{\phi}_n - \frac{1-t_n}{1-y_n} y_n(1-y_n)\boldsymbol{\phi}_n = \sum_{n=1}^{N}(y_n - t_n)\boldsymbol{\phi}_n.$$

Exercise 4.15

We will show that $\boldsymbol{H}$ is positive definite, and that the solution is a minimum.

As a step toward showing that $\boldsymbol{H}$ is positive definite, we first show that $\boldsymbol{R}$ is positive definite. This is easy to show from the definition of positive definiteness:

$$\boldsymbol{a}^T\boldsymbol{R}\boldsymbol{a} = \sum_i a_i y_i(1-y_i)a_i = \sum_i a_i^2 y_i(1-y_i).$$

This expression is positive for every non-zero $\boldsymbol{a}$, since $y_i(1-y_i) > 0$ because the sigmoid function has $0 < y_i < 1$. The Hessian $\boldsymbol{H}$ is positive definite by the same token, since

$$\boldsymbol{u}^T\boldsymbol{H}\boldsymbol{u} = \boldsymbol{u}^T\boldsymbol{\Phi}^T\boldsymbol{R}\boldsymbol{\Phi}\boldsymbol{u} = \underbrace{\boldsymbol{b}^T}_{\boldsymbol{u}^T\boldsymbol{\Phi}^T}\boldsymbol{H}\boldsymbol{b} = \sum_i b_i^2 y_i(1-y_i) > 0.$$

We now argue that the solution is the unique minimum. Notice that the solution might be a subspace, for instance when the data is linearly separable. Expanding a taylor series around the minimum $\boldsymbol{w}^*$ yields the following

$$E(\boldsymbol{w}^* + \Delta\boldsymbol{w}) \simeq E(\boldsymbol{w}^*) + \underbrace{\Delta_{\boldsymbol{w}} E(\boldsymbol{w}^*)\left(\Delta\boldsymbol{w}\right)}_{0} + \left(\Delta\boldsymbol{w}\right)^T \boldsymbol{H}(\boldsymbol{w}^*)\left(\Delta\boldsymbol{w}\right) + \mathcal{O}\left(\|\Delta\boldsymbol{w}\|^3\right)$$

Notice that since the y_is are all functions of the weights $\boldsymbol{w}$, so the Hessian matrix is also a function of $\boldsymbol{w}$. At the minimum, $\Delta_{\boldsymbol{w}} E(\boldsymbol{w}^*)\left(\Delta\boldsymbol{w}\right) = 0$, so the middle term vanishes. Clearly the function is minimized when $\Delta\boldsymbol{w} = \boldsymbol{0}$, since this minimizes the remaining quadratic $\left(\Delta\boldsymbol{w}\right)^T \boldsymbol{H}\left(\Delta\boldsymbol{w}\right)$.

Exercise 4.20

Let block (k, j) be given by the block-matrix $\boldsymbol{B}_{kj} = \sum_{n=1}^{N} y_{nk} \left(I_{kj} - y_{nj}\right) \boldsymbol{\phi}_n \boldsymbol{\phi}_n^T$. We want to show that $\sum_{k=1}^{K} \sum_{j=1}^{K} \boldsymbol{u}_k^T \boldsymbol{B}_{kj} \boldsymbol{u}_j \geq 0$ for every $\boldsymbol{u} = (\boldsymbol{u}_1^T, \boldsymbol{u}_2^T, \ldots, \boldsymbol{u}_K^T)^T$.

Our strategy will be to first write out the sum over k, j and n. Then we will rearrange the sum over n to the outer level, and show that the summand in the sum over n is always nonnegative. Thus the total sum is nonnegative.

We substitute the definition of $\boldsymbol{B}_{kj}$ and move the sum over n to the outer level.

$$\begin{aligned}\sum_k \sum_j \boldsymbol{u}_k^T \boldsymbol{B}_{kj} \boldsymbol{u}_j &= \sum_k \sum_j \boldsymbol{u}_k^T \left[\sum_n y_{nk} \left(I_{kj} - y_{nj}\right) \boldsymbol{\phi}_n \boldsymbol{\phi}_n^T \right] \boldsymbol{u}_j \\ &= \sum_n \left[\sum_k y_{nk} \boldsymbol{u}_k^T \boldsymbol{\phi}_n \boldsymbol{\phi}_n^T \sum_j \left(I_{kj} - y_{nj}\right) \boldsymbol{u}_j \right] \qquad (21)\end{aligned}$$

If every term in the sum over n is nonnegative, then the entire expression will be nonnegative. For every term n, the following manipulation holds.

$$\begin{aligned}\sum_k y_{nk} \boldsymbol{u}_k^T \boldsymbol{\phi}_n \boldsymbol{\phi}_n^T \left[\sum_j \left(I_{kj} - y_{nj}\right) \boldsymbol{u}_j \right] &= \sum_k y_{nk} \boldsymbol{u}_k^T \boldsymbol{\phi}_n \boldsymbol{\phi}_n^T \left[\boldsymbol{u}_k - \sum_j y_{nj} \boldsymbol{u}_j \right] \\ &= \sum_k y_{nk} \boldsymbol{u}_k^T \boldsymbol{\phi}_n \boldsymbol{\phi}_n^T \boldsymbol{u}_k - \sum_k y_{nk} \boldsymbol{u}_k^T \boldsymbol{\phi}_n \boldsymbol{\phi}_n^T \sum_j y_{nj} \boldsymbol{u}_j\end{aligned}$$

In the first equality, we used the relationship $\sum_j I_{kj} \boldsymbol{u}_j = \boldsymbol{u}_k$, since the identity matrix (alternatively the Kronecker delta δ_{kj}) "picks out" the kth term in the sum.

In the last equality, we must show that the second term is smaller than, or equal to, the first term. To show this, we define the function $f(\boldsymbol{u}) = \boldsymbol{u}^T \boldsymbol{\phi}_n \boldsymbol{\phi}_n^T \boldsymbol{u}$. This function is positive semi-definite, and therefore convex, since

$$f(\boldsymbol{u}) = \left(\boldsymbol{u}^T \boldsymbol{\phi}_n\right)\left(\boldsymbol{\phi}_n^T \boldsymbol{u}\right) = \left(\boldsymbol{u}^T \boldsymbol{\phi}_n\right)\left(\boldsymbol{u}^T \boldsymbol{\phi}_n\right) = \alpha^2 \geq 0.$$

Using this function, we write

$$\sum_k y_{nk} \boldsymbol{u}_k^T \boldsymbol{\phi}_n \boldsymbol{\phi}_n^T \boldsymbol{u}_k - \sum_k y_{nk} \boldsymbol{u}_k^T \boldsymbol{\phi}_n \boldsymbol{\phi}_n^T \sum_j y_{nj} \boldsymbol{u}_j = \sum_k y_{nk} f(\boldsymbol{u}_k) - f\left(\sum_k y_{nk} \boldsymbol{u}_k\right).$$

Since $\sum_k p\left(\mathcal{C}_k \mid \boldsymbol{\phi}_n\right) = \sum_k y_{nk} = 1$ and $f(\boldsymbol{u})$ is convex, from Jensen's Inequality we have

$$f\left(\sum_k y_{nk} \boldsymbol{u}_k\right) \leq \sum_k y_{nk} f(\boldsymbol{u}_k).$$

We have showed that the term in the square brackets in Equation (21) nonnegative. Therefore, the total sum over n must be nonnegative, and $\sum_{k=1}^{K} \sum_{j=1}^{K} \boldsymbol{u}_k^T \boldsymbol{B}_{kj} \boldsymbol{u}_j \geq 0$ as claimed.

Exercise 4.22

We can approximate the integral as

$$p(\mathcal{D}) = \int p(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) \, d\boldsymbol{\theta} = \int p(\mathcal{D}, \boldsymbol{\theta}) \, d\boldsymbol{\theta} \simeq p(\mathcal{D}, \boldsymbol{\theta}_{\text{MAP}}) \frac{(2\pi)^{M/2}}{|\boldsymbol{A}|^{1/2}}$$

using Equation (4.135) to approximate the integral in the vicinity of the mode $p(\mathcal{D}, \boldsymbol{\theta}_{\text{MAP}})$. Taking logarithms and using the product rule of probability gives Equation (4.137), revealing the Occam factor.

2.5 Neural networks

Exercise 5.7

We split this exercise into two parts: (1) differentiating the softmax function and (2) differentiating the error function. The result of the first sub-problem will be used to solve the latter.

Differentiating the softmax function First we differentiate $y_k = h(a_k) = \operatorname{softmax}(a_k)$ with respect to a_k to obtain

$$\begin{aligned}\frac{\partial y_k}{\partial a_k} = \frac{\partial}{\partial a_k} \operatorname{softmax}(a_k) = \frac{\partial}{\partial a_k} \left(\frac{\exp a_k}{\sum_j \exp a_j} \right) &= \frac{\exp a_k \sum_{j \neq k} \exp a_j}{\left(\sum_j \exp a_j \right)^2} \qquad \text{(product rule)} \\ &= y_k \frac{\sum_{j \neq k} \exp a_j}{\sum_j \exp a_j} = y_k \left(1 - \frac{\exp a_k}{\sum_j \exp a_j} \right) = y_k (1 - y_k).\end{aligned}$$

In a similar fashion, we differentiate $y_k = \operatorname{softmax}(a_k)$ with respect to a_i to obtain

$$\frac{\partial y_k}{\partial a_i} = \frac{\partial}{\partial a_i} \operatorname{softmax}(a_k) = \frac{-\exp(a_k) \exp(a_i)}{\left(\sum_j \exp a_j \right)^2} = -y_i y_k.$$

These two results may be summarized using the Kronecker delta function δ_{kj} as

$$\frac{\partial y_k}{\partial a_i} = \frac{\partial}{\partial a_i} \operatorname{softmax}(a_k) = y_k \left(\delta_{kj} - y_i \right).$$

Differentiating the error function With the purpose of expressing the error $E(\boldsymbol{w})$ defined by Equation (2.24) as a function of *independent* parameters, we write it as

$$\begin{aligned}E(\boldsymbol{w}) &= -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk}\ln y_k(\boldsymbol{x}_n,\boldsymbol{w})\\ &= -\sum_{n=1}^{N} t_{n1}\ln y_{n1} + t_{n2}\ln y_{n2} + \cdots + t_{nK}\ln y_{nK}\\ &= -\sum_{n=1}^{N}\left(\sum_{k=1}^{K-1} t_{nk}\ln y_{nk} + t_K \ln\left(1-\sum_{k=1}^{K-1} y_{nk}\right)\right),\end{aligned}$$

since $\sum_{k=1}^{K} y_{nk} = 1$ for every n, and so the first $K-1$ parameters uniquely determine the last one. Recall also the constraint, $\sum_{k=1}^{K} t_{nk} = 1$ which we will use later.

Ignoring the summation over n and differentiating with respect to a_j, we obtain

$$\partial_{a_j}E(\boldsymbol{w}) = -\left[\sum_{k\neq j}^{K-1} t_k\frac{1}{y_k}\partial_{a_j}y_k + t_j\frac{1}{y_j}\partial_{a_j}y_j - t_K\frac{1}{\left(1-\sum_{k=1}^{K-1}y_k\right)}\partial_{a_j}\left(1-\sum_{k=1}^{K-1}y_k\right)\right].$$

Now we apply the formulas for the derivatives of the softmax function, and simplify the first two terms. The notation $\sum_{k\neq j}^{K-1}$ means "sum over $k=1,\ldots,K-1$, but skip $k=j$."

$$\begin{aligned}\partial_{a_j}E(\boldsymbol{w}) &= -\left[\sum_{k\neq j}^{K-1} t_k\frac{1}{y_k}(-y_k y_j) + t_j\frac{1}{y_j}y_j(1-y_j) - t_K\frac{1}{\left(1-\sum_{k=1}^{K-1}y_k\right)}\partial_{a_j}\left(1-\sum_{k=1}^{K-1}y_k\right)\right]\\ &= -\left[\sum_{k\neq j}^{K-1} t_k(-y_j) + t_j(1-y_j) - t_K\frac{1}{\left(1-\sum_{k=1}^{K-1}y_k\right)}\left(-y_j\sum_{k\neq j}^{K-1}y_k + y_j(1-y_j)\right)\right]\end{aligned}$$

We simplify further by introducing the jth term back into the sums and performing cancellations in the fractions. These simplifications yield

$$\begin{aligned}\partial_{a_j}E(\boldsymbol{w}) &= -\left[\sum_{k\neq j}^{K-1} t_k(-y_j) + t_j(1-y_j) - t_K\frac{1}{\left(1-\sum_{k=1}^{K-1}y_k\right)}\left(-y_j\sum_{k\neq j}^{K-1}y_k + y_j(1-y_j)\right)\right]\\ &= -\left[-y_j\sum_{k=1}^{K-1}t_k + t_j - t_K\frac{1}{\left(1-\sum_{k=1}^{K-1}y_k\right)}y_j\left(1-\sum_{k=1}^{K-1}y_k\right)\right]\\ &= -\left[-y_j\sum_{k=1}^{K-1}t_k + t_j - t_K y_j\right] = -\left[-y_j\sum_{k=1}^{K}t_k + t_j\right] = y_j - t_j.\end{aligned}$$

This solves the problem. In the last equality, we used the fact that $\sum_{k=1}^{K} t_k = 1$.

Exercise 5.13

The Hessian matrix $\boldsymbol{H}$ is of dimension $W \times W$. Since it's symmetric, i.e. has $\boldsymbol{H} = \boldsymbol{H}^T$, the number of independent parameters consist of the diagonal, plus off-diagonals, i.e.

$$W + (W-1) + (W-2) + \cdots + 2 + 1 = \frac{W(W+1)}{2}.$$

The gradient $\boldsymbol{b} = \nabla_{\boldsymbol{w}} E(\boldsymbol{w})$ has W free parameters, and therefore the total number of free parameters in the second-order Taylor expansion is given by

$$\frac{W(W+1)}{2} + W = \frac{W(W+3)}{2}.$$

Exercise 5.18

We introduce skip-layer connections from the D-dimensional input layer to the K-dimensional output layer. The notation for the D-M-K network is summarized in the diagram below.

$$\begin{tikzcd}
x_i,\ i = 1, \ldots, D \arrow[r, "w_{ji}^{(1)}"] \arrow[rr, bend left, "w_{ki}^{(3)}"] & z_j,\ j = 1, \ldots, M \arrow[r, "w_{kj}^{(2)}"] & y_k,\ k = 1, \ldots, K
\end{tikzcd}$$

The equations outlined in Section 5.3.2 in [Bishop, 2006] mostly remain identical when a skip-layer is added. The exception is Equation (5.64) for the outputs, which becomes

$$y_k = \sum_{j=0}^{M} w_{kj}^{(2)} z_j + \underbrace{\sum_{i=0}^{D} w_{ki}^{(3)} x_i}_{\text{skip-layer}}.$$

The derivatives of $E(\boldsymbol{w})$ with respect to $w_{ji}^{(1)}$ and $w_{kj}^{(2)}$ remain the same, since these weights are not dependent on the $w_{ki}^{(3)}$. Differentating with respect to $w_{ki}^{(3)}$ yields

$$\frac{\partial E_n}{w_{ki}^{(3)}} = (y_n - t_n) \partial_{w_{ki}^{(3)}} y_k = (y_n - t_n) x_i \equiv \delta_k x_i.$$

Exercise 5.20

The Hessian $\boldsymbol{H}$ is $\nabla\nabla E$, where the gradients are taken with respect to the weights. Using the result from Exercise 5.7 for the derivative of the error function with multiclass outputs and softmax activation functions, we obtain

$$\nabla E(\boldsymbol{w}) = \sum_{n=1}^{N} \nabla E_n(\boldsymbol{w}) = \sum_{n=1}^{N} \frac{\partial E_n}{\partial a_n} \nabla a_n = \sum_{n=1}^{N} (y_n - t_n) \nabla a_n.$$

The term ∇a_n represents the gradient of the scalar function $a_n(\boldsymbol{w})$ with respect to the weights. We do not propagate the derivative further back into the network. Differentiating again, we obtain

$$\begin{aligned}
\nabla\nabla E(\boldsymbol{w}) &= \sum_{n=1}^{N} (\nabla a_n)(\nabla y_n)^T + (y_n - t_n)\nabla\nabla y_n \\
&= \sum_{n=1}^{N} y_n(1-y_n)(\nabla a_n)(\nabla a_n)^T + \underbrace{(y_n - t_n)}_{\text{close to zero}}\nabla\nabla y_n \\
&\simeq \sum_{n=1}^{N} y_n(1-y_n)(\nabla a_n)(\nabla a_n)^T = \sum_{n=1}^{N} y_n(1-y_n)\boldsymbol{b}_n\boldsymbol{b}_n^T.
\end{aligned}$$

The term $(y_n - t_n)$ is assumed to be close to zero, for the reasons stated in Section 5.4.2.

