Machine Learning (SS 2020)

Lecture Notes

Thomas Pock

June 8, 2020

The material of these lecture notes is meant as supplemental material to the slides of the lecture. It contains all the derivations and examples presented at the black board and also contains some additional information for the derivations.

1 Probability theory, decision theory, information theory

1.1 Expectation and variance

The expectation of a function $f(x)$ with respect to its distribution $p(x)$ is defined as

$$\mathbb{E}[f] = \int p(x)f(x)\,\mathrm{d}x,$$

The expectation is a linear operator that is

$$\mathbb{E}[\alpha f + \beta g] = \alpha\mathbb{E}[f] + \beta\mathbb{E}[g], \quad \text{for any } \alpha, \beta \in \mathbb{R}$$

This is easily seen from the following computation:

$$\begin{aligned}\mathbb{E}[\alpha f + \beta g] = &\int p(x)(\alpha f(x) + \beta g(x))\,\mathrm{d}x = \\ &\int p(x)\alpha f(x)\,\mathrm{d}x + \int p(x)\beta g(x)\,\mathrm{d}x = \\ &\alpha\int p(x)f(x)\,\mathrm{d}x + \beta\int p(x)g(x)\,\mathrm{d}x = \alpha\mathbb{E}[f] + \beta\mathbb{E}[g].\end{aligned}$$

The variance is defined as the expectation of the quadratic variation of a random variable around its expected value, that is

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

Expanding the squares and using the linearity of the expectation, we can also get the following representation of the variance:

$$\begin{aligned}\mathrm{var}[f] =&\mathbb{E}[(f(x) - \mathbb{E}[f(x)])^2] = \mathbb{E}\left[f(x)^2 - 2f(x)\mathbb{E}[f(x)] + \mathbb{E}[f(x)]^2\right] = \\ &\mathbb{E}[f(x)^2] - 2\mathbb{E}[f(x)]\mathbb{E}[f(x)] + \mathbb{E}[f(x)]^2 = \\ &\mathbb{E}[f(x)^2] - \mathbb{E}[f(x)]^2.\end{aligned}$$

Note however, that from a numerical point of view this representation is less good as it can lead to unwanted numerical cancellations. The variance is a 2-homogeneous operator that is for any $\alpha \in \mathbb{R}$

$$\operatorname{var}[\alpha f] = \mathbb{E}[(\alpha f)^2] - \mathbb{E}[\alpha f]^2 = \alpha^2 \mathbb{E}[f^2] - \alpha^2 \mathbb{E}[f]^2 = \alpha^2 \operatorname{var}[f].$$

The covariance is defined as

$$\operatorname{cov}[x, y] = \mathbb{E}_{x,y}[(x - \mathbb{E}[x])(y - \mathbb{E}[y])].$$

We can also get a different representation by multiplying out the function which is passed to the expectation:

$$\begin{aligned}\operatorname{cov}[x, y] =& \mathbb{E}_{x,y}[(x - \mathbb{E}[x])(y - \mathbb{E}[y])] = \mathbb{E}\left[xy - \mathbb{E}[x]y - \mathbb{E}[y]x + \mathbb{E}[x]\mathbb{E}[y]\right] = \\ & \mathbb{E}_{x,y}[xy] - \mathbb{E}[x]\mathbb{E}[y].\end{aligned}$$

In case x and y are statistically independent, that is $p(x, y) = p(x)p(y)$, we find that

$$\begin{aligned}\mathbb{E}_{x,y}[xy] = \int\int p(x, y)xy \, \mathrm{d}y \, \mathrm{d}x = \int\int p(x)p(y)xy \, \mathrm{d}y \, \mathrm{d}x = \int p(x)x \left(\int p(y)y \, \mathrm{d}y \right) \mathrm{d}x = \\ \int p(x)x\mathbb{E}[y] \, \mathrm{d}x = \mathbb{E}[x]\mathbb{E}[y].\end{aligned}$$

Hence we see that the covariance of two independent random variables is zero.

1.2 The Gaussian distribution

The Gaussian distribution is given by

$$p(x) = \mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2} \right),$$

where μ is the mean and σ^2 is the variance. It is easy to see that $p(x) \geq 0$ and $\int p(x) \, \mathrm{d}x = 1$.

First, we will compute the expected value of the Gaussian distribution.

$$\begin{aligned}\mathbb{E}[x] = \int xp(x) \, \mathrm{d}x = \frac{1}{\sqrt{2\pi\sigma^2}} \int x \exp\left(-\frac{(x - \mu)^2}{2\sigma^2} \right) \mathrm{d}x \overset{z=x-\mu}{=} \frac{1}{\sqrt{2\pi\sigma^2}} \int (z + \mu) \exp\left(-\frac{z^2}{2\sigma^2} \right) \mathrm{d}z = \\ \frac{1}{\sqrt{2\pi\sigma^2}} \int z \exp\left(-\frac{z^2}{2\sigma^2} \right) \mathrm{d}z + \frac{\mu}{\sqrt{2\pi\sigma^2}} \int \exp\left(-\frac{z^2}{2\sigma^2} \right) \mathrm{d}z.\end{aligned}$$

Inspecting the last line, we see that the first term is an odd function ($f(x) = -f(-x)$) and hence integrates to zero. The second term is a standard Gaussian weighted by μ and hence

$$\mathbb{E}[x] = \mu,$$

that is the expected value of the Gaussian is its mean.

Next, we are going to compute the variance of the Gaussian. First, we note that based on the representation formula of the variance we obtain

$$\mathrm{var}[x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2 = \mathbb{E}[x^2] - \mu^2.$$

Hence, in order to compute the variance we are only left with computing the quantity $\mathbb{E}[x^2]$.

$$\mathbb{E}[x^2] = \int x^2 p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \int x^2 \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \mathrm{d}x.$$

In order to compute the integral, we are now going to make the following change of variables:

$$z = \frac{(x-\mu)}{\sqrt{2\sigma^2}} \implies x = \mu + \sqrt{2\sigma^2} z \implies dx = \sqrt{2\sigma^2}\, \mathrm{d}z.$$

Substituting back gives

$$\begin{aligned}\mathbb{E}[x^2] =& \frac{\sqrt{2\sigma^2}}{\sqrt{2\pi\sigma^2}} \int (\mu + \sqrt{2\sigma^2} z)^2 \exp(-z^2)\, \mathrm{d}z = \frac{1}{\sqrt{\pi}} \int (\mu^2 + 2\mu\sqrt{2\sigma^2} z + 2\sigma^2 z^2) \exp(-z^2)\, \mathrm{d}z = \\ & \frac{\mu^2}{\sqrt{\pi}} \int \exp(-z^2)\, \mathrm{d}z + \frac{2\mu\sqrt{2\sigma^2}}{\sqrt{\pi}} \int z \exp(-z^2)\, \mathrm{d}z + \frac{2\sigma^2}{\sqrt{\pi}} \int z^2 \exp(-z^2)\, \mathrm{d}z.\end{aligned}$$

Note again that the middle integral is zero as it is an integral over an odd function. Now using that for any $a > 0$,

$$\int \exp(-ax^2)\, \mathrm{d}x = \sqrt{\frac{\pi}{a}}, \quad \text{and} \quad \int x^2 \exp(-ax^2)\, \mathrm{d}x = \frac{1}{2}\sqrt{\frac{\pi}{a^3}},$$

we obtain

$$\mathbb{E}[x^2] = \mu^2 + \sigma^2 \implies \mathrm{var}[x] = \sigma^2.$$

1.3 Maximum likelihood

Given a dataset $\mathbf{x} = (x_1, ..., x_n)$ drawn independently from the same Gaussian distribution with mean μ and variance σ^2, the joint probability of the entire data set is given by

$$p(\mathbf{x}|\mu, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}(x_n|\mu, \sigma^2),$$

where we have used that for independent random variables x_1 and x_2 the joint probability factorizes in its marginals, that is $p(x_1, x_2) = p(x_1)p(x_2)$.

Instead of maximizing the likelihood function, we will maximize the logarithm of the likelihood function. Using the facts that $\log(ab) = \log(a) + \log(b)$ and $\log(a^b) = b\log(a)$, the log-likelihood function is given by

$$\log(p(\mathbf{x}|\mu, \sigma^2)) = \log\left(\prod_{n=1}^{N} \mathcal{N}(x_n|\mu, \sigma^2)\right) = \sum_{n=1}^{N} \log\left(\mathcal{N}(x_n|\mu, \sigma^2)\right).$$

Inserting the definition of the Gaussian distribution, we obtain

$$\log(p(\mathbf{x}|\mu,\sigma^2)) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2 - \frac{N}{2}\log\sigma^2 - \frac{N}{2}\log(2\pi).$$

Now, the idea of maximum likelihood is to find out the parameters μ and σ^2 of the Gaussian distribution, that maximize the log-likelihood function. That, is we are left in solving the following (concave) maximization problem:

$$\max_{\mu,\sigma^2} -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2 - \frac{N}{2}\log\sigma^2,$$

where we have removed the last term, as it does not depend on μ and σ^2. Computing the derivative with respect to μ and setting it to zeros yields

$$\begin{aligned}
-\frac{2}{2\sigma^2}\sum_{n=1}^{N}\mu - x_n &= 0\\
\sum_{n=1}^{N}\mu &= \sum_{n=1}^{N}x_n\\
N\mu &= \sum_{n=1}^{N}x_n\\
\mu_{ML} &= \frac{1}{N}\sum_{n=1}^{N}x_n,
\end{aligned}$$

which is the usual formula for computing the mean value. Next, we are computing the derivative of the log-likelihood function with respect to σ^2 (by treating σ^2 as the variable) and setting it again to zero.