Exercise 5.25

We split this problem into three parts: computing the gradient and the component $w_j^{(\tau)}$, solving the difference equation, and comparing with regularization.

The gradient and $w_j^{(\tau)}$ The first step is to compute the gradient, which we find to be

$$\nabla_{\boldsymbol{w}} E(\boldsymbol{w}) = \boldsymbol{H}(\boldsymbol{w} - \boldsymbol{w}^*).$$

The component along $w_j^{(\tau)}$ is given by $\boldsymbol{u}_j^T\boldsymbol{w}^{(\tau)}$. The recursive relationship for a single component of the $\boldsymbol{w}^{(\tau)}$ vector becomes

$$\begin{aligned}
\boldsymbol{w}^{(\tau)} &= \boldsymbol{w}^{(\tau-1)} - \rho\boldsymbol{H}(\boldsymbol{w}^{(\tau-1)} - \boldsymbol{w}^*) \\
\boldsymbol{u}_j^T\boldsymbol{w}^{(\tau)} &= \boldsymbol{u}_j^T\boldsymbol{w}^{(\tau-1)} - \rho\boldsymbol{u}_j^T\boldsymbol{H}(\boldsymbol{w}^{(\tau-1)} - \boldsymbol{w}^*) \\
w_j^{(\tau)} &= w_j^{(\tau-1)} - \rho\eta_j(w_j^{(\tau-1)} - w_j^*)
\end{aligned}$$

where we used $\boldsymbol{u}_j^T\boldsymbol{H} = (\boldsymbol{H}\boldsymbol{u}_j)^T = \eta_j\boldsymbol{u}_j^T$.

At this point, we wish to solve the following first order, linear, inhomogeneous difference equation with constant coefficients. To ease notation, we introduce $k := \rho\eta_j$ and $a = w_j^*$.

$$\begin{aligned}
w_j^{(\tau)} &= w_j^{(\tau-1)} - \rho\eta_j(w_j^{(\tau-1)} - w_j^*) \\
y_n &= y_{n-1} - k(y_{n-1} - a)
\end{aligned}$$

Solving the difference equation There exists a rich theory for dealing with equations such as these. Since we know the answer we're after, a proof by induction is also a viable option. However, we will directly solve the problem. First we write out terms as

$$\begin{aligned} y_n - y_{n-1}(1-k) &= ka \\ y_{n-1} - y_{n-2}(1-k) &= ka \\ y_{n-2} - y_{n-3}(1-k) &= ka \\ \vdots &= \vdots \\ y_1 - y_0(1-k) &= ka. \end{aligned}$$

We multiply the first equation by $(1-k)^0$, the second by $(1-k)^1$, the third by $(1-k)^2$, and so forth. Then we sum over the the left-hand side and right-hand side. The sum *telescopes*, and we appeal to the summation of a geometric series to obtain

$$y_n - y_0(1-k)^n = ka\left(1 + (1-k) + \cdots + (1-k)^{n-1}\right) = ka\left(\frac{1-(1-k)^n}{1-(1-k)}\right).$$

Substituting the definitions of a, k and $y_0 = 0$, we see that

$$y_n = a\left(1-(1-k)^n\right) \iff w_j^{(\tau)} = w_j^*\left(1-(1-\rho\eta_j)^\tau\right).$$

Comparing with regularization Convergence is straightforward, since if $|1-\rho\eta_j| < 1$, then the factor $(1-\rho\eta_j)^\tau$ will go to zero as $\tau \to \infty$, and clearly

$$\lim_{\tau\to\infty} w_j^{(\tau)} = w_j^*(1-0) = w_j^*.$$

To show the results related to the size of η_j, consider first the case when $\eta_j \gg (\rho\tau)^{-1}$. This is the same as $\rho\eta_j \gg 1/\tau$. Since $|1-\rho\eta_j| < 1$, we know that $0 < \rho\eta_j < 2$. No matter what value $\rho\eta_j$ has, τ must be huge, so $w_j^{(\tau)} \simeq w_j^*$.

Consider now the case when $\eta_j \ll (\rho\tau)^{-1}$. This is the same as $\rho\eta_j \ll 1/\tau$. No matter what value $\rho\eta_j$ has, τ must be close to zero, so $(1-(1-\rho\eta_j)^\tau)$ is small and $\left|w_j^{(\tau)}\right| \ll |w_j^*|$.

These results are analogous to those in Section 3.5.3, if we identify $(\rho\tau)^{-1}$ with the regularization parameter α. Notice that eigenvalues are denoted by η in this problem, and by λ in Section 3.5.3.

Exercise 5.27

From the problem text, we know that $\mathbb{E}[\boldsymbol{\xi}] = \mathbf{0}$. By "unit covariance", Bishop means that $\operatorname{cov}[\boldsymbol{\xi}] = \operatorname{cov}[\xi_i, \xi_j] = \boldsymbol{I} = \delta_{ij}$. First we see that $y(\boldsymbol{s}(\boldsymbol{x}, \boldsymbol{\xi}))$ takes the simple form

$$y(\boldsymbol{s}(\boldsymbol{x},\boldsymbol{\xi})) = y(\boldsymbol{x}+\boldsymbol{\xi}) \simeq y(\boldsymbol{x}) + \nabla y(\boldsymbol{x})^T\boldsymbol{\xi} + \boldsymbol{\xi}^T\nabla\nabla y(\boldsymbol{x})\boldsymbol{\xi} + \mathcal{O}(\xi_i\xi_j\xi_k)$$

Ignoring cubic terms, we can write the quadratic factor in the error function as

$$\begin{aligned} \left(y(\boldsymbol{s}(\boldsymbol{x},\boldsymbol{\xi})) - t\right)^2 &= \left([y(\boldsymbol{x})-t] + \nabla y(\boldsymbol{x})^T\boldsymbol{\xi} + \boldsymbol{\xi}^T\nabla\nabla y(\boldsymbol{x})\boldsymbol{\xi}\right)^2 \\ &\simeq [y(\boldsymbol{x})-t]^2 + 2\,[y(\boldsymbol{x})-t]\,\nabla y(\boldsymbol{x})^T\boldsymbol{\xi} \\ &+ \left(\nabla y(\boldsymbol{x})^T\boldsymbol{\xi}\right)^2 + 2\,[y(\boldsymbol{x})-t]\,\boldsymbol{\xi}^T\nabla\nabla y(\boldsymbol{x})\boldsymbol{\xi} + \ldots \end{aligned}$$

We drop the last term, since when $y(\boldsymbol{x}) \simeq t + \xi$ the term will be of order $\mathcal{O}(\xi^3)$. Substituting the remaining terms into the error function, we observe that

$$\begin{aligned}
\widetilde{E} &= \frac{1}{2} \iiint \left(y(\boldsymbol{s}(\boldsymbol{x}, \boldsymbol{\xi})) - t\right)^2 p(\boldsymbol{t} \mid \boldsymbol{x}) p(\boldsymbol{x}) p(\boldsymbol{\xi}) \, d\boldsymbol{x} \, d\boldsymbol{t} \, d\boldsymbol{\xi} \\
&= E + \frac{1}{2} \iiint \left(2\left[y(\boldsymbol{x}) - t\right] \nabla y(\boldsymbol{x})^T \boldsymbol{\xi} + \left(\nabla y(\boldsymbol{x})^T \boldsymbol{\xi}\right)^2 \right) p(\boldsymbol{t} \mid \boldsymbol{x}) p(\boldsymbol{x}) p(\boldsymbol{\xi}) \, d\boldsymbol{x} \, d\boldsymbol{t} \, d\boldsymbol{\xi} \\
&= E + \underbrace{0}_{\text{since } \mathbb{E}[\boldsymbol{\xi}] = \boldsymbol{0}} + \frac{1}{2} \iiint \left(\nabla y(\boldsymbol{x})^T \boldsymbol{\xi}\right)^2 p(\boldsymbol{t} \mid \boldsymbol{x}) p(\boldsymbol{x}) p(\boldsymbol{\xi}) \, d\boldsymbol{x} \, d\boldsymbol{t} \, d\boldsymbol{\xi}.
\end{aligned}$$

The final term may then be written as

$$\begin{aligned}
\Sigma &= \frac{1}{2} \iiint \nabla y(\boldsymbol{x})^T \boldsymbol{\xi} \boldsymbol{\xi}^T \nabla y(\boldsymbol{x}) p(\boldsymbol{t} \mid \boldsymbol{x}) p(\boldsymbol{x}) p(\boldsymbol{\xi}) \, d\boldsymbol{x} \, d\boldsymbol{t} \, d\boldsymbol{\xi} \\
&= \frac{1}{2} \int \nabla y(\boldsymbol{x})^T \left(\underbrace{\int \boldsymbol{\xi} \boldsymbol{\xi}^T p(\boldsymbol{\xi}) \, d\boldsymbol{\xi}}_{\text{cov}[\boldsymbol{\xi}] = \boldsymbol{I}} \right) \nabla y(\boldsymbol{x}) p(\boldsymbol{x}) \, d\boldsymbol{x} = \frac{1}{2} \int \left\| \nabla y(\boldsymbol{x}) \right\|^2 p(\boldsymbol{x}) \, d\boldsymbol{x}
\end{aligned}$$

Exercise 5.31

We wish to differentiate $\widetilde{E}(\boldsymbol{w})$ with respect to σ_j. Since $\widetilde{E}(\boldsymbol{w}) = E(\boldsymbol{w}) + \lambda \Omega(\boldsymbol{w})$ and $E(\boldsymbol{w})$ is not a function of σ_j, we need only consider the final term. We observe that

$$\begin{aligned}
\partial_{\sigma_j} \widetilde{E}(\boldsymbol{w}) = \partial_{\sigma_j} \lambda \Omega(\boldsymbol{w}) &= -\partial_{\sigma_j} \lambda \sum_i \ln \left(\sum_{j=1}^{M} \pi_j \mathcal{N}\left(w_i \mid \mu_j, \sigma_j\right) \right) \\
&= -\lambda \sum_i \frac{1}{\sum_{j=1}^{M} \pi_j \mathcal{N}\left(w_i \mid \mu_j, \sigma_j\right)} \partial_{\sigma_j} \left(\sum_{j=1}^{M} \pi_j \mathcal{N}\left(w_i \mid \mu_j, \sigma_j\right) \right) \\
&= \lambda \sum_i \frac{1}{\sum_{j=1}^{M} \pi_j \mathcal{N}\left(w_i \mid \mu_j, \sigma_j\right)} \left(\pi_j \mathcal{N}\left(w_i \mid \mu_j, \sigma_j\right) \left[\frac{1}{\sigma_j} - \frac{(w_i - \mu_j)^2}{\sigma_j^3} \right] \right) \\
&= \lambda \sum_i \gamma_j(w_i) \left[\frac{1}{\sigma_j} - \frac{(w_i - \mu_j)^2}{\sigma_j^3} \right],
\end{aligned}$$

where in the first equality we used $\ln(x)' = x'/x$, and in the second we used the fact that

$$\partial_{\sigma_j} \mathcal{N}\left(w_i \mid \mu_j, \sigma_j\right) = \mathcal{N}\left(w_i \mid \mu_j, \sigma_j\right) \left[\frac{(w_i - \mu_j)^2}{\sigma_j^3} - \frac{1}{\sigma_j} \right]$$

by the chain rule and product rule of differentiation. Finally, we used the definition of $\gamma_j(w_i)$ from Equation (5.140).

Exercise 5.37

We split this problem into two parts, one for $\mathbb{E}\left[\boldsymbol{t} \mid \boldsymbol{x}\right]$ and one for $s^2(\boldsymbol{x})$.

a) Computing $\mathbb{E}\left[\boldsymbol{t} \mid \boldsymbol{x}\right]$ is relatively straightforward. We use the definition of $p(\boldsymbol{t} \mid \boldsymbol{x})$:

$$\begin{aligned}\mathbb{E}\left[\boldsymbol{t} \mid \boldsymbol{x}\right] &= \int \boldsymbol{t} p(\boldsymbol{t} \mid \boldsymbol{x})\, d\boldsymbol{t} = \int \boldsymbol{t} \left[\sum_{k=1}^{K} \pi_k(\boldsymbol{x}) \mathcal{N}\left(\boldsymbol{t} \mid \boldsymbol{\mu}_k(\boldsymbol{x}), \boldsymbol{I}\sigma_k^2(\boldsymbol{x})\right)\right] d\boldsymbol{t} \\ &= \sum_{k=1}^{K} \pi_k(\boldsymbol{x})\, \mathbb{E}\left[\mathcal{N}\left(\boldsymbol{t} \mid \boldsymbol{\mu}_k(\boldsymbol{x}), \boldsymbol{I}\sigma_k^2(\boldsymbol{x})\right)\right] = \sum_{k=1}^{K} \pi_k(\boldsymbol{x}) \boldsymbol{\mu}_k(\boldsymbol{x})\end{aligned}$$

b) Computing $s^2(\boldsymbol{x})$ is a bit more involved, we first decompose the variance as

$$s^2(\boldsymbol{x}) = \mathbb{E}\left[\boldsymbol{t}^T\boldsymbol{t} \mid \boldsymbol{x}\right] - \mathbb{E}\left[\boldsymbol{t} \mid \boldsymbol{x}\right]^T \mathbb{E}\left[\boldsymbol{t} \mid \boldsymbol{x}\right].$$

We will study these two expected values in turn and order. The first term is

$$\mathbb{E}\left[\boldsymbol{t}^T\boldsymbol{t} \mid \boldsymbol{x}\right] = \int \boldsymbol{t}^T\boldsymbol{t} p(\boldsymbol{t} \mid \boldsymbol{x})\, d\boldsymbol{t} = \int \boldsymbol{t}^T\boldsymbol{t} \left[\sum_{k=1}^{K} \pi_k(\boldsymbol{x}) \mathcal{N}\left(\boldsymbol{t} \mid \boldsymbol{\mu}_k(\boldsymbol{x}), \boldsymbol{I}\sigma_k^2(\boldsymbol{x})\right)\right] d\boldsymbol{t}.$$

We interchange summation and integration, and factor the multidimensional Gaussian into one-dimensional factors. This is straightforward since the covariance matrix is diagonal.

$$\sum_{k=1}^{K} \pi_k(\boldsymbol{x}) \int \boldsymbol{t}^T\boldsymbol{t} \underbrace{\prod_{j=1}^{D} \mathcal{N}\left(t_j \mid \mu_{kj}(\boldsymbol{x}), \sigma_k^2(\boldsymbol{x})\right)}_{\text{factored Gaussian}} d\boldsymbol{t}$$

Next we write $\boldsymbol{t}^T\boldsymbol{t}$ as $\sum_{i=1}^{D} t_i^2$ and pull the factored Gaussian into the terms in each sum. The final step is to integrate over each term in the sum over i. When $j \neq i$, the integral is unity, but when $i = j$ we use Equation (1.50) from [Bishop, 2006], which states that $\mathbb{E}[x^2] = \mu^2 + \sigma^2$. The result is

$$\sum_{k=1}^{K} \pi_k(\boldsymbol{x}) \sum_{i=1}^{D} \left(\mu_{ki}^2(\boldsymbol{x}) + \sigma_k^2(\boldsymbol{x})\right) = \sum_{k=1}^{K} \pi_k(\boldsymbol{x}) \left(\boldsymbol{\mu}_k(\boldsymbol{x})^T \boldsymbol{\mu}_k(\boldsymbol{x}) + D\sigma_k^2(\boldsymbol{x})\right). \tag{22}$$

We are happy with the first term in the decomposed variance, and move on to the second. The second term in the decomposed variance is $\mathbb{E}\left[\boldsymbol{t} \mid \boldsymbol{x}\right]^T \mathbb{E}\left[\boldsymbol{t} \mid \boldsymbol{x}\right]$. To simplify notation, we write $\sum_{k=j}^{K} \pi_j(\boldsymbol{x})\boldsymbol{\mu}_j(\boldsymbol{x})$ as $\mathbb{E}\left[\boldsymbol{t} \mid \boldsymbol{x}\right]$. The second term in the decomposed variance may be written as

$$\mathbb{E}\left[\boldsymbol{t} \mid \boldsymbol{x}\right]^T \mathbb{E}\left[\boldsymbol{t} \mid \boldsymbol{x}\right] = \sum_{k=1}^{K} \pi_k(\boldsymbol{x})\, \mathbb{E}\left[\boldsymbol{t} \mid \boldsymbol{x}\right]^T \mathbb{E}\left[\boldsymbol{t} \mid \boldsymbol{x}\right] \tag{23}$$

where we are allowed to add the sum over k, since $\sum_{k=1}^{K} \pi_k(\boldsymbol{x}) = 1$.