$$\begin{aligned}
-\frac{-1}{2(\sigma^2)^2}\sum_{n=1}^{N}(x_n-\mu_{ML})^2 - \frac{N}{2}\frac{1}{\sigma^2} &= 0\\
\sum_{n=1}^{N}(x_n-\mu_{ML})^2 - N\sigma^2 &= 0\\
\sigma^2_{ML} &= \frac{1}{N}\sum_{n=1}^{N}(x_n-\mu_{ML})^2
\end{aligned}$$

An important question is now what is the expected value of μ_{ML} and σ^2_{ML} when treating them as a function of the data set $\mathbf{x}$. First we address the mean value μ_{ML}, where we will make use of the linearity of the expectation operator.

$$\mathbb{E}[\mu_{ML}] = \mathbb{E}\left[\frac{1}{N}\sum_{n=1}^{N}x_n\right] = \frac{1}{N}\mathbb{E}\left[\sum_{n=1}^{N}x_n\right] = \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}\left[x_n\right] = \frac{1}{N}N\mu = \mu.$$

This shows that the mean computed from the maximum likelihood approach is indeed an unbiased estimator. Next we will investigate the expected value of the maximum likelihood estimator σ^2_{ML}.

$$\mathbb{E}[\sigma^2_{ML}] = \mathbb{E}\left[\frac{1}{N}\sum_{n=1}^{N}(x_n - \mu_{ML})^2\right] = \mathbb{E}\left[\frac{1}{N}\sum_{n=1}^{N} x_n^2 - 2x_n\mu_{ML} + \mu^2_{ML}\right] =$$
$$\frac{1}{N}\mathbb{E}\left[\sum_{n=1}^{N} x_n^2\right] - \frac{2}{N}\mathbb{E}[\mu_{ML}\sum_{n=1}^{N} x_n] + \frac{1}{N}\mathbb{E}[N\mu^2_{ML}].$$

Using that $\sum_{n=1}^{N} x_n = N\mu_{ML}$ we obtain

$$\mathbb{E}[\sigma^2_{ML}] = \frac{1}{N}\mathbb{E}\left[\sum_{n=1}^{N} x_n^2\right] - 2\mathbb{E}[\mu^2_{ML}] + \mathbb{E}[\mu^2_{ML}] = \frac{1}{N}\mathbb{E}\left[\sum_{n=1}^{N} x_n^2\right] - \mathbb{E}[\mu^2_{ML}] =$$
$$\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}[x_n^2] - \mathbb{E}[\mu^2_{ML}] = \mathbb{E}[x^2] - \mathbb{E}[\mu^2_{ML}].$$

We also have that

$$\text{var}[x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2 = \mathbb{E}[x^2] - \mu^2, \quad \text{var}[\mu_{ML}] = \mathbb{E}[\mu^2_{ML}] - \mathbb{E}[\mu_{ML}]^2 = \mathbb{E}[\mu^2_{ML}] - \mu^2.$$

Substituting these two expressions into our formula gives

$$\mathbb{E}[\sigma^2_{ML}] = \text{var}[x] + \mu^2 - \text{var}[\mu_{ML}] - \mu^2 = \text{var}[x] - \text{var}[\mu_{ML}].$$

Finally we need an expression of the last term

$$\text{var}[\mu_{ML}] = \text{var}\left[\frac{1}{N}\sum_{n=1}^{N} x_n\right] = \frac{1}{N^2}\,\text{var}\left[\sum_{n=1}^{N} x_n\right].$$

Using our assumption that the x_n are i.i.d, the variance decomposes and gives

$$\text{var}[\mu_{ML}] = \frac{1}{N^2}\,\text{var}\left[\sum_{n=1}^{N} x_n\right] = \frac{1}{N^2}\sum_{n=1}^{N}\text{var}\,[x_n] = \frac{1}{N^2}N\,\text{var}[x] = \frac{1}{N}\,\text{var}[x].$$

Finally, by combining the previous equations we obtain

$$\mathbb{E}[\sigma^2_{ML}] = \text{var}[x] - \text{var}[\mu_{ML}] = \text{var}[x] - \frac{1}{N}\,\text{var}[x] = \frac{N-1}{N}\,\text{var}[x].$$

Hence the ML estimator of the variance is biased by a factor of $(N-1)/N$.

1.4 The expected quadratic loss for regression

Assume we have a function $y(x)$ which is an estimator for the true target variable t. Let us use a standard quadratic loss function $L(t, y(x)) = (y(x) - t)^2$. The expected quadratic loss is given by

$$\mathbb{E}[L] = \int \int (y(x) - t)^2 p(x, t) \, \mathrm{d}t \, \mathrm{d}x.$$

The functional derivative with respect to $y(x)$ is given by

$$\frac{\delta \mathbb{E}[L]}{\delta y(x)} = 2 \int (y(x) - t) p(x, t) \, \mathrm{d}t.$$

As $\mathbb{E}[L]$ is a functional, the optimality condition is given by means of test functions η:

$$\int \frac{\delta \mathbb{E}[L]}{\delta y(x)} \eta \, \mathrm{d}x = 0, \quad \forall \eta,$$

from which it follows that the derivative must vanish, that is

$$2 \int (y(x) - t) p(x, t) \, \mathrm{d}t = 0.$$

Rearranging the equation yields

$$y(x) = \frac{\int t p(x, t) \, \mathrm{d}t}{\int p(x, t) \, \mathrm{d}t} = \frac{\int t p(t|x) p(x) \, \mathrm{d}t}{p(x)} = \int t p(t|x) \, \mathrm{d}t = \mathbb{E}_t[t|x],$$

which is the conditional expectation of t, given x.

This result can also be derived in a different way by rewriting the quadratic loss function in the following way:

$$(y(x)-t)^2 = (y(x)-\mathbb{E}_t[t|x]+\mathbb{E}_t[t|x]-t)^2 = (y(x)-\mathbb{E}_t[t|x])^2+2(y(x)-\mathbb{E}_t[t|x])(\mathbb{E}_t[t|x]-t)+(\mathbb{E}_t[t|x]-t)^2.$$

Substituting this expression in the expectation, we obtain

$$\begin{aligned}\mathbb{E}[L] = &\int \int (y(x) - \mathbb{E}_t[t|x])^2 p(x, t) \, \mathrm{d}t \, \mathrm{d}x + \\ &2 \int \int (y(x) - \mathbb{E}_t[t|x])(\mathbb{E}_t[t|x] - t) p(x, t) \, \mathrm{d}t \, \mathrm{d}x + \\ &\int \int (\mathbb{E}_t[t|x] - t)^2 p(x, t) \, \mathrm{d}t \, \mathrm{d}x.\end{aligned}$$

Using the product rule $p(x, t) = p(t|x)p(x)$ we can modify this expression as follows:

$$\begin{aligned}\mathbb{E}[L] = &\int (y(x) - \mathbb{E}_t[t|x])^2 \left(\int p(t|x) \, \mathrm{d}t \right) p(x) \, \mathrm{d}x + \\ &2 \int (y(x) - \mathbb{E}_t[t|x]) \left(\int (\mathbb{E}_t[t|x] - t) p(t|x) \, \mathrm{d}t \right) p(x) \, \mathrm{d}x + \\ &\int \left(\int (\mathbb{E}_t[t|x] - t)^2 p(t|x) \, \mathrm{d}t \right) p(x) \, \mathrm{d}x.\end{aligned}$$

Next, note that $\int p(t|x)\,\mathrm{d}t = 1$, $\int (\mathbb{E}_t[t|x] - t)^2 p(t|x)\,\mathrm{d}t = \mathrm{var}[t|x]$, and

$$\int (\mathbb{E}_t[t|x] - t)p(t|x)\,\mathrm{d}t = \mathbb{E}_t[t|x] - \int tp(t|x)\,\mathrm{d}t = \mathbb{E}_t[t|x] - \mathbb{E}_t[t|x] = 0,$$

such that we arrive at

$$\mathbb{E}[L] = \int (y(x) - \mathbb{E}_t[t|x])^2 p(x)\,\mathrm{d}x + \int \mathrm{var}[t|x]p(x)\,\mathrm{d}x.$$

As only the first term depends on $y(x)$, this expression is minimized by setting $y(x) = \mathbb{E}_t[t|x]$. The second term is a constant that only depends on the intrinsic noise of the data.

1.5 Entropy

The entropy of a discrete random variable is defined as the expectation of the negative log probability of the random variable, that is

$$H[x] = \mathbb{E}_p[-\log x] = -\sum_x p(x)\log p(x).$$

The base of the logarithm is usually arbitrary and we will not mention it explicitly. Sometimes the base 2 is selected such that the unit of the entropy is measure in "bits".

An interesting question to ask is what is the discrete probability distribution which has the highest entropy (hence hardest to compress)? It will turn out that maximum-entropy distribution is given by the uniform distribution. To derive this result, let us assume that we have a discrete random variable X of $M \geq 1$ states $\{x_1, ..., x_M\}$. We then seek to maximize the entropy of that random variable under the constraint that its probabilities $p(x_i)$ form a valid distribution that is $p(x_i) \geq 0$, $\sum_{i=1}^M p(x_i) = 1$. To keep the notation uncluttered we define $p_i = p(x_i)$ and $p = (p_1, ..., p_M)$.

$$\max_p -\sum_{i=1}^M p_i \log p_i, \quad \text{s.t.} \quad p_i \geq 0, \ \sum_{i=1}^M p_i = 1.$$

First, we note that the constraint $p_i \geq 0$ can be dropped as the entropy diverges to $-\infty$ for $p_i \to 0$. For the equality constraint, we introduce a Lagrange multiplier and the associated Lagrangian becomes

$$L(p, \lambda) = -\sum_{i=1}^M p_i \log p_i + \lambda \left(\sum_{i=1}^M p_i - 1 \right).$$

Differentiating with respect to p_i and λ and setting to zero gives the following nonlinear system of equations

$$-\log p_i - 1 + \lambda = 0.$$
$$\sum_{i=1}^M p_i - 1 = 0.$$

From the first equation we can compute

$$p_i = \exp(\lambda - 1).$$

Substituting back into the second equation yields

$$\sum_{i=1}^{M} \exp(\lambda - 1) = 1 \implies \exp(\lambda - 1) = \frac{1}{M}.$$

Finally, combining with the first equation yields

$$p_i = \frac{1}{M},$$

and hence a uniform distribution.