Finally we're in a position to combine the results. We merge $\mathbb{E}\left[\boldsymbol{t}^T\boldsymbol{t} \mid \boldsymbol{x}\right]$ from Equation (22) with $\mathbb{E}\left[\boldsymbol{t} \mid \boldsymbol{x}\right]^T \mathbb{E}\left[\boldsymbol{t} \mid \boldsymbol{x}\right]$ from Equation (23) to obtain

$$\sum_{k=1}^{K} \pi_k(\boldsymbol{x}) \left[D\sigma_k^2(\boldsymbol{x}) + \boldsymbol{\mu}_k(\boldsymbol{x})^T \boldsymbol{\mu}_k(\boldsymbol{x}) - \mathbb{E}\left[\boldsymbol{t} \mid \boldsymbol{x}\right]^T \mathbb{E}\left[\boldsymbol{t} \mid \boldsymbol{x}\right]\right] =$$
$$\sum_{k=1}^{K} \pi_k(\boldsymbol{x}) \left[D\sigma_k^2(\boldsymbol{x}) + \left\|\boldsymbol{\mu}_k(\boldsymbol{x}) - \mathbb{E}\left[\boldsymbol{t} \mid \boldsymbol{x}\right]\right\|^2\right].$$

This is the result we're after. Notice that $D\sigma_k^2(\boldsymbol{x})$ is correct, not $\sigma_k^2(\boldsymbol{x})$ as Equation (5.160) states. This is a typo in the book.

2.6 Kernel methods

Exercise 6.3

First we observe that the Euclidean distance $d(\boldsymbol{x}, \boldsymbol{x}')$ may be expressed as a kernel function:

$$\begin{aligned} d(\boldsymbol{x}, \boldsymbol{x}') &= (\boldsymbol{x} - \boldsymbol{x}')^T(\boldsymbol{x} - \boldsymbol{x}') = \boldsymbol{x}^T\boldsymbol{x} - 2\boldsymbol{x}^T\boldsymbol{x}' + \boldsymbol{x}'^T\boldsymbol{x}' \\ &= \left(\boldsymbol{x}^T\boldsymbol{x}, i\sqrt{2}\boldsymbol{x}, 1\right)^T \left(1, i\sqrt{2}\boldsymbol{x}', \boldsymbol{x}'^T\boldsymbol{x}'\right) = \boldsymbol{\phi}\left(\boldsymbol{x}\right)^T \boldsymbol{\phi}\left(\boldsymbol{x}'\right) = k(\boldsymbol{x}, \boldsymbol{x}') \end{aligned}$$

where $i \equiv \sqrt{-1}$ denotes the imaginary unit. We can extend the above to an arbitrary kernel $k(\boldsymbol{x}, \boldsymbol{x}')$.

The nearest neighbor rule assigns a point $\boldsymbol{x}$ to the class of nearest neighbor. Instead of using $d(\boldsymbol{x}, \boldsymbol{x}')$, we use a kernel $k(\boldsymbol{x}, \boldsymbol{x}')$ to measure similarity. The algorithm becomes:

For a new point $\boldsymbol{x}$, assign to it the class of the point $\boldsymbol{x}^*$ defined by

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}' \in \mathcal{D}} k(\boldsymbol{x}, \boldsymbol{x}').$$

Exercise 6.7

Verifying Equation (6.17) We assume that $k_1(\boldsymbol{x}, \boldsymbol{x}')$ and $k_2(\boldsymbol{x}, \boldsymbol{x}')$ are valid kernels. In other words, there exist functions $\boldsymbol{\phi}_1(\cdot)$ and $\boldsymbol{\phi}_2(\cdot)$ so that

$$\begin{aligned} k_1(\boldsymbol{x}, \boldsymbol{x}') &= \boldsymbol{\phi}_1\left(\boldsymbol{x}\right)^T \boldsymbol{\phi}_1\left(\boldsymbol{x}'\right) \\ k_2(\boldsymbol{x}, \boldsymbol{x}') &= \boldsymbol{\phi}_2\left(\boldsymbol{x}\right)^T \boldsymbol{\phi}_2\left(\boldsymbol{x}'\right). \end{aligned}$$

The sum of these kernels is also a kernel, since

$$\begin{aligned} k(\boldsymbol{x}, \boldsymbol{x}') &= k_1(\boldsymbol{x}, \boldsymbol{x}') + k_2(\boldsymbol{x}, \boldsymbol{x}') \\ &= \boldsymbol{\phi}_1\left(\boldsymbol{x}\right)^T \boldsymbol{\phi}_1\left(\boldsymbol{x}'\right) + \boldsymbol{\phi}_2\left(\boldsymbol{x}\right)^T \boldsymbol{\phi}_2\left(\boldsymbol{x}'\right) \\ &= \underbrace{\left(\boldsymbol{\phi}_1\left(\boldsymbol{x}\right), \boldsymbol{\phi}_2\left(\boldsymbol{x}\right)\right)^T}_{\boldsymbol{\phi}(\boldsymbol{x})^T} \underbrace{\left(\boldsymbol{\phi}_1\left(\boldsymbol{x}'\right), \boldsymbol{\phi}_2\left(\boldsymbol{x}'\right)\right)}_{\boldsymbol{\phi}(\boldsymbol{x}')} = \boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{\phi}(\boldsymbol{x}'). \end{aligned}$$

Verifying Equation (6.18) Again we assume that

$$\begin{aligned}
k_1(\boldsymbol{x},\boldsymbol{x}') &= \boldsymbol{\phi}_1\left(\boldsymbol{x}\right)^T \boldsymbol{\phi}_1\left(\boldsymbol{x}'\right)\\
k_2(\boldsymbol{x},\boldsymbol{x}') &= \boldsymbol{\phi}_2\left(\boldsymbol{x}\right)^T \boldsymbol{\phi}_2\left(\boldsymbol{x}'\right).
\end{aligned}$$

The product of these kernels is also a kernel. We write out to obtain a quadratic form

$$k(\boldsymbol{x},\boldsymbol{x}') = k_1(\boldsymbol{x},\boldsymbol{x}')k_2(\boldsymbol{x},\boldsymbol{x}') = \boldsymbol{\phi}_1\left(\boldsymbol{x}\right)^T \boldsymbol{\phi}_1\left(\boldsymbol{x}'\right) \boldsymbol{\phi}_2\left(\boldsymbol{x}\right)^T \boldsymbol{\phi}_2\left(\boldsymbol{x}'\right)$$

Since $\boldsymbol{a}\boldsymbol{b}^T = \boldsymbol{b}\boldsymbol{a}^T$ we can write this as

$$\begin{aligned}
\boldsymbol{\phi}_1\left(\boldsymbol{x}\right)^T \boldsymbol{\phi}_1\left(\boldsymbol{x}'\right) \boldsymbol{\phi}_2\left(\boldsymbol{x}\right)^T \boldsymbol{\phi}_2\left(\boldsymbol{x}'\right) &= \boldsymbol{\phi}_1\left(\boldsymbol{x}\right)^T \boldsymbol{\phi}_2\left(\boldsymbol{x}\right) \boldsymbol{\phi}_1\left(\boldsymbol{x}'\right)^T \boldsymbol{\phi}_2\left(\boldsymbol{x}'\right)\\
&= \underbrace{\boldsymbol{\phi}_1\left(\boldsymbol{x}\right)^T \boldsymbol{\phi}_2\left(\boldsymbol{x}\right)}_{\phi(\boldsymbol{x})^T} \underbrace{\boldsymbol{\phi}_1\left(\boldsymbol{x}'\right)^T \boldsymbol{\phi}_2\left(\boldsymbol{x}'\right)}_{\phi(\boldsymbol{x}')} = \phi(\boldsymbol{x})^T\phi(\boldsymbol{x}').
\end{aligned}$$

Notice that $\phi(\boldsymbol{x}) = \phi(\boldsymbol{x})^T := \boldsymbol{\phi}_1\left(\boldsymbol{x}\right)^T \boldsymbol{\phi}_2\left(\boldsymbol{x}\right)$ is a scalar, which is a valid kernel because it's an inner product in a one-dimensional feature space.

Exercise 6.11

Consider first $\exp(x_i x_i'/\sigma^2)$. Using the power series $\exp(z) = \sum_{j=0} z^j/j!$, the exponential may be written as

$$\begin{aligned}
\exp(x_i x_i'/\sigma^2) &= 1 + \left(\frac{x_i x_i'}{\sigma^2}\right) + \frac{1}{2}\left(\frac{x_i x_i'}{\sigma^2}\right)^2 + \frac{1}{6}\left(\frac{x_i x_i'}{\sigma^2}\right)^3 + \cdots\\
&= \left(1, \frac{x_i}{\sigma}, \frac{1}{\sqrt{2}}\left(\frac{x_i}{\sigma}\right)^2, \frac{1}{\sqrt{6}}\left(\frac{x_i}{\sigma}\right)^3, \dots\right)^T\\
&\quad \left(1, \frac{x_i'}{\sigma}, \frac{1}{\sqrt{2}}\left(\frac{x_i'}{\sigma}\right)^2, \frac{1}{\sqrt{6}}\left(\frac{x_i'}{\sigma}\right)^3, \dots\right) = \boldsymbol{\phi}(x_i)^T\boldsymbol{\phi}(x_i')
\end{aligned}$$

Extending this, we find that

$$\begin{aligned}
\exp(\boldsymbol{x}^T\boldsymbol{x'}/\sigma^2) &= \exp(x_1 x_1'/\sigma^2)\exp(x_2 x_2'/\sigma^2)\cdots\exp(x_M x_M'/\sigma^2)\\
&= \boldsymbol{\phi}(x_1)^T\boldsymbol{\phi}(x_1')\boldsymbol{\phi}(x_2)^T\boldsymbol{\phi}(x_2')\cdots\boldsymbol{\phi}(x_M)^T\boldsymbol{\phi}(x_M')
\end{aligned}$$

This is a valid kernel by Equation (6.18) in [Bishop, 2006], since each factor in the product is valid. Each term in the product consists of a mapping to a feature space of infinite dimensionality, given by

$$\boldsymbol{\phi}(x_i) = \left(1, \frac{x_i}{\sigma}, \frac{1}{\sqrt{2}}\left(\frac{x_i}{\sigma}\right)^2, \frac{1}{\sqrt{6}}\left(\frac{x_i}{\sigma}\right)^3, \dots, \frac{1}{\sqrt{k!}}\left(\frac{x_i}{\sigma}\right)^k\right).$$

Exercise 6.15

The entries of the *Gram matrix* $\boldsymbol{K}_{nm}$ are given by $\boldsymbol{\phi}(\boldsymbol{x}_n)^T\boldsymbol{\phi}(\boldsymbol{x}_m) = k(\boldsymbol{x}_n, \boldsymbol{x}_m)$. The kernel $k(\boldsymbol{x}_n, \boldsymbol{x}_m)$ is valid if and only if $\boldsymbol{K}$ is positive semidefinite. Therefore we may safely assume that $\boldsymbol{K}$ is positive semidefinite, which is equivalent to its leading principal minors being non-negative. In particular, the $\det(\boldsymbol{K}) \geq 0$.

In the 2 × 2 case, the Gram matrix is

$$\boldsymbol{K} = \begin{pmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & k(\boldsymbol{x}_1, \boldsymbol{x}_2) \\ k(\boldsymbol{x}_2, \boldsymbol{x}_1) & k(\boldsymbol{x}_2, \boldsymbol{x}_2) \end{pmatrix}.$$

The determinant becomes

$$\det(\boldsymbol{K}) = k(\boldsymbol{x}_1, \boldsymbol{x}_1)k(\boldsymbol{x}_2, \boldsymbol{x}_2) - k(\boldsymbol{x}_1, \boldsymbol{x}_2)^2 \geq 0,$$

and this is equivalent to the Cauchy-Schwartz inequality. We used the symmetry of the kernel function in the expression for the determinant, i.e. $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = k(\boldsymbol{x}_2, \boldsymbol{x}_1)$.

Exercise 6.18

We use a Parzen density estimator with a Gaussian component density function to model the joint distribution. Since the covariance matrix is diagonal, the Gaussian component density factors as

$$f(x - x_n, t - t_n) = \mathcal{N}(x \mid x_n, \sigma^2)\mathcal{N}(t \mid t_n, \sigma^2).$$

We find expressions for $g(x - x_n)$ and $k(x, x_n)$ as

$$\begin{aligned} g(x - x_n) &= \int f(x - x_n, t)\, dt = \mathcal{N}(x \mid x_n, \sigma^2) \\ k(x, x_n) &= \frac{g(x - x_n)}{\sum_m^N g(x - x_m)} = \frac{\mathcal{N}(x \mid x_n, \sigma^2)}{\sum_m^N \mathcal{N}(x \mid x_m, \sigma^2)}. \end{aligned}$$

We will solve the problem by expressing the quantities $p(t \mid x)$, $\mathbb{E}[t \mid x]$ and $\mathrm{var}[t \mid x]$ as functions of the normal distributions, which are proportional the the kernel as seen above.

The conditional density $p(t \mid x)$ becomes

$$\begin{aligned} p(t \mid x) = \frac{p(t, x)}{p(x)} &= \frac{\frac{1}{N}\sum_n f(x - x_n, t - t_n)}{\int \frac{1}{N}\sum_m f(x - x_m, t - t_m)\, dt} \\ &= \frac{\sum_n \mathcal{N}(x \mid x_n, \sigma^2)\mathcal{N}(t \mid t_n, \sigma^2)}{\sum_m \mathcal{N}(x \mid x_m, \sigma^2)} = \sum_n k(x, x_n)\mathcal{N}(t \mid t_n, \sigma^2). \end{aligned}$$

The conditional expectation $\mathbb{E}\left[t \mid x\right]$ can be read off from Equation (6.45), or we can take take expected value of the above. Either way, the result is

$$\mathbb{E}\left[t \mid x\right] = \sum_n k(x, x_n) t_n = \sum_n \left[\frac{\mathcal{N}(x \mid x_n, \sigma^2)}{\sum_m^N \mathcal{N}(x \mid x_m, \sigma^2)} \right] t_n.$$

The conditional variance $\operatorname{var}\left[t \mid x\right]$ can be calculated using $\operatorname{var}\left[t \mid x\right] = \mathbb{E}\left[t^2 \mid x\right] - \mathbb{E}\left[t \mid x\right]^2$. We know the second term already from the previous sub-problem, so let's focus on the second term. First we use for the conditional density above, and then we use Equation (1.50) from [Bishop, 2006].

$$\begin{aligned}\mathbb{E}\left[t^2 \mid x\right] &= \int p(t \mid x) t^2 \, dt = \int \frac{p(t, x)}{p(x)} t^2 \, dt = \int \left[\sum_n k(x, x_n) \mathcal{N}(t \mid t_n, \sigma^2) \right] t^2 \, dt \\ &= \sum_n k(x, x_n) \int \mathcal{N}(t \mid t_n, \sigma^2) t^2 \, dt = \sum_n k(x, x_n)(t_n^2 + \sigma^2) = \sigma^2 + \sum_n k(x, x_n) t_n^2\end{aligned}$$

Combining these results, the conditional variance becomes

$$\operatorname{var}\left[t \mid x\right] = \mathbb{E}\left[t^2 \mid x\right] - \mathbb{E}\left[t \mid x\right]^2 = \sigma^2 + \sum_n k(x, x_n) t_n^2 - \left(\sum_n k(x, x_n) t_n \right)^2.$$

Exercise 6.24

The matrix $\boldsymbol{W}$ is positive definite since

$$\boldsymbol{x}^T \boldsymbol{W} \boldsymbol{x} = \sum_i x_i W_{ii} x_i = \sum_i x_i^2 W_{ii}.$$

For a vector $\boldsymbol{x} \neq \boldsymbol{0}$, the x_i^2 terms are positive, and the W_{ii} terms are positive. Therefore their product is positive, and the sum of positive terms is always positive.