Next, we are going to show that in case of continuous distributions, the distribution that maximizes the entropy among all distributions with fixed mean μ and fixed variance σ^2 is the Gaussian distribution. To show this, we are going to solve to following optimization problem:

$$\max_p -\int p(x)\log p(x)\,\mathrm{d}x, \quad \text{s.t.} \quad p(x) \geq 0,\ \int p(x)\,\mathrm{d}x = 1,\ \int xp(x)\,\mathrm{d}x = \mu,\ \int x^2 p(x)\,\mathrm{d}x = \sigma^2+\mu^2.$$

As before, we can skip the $p(x) \geq 0$ constraint. We consider the following Lagrangian:

$$L(p, \lambda_0, \lambda_1, \lambda_2) = -\int p(x)\log p(x)\,\mathrm{d}x + \lambda_0\left(\int p(x)\,\mathrm{d}x - 1\right) + \lambda_1\left(\int xp(x)\,\mathrm{d}x - \mu\right) + \lambda_2\left(\int x^2 p(x)\,\mathrm{d}x - (\sigma^2+\mu^2)\right).$$

Differentiating with respect to $p(x)$ gives

$$-\log p(x) - 1 + \lambda_0 + x\lambda_1 + x^2\lambda_2 = 0 \implies p(x) = e^{(\lambda_0 + x\lambda_1 + x^2\lambda_2 - 1)},$$

from which we see that the sought distribution will be the exponential of a quadratic form. It remains to compute the multipliers $\lambda_0, ..., \lambda_2$, which will be done from the constraint equations. For this we will make use of the following known integral identities for $a > 0$:

$$\int e^{(-ax^2+bx)}\,\mathrm{d}x = \sqrt{\frac{\pi}{a}}e^{\left(\frac{b^2}{4a}\right)}, \quad \int xe^{(-ax^2+bx)}\,\mathrm{d}x = \frac{\sqrt{\pi}b}{2a^{3/2}}e^{\left(\frac{b^2}{4a}\right)}, \quad \int x^2 e^{(-ax^2+bx)}\,\mathrm{d}x = \frac{\sqrt{\pi}(2a+b^2)}{4a^{5/2}}e^{\left(\frac{b^2}{4a}\right)}.$$

Comparing the coefficients with our exponential distribution, one can easily see that $b = \lambda_1$ and $a = -\lambda_2$. Now we are going to evaluate the integrals of our three constraint equations:

$$\int p(x)\,\mathrm{d}x = 1 \implies e^{\lambda_0 - 1}\int e^{(x\lambda_1 + x^2\lambda_2)}\,\mathrm{d}x = e^{\lambda_0 - 1}\sqrt{\frac{\pi}{-\lambda_2}}e^{-\frac{\lambda_1^2}{4\lambda_2}} = 1,$$

$$\int xp(x)\,\mathrm{d}x = \mu \implies e^{\lambda_0 - 1}\int xe^{(x\lambda_1 + x^2\lambda_2)}\,\mathrm{d}x = e^{\lambda_0 - 1}\frac{\sqrt{\pi}b}{2(-\lambda_2)^{3/2}}e^{-\frac{\lambda_1^2}{4\lambda_2}} = \mu,$$

$$\int x^2 p(x)\,\mathrm{d}x = \sigma^2 + \mu^2 \implies e^{\lambda_0 - 1}\int x^2 e^{(x\lambda_1 + x^2\lambda_2)}\,\mathrm{d}x = e^{\lambda_0 - 1}\frac{\sqrt{\pi}(\lambda_1^2 - 2\lambda_2)}{4(-\lambda_2)^{5/2}}e^{-\frac{\lambda_1^2}{4\lambda_2}} = \sigma^2 + \mu^2.$$

From the first equation we can express the product of the two factors which are common to all three equations as

$$e^{\lambda_0 - 1} e^{-\frac{\lambda_1^2}{4\lambda_2}} = \sqrt{\frac{-\lambda_2}{\pi}}.$$

Substituting this expression into the second equation gives

$$\sqrt{\frac{-\lambda_2}{\pi}} \frac{\sqrt{\pi}\lambda_1}{2(-\lambda_2)^{3/2}} = -\frac{\lambda_1}{2\lambda_2} = \mu.$$

Similarly, the third equation gives

$$\sqrt{\frac{-\lambda_2}{\pi}} \frac{\sqrt{\pi}(\lambda_1^2 - 2\lambda_2)}{4(-\lambda_2)^{5/2}} = \frac{\lambda_1^2 - 2\lambda_2}{4\lambda_2^2} = \sigma^2 + \mu^2.$$

Combining the last two equations gives the expression of $-\lambda_2$ as

$$\frac{\lambda_1^2 - 2\lambda_2}{4\lambda_2^2} = \sigma^2 + \left(-\frac{\lambda_1}{2\lambda_2}\right)^2 \implies -\lambda_2 = \frac{1}{2\sigma^2},$$

from which we see that $-\lambda_2 \geq 0$. Substituting this expression back in the second equation gives

$$\lambda_1 \sigma^2 = \mu \implies \lambda_1 = \frac{\mu}{\sigma^2}.$$

and from the first equation we obtain

$$e^{\lambda_0 - 1} e^{\frac{\mu^2}{2\sigma^2}} = \frac{1}{\sqrt{2\pi\sigma^2}} \implies e^{\lambda_0 - 1} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\mu^2}{2\sigma^2}}.$$

Finally, putting everything together gives the desired result

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\mu^2}{2\sigma^2}} e^{(\frac{\mu x}{\sigma^2} - \frac{x^2}{2\sigma^2})} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

1.6 The Kullback-Leibler divergence

Given two continuous probability distributions $p(x)$ and $q(x)$, the KL divergence is defined as the expected value of the quantity $\log \frac{q(x)}{p(x)}$ with respect to the distribution $p(x)$:

$$KL(p||q) = \mathbb{E}_p[-\log \frac{q(x)}{p(x)}] = -\int p(x) \log \frac{q(x)}{p(x)} \, \mathrm{d}x.$$

In order to show that $KL(p||q) \geq 0$ for all p, q we will make use of the convexity of the KL divergence in its second argument q.

A function $f(x)$ is said convex if it is defined over a convex set C and if for any $x, y \in C$ and $\lambda \in [0, 1]$ it holds that

$$f(\lambda x + (1 - \lambda y)) \leq \lambda f(x) + (1 - \lambda) f(y).$$

Intuitively, the function value $f(\lambda x + (1-\lambda y))$ always lies below the chord $\lambda f(x) + (1-\lambda) f(y)$ which is spanned between the points $(x, f(x))$ and $(y, f(y)$.

The generalization of the definition of convexity to several points x_i (possibly infinitely many) is known under the name "Jensen's inequality". Let f be a convex function and let $x_i \in C$, $i = 1, ..., n$. Moreover, let $\lambda = (\lambda_1, ..., \lambda_n)$ such that $\lambda_i \geq 0$ and $\sum_{i=1}^n \lambda_i = 1$, Then it holds that

$$f\left(\sum_{i=1}^n \lambda_i x_i\right) \leq \sum_{i=1}^n \lambda_i f(x_i).$$

In case of continuous distributions $p(x)$, Jensen's inequality is given by

$$f\left(\int x p(x)\, \mathrm{d}x\right) \leq \int p(x) f(x)\, \mathrm{d}x.$$

Observe that Jensen's inequality can also be written in terms of expected values:

$$f\left(\mathbb{E}_p[x]\right) \leq \mathbb{E}_p[f(x)].$$

Now, as $-\log(x)$ is a convex function (this is easily see from the fact that the second derivative $f''(x) = 1/x^2 > 0$ for all $x > 0$), we can apply Jensen's inequality to the KL divergence.

$$KL(p||q) = \mathbb{E}_p[-\log \frac{q(x)}{p(x)}] \geq -\log\left(\mathbb{E}_p\left[\frac{q(x)}{p(x)}\right]\right) = -\log\left(\int p(x)\frac{q(x)}{p(x)}\, \mathrm{d}x\right) = -\log\left(\underbrace{\int q(x)\, \mathrm{d}x}_{=1}\right) = -\log 1 = 0.$$

Hence $KL(p||q) \geq 0$ and $KL(p||q) = 0$ if and only if $p = 0$.