We can also appeal to the relationship with eigenvalues, since the eigenvalues of a diagonal matrix are the diagonal elements. A matrix is positive definite if and only if every eigenvalue is positive.

To see that the sum of two positive definitive matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ is positive definite, write

$$\boldsymbol{x}^T (\boldsymbol{A} + \boldsymbol{B}) \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} + \boldsymbol{x}^T \boldsymbol{B} \boldsymbol{x}.$$

By assumption, both terms are positive, and the sum of positive terms is positive.

2.7 Sparse Kernel Machines

Exercise 7.4 and 7.5

We solve these problems, but not in the order that they're given.

The distance from the margin to the nearest point is ρ. Since $\rho = t_i y_i / \left\| \boldsymbol{w} \right\|$, where i is the index of a support vector data point (there are at least two), we have

$$\frac{1}{\rho^2} = \frac{\left\| \boldsymbol{w} \right\| \left\| \boldsymbol{w} \right\|}{(t_i y_i)^2} = \boldsymbol{w}^T \boldsymbol{w}$$

where we used Equation (7.4) from the book, i.e. $t_i y_i = 1$, which is valid for data points corresponding to support vectors. This solves **the second part of Exercise 7.5**.

We now expand the inner product and apply Equation (7.13) to get

$$\frac{1}{\rho^2} = \boldsymbol{w}^T \boldsymbol{w} = \sum_m a_m t_m \sum_n a_n t_n k(\boldsymbol{x}_m, \boldsymbol{x}_n) = \sum_m a_m t_m \left(y(\boldsymbol{x}_m) - b \right). \tag{24}$$

Recall that for every $(\boldsymbol{x}_n, t_n)$ data point, either

a) The constraint is inactive and $a_n = 0$. The point is not a support vector.
b) The constraint is active and $y(\boldsymbol{x}_n) t_n = 1$. The point is a support vector.

Combining the above with Equation (7.9), we observe that

$$\frac{1}{\rho^2} = \sum_m a_m t_m \left(y_m - b \right) = \sum_m a_m t_m y_m - b \underbrace{\sum_m a_m t_m}_{0} = \sum_m a_m. \tag{25}$$

In words: each term $a_m \cdot t_m y_m$ in the sum from $m = 1, \ldots, N$ is either $0 \cdot t_m y_m$ or $a_m \cdot 1$, which is equal to summing over the $\{a_m\}$. This solves **Exercise 7.4**

Finally we make the observation that

$$2\tilde{L}(\boldsymbol{a}) = 2 \sum_m a_m - \sum_m a_m t_m \sum_n a_n t_n k(\boldsymbol{x}_m, \boldsymbol{x}_n),$$

and use Equations (24) and (25) to write the second term as $\sum_m a_m$. This solves **the first part of Exercise 7.5**. We have solved the exercises.

Exercise 7.10

TODO

Exercise 7.12

TODO

Exercise 7.16

In this problem, we will omit the subscripts to ease notation. Differentiating $\lambda(\alpha)$ twice by making use of Equation (7.100), we obtain the second derivative, which is

$$\lambda''(\alpha) = \frac{-\left(\frac{s}{\alpha}\right)^2 - 4\left(\frac{s^2}{\alpha} - (q^2 - s)\right)}{2(\alpha + s)}.$$

We know that $\alpha \geq 0$, and since $\boldsymbol{C}$ is positive semidefinite, it's inverse is also positive semidefinite and so $s \geq 0$ too. The denominator in $\lambda''(\alpha)$ must therefore be positive.

The first term in the numerator is also positive, and in the stationary point when $\alpha^* = s^2/(q^2 - s)$, the second term in the numerator becomes

$$\frac{s^2}{\alpha^*} - (q^2 - s) = s^2\left(\frac{q^2 - s}{s^2} - (q^2 - s)\right) = 0.$$

Since $\lambda''(\alpha^*) \leq 0$ at the stationary point, it's indeed a maximum point.

2.8 Graphical Models

Exercise 8.3

Showing that $p(a,b) \neq p(a)p(b)$. We first show that $p(a,b) \neq p(a)p(b)$. To do so, we only need to produce a counterexample. In other words, we must provide values a_i and b_j such that $p(a = a_i, b = b_j) \neq p(a = a_i)p(b = b_j)$. Specifically, we will show that $p(a = 1, b = 1)$ is not equal to $p(a = 1)p(b = 1)$.

We first compute the marginal probabilities as:

$$p(a = 1) = \sum_b \sum_c p(a = 1, b, c) = 0.192 + 0.064 + 0.048 + 0.096 = 0.4$$
$$p(b = 1) = \sum_a \sum_c p(a, b = 1, c) = 0.048 + 0.216 + 0.048 + 0.096 = 0.408$$

The joint probability is given by

$$p(a = 1, b = 1) = \sum_c p(a = 1, b = 1, c) = 0.048 + 0.096 = 0.144$$

Clearly this is a counterexample, since $0.4 \cdot 0.408 = 0.1632 \neq 0.144$. As a result, the distribution over a and b is marginally dependent, i.e. $p(a,b) \neq p(a)p(b)$.

Showing that $p(a,b \mid c) = p(a \mid c)p(b \mid c)$. To show that a and b are marginally independent when conditioned on c, we must show that $p(a,b \mid c) = p(a \mid c)p(b \mid c)$ for every value of c. In this problem, we must show that it holds for both $c = 0$ and $c = 1$.

We will only show the computation when $c = 0$, since the computations are completely identical when $c = 1$. First we start by computing the probability

$$p(c = 0) = 0.192 + 0.048 + 0.192 + 0.048 = 0.48.$$

We want to find $p(a \mid c = 0)$ and $p(b \mid c = 0)$ for all values of a and b. Using the product rule and sum rule, we obtain

$$\begin{aligned}
p(a = 0 \mid c = 0) &= \frac{p(a = 0, c = 0)}{p(c = 0)} = \frac{\sum_b p(a = 0, b, c = 0)}{p(c = 0)} = \frac{0.192 + 0.048}{0.48} = 0.5 \\
p(a = 1 \mid c = 0) &= 1 - p(a = 0 \mid c = 0) = 0.5 \\
p(b = 0 \mid c = 0) &= \frac{p(b = 0, c = 0)}{p(c = 0)} = \frac{\sum_a p(a, b = 0, c = 0)}{p(c = 0)} = \frac{0.192 + 0.192}{0.48} = 0.8 \\
p(b = 1 \mid c = 0) &= 1 - p(b = 0 \mid c = 0) = 0.2
\end{aligned}$$

Next we evaluate the joint distribution for all values of a and b. When $c = 0$, we use Bayes theorem to write

$$p(a, b \mid c = 0) = \frac{p(a, b, c = 0)}{p(c = 0)}.$$

Structuring the computations in tabular form, we obtain the following numbers:

$$\begin{aligned}
\frac{p(a = 0, b = 0, c = 0)}{p(c = 0)} &= \frac{0.192}{0.48} = 0.4 \\
\frac{p(a = 0, b = 1, c = 0)}{p(c = 0)} &= \frac{0.048}{0.48} = 0.1 \\
\frac{p(a = 1, b = 0, c = 0)}{p(c = 0)} &= \frac{0.192}{0.48} = 0.4 \\
\frac{p(a = 1, b = 1, c = 0)}{p(c = 0)} &= \frac{0.048}{0.48} = 0.1
\end{aligned}$$

Directly evaluating every possibility, we notice that indeed

$$\begin{aligned}
p(a = 0, b = 0 \mid c = 0) &= 0.4 \qquad & p(a = 0 \mid c = 0)p(b = 0 \mid c = 0) &= 0.5 \cdot 0.8 = 0.4 \\
p(a = 0, b = 1 \mid c = 0) &= 0.1 \qquad & p(a = 0 \mid c = 0)p(b = 1 \mid c = 0) &= 0.5 \cdot 0.2 = 0.1 \\
p(a = 1, b = 0 \mid c = 0) &= 0.4 \qquad & p(a = 1 \mid c = 0)p(b = 0 \mid c = 0) &= 0.5 \cdot 0.8 = 0.4 \\
p(a = 1, b = 1 \mid c = 0) &= 0.1 \qquad & p(a = 1 \mid c = 0)p(b = 1 \mid c = 0) &= 0.5 \cdot 0.2 = 0.1
\end{aligned}$$

We have shown that $p(a, b \mid c = 0) = p(a \mid c = 0)p(b \mid c = 0)$ for all of the $2^2 = 4$ possible values of a and b. Analogous computations for $c = 1$ reveal that $p(a, b \mid c) = p(a \mid c)p(b \mid c)$ in general. We omit these tedious calculations, as they are indistinguishable from those already shown.

Exercise 8.6

The logical OR function, i.e. OR : $\{0, 1\}^M \to \{0, 1\}$, returns 1 if any one of the M arguments is 1. It only returns 0 if every one of the M arguments is 0.

Reduction to the logical OR function. The probabilistic OR function is given by

$$p(y = 1 \mid x_1, \ldots, x_M) = 1 - (1 - \mu_0) \prod_{i=1}^{M} (1 - \mu_i)^{x_i} .$$

Notice that when $\mu_0 = 0$ and $\mu_i = 1$ for $i = 1, \ldots, M$ the above reduces to

$$p(y = 1 \mid x_1, \ldots, x_M) = 1 - \prod_{i=1}^{M} 0^{x_i} = \begin{cases} 0 & \text{if every } x_i = 0 \\ 1 & \text{else} \end{cases} ,$$

which is exactly the logical OR function, since we take 0^0 to be 1. We observe that when $0 \leq \mu_i \leq 1$, then $0 \leq (1-\mu_0) \prod_{i=1}^{M} (1 - \mu_i)^{x_i} \leq 1$, and the returned value is between 0 and 1 inclusive. Furthermore, if $\mu_0 \approx 0$ and $\mu_i \approx 1$ for $i = 1, \ldots, M$, then the probabilistic OR function is "close" to the logical OR function.

Interpretation of the μ_is. We can interpret μ_0 by observing that

$$p(y = 1 \mid \boldsymbol{x} = \boldsymbol{0}) = 1 - (1 - \mu_0) = \mu_0.$$

Therefore the parameter μ_0 may be interpreted as the probability that $y = 1$ when every $x_i = 0$. The logical OR function has zero probability of this, so setting $\mu_0 \approx 0$ yields a function whose behavior resembles the logical OR.

We can interpret the μ_is similarly by noting that

$$p(y = 1 \mid x_i = 1, \boldsymbol{x}_{\{j \neq i\}} = \boldsymbol{0}) = 1 - (1 - \mu_i) = \mu_i.$$

The interpretation is that μ_i is the probability that $y = 1$ when $x_i = 1$ and every other $x_j = 0$. In the logical OR function this probability is one, so setting $\mu_0 \approx 1$ yields a function whose behavior resembles the logical OR.

Exercise 8.8

Recall that the meaning of the $\perp\!\!\!\perp$ symbol for conditional independence:

$$\begin{aligned} a \perp\!\!\!\perp b \mid d &\iff \quad p(a, b \mid d) = p(a \mid d) p(b \mid d) \\ a \perp\!\!\!\perp b, c \mid d &\iff p(a, b, c \mid d) = p(a \mid d) p(b, c \mid d). \end{aligned}$$

We will show that $a \perp\!\!\!\perp b, c \mid d$ implies $a \perp\!\!\!\perp b \mid d$. From the sum rule, we know that

$$p(a, b \mid d) = \sum_c p(a, b, c \mid d).$$

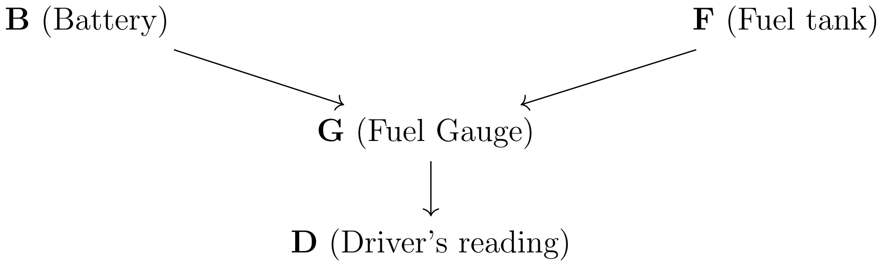
Appealing to the conditional independence $a \perp\!\!\!\perp b, c \mid d$, we write the above as

$$p(a, b \mid d) = \sum_c p(a, b, c \mid d) = p(a \mid d) \sum_c p(b, c \mid d) = p(a \mid d) p(b \mid d).$$

We have shown that $a \perp\!\!\!\perp b, c \mid d$ implies $a \perp\!\!\!\perp b \mid d$.

Exercise 8.11

This is a lengthy problem, as the computations require some care. We add a **D** node to the graph given in Figure 8.21 in [Bishop, 2006], obtaining the following structure.



We split the problem into three parts: we (1) evaluate $p(F = 0 \mid D = 0)$, (2) evaluate $p(F = 0 \mid D = 0, B = 0)$, and (3) compare the results and discuss.

Evaluating $p(F = 0 \mid D = 0)$**.** From Bayes theorem, we have

$$p(F = 0 \mid D = 0) = \frac{p(D = 0 \mid F = 0)p(F = 0)}{p(D = 0)}.$$

We evaluate the three terms in the equation above in turn.

a) The denominator $p(D = 0)$ may be evaluated using the sum rule, and the graph factorization property, as

$$\begin{aligned} p(D = 0) &= \sum_{B,F,G} p(D = 0, B, F, G) \\ &= \sum_{B,F,G} p(D = 0 \mid G)p(G \mid B, F)p(B)p(F) = 0.352. \end{aligned}$$

The sum ranges over $2^3 = 8$ terms. To reduce the probability of error when computing by hand, this sum (and every other sum in this problem) was computed in the accompanying `ch8_problem_11.py` Python script.

b) The first factor in the numerator is $p(D = 0 \mid F = 0)$, which evaluates to

$$\begin{aligned} p(D = 0 \mid F = 0) &= \sum_{B,G} p(D = 0, B, G \mid F = 0) \\ &= \sum_{B,G} p(D = 0 \mid G)p(G \mid F = 0, B)p(B) = 0.748. \end{aligned}$$

c) The second factor is simply $p(F = 0) = 0.1$.

Combining the three results above, we find that

$$p(F = 0 \mid D = 0) = \frac{p(D = 0 \mid F = 0)p(F = 0)}{p(D = 0)} = \frac{0.748 \cdot 0.1}{0.352} = 0.2125.$$

Evaluating $P(F = 0 \mid D = 0, B = 0)$. We solve this in a slightly different way, using the product rule instead of Bayes theorem. In the numerator, we will have a joint distribution $p(D = 0, B = 0, F = 0)$ instead of the product $p(D = 0, B = 0 \mid F = 0)p(F = 0)$. Either approach would produce the same result, of course.

$$P(F = 0 \mid D = 0, B = 0) = \frac{p(D = 0, B = 0, F = 0)}{p(D = 0, B = 0)}.$$

We evaluate the denominator and the numerator.

a) The denominator $p(D = 0, B = 0)$ may be evaluated using the sum rule and the graph factorization. The sum ranges over 2^2 terms, and evaluates to

$$\begin{aligned} p(D = 0, B = 0) &= \sum_{F,G} p(D = 0, B = 0, F, G) \\ &= \sum_{F,G} p(D = 0 \mid G)p(G \mid B = 0, F)p(B = 0)p(F) = 0.0748. \end{aligned}$$

b) The numerator evaluates to

$$\begin{aligned} p(D = 0, B = 0, F = 0) &= \sum_{G} p(D = 0, B = 0, G, F = 0) \\ &= \sum_{G} p(D = 0 \mid G)p(G \mid F = 0, B = 0)p(B = 0)p(F = 0) \\ &= 0.0082 \end{aligned}$$

Combining the two results above above, we find that

$$P(F = 0 \mid D = 0, B = 0) = \frac{p(D = 0, B = 0, F = 0)}{p(D = 0, B = 0)} = \frac{0.0082}{0.0748} \approx 0.1096.$$

Discussion. When the driver informs us that the fuel gauge reads empty, the probability of the tank actually being empty is 0.2125. If we also know that if the battery is flat, the probability of the tank actually being empty *decreases* to 0.1096. Finding out that the battery is flat *explains away* the observation that the driver reads the fuel gauge as empty. Intuitively, this is because we now have another probable reason for the gauge showing empty, which we had relatively little prior reason to believe was true.