2 Probability distributions

2.1 The Bernoulli distribution

The Bernoulli distribution is the distribution of a binary random variable x that can take the value $x = 1$ or $x = 1$. Let $\mu \in [0, 1]$ be the probability of $x = 1$, the Bernoulli distribution is then given by

$$Bern(x|\mu) = \mu^x (1-\mu)^{1-x}.$$

Let us rapidly observe that $Bern(x = 1|\mu) = \mu$ and $Bern(x = 0|\mu) = (1-\mu)$. Next we compute the expected value, which is given by

$$\mathbb{E}[x] = 1 \cdot \mu^1 (1-\mu)^{1-1} + 0 \cdot \mu^0 (1-\mu)^{1-0} = \mu,$$

and the variance which is given by

$$\mathrm{var}[x] = \mathbb{E}[(x-\mu)^2] = (1-\mu)^2 \cdot \mu^1 (1-\mu)^{1-1} + (0-\mu)^2 \cdot \mu^0 (1-\mu)^{1-0} = (1-\mu)^2\mu + \mu^2(1-\mu) = \mu(1-\mu).$$

Next we will perform a maximum likelihood estimation of the parameter μ given a dataset $\mathcal{D} = \{x_1, ..., x_N\}$, where each $x_n \in \{0, 1\}$. Assuming that all samples in $\mathcal{D}$ are i.i.d, the joint probability distribution is given by the likelihood function

$$p(\mathcal{D}|\mu) = \prod_{n=1}^{N} p(x_n|\mu) = \prod_{n=1}^{N} \mu^{x_n}(1-\mu)^{1-x_n}.$$

The log-likelihood is given by

$$\log p(\mathcal{D}|\mu) = \sum_{n=1}^{N} \log p(x_n|\mu) = \sum_{n=1}^{N} x_n \log \mu + (1-x_n)\log(1-\mu).$$

In order to determine the most likely parameter μ we are now going to maximize the log-likelihood function. Computing its derivative and setting it to zeros yields

$$\sum_{n=1}^{N} x_n \frac{1}{\mu} + (1-x_n)\frac{1}{1-\mu}(-1) = 0.$$

Solving this equation for μ gives the somewhat expected result

$$\mu_{ML} = \frac{1}{N}\sum_{n=1}^{N} x_n.$$

Denoting by $m = \#\{x_n : x_n = 1\}$, i.e. the number of experiments where $x_n = 1$, the maximum likelihood estimate for μ can also be written as

$$\mu_{ML} = \frac{m}{N}.$$

2.2 Binomial distribution

Let us again consider the binary experiment (for example flipping a coin) where $x = 1$ with probability μ and $x = 0$ with probability $1 - \mu$. We could ask the question, what is the probability to see m times the value $x = 1$ when repeating the experiment N times. From the previous exposition, we see that the probability such an outcome is

$$\mu^m(1-\mu)^{N-m}.$$

Figure 1 visualizes a binary experiment with $N = 3$ repetitions. Observe that there are actually 3 possibilities that the experiment gives $m = 2$ times the value $x = 1$. Hence, in order to compute the probability of seeing $m = 2$ times the value $x = 1$ we need to add up these 2 possibilities, hence

$$p(m = 2|\mu, N = 3) = 3\mu^m(1-\mu)^{N-m}.$$

For more general N and m we obtain the well-known Binomial distribution

$$Bin(m|N, \mu) = \binom{N}{m}\mu^m(1-\mu)^{N-m},$$

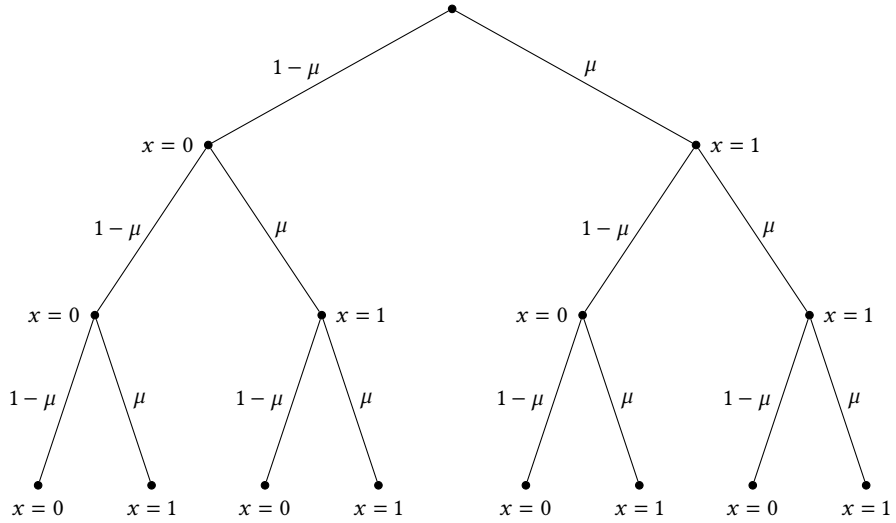

Figure 1: Probability tree for a binary experiment with $N = 3$ repetitions. Observe that there are three possibilities to see $m = 2$ times the values $x = 1$.

where the binomial coefficient

$$\binom{N}{m} = \frac{N!}{(N-m)!m!},$$

counts the number of ways of choosing m objects out of N objects. To see that the Binomial distribution is properly normalized, we write

$$\sum_{m=0}^{N} Bin(m|N,\mu) = \sum_{m=0}^{N} \binom{N}{m} \mu^m (1-\mu)^{N-m} = ((1-\mu)+\mu)^N = 1,$$

where we have used the famous Binomial theorem

$$(a+b)^k = \sum_{l=0}^{k} \binom{k}{l} a^{k-l} b^l.$$

Next, we can also compute the expected value and the variance of the Binomial distribution. For the expected value we have

$$\mathbb{E}[m] = \sum_{m=0}^{N} mBin(m|N,\mu) = \sum_{m=1}^{N} mBin(m|N,\mu) = \sum_{m=1}^{N} m \binom{N}{m} \mu^m (1-\mu)^{N-m}.$$

Next we note that

$$m\binom{N}{m} = m\frac{N!}{(N-m)!m!} = N\frac{(N-1)!}{((N-1)-(m-1))!(m-1)!} = N\binom{N-1}{m-1},$$

such that

$$\mathbb{E}[m] = N\sum_{m=1}^{N} \binom{N-1}{m-1} \mu^m (1-\mu)^{N-m} = N\mu \sum_{m=1}^{N} \binom{N-1}{m-1} \mu^{m-1} (1-\mu)^{(N-1)-(m-1)}.$$

Finally by a change of variables $n = m - 1$ we obtain

$$\mathbb{E}[m] = N\mu \sum_{n=0}^{N-1} \binom{N-1}{n} \mu^n (1-\mu)^{(N-1)-n} = N\mu \sum_{n=0}^{N-1} Bin(n|N-1,\mu) = N\mu.$$

For the variance we first make use of the fact that

$$\mathrm{var}[m] = \mathbb{E}[m^2] - \mathbb{E}[m]^2 = \mathbb{E}[m^2] - N^2\mu^2.$$

In order to compute $\mathbb{E}[m^2]$ we proceed in the same way as before to obtain

$$\mathbb{E}[m^2] = \sum_{m=1}^{N} m^2 Bin(m|N,\mu) = N\mu \sum_{m=1}^{N-1} mBin(m-1|N-1,\mu) \overset{n=m-1,\, M=N-1}{=} N\mu \sum_{n=0}^{M} (n+1)Bin(n|M,\mu).$$

Then by expanding the expression $(n+1)$ we have

$$\mathbb{E}[m^2] = N\mu + N\mu \sum_{n=0}^{M} nBin(n|M,\mu) = N\mu + N\mu M\mu = N\mu + N^2\mu^2 - N\mu^2.$$

Finally, the variance is given by

$$\mathrm{var}[m] = N\mu + N^2\mu^2 - N\mu^2 - N^2\mu^2 = N\mu(1-\mu).$$

2.3 The Gaussian distribution

Consider two uniformly distributed but independent random variables X, Y, both are defined in the interval $[0,1]$, that is their density functions are given by the box functions:

$$p_X(t) = p_Y(t) = \begin{cases} 1 & \text{if } 0 \le t < 1 \\ 0 & \text{else.} \end{cases}$$

One could ask, what is the distribution of the average of these two variables? Let us define $Z = (X+Y)/2$. Since both variables are independent, their joint probability density is given by $p_{X,Y}(x,y) = p_X(x)p_Y(y)$. Then

$$P_Z(Z \le z) = P_Z\left(\frac{X+Y}{2} \le z\right) = \int_{-\infty}^{\infty} \left(\int_{-\infty}^{2z-x} p_X(x)p_Y(y)\, \mathrm{d}y \right) \mathrm{d}x.$$

Let us perform the following change of variables: $y = f(t) = 2t - x$ which implies that

$$\mathrm{d}y = 2\, \mathrm{d}t,\ \ t = f^{-1}(y) = \frac{y+x}{2}.$$

Moreover, the integration bound $\bar{y} = 2z - x$ has to be transformed into an integration bound of the new variable that is $\bar{t} = f^{-1}(\bar{y}) = z$. Therfore,

$$P_Z(z \le Z) = \int_{-\infty}^{\infty} \left(\int_{-\infty}^{z} 2p_X(x)p_Y(2t-x)\, \mathrm{d}t \right) \mathrm{d}x = \int_{-\infty}^{z} \left(\underbrace{2\int_{-\infty}^{\infty} p_X(x)p_Y(2t-x)\, \mathrm{d}x}_{=p_Z(t)} \right) \mathrm{d}t.$$

Observe that $p_Z(t)$ is nothing else than the convolution of two box functions and a scaling by a factor of two. The resulting function is a piecewise linear triangle-shaped function which is given by

$$p_Z(t) = \begin{cases} 0 & \text{if } t < 0 \\ 4t & \text{if } 0 \leq t \leq \frac{1}{2} \\ 4(1-t) & \text{if } \frac{1}{2} \leq t \leq 1 \\ 0 & \text{if } t > 1. \end{cases}$$

This process can be repeated and the resulting denisty function will become increasingly smooth. The central limit theorem states that in the limit $Z = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} X_n$ its distribution will tend towards a Gaussian distribution, even if the random variables are not normally distributed, see also figure 2.