Exercise 8.16

We wish to evaluate $p(x_n \mid x_N)$ for every $n = 1, 2, \ldots, N - 1$. We will make some observations, and then sketch an algorithm.

Observations. By using the product rule and then marginalizing over every variable apart from x_n, we obtain the following expression for the conditional probability.

$$\begin{aligned} p(x_n \mid x_N) &= \sum_{x_1} \cdots \sum_{x_{n-1}} \sum_{x_{n+1}} \cdots \sum_{x_N} \frac{p(\boldsymbol{x})}{p(x_N)} \\ &= \frac{1}{Z} \sum_{x_1} \cdots \sum_{x_{n-1}} \sum_{x_{n+1}} \cdots \sum_{x_N} \psi_{1,2}(x_1, x_2) \psi_{2,3}(x_2, x_3) \cdots \frac{\psi_{N-1,N}(x_{N-1}, x_N)}{p(x_N)}. \end{aligned}$$

The equation above only differs from Equation (8.52) because of the marginal distribution $p(x_N)$ in the denominator in the last term. Assuming that we know $p(x_N)$, we group terms as in Equation (8.52), writing them as

$$\frac{1}{Z} \mu_\alpha(x_n) \underbrace{\left[\sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \cdots \left[\sum_{x_N} \frac{\psi_{N-1,N}(x_{N-1}, x_N)}{p(x_N)} \right] \cdots \right]}_{\mu_\beta^*(x_n)}.$$

The difference between the $\mu_\beta^*(x_n)$ introduced here and the $\mu_\beta(x_n)$ in the book is the denominator in the innermost parenthesis.

Algorithm sketch. We sketch an $\mathcal{O}(NK^2)$ algorithm which uses message passing.

1) The first step involves finding $p(x_N)$. Since $p(x_N) = \mu_\alpha(x_N)\mu_\beta(x_N)/Z = \mu_\alpha(x_N)/Z$, we compute $\mu_\alpha(x_2)$, then $\mu_\alpha(x_3)$, and so forth by use of the recursive relationship detailed in Equation (8.55). This step is $\mathcal{O}(NK^2)$, and every $\mu_\alpha(x_n)$ is stored.
2) Having obtained $p(x_N)$, we compute the $\mu_\beta^*(x_n)$ by the following equations, which give an initial value and a recursive relationship.

$$\begin{aligned} \mu_\beta^*(x_{N-1}) &= \sum_{x_N} \frac{\psi_{N-1,N}(x_{N-1}, x_N)}{p(x_N)} \\ \mu_\beta^*(x_n) &= \sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \mu_\beta^*(x_{n+1}) \end{aligned}$$

   This step is also $\mathcal{O}(NK^2)$, and every $\mu_\beta^*(x_n)$ is stored.
3) Now we can compute $p(x_n \mid x_N)$ for every $n = 1, \ldots, N-1$ using

$$p(x_n \mid x_N) = \frac{1}{Z} \mu_\alpha(x_n) \mu_\beta^*(x_n).$$

   Since $\sum_{x_n} p(x_n \mid x_N) = 1$, we compute Z as $\sum_{x_n} \mu_\alpha(x_n)\mu_\beta^*(x_n)$.

The algorithm runs in $\mathcal{O}(NK^2)$ time, where N is the number of variable nodes and K is the number of states per variable. We propagate messages forward, then backwards, evaluate Z and compute the conditional probabilities $p(x_n \mid x_N)$.

Exercise 8.21

The solution to this problem has somewhat dense notation. Drawing and working through a concrete example while reading is highly recommended. From the sum rule, we have

$$p(\boldsymbol{x}_s) = \sum_{\boldsymbol{x} \setminus \boldsymbol{x}_s} p(\boldsymbol{x}).$$

The joint distribution $p(\boldsymbol{x})$ can be factored as the product of:

- $f_s(\boldsymbol{x}_s)$, the factor defined over the variables $\boldsymbol{x}_s$. This factor is not marginalized out.
- The factors "two steps" away from $f_s(\boldsymbol{x}_s)$ in the graph, i.e. the "grandchildren." More formally, for every variable node index $i \in \mathrm{ne}(f_s)$ adjacent to f_s, the product of the factors associated with the factor nodes $\ell \in \mathrm{ne}(x_i) \setminus f_s$.

We substitute this factorization of $p(\boldsymbol{x})$ into the above to obtain

$$p(\boldsymbol{x}_s) = \sum_{\boldsymbol{x} \setminus \boldsymbol{x}_s} f_s(\boldsymbol{x}_s) \left[\prod_{i \in \mathrm{ne}(f_s)} \left(\prod_{\ell \in \mathrm{ne}(x_i) \setminus f_s} F_\ell(x_i, X_\ell) \right) \right],$$

where $F_\ell(x_i, X_\ell)$ denotes the factors associated with factor node ℓ and it's sub-tree, adjacent to $x_i \in \boldsymbol{x}_s$. Next we bring the factor $f_s(\boldsymbol{x}_s)$ out of the sum, and interchange summation and products. Interchanging summation and products is valid since the products are over disjoint factors. We obtain

$$p(\boldsymbol{x}_s) = f_s(\boldsymbol{x}_s) \prod_{i \in \mathrm{ne}(f_s)} \left(\prod_{\ell \in \mathrm{ne}(x_i) \setminus f_s} \sum_{X_\ell} F_\ell(x_i, X_\ell) \right),$$

since the variables $\boldsymbol{x} \setminus \boldsymbol{x}_s$ is the union of every set X_ℓ. Recall that the variables X_ℓ belong to the sub-trees associated with the factors "two steps" away from $f_s(\boldsymbol{x}_s)$. Using the definition of $\mu_{f_\ell \to x_i}(x_i)$ from Equation (8.64) followed by Equation (8.69), we obtain

$$\begin{aligned} p(\boldsymbol{x}_s) &= f_s(\boldsymbol{x}_s) \prod_{i \in \mathrm{ne}(f_s)} \left(\prod_{\ell \in \mathrm{ne}(x_i) \setminus f_s} \sum_{X_\ell} F_\ell(x_i, X_\ell) \right) \\ &= f_s(\boldsymbol{x}_s) \prod_{i \in \mathrm{ne}(f_s)} \left(\prod_{\ell \in \mathrm{ne}(x_i) \setminus f_s} \mu_{f_\ell \to x_i}(x_i) \right) \\ &= f_s(\boldsymbol{x}_s) \prod_{i \in \mathrm{ne}(f_s)} \mu_{x_i \to f_s}(x_i). \end{aligned}$$

This is equivalent to Equation (8.72), which is what we wanted to show.

2.9 Mixture Models and EM

Exercise 9.2

Recall the general Robbins-Monro algorithm: given $p(\theta, z)$ we compute the root of the *regression function*

$$f(\theta) = \mathbb{E}_z [z \mid \theta] = \int zp(z \mid \theta)\, d\theta$$

by iterating the recursive equation

$$\theta^N = \theta^{N-1} - a_{N-1} z \left(\theta^{N-1}\right). \tag{26}$$

In this case, let's consider θ as a specific $\boldsymbol{\mu}_k$. We fixate k, and consider the points in the data stream $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \ldots$ for which $r_{nk} = 1$, i.e. points which are closest to cluster k.

The cluster prototype $\boldsymbol{\mu}_k$ which is closest to a point $\boldsymbol{x}_n$ is a function of the previously seen data $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{n-1}$, but upon seeing a data point $\boldsymbol{x}_n$ we can always compute the closest cluster. In other words, we assume that we know which data points belong to k, but remark that we do not really know this "ahead of time." We wish to find the root of

$$\sum_{\boldsymbol{x}_n \mid k} (\boldsymbol{x}_n - \boldsymbol{\mu}_k).$$

where $\boldsymbol{x}_n \mid k$ denotes the data points that are assigned to cluster k. Let N_k be the total number of data points falling closest to $\boldsymbol{\mu}_k$. As N_k goes to infinity, we have

$$-\lim_{N_k \to \infty} \frac{1}{N_k} \sum_{\boldsymbol{x}_k \mid k} (\boldsymbol{x}_n - \boldsymbol{\mu}_k) = -\int (\boldsymbol{x} - \boldsymbol{\mu}_k)\, p(\boldsymbol{x} \mid k)\, d\boldsymbol{x} = \mathbb{E}\left[-(\boldsymbol{x} - \boldsymbol{\mu}_k) \mid k\right],$$

so clearly we're trying to minimize a regression function. We now identify $\boldsymbol{\mu}_k$ as θ and $-(\boldsymbol{x} - \boldsymbol{\mu}_k)$ as z in Equation (26), which leads us to the desired sequential update rule

$$\boldsymbol{\mu}_k^{\text{new}} = \boldsymbol{\mu}_k^{\text{old}} + \eta_n (\boldsymbol{x} - \boldsymbol{\mu}_k).$$

Exercise 9.4

Recall that in the original EM algorithm in Section 9.3 we wish to maximize $p(\boldsymbol{X} \mid \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. In other words, we want to compute

$$\max_{\boldsymbol{\theta}} \ln p(\boldsymbol{X} \mid \boldsymbol{\theta}) = \max_{\boldsymbol{\theta}} \ln \left(\sum_{\boldsymbol{Z}} p(\boldsymbol{X}, \boldsymbol{Z} \mid \boldsymbol{\theta}) \right)$$

but we settle for maximizing $\mathbb{E}_{\boldsymbol{Z}} [\ln p(\boldsymbol{X}, \boldsymbol{Z} \mid \boldsymbol{\theta})]$ instead.

In this problem, we wish to maximize $p(\boldsymbol{\theta} \mid \boldsymbol{X}) \propto p(\boldsymbol{X} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. In other words, we want to compute

$$\max_{\boldsymbol{\theta}} \ln \left(p(\boldsymbol{X} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})\right) = \max_{\boldsymbol{\theta}} \ln \left(\sum_{\boldsymbol{Z}} p(\boldsymbol{X}, \boldsymbol{Z} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) \right)$$

but we settle for maximizing $\mathbb{E}_{\boldsymbol{Z}}\left[\ln p(\boldsymbol{X}, \boldsymbol{Z} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})\right]$ instead. Expanding the logarithm and writing out the expected value explicitly, the expression becomes

$$\sum_{\boldsymbol{Z}} p(\boldsymbol{Z} \mid \boldsymbol{X}, \boldsymbol{\theta})\left(\ln p(\boldsymbol{X}, \boldsymbol{Z} \mid \boldsymbol{\theta}) + \ln p(\boldsymbol{\theta})\right) = Q + \ln p(\boldsymbol{\theta}).$$

The **E step** remains the same, since the expected value is to be taken over the same posterior distribution $p(\boldsymbol{Z} \mid \boldsymbol{X}, \boldsymbol{\theta})$ as in the case when we're maximizing the likelihood $p(\boldsymbol{X} \mid \boldsymbol{\theta})$. The **M step** is altered from maximizing Q to maximizing $Q + \ln p(\boldsymbol{\theta})$, since we wish to maximize $p(\boldsymbol{X} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})$ and not $p(\boldsymbol{X} \mid \boldsymbol{\theta})$.

Exercise 9.10

Applying the product rule to the conditional probability yields

$$p(\boldsymbol{x}_b \mid \boldsymbol{x}_a) = \frac{p(\boldsymbol{x}_b, \boldsymbol{x}_a)}{p(\boldsymbol{x}_a)} = \frac{p(\boldsymbol{x})}{p(\boldsymbol{x}_a)}.$$

We already have an expression for the numerator, given in the problem statement. The denominator found by substituting the definition and applying the sum rule.

$$p(\boldsymbol{x}_a) = \sum_{\boldsymbol{x}_b} p(\boldsymbol{x}_a, \boldsymbol{x}_b) = \sum_{\boldsymbol{x}_b} \sum_{j=1}^{K} \pi_j p(\boldsymbol{x}_a, \boldsymbol{x}_b \mid j) = \sum_{j=1}^{K} \pi_j p(\boldsymbol{x}_a \mid j).$$

Substituting this into the above fraction and diving every term in the sum, we obtain

$$p(\boldsymbol{x}_b \mid \boldsymbol{x}_a) = \frac{\sum_{k=1}^{K} \pi_k p(\boldsymbol{x}_a, \boldsymbol{x}_b \mid k)}{\sum_{j=1}^{K} \pi_j p(\boldsymbol{x}_a \mid j)} = \sum_{k=1}^{K} \left(\frac{\pi_k}{\sum_{j=1}^{K} \pi_j p(\boldsymbol{x}_a \mid j)} \right) p(\boldsymbol{x} \mid k).$$

From the expression above we observe that the mixing coefficients in the conditional distribution become $\pi_k / \left(\sum_{j=1}^{K} \pi_j p(\boldsymbol{x}_a \mid j) \right)$, while the component densities remain unchanged.

Notice that if $\boldsymbol{x}_a = \boldsymbol{x}$ and $\boldsymbol{x}_b = \emptyset$, the above reduces to the original expression for $p(\boldsymbol{x})$.

Exercise 9.12

In this problem we use the notation $p(\boldsymbol{x} \mid \boldsymbol{\mu}_k)$ instead of $p(\boldsymbol{x} \mid k)$. Furthermore, we will assume that by "denote the mean and covariance of $p(\boldsymbol{x} \mid k)$ as $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$" the problem text is interpreted as stating that $\mathbb{E}\left[\boldsymbol{x} \mid \boldsymbol{\mu}_k\right] = \boldsymbol{\mu}_k$ and $\operatorname{cov}\left[\boldsymbol{x} \mid \boldsymbol{\mu}_k\right] = \boldsymbol{\Sigma}_k$. In other words, the expected value and variance is taken over the variable $\boldsymbol{x}$.

The mean. The expected value of the mixture distribution is readily found to be

$$\begin{aligned} \mathbb{E}[\boldsymbol{x}] &= \sum_{\boldsymbol{x}} \boldsymbol{x} p(\boldsymbol{x}) = \sum_{\boldsymbol{x}} \boldsymbol{x} \sum_{k=1}^{K} \pi_k p(\boldsymbol{x} \mid \boldsymbol{\mu}_k) = \sum_{k=1}^{K} \pi_k \sum_{\boldsymbol{x}} \boldsymbol{x} p(\boldsymbol{x} \mid \boldsymbol{\mu}_k) \\ &= \sum_{k=1}^{K} \pi_k \, \mathbb{E}\left[\boldsymbol{x} \mid \boldsymbol{\mu}_k\right] = \sum_{k=1}^{K} \pi_k \boldsymbol{\mu}_k. \end{aligned}$$

The covariance. To evaluate the covariance of the mixture distribution, recall Equation (1.42) in [Bishop, 2006], which states the covariance identity

$$\operatorname{cov}[\boldsymbol{x}] = \mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^T\right] - \mathbb{E}[\boldsymbol{x}]\,\mathbb{E}[\boldsymbol{x}]^T .$$

We explore the last term $\mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^T\right]$, since we've already found that $\mathbb{E}[\boldsymbol{x}] = \sum_{k=1}^K \pi_k \boldsymbol{\mu}_k$ in the previous sub-problem.