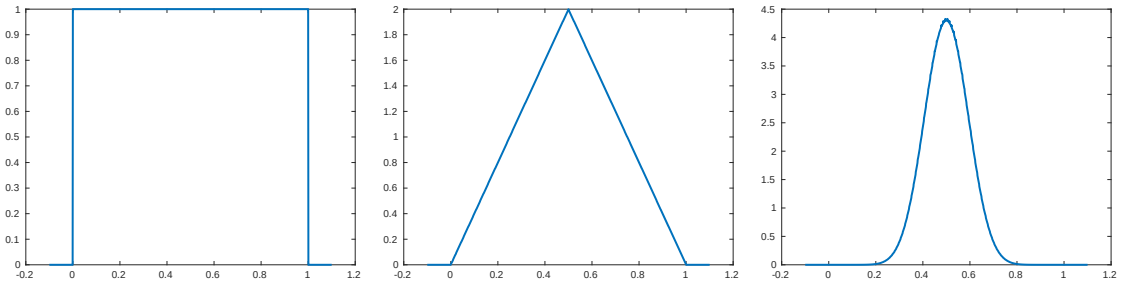

Figure 2: Left: Uniform distribution, middle: average of two uniform distributions, right: average of 10 uniform distributions.

.

2.4 The multivariate Gaussian

Recall that the multivariate Gaussian distribution is defined as

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

The main characteristic of the Gaussian is that the exponent is a quadratic function of the form

$$\Delta^2 = (\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}),$$

where Δ is called the Mahalanobis distance. It can be seen as the Euclidean distance measured ion the metric Σ^{-1}. Note that without loss of generality, Σ^{-1} is a symmetric matrix. If Σ^{-1} is not symmetric then any antisymmetric component would disappear from the quadratic form. Indeed,

$$\boldsymbol{x}^T\Sigma^{-1}\boldsymbol{x} = \left(\boldsymbol{x}^T\Sigma^{-1}\boldsymbol{x}\right)^T = \boldsymbol{x}^T(\boldsymbol{x}^T\Sigma^{-1})^T = \boldsymbol{x}^T\Sigma^{-T}\boldsymbol{x} \implies \Sigma^{-1} = \Sigma^{-T}$$

The well-known spectral decomposition theorem allows to write $\Sigma = U\Lambda U^T$, where $U = [u_1, ..., u_D]$ is an orthogonal matrix whose columns are the eigenvectors of Σ. The matrix

$\mathbf{\Lambda} = \text{diag}(\lambda_1, ..., \lambda_D)$ is a diagonal matrix consisting of the eigenvalues λ_i. As Σ is symmetric, all eigenvalues are real-valued. The matrix Σ is positive definite if and only if all eigenvalues are positive.

Geometrically, the matrix U defines a rotation of the space and the matrix $\mathbf{\Lambda}$ performs a scaling along the respective rotation exes. Based on the spectral decomposition theorem the matrix Σ has the following expansion in terms of its eigenvectors and eigenvalues.

$$\Sigma = \sum_{i=1}^{D} \lambda_i u_i u_i^T .$$

Moreover as $U^{-1} = U^T$ and $(AB)^{-1} = B^{-1}A^{-1}$, its inverse is computed as $\Sigma^{-1} = U\mathbf{\Lambda}^{-1}U^T$ and its expansion reads

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

Let us finally observe that the normalization constant of the Gaussian is correct. Let us assume that $\mu = 0$ (no translation) and that $\Sigma = \mathbf{\Lambda}$ (only scaling, no rotation). Then

$$p(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^D \prod_{i=1}^{D} \lambda_i}} \exp\left\{-\frac{1}{2}\sum_{i=1}^{D} \frac{x_i^2}{\lambda_i}\right\} = \prod_{i=1}^{D} \frac{1}{\sqrt{2\pi\lambda_i}} \exp\left\{-\frac{x_i^2}{2\lambda_i}\right\}.$$

Integrating this expression can now be done along each dimension separately

$$\int p(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} = \prod_{i=1}^{D} \underbrace{\int \frac{1}{\sqrt{2\pi\lambda_i}} \exp\left\{-\frac{x_i^2}{2\lambda_i}\right\} \mathrm{d}x_i}_{=1} = 1.$$

2.5 Conditional Gaussian

Consider a Gaussian distribution $p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \Sigma)$, whose vector $\boldsymbol{x}$ can be grouped into two blocks of variables, hence

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_a \\ \boldsymbol{x}_b \end{pmatrix}, \quad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix}, \quad \mathbf{\Lambda} = \Sigma^{-1} = \begin{pmatrix} \mathbf{\Lambda}_{aa} & \mathbf{\Lambda}_{ab} \\ \mathbf{\Lambda}_{ba} & \mathbf{\Lambda}_{bb} \end{pmatrix},$$

Because of symmetry of Σ and $\mathbf{\Lambda}$, we have $\Sigma_{ab} = \Sigma_{ba}$ and $\mathbf{\Lambda}_{ab} = \mathbf{\Lambda}_{ba}$.

We are now going to compute the conditional distribution $p(\boldsymbol{x}_a|\boldsymbol{x}_b)$, which will turn out to be again a certain Gaussian distribution of the form $\mathcal{N}(\boldsymbol{x}_a|\mu_{a|b}, \Sigma_{a|b})$. For this we perform the following quadratic expansion

$$\begin{aligned} -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) = &-\frac{1}{2}(\boldsymbol{x}_a-\boldsymbol{\mu}_a)^T\mathbf{\Lambda}_{aa}(\boldsymbol{x}_a-\boldsymbol{\mu}_a) \\ &-\frac{1}{2}(\boldsymbol{x}_a-\boldsymbol{\mu}_a)^T\mathbf{\Lambda}_{ab}(\boldsymbol{x}_b-\boldsymbol{\mu}_b) \\ &-\frac{1}{2}(\boldsymbol{x}_b-\boldsymbol{\mu}_b)^T\mathbf{\Lambda}_{ba}(\boldsymbol{x}_a-\boldsymbol{\mu}_a) \\ &-\frac{1}{2}(\boldsymbol{x}_b-\boldsymbol{\mu}_b)^T\mathbf{\Lambda}_{bb}(\boldsymbol{x}_b-\boldsymbol{\mu}_b). \end{aligned} \tag{2.1}$$

For a general Gaussian, we remark that

$$-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) = -\frac{1}{2}\boldsymbol{x}^T\Sigma^{-1}\boldsymbol{x} + \boldsymbol{x}^T\Sigma^{-1}\boldsymbol{\mu} + const,$$

which is the sum of a quadratic term in $\boldsymbol{x}$, a linear term in $\boldsymbol{x}$ and a constant term *const* which means all terms that do not depend on $\boldsymbol{x}$. The aim of this remark is that whenever we see such a quadratic form, we can easily "read off" the covariance matrix Σ and the mean vector $\boldsymbol{\mu}$. This concept is known under the name "completing the squares".

Applying this procedure to (2.1), where we regard $\boldsymbol{x}_b$ as a constant, we can easily identify the quadratic term in $\boldsymbol{x}_a$ as

$$-\frac{1}{2}\boldsymbol{x}_a^T\boldsymbol{\Lambda}_{aa}\boldsymbol{x}_a \implies \Sigma_{a|b} = \boldsymbol{\Lambda}_{aa}^{-1}.$$

Moreover, all linear terms involving $\boldsymbol{x}_a$ in (2.1) are given by

$$\boldsymbol{x}_a^T\left(\boldsymbol{\Lambda}_{aa}\boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{ab}(\boldsymbol{x}_b - \boldsymbol{\mu}_b)\right) \implies \boldsymbol{\mu}_{a|b} = \boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{aa}^{-1}\boldsymbol{\Lambda}_{ab}(\boldsymbol{x}_b - \boldsymbol{\mu}_b).$$

Consider the following identity for the inverse of a block-matrix:

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} = \begin{pmatrix} M & -MBD^{-1} \\ -D^{-1}CM & D^{-1} + D^{-1}CMBD^{-1} \end{pmatrix},$$

where the matrix $M = (A - BD^{-1}C)^{-1}$ is the so-called Schur-complement. Applying this to the definition of the matrix $\boldsymbol{\Lambda}_{aa}$, we get that

$$\boldsymbol{\Lambda}_{aa} = (\Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba})^{-1}, \quad \boldsymbol{\Lambda}_{ab} = -(\Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba})^{-1}\Sigma_{ab}\Sigma_{bb}^{-1}.$$

From this is follows that the condition Gaussian has a mean vector and a covariance matrix given by

$$\boldsymbol{\mu}_{a|b} = \boldsymbol{\mu}_a + \Sigma_{ab}\Sigma_{bb}^{-1}(\boldsymbol{x}_b - \boldsymbol{\mu}_b), \quad \Sigma_{a|b} = \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba}.$$

Note that the mean is linear in $\boldsymbol{x}_b$ but the covariance is independent of $\boldsymbol{x}_b$.

2.6 Marginal Gaussian

Next, we will compute the marginal distribution $p(\boldsymbol{x}_a)$ which will again turn out to be a Gaussian of the form $\mathcal{N}(\boldsymbol{x}_a|\mu_a, \Sigma_a)$. In general, the marginal distribution over one variable is computed by "integrating out" the other variable, that is

$$p(\boldsymbol{x}_a) = \int p(\boldsymbol{x}_a, \boldsymbol{x}_b)\, \mathrm{d}\boldsymbol{x}_b.$$

We again write down the expanded quadratic form of the joint probability distribution

$$\begin{aligned} -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) = &-\frac{1}{2}(\boldsymbol{x}_a - \boldsymbol{\mu}_a)^T\boldsymbol{\Lambda}_{aa}(\boldsymbol{x}_a - \boldsymbol{\mu}_a) \\ &-\frac{1}{2}(\boldsymbol{x}_a - \boldsymbol{\mu}_a)^T\boldsymbol{\Lambda}_{ab}(\boldsymbol{x}_b - \boldsymbol{\mu}_b) \\ &-\frac{1}{2}(\boldsymbol{x}_b - \boldsymbol{\mu}_b)^T\boldsymbol{\Lambda}_{ba}(\boldsymbol{x}_a - \boldsymbol{\mu}_a) \\ &-\frac{1}{2}(\boldsymbol{x}_b - \boldsymbol{\mu}_b)^T\boldsymbol{\Lambda}_{bb}(\boldsymbol{x}_b - \boldsymbol{\mu}_b). \end{aligned}$$

Since our aim is now to integrate out the $\boldsymbol{x}_b$ variable, we will rewrite the above quadratic form in terms of a quadratic function of $\boldsymbol{x}_b$.

$$-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) = -\frac{1}{2}\boldsymbol{x}_b^T\Lambda_{bb}\boldsymbol{x}_b + \boldsymbol{x}_b^T\underbrace{\left(\Lambda_{bb}\boldsymbol{\mu}_b - \Lambda_{ba}(\boldsymbol{x}_a-\boldsymbol{\mu}_a)\right)}_{=\boldsymbol{m}} + const = -\frac{1}{2}\boldsymbol{x}_b^T\Lambda_{bb}\boldsymbol{x}_b + \boldsymbol{x}_b^T\boldsymbol{m} + const,$$

where *const* again means function that do not depend on $\boldsymbol{x}_b$. By neglecting the constant terms, the remaining two terms can be rewritten as standard quadratic form used inside a Gaussian plus another constant term not depending on $\boldsymbol{x}_b$:

$$-\frac{1}{2}\boldsymbol{x}_b^T\Lambda_{bb}\boldsymbol{x}_b + \boldsymbol{x}_b^T\boldsymbol{m} = -\frac{1}{2}(\boldsymbol{x}_b - \Lambda_{bb}^{-1}\boldsymbol{m})^T\Lambda_{bb}(\boldsymbol{x}_b - \Lambda_{bb}^{-1}\boldsymbol{m}) + \frac{1}{2}\boldsymbol{m}^T\Lambda_{bb}^{-1}\boldsymbol{m}.$$

As the first term is the quadratic form of a standard Gaussian, we can easily integrate it out so that the remaining terms are given by

$$\frac{1}{2}\boldsymbol{m}^T\Lambda_{bb}^{-1}\boldsymbol{m} - \frac{1}{2}\boldsymbol{x}_a^T\Lambda_{aa}\boldsymbol{x}_a + \boldsymbol{x}_a^T(\Lambda_{aa}\boldsymbol{\mu}_a + \Lambda_{ab}\boldsymbol{\mu}_b) + const =$$

$$\frac{1}{2}(\Lambda_{bb}\boldsymbol{\mu}_b - \Lambda_{ba}(\boldsymbol{x}_a-\boldsymbol{\mu}_a))^T\Lambda_{bb}^{-1}(\Lambda_{bb}\boldsymbol{\mu}_b - \Lambda_{ba}(\boldsymbol{x}_a-\boldsymbol{\mu}_a)) - \frac{1}{2}\boldsymbol{x}_a^T\Lambda_{aa}\boldsymbol{x}_a + \boldsymbol{x}_a^T(\Lambda_{aa}\boldsymbol{\mu}_a + \Lambda_{ab}\boldsymbol{\mu}_b) + const =$$

$$-\frac{1}{2}\boldsymbol{x}_a^T(\Lambda_{aa} - \Lambda_{ab}\Lambda_{bb}^{-1}\Lambda_{ba})\boldsymbol{x}_a + \boldsymbol{x}_a^T(\Lambda_{aa} - \Lambda_{ab}\Lambda_{bb}^{-1}\Lambda_{ba})\boldsymbol{\mu}_a + const,$$

where again *const* is a placeholder for functions not depending on $\boldsymbol{x}_a$. As before, we can "read off" the covariance matrix and the mean vector. They are given by (again invoking the Schur complement).

$$\Sigma_a = \left(\Lambda_{aa} - \Lambda_{ab}\Lambda_{bb}^{-1}\Lambda_{ba}\right)^{-1} = \Sigma_{aa}, \quad \boldsymbol{\mu}_a = \boldsymbol{\mu}_a.$$

It follows that the marginal Gaussian has mean $\boldsymbol{\mu}_a$ and covariance matrix Σ_{aa}.

2.7 Maximum likelihood for the Gaussian

Suppose, we have given i.i.d training examples $\boldsymbol{X} = \{\boldsymbol{x}_n\}_{n=1}^N$, $x_n \in \mathbb{R}^D$ sampled from a multivariate Gaussian

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

with unknown parameters $\boldsymbol{\mu}$ and Σ. The idea of maximum likelihood is write down the joint likelihood function over the dataset and maximizing the likelihood function with respect to the mean vector $\boldsymbol{\mu}$ and the covariance matrix Σ.

$$\max_{\boldsymbol{\mu},\Sigma}\left\{p(\boldsymbol{X}|\boldsymbol{\mu},\Sigma) := \prod_{n=1}^N p(\boldsymbol{x}_n|\boldsymbol{\mu},\Sigma)\right\}.$$

In order to simplify the maximization problem, we maximize instead the log-likelihood function which is given by (by ignoring constant terms)

$$\log p(\boldsymbol{X}|\boldsymbol{\mu}, \Sigma) = -\frac{N}{2}\log|\Sigma| - \frac{1}{2}\sum_{n=1}^{N}(\boldsymbol{x}_n - \boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}) + const.$$

Let us first compute the optimal $\boldsymbol{\mu}$ vector. For this we take the derivative of the log-likelihood function with respect to $\boldsymbol{\mu}$ and set it to zero:

$$\sum_{n=1}^{N}\Sigma^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}) = 0.$$

Since Σ is assumed to positive definite we can multiply the whole equation from left by Σ and we easily obtain that

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n.$$

Next we compute the derivative with respect to Σ. In order to do so we first rewrite our objective function a little bit. Since both Σ and Σ^{-1} appear in the objective we apply the following well-known identity:

$$|A| = \frac{1}{|A^{-1}|}.$$

Hence our objective function is given by (by removing constant terms and multiplying by 2)

$$N\log|\Sigma^{-1}| - \sum_{n=1}^{N}(\boldsymbol{x}_n - \boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}).$$

Next we need the following identity:

$$(\boldsymbol{x}_n - \boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}) = \mathrm{tr}\left(\Sigma^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu})(\boldsymbol{x}_n - \boldsymbol{\mu})^T\right),$$

which yields the objective

$$N\log|\Sigma^{-1}| - \sum_{n=1}^{N}\mathrm{tr}\left(\Sigma^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu})(\boldsymbol{x}_n - \boldsymbol{\mu})^T\right) +$$

Finally we recall for A and B being two matrices the following differentiation rules:

$$\partial_A|A| = |A|A^{-1} \quad \text{(Jacobi's formula)}, \quad \text{and } \partial_A\,\mathrm{tr}\,(AB) = B^T.$$

Using these rules, we obtain for the gradient with respect to Σ^{-1}:

$$N\Sigma - \sum_{n=1}^{N}(\boldsymbol{x}_n - \boldsymbol{\mu})(\boldsymbol{x}_n - \boldsymbol{\mu})^T = 0,$$

and hence

$$\Sigma = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}_n - \boldsymbol{\mu})(\boldsymbol{x}_n - \boldsymbol{\mu})^T$$

2.8 The Student's t distribution

Let us consider a univariate Gaussian distribution with mean μ and precision τ:

$$\mathcal{N}(x|\mu,\tau^{-1}) = \left(\frac{\tau}{2\pi}\right)^{\frac{1}{2}} \exp\left(-\frac{\tau}{2}(x-\mu)^2\right).$$

The correct conjugate prior for the precision τ of the univariate Gaussian is given by the gamma distribution

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

where a, b are parameters and $\Gamma(a)$ is the gamma function defined by

$$\Gamma(a) = \int_0^\infty z^{a-1} \exp(-z)\, \mathrm{d}z.$$

We now consider a joint density of the univariate Gaussian together with the gamma distribution

$$p(x,\tau|\mu,a,b) = \mathcal{N}(x|\mu,\tau^{-1}) Gam(\tau|a,b).$$

In order to get the marginal distribution $p(x|\mu,a,b)$, we need to apply the sum formula, which means integrating out the precision:

$$\begin{aligned} p(x|\mu,a,b) &= \int_0^\infty \mathcal{N}(x|\mu,\tau^{-1}) Gam(\tau|a,b)\, \mathrm{d}\tau \\ &= \int_0^\infty \left(\frac{\tau}{2\pi}\right)^{\frac{1}{2}} \exp\left(-\frac{\tau}{2}(x-\mu)^2\right) \frac{1}{\Gamma(a)} b^a \tau^{a-1} \exp(-b\tau)\, \mathrm{d}\tau \\ &= \frac{b^a}{(2\pi)^{\frac{1}{2}}\Gamma(a)} \int_0^\infty \tau^{a-\frac{1}{2}} \exp\left(-\tau\left(b + \frac{(x-\mu)^2}{2}\right)\right) \mathrm{d}\tau. \end{aligned}$$

In order to solve the integral, we perform the following change of variables

$$z = \tau\left(b + \frac{(x-\mu)^2}{2}\right), \qquad \mathrm{d}z = \left(b + \frac{(x-\mu)^2}{2}\right) \mathrm{d}\tau.$$

Hence, the integral becomes

$$\frac{b^a}{(2\pi)^{\frac{1}{2}}\Gamma(a)} \left(b + \frac{(x-\mu)^2}{2}\right)^{-a-\frac{1}{2}} \underbrace{\int_0^\infty z^{a-\frac{1}{2}} \exp(-z)\, \mathrm{d}z}_{=\Gamma(a+\frac{1}{2})}.$$