In turn we use the definition of the expected value, then the definition of $p(\boldsymbol{x})$, the definition of the conditional expected value, the covariance identity, and finally the variables given in the problem statement:

$$\begin{aligned}
\mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^T\right] &= \sum_{\boldsymbol{x}} p(\boldsymbol{x})\boldsymbol{x}\boldsymbol{x}^T = \sum_{\boldsymbol{x}} \left(\sum_{k=1}^K \pi_k p(\boldsymbol{x} \mid \boldsymbol{\mu}_k) \right) \boldsymbol{x}\boldsymbol{x}^T \\
&= \sum_{k=1}^K \pi_k \sum_{\boldsymbol{x}} p(\boldsymbol{x} \mid \boldsymbol{\mu}_k)\boldsymbol{x}\boldsymbol{x}^T = \sum_{k=1}^K \pi_k \, \mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^T \mid \boldsymbol{\mu}_k\right] \\
&= \sum_{k=1}^K \pi_k \left(\operatorname{cov}[\boldsymbol{x} \mid \boldsymbol{\mu}_k] + \mathbb{E}[\boldsymbol{x} \mid \boldsymbol{\mu}_k]\,\mathbb{E}[\boldsymbol{x} \mid \boldsymbol{\mu}_k]^T \right) && \text{(covariance identity)} \\
&= \sum_{k=1}^K \pi_k \left(\boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T \right)
\end{aligned}$$

Combining the above expression for $\mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^T\right]$ with the covariance identity yields Equation (9.50) from the book, and this is what the problem asked us to show.

Exercise 9.14

In general, from the sum and product rule, we know that

$$p(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_{\boldsymbol{z}} \underbrace{p(\boldsymbol{x} \mid \boldsymbol{z}, \boldsymbol{\mu}) p(\boldsymbol{z} \mid \boldsymbol{\pi})}_{p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\mu}, \boldsymbol{\pi})} .$$

By substituting the expressions for $p(\boldsymbol{x} \mid \boldsymbol{z}, \boldsymbol{\mu})$ and $p(\boldsymbol{z} \mid \boldsymbol{\pi})$, we see that

$$p(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_{\boldsymbol{z}} \underbrace{\left(\prod_{k=1}^K p(\boldsymbol{x} \mid \boldsymbol{\mu}_k)^{z_k} \right)}_{p(\boldsymbol{x} \mid \boldsymbol{z}, \boldsymbol{\mu})} \underbrace{\left(\prod_{k=1}^K \pi_k^{z_k} \right)}_{p(\boldsymbol{z} \mid \boldsymbol{\pi})} = \sum_{\boldsymbol{z}} \prod_{k=1}^K \left[p(\boldsymbol{x} \mid \boldsymbol{\mu}_k) \pi_k \right]^{z_k} .$$

The sum is taken over states of $\boldsymbol{z}$, which are unit vectors, e.g. $(1, 0, 0, \ldots)$, $(0, 1, 0, \ldots)$ and $(0, 0, 1, \ldots)$ and so forth. There are K such states, and therefore K terms in the sum. In the first term of the sum, i.e. the first state of $\boldsymbol{z}$, the only term in the product which is not 1 is $p(\boldsymbol{x} \mid \boldsymbol{\mu}_1)\pi_1$. Likewise, the second term becomes $p(\boldsymbol{x} \mid \boldsymbol{\mu}_2)\pi_2$ and the third $p(\boldsymbol{x} \mid \boldsymbol{\mu}_3)\pi_3$ and so forth. Therefore only one factor is each product remain, yielding

$$p(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_{\boldsymbol{z}} \prod_{k=1}^K \left[p(\boldsymbol{x} \mid \boldsymbol{\mu}_k) \pi_k \right]^{z_k} = \sum_{k=1}^K p(\boldsymbol{x} \mid \boldsymbol{\mu}_k) \pi_k .$$

The equation above is the result which the problem asked us to show.

Exercise 9.18

We keep the results of Exercise 9.4 in mind: we know the **E step** will be unchanged in the sense that we take expected value over the same distribution $p(\boldsymbol{Z} \mid \boldsymbol{X}, \boldsymbol{\theta})$. The problem is finding the expected value and maximizing it, i.e. finding explicit equations for ***pi*** and the $\boldsymbol{\mu}_k$ which maximize the expected complete-data log likelihood.

The expected value. The expectation becomes

$$\mathbb{E}_{\boldsymbol{Z}}\left[\ln p(\boldsymbol{X}, \boldsymbol{Z} \mid \boldsymbol{\mu}, \boldsymbol{\pi}) + \ln p(\boldsymbol{\mu}, \boldsymbol{\pi})\right] = \mathbb{E}_{\boldsymbol{Z}}\left[\ln p(\boldsymbol{X}, \boldsymbol{Z} \mid \boldsymbol{\mu}, \boldsymbol{\pi})\right] + \ln p(\boldsymbol{\mu}, \boldsymbol{\pi}).$$

Let's investigate the last term, which is due to the priors over the model parameters. While $\boldsymbol{\pi}$ is a K-dimensional vector, $\boldsymbol{\pi}$ represents a set of K vectors $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K\}$.

The vector $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$ is subject to the constraint $\sum_k \pi_k = 1$. The probability of $\boldsymbol{\pi}$ is given by Equation (2.38) for the Dirichlet distribution from the book, i.e.

$$p(\boldsymbol{\pi} \mid \boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_K)} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1} \propto \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}. \tag{27}$$

In the set of vectors $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K\}$, the individual vectors $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$ are independent, so we can factor the joint distribution as $p(\boldsymbol{\mu}) = p(\boldsymbol{\mu}_1) \cdots p(\boldsymbol{\mu}_K)$. We assume that the entries of the kth vector $\boldsymbol{\mu}_k$ are independent given a_k and b_k, so that $p(\mu_{ki} \mid a_k, b_k)$ is given by Equation (2.13), i.e.

$$p(\mu_{ki} \mid a_k, b_k) = \frac{\Gamma(a_k + b_k)}{\Gamma(a_k)\Gamma(b_k)} \mu_{ki}^{a_k - 1}(1 - \mu_{ki})^{b_k - 1} \propto \mu_{ki}^{a_k - 1}(1 - \mu_{ki})^{b_k - 1}$$

and therefore

$$p(\boldsymbol{\mu}_k \mid a_k, b_k) = \prod_{i=1}^{D} p(\mu_{ki} \mid a_k, b_k) \propto \prod_{i=1}^{D} \mu_{ki}^{a_k - 1}(1 - \mu_{ki})^{b_k - 1}.$$

Combining the equation above with the factorization $p(\boldsymbol{\mu}) = p(\boldsymbol{\mu}_1) \cdots p(\boldsymbol{\mu}_K)$, we observe that the probability of the set $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K\}$ becomes

$$p(\boldsymbol{\mu}) = p(\boldsymbol{\mu}_1) \cdots p(\boldsymbol{\mu}_K) \propto \prod_{k=1}^{K} \prod_{i=1}^{D} \mu_{ki}^{a_k - 1}(1 - \mu_{ki})^{b_k - 1}. \tag{28}$$

Note that we have ignored multiplicative constants independent of $\boldsymbol{\pi}$ and the $\boldsymbol{\mu}_k$s, since when we take logarithms these constants will become additive. Additive constants disappear upon differentiating, and can safely be ignored as they will play no role in the following maximization step.

Finally we take the logarithm of $p(\boldsymbol{\mu}, \boldsymbol{\pi}) = p(\boldsymbol{\mu})p(\boldsymbol{\pi})$. Using Equations (27) and (28) for the priors over $\boldsymbol{\pi}$ and $\boldsymbol{\mu}$, we then obtain:

$$\begin{aligned}
\ln p(\boldsymbol{\mu}, \boldsymbol{\pi}) &= \ln\left(p(\boldsymbol{\mu})p(\boldsymbol{\pi})\right) = \ln p(\boldsymbol{\mu}) + \ln p(\boldsymbol{\pi}) \\
\ln p(\boldsymbol{\pi} \mid \boldsymbol{\alpha}) &\propto \sum_{k=1}^{K} (\alpha_k - 1) \ln \pi_k \\
\ln p(\boldsymbol{\mu}) &\propto \sum_{k=1}^{K} \sum_{i=1}^{D} (a_k - 1) \ln \mu_{ki} + (b_k - 1) \ln(1 - \mu_{ki})
\end{aligned}$$

The maximization. We ignore constants and explicitly give the equation for the complete-data log likelihood, which becomes

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{Z}}\left[\ln p(\boldsymbol{X}, \boldsymbol{Z} \mid \boldsymbol{\mu}, \boldsymbol{\pi})\right] + \ln p(\boldsymbol{\mu}, \boldsymbol{\pi}) \\
&= \sum_{n=1}^{N} \sum_{k=1}^{k} \gamma(z_{nk}) \left[\ln \pi_k + \sum_{i=1}^{D} x_{ni} \ln \mu_{ki} + (1 - x_{ni}) \ln(1 - \mu_{ki})\right] \\
&+ \sum_{k=1}^{K} (\alpha_k - 1) \ln \pi_k + \sum_{k=1}^{K} \sum_{i=1}^{D} (a_k - 1) \ln \mu_{ki} + (b_k - 1) \ln(1 - \mu_{ki}).
\end{aligned} \tag{29}$$

Notice that the first term with the double sum is identical to Equation (9.55) in the book.

Maximizing w.r.t. $\boldsymbol{\pi}$. We introduce a Lagrange multiplier for the constraint $\sum_k \pi_k = 1$, and then differentiate (29) with respect to π_k to obtain

$$\sum_n \gamma(z_{nk}) \frac{1}{\pi_k} + (\alpha_k - 1) \frac{1}{\pi_k} + \lambda = 0.$$

Multiplying by π_k, summing both sides over $k = 1, \ldots, K$ and using $\sum_k \pi_k = 1$ lets us solve for the Lagrange multiplier λ. Substituting this back into the equation and using Equation (9.57) from the book and the fact that $\sum_k N_k = N$ then yields the solution

$$\pi_k = \frac{N_k + \alpha_k - 1}{N + \sum_j (\alpha_j - 1)} = \frac{N_k + \alpha_k - 1}{N + \alpha_0 - K}.$$

Notice the similarity with Equation (9.60), which represents the same equation with no prior $p(\boldsymbol{\pi} \mid \boldsymbol{\alpha})$. In fact, if we set $(\alpha_1, \ldots, \alpha_K) = (1, \ldots, 1)$, then $\alpha_0 = \sum_k \alpha_k = K$ and the expression above reduces exactly to Equation (9.60).

Maximizing w.r.t. $\boldsymbol{\mu}_k$. We differentiate Equation (29) above with respect to μ_{ki}:

$$\sum_n \gamma(z_{nk}) \left(\frac{x_{ni}}{\mu_{ki}} - \frac{1 - x_{ni}}{1 - \mu_{ki}}\right) + \frac{a_k - 1}{\mu_{ki}} - \frac{b_k - 1}{1 - \mu_{ki}} = 0$$

The common denominator is $\mu_{ki}(1-\mu_{ki})$. Multiplying the fractions to obtain a common denominator and then multiplying both sides of the equation by it yields

$$\sum_n \gamma(z_{nk})\left(x_{ni}-\mu_{ki}\right)+(a_k-1)+\mu_{ki}(2-a_k-b_k)=0.$$

We solve for μ_{ki} and make use of Equations (9.57) and (9.58), to get

$$\mu_{ki}=\frac{\sum_n \gamma(z_{nk})x_{ni}+(a_k-1)}{N_k+(a_k-1)+(b_k-1)} \quad \text{or} \quad \boldsymbol{\mu}_k=\frac{N_k\bar{\boldsymbol{x}}_k+(a_k-1)}{N_k+(a_k-1)+(b_k-1)}.$$

Notice that if $a_k=b_k=1$, then this reduces to the equation with no prior.

2.10 Approximate Inference

Exercise 10.4

The (reversed) Kullback-Leibler divergence is defined as

$$KL(p||q)=-\int p(\boldsymbol{x})\ln\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\,d\boldsymbol{x}.$$

In this problem $q(\boldsymbol{x})=\mathcal{N}(\boldsymbol{x}\mid\boldsymbol{\mu},\boldsymbol{\Sigma})$, and therefore

$$\begin{aligned}\partial_{\boldsymbol{\mu}}q(\boldsymbol{x})&=\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}\right)q(\boldsymbol{x})\\ \partial_{\boldsymbol{\Sigma}}\ln q(\boldsymbol{x})&=-\frac{1}{2}\boldsymbol{\Sigma}^{-T}+\frac{1}{2}\boldsymbol{\Sigma}^{T}\left(\boldsymbol{x}-\boldsymbol{\mu}\right)\left(\boldsymbol{x}-\boldsymbol{\mu}\right)^T\boldsymbol{\Sigma}^{-T}.\end{aligned}$$

The first equation is essentially differentiation of a quadratic form with respect to the vector, which we assume is known and straightforward. The second equation is more involved, and the result presented here is taken from *the matrix cookbook*[1].

Differentiating the KL divergence with respect to $\boldsymbol{\mu}$ and setting it equal to zero yields

$$\int p(\boldsymbol{x})\boldsymbol{\Sigma}^{-1}\boldsymbol{x}\,d\boldsymbol{x}=\int p(\boldsymbol{x})\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\,d\boldsymbol{x}.$$

Taking constants out of the integrals and left-multiplying with $\boldsymbol{\Sigma}$ (which we assume to be invertible) yields $\boldsymbol{\mu}=\int p(\boldsymbol{x})\boldsymbol{x}\,d\boldsymbol{x}$, which solves the first part of the problem.

To solve the second part, we differentiate with respect to $\boldsymbol{\Sigma}$ and equate the result to zero:

$$\int p(\boldsymbol{x})\boldsymbol{\Sigma}^{-T}+p(\boldsymbol{x})\boldsymbol{\Sigma}^{T}\left(\boldsymbol{x}-\boldsymbol{\mu}\right)\left(\boldsymbol{x}-\boldsymbol{\mu}\right)^T\boldsymbol{\Sigma}^{-T}\,d\boldsymbol{x}=\boldsymbol{0}.$$

Right-multiplying by $\boldsymbol{\Sigma}^{-T}$ and using $\boldsymbol{\Sigma}^T=\boldsymbol{\Sigma}$, we then solve for $\boldsymbol{\Sigma}^{-1}$ and obtain

$$\boldsymbol{\Sigma}^{-1}=\int p(\boldsymbol{x})\left(\boldsymbol{x}-\boldsymbol{\mu}\right)\left(\boldsymbol{x}-\boldsymbol{\mu}\right)^T\,d\boldsymbol{x}=\mathbb{E}\left[\left(\boldsymbol{x}-\boldsymbol{\mu}\right)\left(\boldsymbol{x}-\boldsymbol{\mu}\right)^T\right]=\operatorname{cov}[\boldsymbol{x}].$$

[1] `https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf`

Exercise 10.8

The mean of the variational posterior distribution for τ is given by

$$\mathbb{E}[\tau] = \frac{a_N}{b_N}$$

We will examine both of these terms as $N \to \infty$. Clearly $a_N \simeq N/2$ as N goes to infinity.

The expression for b_N requires more finesse. By the linearity of the expected value

$$b_N \simeq \frac{1}{2}\left[\sum_{n=1}^{N}\left(x_n^2 - 2x_n\,\mathbb{E}[\mu] + \mathbb{E}[\mu^2]\right) + \lambda_0\left(\mathbb{E}[\mu^2] - 2\mu_0\,\mathbb{E}[\mu] + \mu_0^2\right)\right]. \tag{30}$$

As $N \to \infty$, we see that $\mathbb{E}[\mu] = \bar{x}$, and that (from Equation (1.50) in the book)

$$\mathbb{E}[\mu^2] = \mu^2 + \lambda_N^{-1} \simeq \bar{x}^2 + \frac{1}{(\lambda_0 + N)\,\mathbb{E}[\tau]} \simeq \bar{x}^2.$$

Substituting the expressions for $\mathbb{E}[\mu]$ and $\mathbb{E}[\mu^2]$ back into Equation (30) yields

$$b_N \simeq \frac{1}{2}\left[\sum_{n=1}^{N}(x_n - \bar{x})^2 + \lambda_0(\bar{x} - \mu_0)^2\right] \simeq \frac{1}{2}\sum_{n=1}^{N}(x_n - \bar{x})^2,$$

where the final asymptotic equality holds because $(\bar{x} - \mu_0)^2$ remains constant as $N \to \infty$. We are now in a position to see that the expected value of the variational posterior distribution over the precision, i.e. $1/\,\mathbb{E}[\tau]$, asymptotically equals the sample variance. In other words, we have

$$\frac{1}{\mathbb{E}[\tau]} = \frac{b_N}{a_N} \simeq \frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})^2.$$

Using the above, we readily observe that $\operatorname{var}[\tau]$ goes to zero. We have

$$\operatorname{var}[\tau] = \frac{a_N}{b_N^2} = \frac{\mathbb{E}[\tau]}{b_N} \simeq 0$$

since $\mathbb{E}[\tau]$ converges but b_N grows as $N \to \infty$.