Finally we let $\nu = 2a$ denote the "degrees of freedom" and $\lambda = a/b$ the "variance". The final distribution , which is called the "student's t distribution, becomes

$$p(x|\mu,\lambda,\nu) = \frac{\Gamma(\frac{\nu}{2}+\frac{1}{2})}{\Gamma(\frac{\nu}{2})} \left(\frac{\lambda}{\pi\nu}\right)^{\frac{1}{2}} \left(1 + \frac{\lambda(x-\mu)^2}{\nu}\right)^{-\frac{\nu}{2}-\frac{1}{2}}.$$

3 Linear Regression

3.1 Maximum likelihood and least squares solution for linear regression

Let us consider a linear regression model of the form

$$y(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}),$$

where $\boldsymbol{w} \in \mathbb{R}^M$ is the parameter vector (including bias component) and $\boldsymbol{\phi}(\boldsymbol{x}) \in \mathbb{R}^M$ are the features (again containing a bias) computed from $\boldsymbol{x} \in \mathbb{R}^D$. Furthermore, we consider the case of zero-mean Gaussian noise that is function $y(\boldsymbol{x}, \boldsymbol{w})$ with additive Gaussian noise, that is the observed targets t are given by the deterministic function $y(\boldsymbol{x}, \boldsymbol{w})$ plus some zero-mean Gaussian noise $\eta \sim \mathcal{N}(0, \beta^{-1})$ with precision β.

$$t = y(\boldsymbol{x}, \boldsymbol{w}) + \eta.$$

In other words, the target values t can be seen as being drawn from a Gaussian distribution

$$t(\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}), \beta^{-1}).$$

We now consider a data set of inputs $\boldsymbol{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$ with corresponding target values $\boldsymbol{t} = \{t_1, ..., t_N\}$. Since the target values are drawn independently from a Gaussian distribution, the likelihood function is given by

$$p(\boldsymbol{t}|X, \boldsymbol{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n), \beta^{-1}).$$

As usual, we consider the log-likelihood function which is given by

$$\log p(\boldsymbol{t}|\boldsymbol{w}, \beta) = \frac{N}{2} \log \beta - \frac{N}{2} \log(2\pi) - \beta E_D(\boldsymbol{w}), \quad E_D(\boldsymbol{w}) = \frac{1}{2} \sum_{n=1}^{N} (t_n - \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n))^2,$$

where $E_D(\boldsymbol{w})$ is the sum of squares error function.

Let us first maximize the log-likelihood function with respect to the the precision β. Computing the derivative wrt. β and setting it to zero gives for the variance β^{-1}.

$$\frac{N}{2} \beta^{-1} - E_D(\boldsymbol{w}) = 0 \implies \beta_{ML}^{-1} = \frac{2}{N} E_D(\boldsymbol{w}).$$

Next, in order to compute the maximum likelihood parameter vector $\boldsymbol{w}$, we see that we equivalently need to minimize the sum of squares error function

$$E_D(\boldsymbol{w}) = \frac{1}{2} \sum_{n=1}^{N} (t_n - \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n))^2.$$

In order to simplify the computation of the gradient with respect to $\boldsymbol{w}$, we introduce

$$\Phi = \begin{pmatrix} \boldsymbol{\phi}(x_1)^T \\ \vdots \\ \boldsymbol{\phi}(\boldsymbol{x}_N)^T \end{pmatrix}, \quad \boldsymbol{t} = \begin{pmatrix} t_1 \\ \vdots \\ t_N \end{pmatrix}.$$

Hence the sum of squares error function can be written as a squared ℓ_2 norm

$$E_D(\boldsymbol{w}) = \frac{1}{2}\|\Phi\boldsymbol{w} - \boldsymbol{t}\|_2^2.$$

Now, computing the gradient with respect to $\boldsymbol{w}$ and setting it to zero gives

$$\nabla E_D(\boldsymbol{w}) = \Phi^T(\Phi\boldsymbol{w} - \boldsymbol{t}) = 0 \implies \boldsymbol{w}_{ML} = (\Phi^T\Phi)^{-1}\Phi^T\boldsymbol{t} = \Phi^\dagger\boldsymbol{t},$$

where $\Phi^\dagger$ denotes the well-known Moore-Penrose pseudo inverse, which can be computed, e.g. using the singular value decomposition (SVD).

Finally, let us have a closer look onto the bias parameter w_0. For this, we rewrite the sum of squares error function as

$$E_D(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}(t_n - w_0 - \sum_{j=1}^{M-1} w_j\phi_j(\boldsymbol{x}_n))^2.$$

Computing the derivative (now only with respect to w_0) and setting it to zero yields

$$w_0 = \bar{t} - \sum_{j=1}^{M-1} w_j\bar{\phi}_j, \quad \bar{t} = \frac{1}{N}\sum_{n=1}^{N} t_n, \quad \bar{\phi}_j = \frac{1}{N}\sum_{n=1}^{N}\phi_j(\boldsymbol{x}_n).$$

Hence , we see that the role of the bias parameter w_0 is to compensate for the error between the average target value $\bar{t}$ and the value predicted by the average features $\bar{\phi}_j$.

3.2 Bias-variance decomposition

Let us recall the expected quadratic loss function for regression (see above) defined by

$$\mathbb{E}[L] = \int\int (y(\boldsymbol{x}) - t)^2 p(\boldsymbol{x}, t)\,\mathrm{d}t\,\mathrm{d}\boldsymbol{x}.$$

We have already shown that this function can be reformulated as

$$\mathbb{E}[L] = \int (y(\boldsymbol{x}) - h(\boldsymbol{x}))^2 p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} + \underbrace{\int (h(\boldsymbol{x}) - t)^2 p(\boldsymbol{x}, t)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t}_{noise},$$

where we denote by

$$h(\boldsymbol{x}) = \mathbb{E}[t|\boldsymbol{x}] = \int t p(t|\boldsymbol{x})\,\mathrm{d}t,$$

the conditional expectation or regression function. Recall that the second term in the expected quadratic loss function is independent of $y(\boldsymbol{x})$ and hence captures the intrinsic noise in the training data. It also defines a lower bound to the minimum of the expected quadratic loss, which is obtained if $y(\boldsymbol{x}) = h(\boldsymbol{x})$. This is however only possible if we had an infinite amount of data.

Here we want to look more closely at the first expression and show that it can be further decomposed into two expressions, one of which can be linked to the bias of the model and the second to the variance of the model.

We assume that we have given training data, which can be partitioned into an ensemble of data sets $\mathcal{D}$, each of size N. For each data set $\mathcal{D}$, we can run a learning algorithm (e.g. maximum likelihood) in order to obtain a particular regression function $y(\boldsymbol{x};\mathcal{D})$. Clearly, we would be interested in comparing this particular regressor to the expected regressor $\mathbb{E}_D[y(\boldsymbol{x};\mathcal{D}]$. In order to do this, we rewrite the integrand of the first term of the expected quadratic loss functions as

$$\begin{aligned}(y(\boldsymbol{x};\mathcal{D})-h(\boldsymbol{x}))^2 =&(y(\boldsymbol{x};\mathcal{D}) \underbrace{-\mathbb{E}_D[y(\boldsymbol{x};\mathcal{D}]+\mathbb{E}_D[y(\boldsymbol{x};\mathcal{D}]}_{=0} -h(\boldsymbol{x}))^2\\ =&(y(\boldsymbol{x};\mathcal{D})-\mathbb{E}_D[y(\boldsymbol{x};\mathcal{D}])^2+(\mathbb{E}_D[y(\boldsymbol{x};\mathcal{D}]-h(\boldsymbol{x}))^2\\ &+2(y(\boldsymbol{x};\mathcal{D})-\mathbb{E}_D[y(\boldsymbol{x};\mathcal{D}])(\mathbb{E}_D[y(\boldsymbol{x};\mathcal{D}]-h(\boldsymbol{x})).\end{aligned}$$

Now, taking the expectation of the expanded expression with respect to $\mathcal{D}$, one obtains

$$\begin{aligned}&\mathbb{E}_{\mathcal{D}}\left[(y(\boldsymbol{x};\mathcal{D})-\mathbb{E}_D[y(\boldsymbol{x};\mathcal{D}])^2+(\mathbb{E}_D[y(\boldsymbol{x};\mathcal{D}]-h(\boldsymbol{x}))^2\right.\\ &\left.+2(y(\boldsymbol{x};\mathcal{D})-\mathbb{E}_D[y(\boldsymbol{x};\mathcal{D}])(\mathbb{E}_D[y(\boldsymbol{x};\mathcal{D}]-h(\boldsymbol{x}))\right]=\\ &\mathbb{E}_{\mathcal{D}}\left[(y(\boldsymbol{x};\mathcal{D})-\mathbb{E}_D[y(\boldsymbol{x};\mathcal{D}])^2\right]+\mathbb{E}_{\mathcal{D}}\left[(\mathbb{E}_D[y(\boldsymbol{x};\mathcal{D}]-h(\boldsymbol{x}))^2\right]+\\ &2\mathbb{E}_{\mathcal{D}}\left[(y(\boldsymbol{x};\mathcal{D})-\mathbb{E}_D[y(\boldsymbol{x};\mathcal{D}])(\mathbb{E}_D[y(\boldsymbol{x};\mathcal{D}]-h(\boldsymbol{x}))\right]=\\ &\mathbb{E}_{\mathcal{D}}\left[(y(\boldsymbol{x};\mathcal{D})-\mathbb{E}_D[y(\boldsymbol{x};\mathcal{D}])^2\right]+(\mathbb{E}_D[y(\boldsymbol{x};\mathcal{D}]-h(\boldsymbol{x}))^2+\\ &2\underbrace{(\mathbb{E}_{\mathcal{D}}[y(\boldsymbol{x};\mathcal{D})]-\mathbb{E}_D[y(\boldsymbol{x};\mathcal{D}])}_{=0}(\mathbb{E}_D[y(\boldsymbol{x};\mathcal{D}]-h(\boldsymbol{x}))=\\ &\underbrace{\mathbb{E}_{\mathcal{D}}\left[(y(\boldsymbol{x};\mathcal{D})-\mathbb{E}_D[y(\boldsymbol{x};\mathcal{D}])^2\right]}_{variance}+\underbrace{(\mathbb{E}_D[y(\boldsymbol{x};\mathcal{D}]-h(\boldsymbol{x}))^2}_{(bias)^2}.\end{aligned}$$

The first term is the expected value of the quadratic variation of the regression function $y(\boldsymbol{x};\mathcal{D})$ around its expected regression function $\mathbb{E}_D[y(\boldsymbol{x};\mathcal{D}]$, hence the variance. The second term is the quadratic distance of the expected regression function $\mathbb{E}_D[y(\boldsymbol{x};\mathcal{D}]$ from its true value $h(\boldsymbol{x})=\mathbb{E}[t|\boldsymbol{x}]$, which is the quadratic bias.