Exercise 10.11

We will make two assumptions in this problem: (1) a Lagrange multiplier can be used to optimize a *functional* $F[y]$ in the same way that it is used to optimize a *function* $y(x)$, and (2) the variation $\delta F[y]/\delta y$ can be taken irrespective of whether the function $y(x)$ is defined over real numbers of integers. We assume these to be true, but a rigorous approach would be to investigate these assumptions thoroughly—which we will not.

The functional we wish to optimize becomes

$$F[q(m)] = \sum_m \sum_{\boldsymbol{Z}} q\left(\boldsymbol{Z} \mid m\right) q(m) \ln\left[\frac{p(\boldsymbol{Z}, \boldsymbol{X}, m)}{q\left(\boldsymbol{Z} \mid m\right) q(m)}\right] + \lambda\left(\sum_m q(m) - 1\right).$$

Taking the variation with respect to $q(m)$ and requiring that the functional derivative vanishes gives the equation

$$\sum_{\boldsymbol{Z}} q\left(\boldsymbol{Z} \mid m\right) \ln\left[\frac{p(\boldsymbol{Z},\boldsymbol{X} \mid m)}{q\left(\boldsymbol{Z} \mid m\right)}\right] + \sum_{\boldsymbol{Z}} q\left(\boldsymbol{Z} \mid m\right) \ln\left[\frac{p(m)}{q(m)}\right] - \sum_{\boldsymbol{Z}} q\left(\boldsymbol{Z} \mid m\right) + \lambda = 0$$

where we have used $p(\boldsymbol{Z},\boldsymbol{X},m) = p(\boldsymbol{Z},\boldsymbol{X} \mid m)p(m)$. The first term is $\mathcal{L}_m$, in the second only the logarithm remains when we perform the sum (since the probability mass function sums to unity). The equation above therefore becomes

$$\mathcal{L}_m + \ln\frac{p(m)}{q(m)} - 1 + \lambda = 0,$$

which means that $q(m) = p(m)e^{\lambda-1+\mathcal{L}_m} \propto p(m)e^{\mathcal{L}_m}$, which we wanted to show.

Exercise 10.12

We detail some of the steps going from Equation (10.41) to (10.49). Starting with

$$\ln q^\star(\boldsymbol{Z}) = \mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda}}[\ln p(\boldsymbol{X},\boldsymbol{Z},\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda})] + \text{const}$$

we obtain Equation (10.44) by noticing that only $p(\boldsymbol{Z} \mid \boldsymbol{\pi})$ and $p(\boldsymbol{X} \mid \boldsymbol{Z},\boldsymbol{\mu},\boldsymbol{\Lambda})$ in (10.41) are functions of $\boldsymbol{Z}$. The expected value of the first term becomes

$$\mathbb{E}_{\boldsymbol{\pi}}[\ln p(\boldsymbol{Z} \mid \boldsymbol{\pi})] = \mathbb{E}_{\boldsymbol{\pi}}\left[\ln\prod_{n=1}^{N}\prod_{k=1}^{K}\boldsymbol{\pi}_k^{z_{nk}}\right] = \mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\ln\boldsymbol{\pi}_k\right] = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\,\mathbb{E}_{\boldsymbol{\pi}}\left[\ln\boldsymbol{\pi}_k\right]$$

and since $\ln p(\boldsymbol{x}_n, \boldsymbol{z}_n \mid, \boldsymbol{\mu}, \boldsymbol{\Lambda}) \propto \ln|\boldsymbol{\Lambda}_k| - D\ln(2\pi) - \mathbb{E}\left[(\boldsymbol{x}_n - \boldsymbol{\mu})^T\boldsymbol{\Lambda}_k(\boldsymbol{x}_n - \boldsymbol{\mu})\right]$ the expected value of the second term becomes

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\mu},\boldsymbol{\Lambda}}[\ln p(\boldsymbol{X} \mid \boldsymbol{Z},\boldsymbol{\mu},\boldsymbol{\Lambda})] &= \mathbb{E}_{\boldsymbol{\mu},\boldsymbol{\Lambda}}\left[\ln\prod_{n=1}^{N}\prod_{k=1}^{K}\mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1})^{z_{nk}}\right] \\
&= \mathbb{E}_{\boldsymbol{\mu},\boldsymbol{\Lambda}}\left[\sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\ln\mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1})\right] \\
&= \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\frac{1}{2}\left(\mathbb{E}_{\boldsymbol{\Lambda}}[\ln|\boldsymbol{\Lambda}_k|] - D\ln(2\pi) - \mathbb{E}_{\boldsymbol{\mu},\boldsymbol{\Lambda}}\left[(\boldsymbol{x}_n - \boldsymbol{\mu})^T\boldsymbol{\Lambda}_k(\boldsymbol{x}_n - \boldsymbol{\mu})\right]\right).
\end{aligned}$$

Combining the expressions for the two terms $\mathbb{E}_{\boldsymbol{\pi}}[\ln p(Z \mid \boldsymbol{\pi})]$ and $\mathbb{E}_{\boldsymbol{\mu},\boldsymbol{\Lambda}}[\ln p(\boldsymbol{X} \mid \boldsymbol{Z},\boldsymbol{\mu},\boldsymbol{\Lambda})]$ yields Equation (10.46) in the book.

In going from Equation (10.47) to (10.48), we wish to choose a normalization constant C so that

$$C\sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n) = C\sum_{z_{nk}\in\boldsymbol{z}_n}\left(\prod_{k=1}^{K}\rho_{nk}^{z_{nk}}\right) = C\sum_{k=1}^{K}\rho_{nk} = 1.$$

This ensures normalization. Multiplying with C takes us from (10.47) to (10.48).

We have explained the major steps going from Equation (10.41) to (10.49).

Exercise 10.19

We immediately integrate out $q(\boldsymbol{\pi}) = \mathrm{Dir}(\boldsymbol{q} \mid \boldsymbol{\alpha})$, leaving us with

$$\iint \pi_k \mathcal{N}\left(\widehat{\boldsymbol{x}} \mid \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}\right) \underbrace{\mathcal{N}\left(\boldsymbol{\mu}_k \mid \boldsymbol{m}_k, (\beta_k \boldsymbol{\Lambda}_k)^{-1}\right) \mathcal{W}\left(\boldsymbol{\Lambda}_k \mid \boldsymbol{W}_k, \nu_k\right)}_{q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)} \, d\boldsymbol{\mu}_k \, d\boldsymbol{\Lambda}_k$$

where the explicit form of $q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)$ is given by Equation (10.59) in the book.

We then structure the integral as

$$\pi_k \int \mathcal{W}\left(\boldsymbol{\Lambda}_k \mid \boldsymbol{W}_k, \nu_k\right) \left[\int \mathcal{N}\left(\widehat{\boldsymbol{x}} \mid \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}\right) \mathcal{N}\left(\boldsymbol{\mu}_k \mid \boldsymbol{m}_k, (\beta_k \boldsymbol{\Lambda}_k)^{-1}\right) \, d\boldsymbol{\mu}_k \right] d\boldsymbol{\Lambda}_k$$

and since $p(\widehat{\boldsymbol{x}}) = \int p(\widehat{\boldsymbol{x}} \mid \boldsymbol{\mu}_k) p(\boldsymbol{\mu}_k) \, d\boldsymbol{\mu}_k$ we appeal to Equations (2.113), (2.114) and (2.115) for the marginal distribution $p(\widehat{\boldsymbol{x}})$. After some work we then obtain

$$\pi_k \int \mathcal{W}\left(\boldsymbol{\Lambda}_k \mid \boldsymbol{W}_k, \nu_k\right) \mathcal{N}\left(\widehat{\boldsymbol{x}} \mid \boldsymbol{m}_k, \boldsymbol{\Lambda}_k^{-1} + \beta^{-1} \boldsymbol{\Lambda}_k^{-1}\right) \, d\boldsymbol{\Lambda}_k \tag{31}$$

The one dimensional case Let's consider first a one dimensional case of (31). We ignore the constant π_k and drop the subscript on k to ease notation. The Wishart distribution reduces to the gamma distribution, and in a single dimension we have

$$\mathcal{W}\left(\lambda \mid w, \nu\right) = \mathrm{Gam}(\lambda \mid a = \nu/2, b = w^{-1}/2)$$

so that Equation (31) becomes

$$\int_0^\infty \mathrm{Gam}(\lambda \mid a = \nu/2, b = w^{-1}/2) \mathcal{N}\left(x \mid m, \lambda^{-1}(1 + \beta^{-1})\right) \, d\lambda.$$

Using Equations (2.42) and (2.146) for the Normal distribution and the Gamma distribution respectively, the integral explicitly becomes

$$\frac{1}{\Gamma(\nu/2)} \left(\frac{1}{2} w^{-1}\right)^{\nu/2} \frac{1}{(2\pi(1+\beta^{-1}))^{1/2}} \int_0^\infty \lambda^{(\nu-1)/2} \exp\left(-\frac{1}{2} w^{-1} \lambda - \frac{1}{2} \frac{\lambda (x-m)^2}{1+\beta^{-1}}\right) d\lambda. \tag{32}$$

Following Equation (2.158) in the book, we change variables to

$$z = \lambda \underbrace{\left[\frac{1}{2} w^{-1} + \frac{1}{2} \frac{(x-m)^2}{1+\beta^{-1}}\right]}_{\alpha}$$

and use the gamma function $\Gamma(\cdot)$, defined by Equation (1.141) to obtain

$$\frac{1}{\Gamma(\nu/2)} \left(\frac{1}{2} w^{-1}\right)^{\nu/2} (2\pi(1+\beta^{-1}))^{-1/2} \alpha^{-\nu/2-1/2} \underbrace{\int_0^\infty z^{\frac{\nu-1}{2}} \exp(-z) \, dz}_{\Gamma(\nu/2+1/2)}.$$

Rearranging the factors and multiplying with 1 (last factor), we obtain

$$\frac{\Gamma(\nu/2+1/2)}{\Gamma(\nu/2)}\left(\frac{1}{2w}\right)^{\nu/2}\left(\frac{1}{2\pi}\right)^{1/2}\left(\frac{\beta+1}{\beta}\right)^{-1/2}\alpha^{-\nu/2-1/2}\left(\frac{\beta+1}{\beta}\right)^{-\nu/2-1/2+\nu/2+1/2}$$

we now pull the last factor into the others, obtaining

$$\frac{\Gamma(\nu/2+1/2)}{\Gamma(\nu/2)}\left(\frac{\beta+1}{\beta 2w}\right)^{\nu/2}\left(\frac{1}{2\pi}\right)^{1/2}\left[\frac{\beta+1}{\beta 2w}+\frac{(x-m)^2}{2}\right]^{-\nu/2-1/2}.$$

We compare this with the last term in (2.158) and identifying the term as

$$\lambda = a/b = \frac{\nu\beta w}{1+\beta} \qquad \nu = 2a = \nu$$

This is indeed equal to

$$\mathrm{St}(x \mid \mu, \frac{\nu\beta w}{1+\beta}, \nu),$$

which is a one-dimensional variant of (10.81) and (10.82).

The D-dimensional case We now move to the general case of (31), and again we ignore the constant π_k and drop the subscript on k to ease notation.

Using Equations (2.155) and (2.43) for the definition of the Wishart distribution and Gaussian, the integral explicitly becomes

$$\frac{B}{(2\pi)^{D/2}}\int |\boldsymbol{\Lambda}|^{(\nu-D-1)/2}\exp\left(-\frac{1}{2}\operatorname{Tr}(\boldsymbol{W}^{-1}\boldsymbol{\Lambda})\right)\frac{1}{|\boldsymbol{\Lambda}^{-1}|^{1/2}(1+\beta^{-1})^{1/2}}$$
$$\exp\left(-\frac{1}{2(1+\beta^{-1})}(\widehat{\boldsymbol{x}}-\boldsymbol{m})^T\boldsymbol{\Lambda}(\widehat{\boldsymbol{x}}-\boldsymbol{m})\right)d\boldsymbol{\Lambda}.$$

We decompose $\boldsymbol{\Lambda}$ as into it's eigenvalues and eigenvectors

$$\boldsymbol{\Lambda} = \boldsymbol{UEU}^T = \sum_i^D \lambda_i \boldsymbol{u}_i\boldsymbol{u}_i^T$$

and since $\boldsymbol{\Lambda}$ is positive definite every eigenvalue is non-negative.

$$\begin{aligned}
d\boldsymbol{\Lambda} &= d\boldsymbol{\Lambda}(?)\\
|\boldsymbol{\Lambda}^{-1}|^{1/2} &= \prod_i^D \lambda_i^{-1/2}\\
|\boldsymbol{\Lambda}| &= \prod_i^D \lambda_i\\
\operatorname{tr}\left(\boldsymbol{W}^{-1}\boldsymbol{\Lambda}\right) &= \sum_i^D \lambda_i \boldsymbol{u}_i^T\boldsymbol{W}^{-1}\boldsymbol{u}_i\\
(\widehat{\boldsymbol{x}}-\boldsymbol{m})^T\boldsymbol{\Lambda}(\widehat{\boldsymbol{x}}-\boldsymbol{m}) &= \sum_i^D \lambda_i \underbrace{(\widehat{\boldsymbol{x}}-\boldsymbol{m})^T\boldsymbol{u}_i}_{y_i}(\widehat{\boldsymbol{x}}-\boldsymbol{m})\boldsymbol{u}_i^T = \sum_i^D \lambda_i y_i^2
\end{aligned}$$

Wishart becomes

$$|\boldsymbol{\Lambda}|^{(\nu-D-1)/2} \exp\left(-\frac{1}{2}\operatorname{Tr}(\boldsymbol{W}^{-1}\boldsymbol{\Lambda})\right) = \prod_i^D \lambda_i^{(\nu-D-1)/2} \exp\left(-\frac{1}{2}\lambda_i \boldsymbol{u}_i^T \boldsymbol{W}^{-1}\boldsymbol{u}_i\right)$$

Gaussian becomes

$$\exp\left(-\frac{1}{2(1+\beta^{-1})}(\widehat{\boldsymbol{x}}-\boldsymbol{m})^T\boldsymbol{\Lambda}(\widehat{\boldsymbol{x}}-\boldsymbol{m})\right) = \prod_i^D \exp\left(\frac{-\lambda_i y_i^2}{2(1+\beta^{-1})}\right)$$

The integral becomes

$$\frac{B}{(2\pi)^{D/2}}\frac{1}{(1+\beta^{-1})^{1/2}}\prod_i^D \int \lambda_i^{(\nu-D)/2} \exp\left(-\frac{1}{2}\lambda_i \boldsymbol{u}_i^T \boldsymbol{W}^{-1}\boldsymbol{u}_i - \frac{\lambda_i y_i^2}{2(1+\beta^{-1})}\right) d\lambda_i$$

Solving yields

$$\frac{B}{(2\pi)^{D/2}}\frac{1}{(1+\beta^{-1})^{1/2}}\prod_i^D \left(\frac{1}{2}\boldsymbol{u}_i^T \boldsymbol{W}^{-1}\boldsymbol{u}_i + \frac{y_i^2}{2(1+\beta^{-1})}\right)^{\frac{\nu-D}{2}+1} \Gamma\left(\frac{D-\nu}{2}-1\right)$$

I AM STUCK HERE.

1D case

$$\frac{1}{\Gamma(\nu/2)}\left(\frac{1}{2}w^{-1}\right)^{\nu/2} \frac{1}{(2\pi(1+\beta^{-1}))^{1/2}} \int_0^\infty \lambda^{(\nu-1)/2} \exp\left(-\frac{1}{2}w^{-1}\lambda - \frac{1}{2}\frac{\lambda(x-m)^2}{1+\beta^{-1}}\right) d\lambda. \tag{33}$$

Want to show

$$\frac{\Gamma((\nu+1-D)/2)}{\Gamma((\nu+1)/2-D)}\frac{|\boldsymbol{W}|^{1/2}}{\pi^{D/2}}\left(\frac{\beta}{1+\beta}\right)\frac{1}{(\nu+1-D)^D}\left[1+\Delta^2\right]^{D-\frac{\nu+1}{2}}$$

where

$$\Delta^2 = \frac{\beta}{1+\beta}(\widehat{\boldsymbol{x}}-\boldsymbol{m})^T\boldsymbol{W}(\widehat{\boldsymbol{x}}-\boldsymbol{m})$$

Exercise 10.23

TODO

Exercise 10.28

TODO

Exercise 10.34

TODO

2.11 Sampling Methods

Exercise 11.4

We start with Equations (11.10) and (11.11), square them and add them together. The result is the following relationship between variables (z_1, z_2) and (y_1, y_2).

$$r^2 = z_1^2 + z_2^2 = \exp\left(-\frac{1}{2}\left(y_1^2 + y_2^2\right)\right)$$

Substitute the equation above back into Equations (11.10) and (11.11) to obtain

$$z_1 = y_1 \frac{\exp\left(-\frac{1}{4}\left(y_1^2 + y_2^2\right)\right)}{\left(y_1^2 + y_2^2\right)^{1/2}} \qquad z_2 = y_2 \frac{\exp\left(-\frac{1}{4}\left(y_1^2 + y_2^2\right)\right)}{\left(y_1^2 + y_2^2\right)^{1/2}}.$$

Differentiating these equations yield the following relations

$$\begin{aligned}
\frac{\partial z_i}{\partial y_i} &= \frac{\left(y_j^2 - y_i^4/2 - y_i^2 y_j^2/2\right) \exp\left(-\frac{1}{4}\left(y_i^2 + y_j^2\right)\right)}{\left(y_i^2 + y_j^2\right)^{3/2}} \\
\frac{\partial z_i}{\partial y_j} &= -\frac{y_i y_j \left(y_i^2/2 + y_j^2/2 + 1\right) \exp\left(-\frac{1}{4}\left(y_i^2 + y_j^2\right)\right)}{\left(y_i^2 + y_j^2\right)^{3/2}}.
\end{aligned}$$

Notice the symmetry $(y_i, y_j) \leftrightarrow (y_j, y_i)$. Symbolic manipulations will reveal that the absolute value of the determinant then becomes

$$\left|\frac{\partial(z_1, z_2)}{\partial(y_1, y_2)}\right| = \left|\frac{\partial z_1}{\partial y_1}\frac{\partial z_2}{\partial y_2} - \frac{\partial z_1}{\partial y_2}\frac{\partial z_2}{\partial y_1}\right| = \frac{1}{2}\exp\left(-\frac{1}{2}\left(y_i^2 + y_j^2\right)\right)$$

which corresponds perfectly with Equation (11.12) and solves the problem.

Exercise 11.7

The following diagram shows the change of variables and the associated probability density functions.

$$\begin{tikzcd}
p_y(y) \arrow[r, bend left, "f"] & q_z(z) \arrow[l, bend left, "h"]
\end{tikzcd}$$

In this problem $p_y(y)$ is uniform on $(0, 1)$ and $z = f(y) = b\tan(y) + c$. The inverse of $f(y)$ is simply $y = h(z) = \tan^{-1}((z - c)/b)$. We wish to find $q_z(z)$, which we do by evaluating

$$q_z(z) = \frac{dy}{dz} = \frac{d}{dz}\left(\tan^{-1}\left(\frac{z - c}{b}\right)\right) = \frac{1}{1 + \left(\frac{z-c}{b}\right)^2}\frac{1}{b}.$$

Here c is a shift parameter and b is a scale parameter. Multiplying by k and ignoring the last factor $1/b$ allows the unnormalized distribution to completely overlap the gamma distribution in Figure 11.5. This is required in rejection sampling.

Exercise 11.12

The standard Gibbs sampler is not ergodic in this case, and would not sample from the shown mixture distribution correctly. Instead it would get stuck on one of the two distributions. To see this, notice that $p(x_1 \mid x_2)$ and $p(x_2 \mid x_1)$ are both zero outside of the initial distribution in the mixture, hence the standard Gibbs sampler has no way of moving from one distribution to the other and is forever stuck.

Exercise 11.15

Starting from Equation (11.58) in the book, we see that

$$\frac{dz_i}{d\tau} = \frac{\partial H}{\partial r_i} = \frac{\partial}{\partial r_i}\left[E(\boldsymbol{z}) + K(\boldsymbol{r})\right] = \frac{\partial K(\boldsymbol{r})}{\partial r_i} = r_i.$$

Where we have used Equations (11.56) and (11.57). The first term in the chain of equalities above equals the last, showing that (11.53) is indeed equivalent to (11.58).

The next sub problem is trivial, since we have

$$\frac{\partial H}{\partial z_i} = \frac{\partial}{\partial z_i}\left[E(\boldsymbol{z}) + K(\boldsymbol{r})\right] = \frac{\partial E(\boldsymbol{z})}{\partial z_i}.$$

2.12 Continuous Latent Variables

Exercise 12.4

If we let the distribution of $\boldsymbol{z}$ by given by $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z} \mid \boldsymbol{m}, \boldsymbol{\Sigma})$, then the marginal distribution over $\boldsymbol{x}$ becomes

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{W}\boldsymbol{m} + \boldsymbol{\mu}, \sigma^2\boldsymbol{I} + \boldsymbol{W}\boldsymbol{\Sigma}^{-1}\boldsymbol{W}^T).$$

There is redundancy in the term $\boldsymbol{W}\boldsymbol{m} + \boldsymbol{\mu}$ as we have the freedom to choose both $\boldsymbol{m}$ and $\boldsymbol{\mu}$ without altering the conditional covariance. Setting $\boldsymbol{m} = \boldsymbol{0}$ lets us use $\boldsymbol{\mu}$ alone to control the conditional mean.

Having set $\boldsymbol{m} = \boldsymbol{0}$, we see that there is a redundancy in the conditional covariance too. Making use of the Cholesky decomposition we can write $\boldsymbol{\Sigma}^{-1}$ as $\boldsymbol{R}^T\boldsymbol{R}$, so the conditional covariance becomes $\sigma^2\boldsymbol{I} + \boldsymbol{W}\boldsymbol{R}^T\boldsymbol{R}\boldsymbol{W}^T = \sigma^2\boldsymbol{I} + \boldsymbol{W}\boldsymbol{R}^T\left(\boldsymbol{W}\boldsymbol{R}^T\right)^T$. We have the freedom to choose both $\boldsymbol{W}$ and $\boldsymbol{R}$ without altering the conditional mean. Setting $\boldsymbol{R} = \boldsymbol{I}$ lets us use $\boldsymbol{W}$ alone to control the conditional covariance.

Setting $\boldsymbol{m} = \boldsymbol{0}$ and $\boldsymbol{R} = \boldsymbol{I}$ we recover

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}, \sigma^2 \boldsymbol{I} + \boldsymbol{W}\boldsymbol{W}^T).$$

Therefore "enriching" the model with $\boldsymbol{m}$ and $\boldsymbol{R}$ did not actually produce a more complex model, only redundant parameters. In terms of re-defining variables, the new definitions would be

$$\widehat{\boldsymbol{\mu}} = \boldsymbol{W}\boldsymbol{m} + \boldsymbol{\mu} \qquad \widehat{\boldsymbol{W}}\widehat{\boldsymbol{W}}^T = \boldsymbol{W}\boldsymbol{\Sigma}^{-1}\boldsymbol{W}^T.$$

Exercise 12.6

The node $\boldsymbol{x}$ can be split up into nodes $x_1, \ldots, x_D$ since the covariance $\sigma^2 \boldsymbol{I}$ is diagonal. Hence the paths from x_i to x_j when $i \neq j$ are tail-to-tail at the observed node $\boldsymbol{z}$, the path is blocked and therefore the conditional independence $x_i \perp\!\!\!\perp x_j \mid \boldsymbol{z}$ holds. Algebraically, the distribution $p(\boldsymbol{x} \mid \boldsymbol{z})$ factorizes into $p(x_1 \mid \boldsymbol{z}) \cdots p(x_D \mid \boldsymbol{z})$ due to the diagonal covariance.

Exercise 12.10

Differentiating Equation (12.43) for the log likelihood once yields

$$\boldsymbol{C}^{-1} \sum_n (\boldsymbol{x}_n - \boldsymbol{\mu}),$$

and equating this to $\boldsymbol{0}$ and solving yields $\boldsymbol{\mu}_{\mathrm{ML}}$. Differentiating again we obtain

$$-\sum_n \boldsymbol{C}^{-1} = -N\boldsymbol{C}^{-1}.$$

A quadratic form $-\boldsymbol{y}^T \boldsymbol{C}^{-1} \boldsymbol{y}$ has a unique maximum at $\boldsymbol{y} = \boldsymbol{0}$ if $\boldsymbol{C}^{-1}$ is positive definite. The inverse of $\boldsymbol{C}^{-1}$ is clearly positive definite, since from (12.36)

$$\boldsymbol{y}^T \boldsymbol{C} \boldsymbol{y} = \left(\boldsymbol{W}^T \boldsymbol{y}\right)^T \left(\boldsymbol{W}^T \boldsymbol{y}\right) + \sigma^2 \boldsymbol{y}^T \boldsymbol{y} > 0 \text{ when } \boldsymbol{y} \neq \boldsymbol{0}.$$

The inverse of a positive definite matrix is also positive definite, which can be shown by writing the matrix in it's eigendecomposition. Since $\boldsymbol{C}^{-1}$ is positive definite $-\boldsymbol{y}^T \boldsymbol{C}^{-1} \boldsymbol{y}$ is always negative except for when $\boldsymbol{y} = \boldsymbol{0}$. Hence the maximum likelihood solution $\boldsymbol{\mu}_{\mathrm{ML}}$ represents a unique maximum.

Exercise 12.14

We consider first when $M = D - 1$ in Equation (12.51), then we have

$$\begin{aligned} D(D-1) + 1 - \frac{(D-1)(D-2)}{2} &= D(D-1) + 1 - \frac{(D-1)D}{2} + (D-1) \\ &= \frac{(D-1)D}{2} + D = \frac{(D+1)D}{2} = D + (D-1) + \cdots + 1 \end{aligned}$$

which recovers the standard result for the degrees of freedom in symmetric a $D \times D$ matrix.

When $M = 0$ Equation (12.51) simply equals 1. This represents the degrees of freedom in the a Gaussian with isotropic covariance matrix $\sigma^2 \boldsymbol{I}$.

Exercise 12.18

The factor analysis model is equal to the PPCA model, except for the fact that the conditional variance $\sigma^2 \boldsymbol{I}$ is replaced by the diagonal matrix $\boldsymbol{\Psi}$. One free parameter σ^2 is replaced by the D parameters in $\boldsymbol{\Psi}$. Following the logic leading up to Equation (12.51), the factor analysis model has

$$DM + D - M(M-1)/2$$

independent parameters in the covariance (and D in the mean). The difference is the 1 in the middle of (12.51) becoming a D.

Exercise 12.23

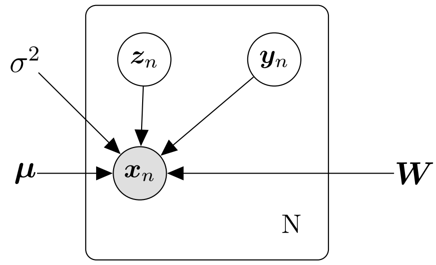
The distribution over $\boldsymbol{z}$ remains $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z} \mid \boldsymbol{0}, \boldsymbol{I})$. We let $\boldsymbol{y} \in \{0,1\}^K$ be a latent multinomial variable representing which of the K mixture PPCA models an observation $\boldsymbol{x}_n$ is drawn from. The latent distribution over $\boldsymbol{y}$ governed by a parameter $\boldsymbol{\pi}$, so that

$$p(y_k = 1) = \pi_k^{y_k}.$$

The conditional distribution becomes

$$p(\boldsymbol{x} \mid \boldsymbol{z}, \boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{W}_k \boldsymbol{z} + \boldsymbol{\mu}_k, \sigma_k^2 \boldsymbol{I}\right). \tag{34}$$

The probabilistic graphical model below is inspired by Figures 12.10 and 9.6 on pages 574 and 433 respectively. Below $\sigma^2 = \{\sigma_1^2, \ldots, \sigma_K^2\}$, $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K\}$ and $\boldsymbol{W} = \{\boldsymbol{W}_1, \ldots, \boldsymbol{W}_K\}$.



If the variances mixture parameters $\{\sigma_k^2\}$, $\{\boldsymbol{\mu}_k\}$ and $\{\boldsymbol{W}_k\}$ are shared, then the mixture model above reduces to the ordinary PPCA model. To see this, notice that Equation (34) reduces to the ordinary PPCA model and the latent variable $\boldsymbol{y}$ becomes superfluous.

2.13 Sequential Data

Exercise 13.8

The first part of the problem asks us to show Equations (13.22) and (13.23). To show Equation (13.22), we combine Equation (13.9) with the multinomial probability density function, given by

$$p(\boldsymbol{x}_n \mid \boldsymbol{\phi}_k) = \prod_{i=1}^{D} \mu_{ik}^{x_{ni}}.$$

We drop the subscript on n, and see that

$$p(\boldsymbol{x} \mid \boldsymbol{z}) = \prod_{k=1}^{K} p(\boldsymbol{x} \mid \boldsymbol{\phi}_k)^{z_k} = \prod_{k=1}^{K} \left(\prod_{i=1}^{D} \mu_{ik}^{x_i} \right)^{z_k} = \prod_{i=1}^{D} \prod_{k=1}^{K} \mu_{ik}^{x_i z_k}.$$

To show Equation (13.23), we maximize Equation (13.17) with respect to μ_{ik}. We use a Lagrange multiplier to account for the constraint $\sum_i \mu_{ik} = 1$ for every $k = 1, \ldots, K$. Only the final term in Equation (13.17) and the constraint term is a function of μ_{ik}, so the Lagrange function becomes

$$L(\mu_{ik}) = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \sum_{i=1}^{D} x_{ni} \ln(\mu_{ik}) - \lambda \left(\sum_{i} \mu_{ik} - 1 \right) + \text{const.}$$

We differentiate this with respect to μ_{ik} and solve for the Lagrange multiplier λ by summing both sides over i. We obtain

$$\frac{1}{\mu_{ik}} \left[\sum_{n=1}^{N} \gamma(z_{nk}) x_{ni} \right] - \lambda = 0 \quad \Rightarrow \quad \lambda = \sum_{i=1}^{D} \sum_{n=1}^{N} \gamma(z_{nk}) x_{ni}.$$

Solving for μ_{ik} yields

$$\mu_{ik} = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) x_{ni}}{\sum_{n=1}^{N} \gamma(z_{nk}) \sum_{i=1}^{D} x_{ni}} = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) x_{ni}}{\sum_{n=1}^{N} \gamma(z_{nk})}.$$

The second part of the problem asks us to consider the case when $\boldsymbol{x}$ has multiple binary outputs governed by Bernoulli distributions. In this case the conditional distribution of $\boldsymbol{x}_n$ is given by

$$p(\boldsymbol{x}_n \mid \boldsymbol{\phi}_k) = \prod_{i=1}^{D} \mu_{ik}^{x_{ni}} \left(1 - \mu_{ik}\right)^{(1 - x_{ni})}$$

and there is no summation constraint on μ_{ik}, but of course it has to be in the range $[0, 1]$. The optimization of Equation (13.17) becomes

$$L(\mu_{ik}) = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \sum_{i=1}^{D} x_{ni} \ln\left(\mu_{ik}\right) + \left(1 - x_{ni}\right) \ln\left(1 - \mu_{ik}\right)$$

where no Lagrange multiplier is present. Differentiating with respect to μ_{ik} yields

$$\partial_{\mu_{ik}} L(\mu_{ik}) = \sum_{n=1}^{N} \gamma(z_{nk}) \left[\frac{x_{ni}}{\mu_{ik}} - \frac{1 - x_{ni}}{1 - \mu_{ik}} \right] = 0$$

and solving this yields

$$\mu_{ik} = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) x_{ni}}{\sum_{n=1}^{N} \gamma(z_{nk})}$$

which is exactly the same as Equation (13.23).

References

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer, New York.

[VanderPlas, 2014] VanderPlas, J. (2014). Frequentism and Bayesianism: A Python-driven Primer.