In summary, the expected quadratic loss function cane be decomposed in to three terms:

$$\boxed{\text{expected loss} = (bias)^2 + variance + noise,}$$

4 Linear classification

We consider here the following linear model for 2-class classification:

$$y(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0,$$

where $\boldsymbol{x}\in\mathbb{R}^D$ is the input vector, $\boldsymbol{w}\in\mathbb{R}^D$ is the parameter vector and w_0 is the bias parameter. An input vector $\boldsymbol{x}$ will be assigned to class C_1 if $y(\boldsymbol{x})\geq 0$ and it will be assigned to class C_2 if

$y(\boldsymbol{x}) < 0$. An important remark is that the classification does not change if we multiply both $\boldsymbol{w}$ and w_0 by a scalar $\alpha \in \mathbb{R}$.

We first note that the function $y(\boldsymbol{x}) = 0$ represents a hyperplane and that $\boldsymbol{w}$ is the normal vector of the hyperplane. To see this assume that we have two points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ lying on the hyperplane, hence $y(\boldsymbol{x}_1) = y(\boldsymbol{x}_2) = 0$. Then,

$$\boldsymbol{w}^T\boldsymbol{x}_1 + w_0 = \boldsymbol{w}^T\boldsymbol{x}_1 + w_0 = 0 \implies \boldsymbol{w}^T(\boldsymbol{x}_1 - \boldsymbol{x}_2) = 0,$$

from which we see that $\boldsymbol{w}$ is orthogonal to the vector $(\boldsymbol{x}_1 - \boldsymbol{x}_2)$ which lies on the hyperplane.

It is also very instructive to compute the projection of an arbitrary point to the hyperplane. Let us denote by $\boldsymbol{x}_\perp$ the orthogonal projection of a point $\boldsymbol{x}$ to the hyperplane. From simple geometric reasoning (see the picture in the slides), we see that

$$\boldsymbol{x}_\perp = \boldsymbol{x} + \mu\boldsymbol{w}.$$

In words, the point $\boldsymbol{x}_\perp$ is obtained by starting from $\boldsymbol{x}$ and going into the direction $\boldsymbol{w}$ times an unknown factor $\mu \in \mathbb{R}$. Depending on the orientation of $\boldsymbol{w}$, this factor could of course also be negative. The factor μ can be computed by observing that the point $\boldsymbol{x}_\perp$ has to lie on the hyperplane, that is

$$0 = \boldsymbol{w}^T\boldsymbol{x}_\perp + w_0 = \boldsymbol{w}^T\boldsymbol{x} + \mu \underbrace{\boldsymbol{w}^T\boldsymbol{w}}_{=\|\boldsymbol{w}\|_2^2} + w_0 = y(\boldsymbol{x}) + \mu\|\boldsymbol{w}\|_2^2.$$

From this equation, the parameter μ is computed as

$$\mu = -\frac{y(\boldsymbol{x})}{\|\boldsymbol{w}\|_2^2}.$$

Substituting this expression back in the original equation gives

$$\boldsymbol{x}_\perp = \boldsymbol{x} + \mu\boldsymbol{w} = \boldsymbol{x} - \frac{y(\boldsymbol{x})}{\|\boldsymbol{w}\|_2^2}\boldsymbol{w} = \boldsymbol{x} + r(\boldsymbol{x})\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2},$$

where we have defined the signed normal distance

$$r(\boldsymbol{x}) = \frac{-y(\boldsymbol{x})}{\|\boldsymbol{w}\|_2},$$

along the normal direction $\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}$. From this equation we can also easily compute the signed normal distance of the hyperplane from the origin by simply using $\boldsymbol{x} = 0$. It gives

$$r(0) = \frac{-w_0}{\|\boldsymbol{w}\|_2^2},$$

hence the bias parameter in some sense controls the distance of the hyperplane from the origin.

4.1 Fisher's linear discriminant function

Assume we are using a standard linear model for classification

$$y = \boldsymbol{w}^T \boldsymbol{x},$$

where we assume that the $\boldsymbol{w}$ and $\boldsymbol{x}$ vectors are such that they include a bias component. The idea of the Fisher linear discriminant function is to compute $\boldsymbol{w}$ such that the projection $\boldsymbol{w}^T\boldsymbol{x}$ allows for an easy separation. We assume that we have given training data $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$ with $N = N_1 + N_2$, where N_1 refers to the number of samples from the first class C_1 and N_2 to the number of samples from the second class C_2.

The first idea is that the projections of the means of the two classes

$$m_k = \boldsymbol{w}^T \boldsymbol{m}_k, \quad \boldsymbol{m}_k = \frac{1}{N_k} \sum_{n \in C_k} \boldsymbol{x}_n, \quad k = 1, 2,$$

are well separated, that is we could try to maximize

$$\max_{\boldsymbol{w}} m_2 - m_1 = \max_{\boldsymbol{w}} \boldsymbol{w}^T (\boldsymbol{m}_2 - \boldsymbol{m}_1), \quad \text{s.t. } \|\boldsymbol{m}\| = 1$$

where the constraint is introduced in order to make the optimization problem well-posed. The Lagrangian function of this constrained optimization problem is given by

$$\mathcal{L}(\boldsymbol{w}, \lambda) = \boldsymbol{w}^T (\boldsymbol{m}_2 - \boldsymbol{m}_1) + \lambda(\|\boldsymbol{m}\|^2 - 1),$$

where $\lambda \in \mathbb{R}$ is the Lagrange multiplier. From the Lagrange-multiplier theorem we know that there exists a Lagrange multiplier λ^* such that

$$\nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \lambda^*) = (\boldsymbol{m}_2 - \boldsymbol{m}_1) + 2\lambda^* \boldsymbol{w} = 0.$$

Hence the vector $\boldsymbol{w}$ that maximizes the distance of the means is given by

$$\boldsymbol{w} = -\frac{1}{2\lambda^*}(\boldsymbol{m}_2 - \boldsymbol{m}_1) \propto (\boldsymbol{m}_2 - \boldsymbol{m}_1).$$

where the last step indicates that we are usually not interested in the length of $\boldsymbol{w}$ but only its direction.

This solution of $\boldsymbol{w}$ leads to a maximal separation of the means of the two classes but it does not take care of the intra-class variances of the two classes. This could lead to a projection with quite overlapping classes. In order to overcome this problem, the final idea of the Fisher linear discriminant is to maximize instead

$$J(\boldsymbol{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2},$$

where

$$s_k^2 = \sum_{n \in C_k} (\boldsymbol{w}^T \boldsymbol{x}_n - m_k)^2, \quad k = 1, 2,$$

denote the intra-class co-variance. Hence, the Fisher criterion maximizes the distance between the means but at the same time minimizes (maximizes the inverse of) the sum of the intra-class variances. next, we rewrite the Fisher criterion such that it is more explicit in the hyperplane parameters $\boldsymbol{w}$. For this we observe that

$$(m_2 - m_1)^2 = (m_2 - m_1)(m_2 - m_1)^T = \boldsymbol{w}^T(\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{w}^T(\boldsymbol{m}_2 - \boldsymbol{m}_1))^T = \boldsymbol{w}^T \underbrace{(\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T}_{S_B} \boldsymbol{w},$$

where S_B denotes the between-class co-variance. Similarly,

$$\begin{aligned}
s_1^2 + s_2^2 &= \sum_{n \in C_1} (\boldsymbol{w}^T \boldsymbol{x}_n - m_1)^2 + \sum_{n \in C_2} (\boldsymbol{w}^T \boldsymbol{x}_n - m_2)^2 \\
&= \sum_{n \in C_1} (\boldsymbol{w}^T (\boldsymbol{x}_n - \boldsymbol{m}_1))^2 + \sum_{n \in C_2} (\boldsymbol{w}^T (\boldsymbol{x}_n - \boldsymbol{m}_2))^2 \\
&= \sum_{n \in C_1} \boldsymbol{w}^T (\boldsymbol{x}_n - \boldsymbol{m}_1)(\boldsymbol{w}^T (\boldsymbol{x}_n - \boldsymbol{m}_1))^T + \sum_{n \in C_2} (\boldsymbol{w}^T (\boldsymbol{x}_n - \boldsymbol{m}_2))(\boldsymbol{w}^T (\boldsymbol{x}_n - \boldsymbol{m}_2))^T \\
&= \boldsymbol{w}^T \underbrace{\left(\sum_{n \in C_1} (\boldsymbol{x}_n - \boldsymbol{m}_1)(\boldsymbol{x}_n - \boldsymbol{m}_1)^T + \sum_{n \in C_2} (\boldsymbol{x}_n - \boldsymbol{m}_2)(\boldsymbol{x}_n - \boldsymbol{m}_2)^T \right)}_{S_W} \boldsymbol{w},
\end{aligned}$$

where S_W denotes the sum-of-within-classes co-variance. With this definitions, the Fisher criterion now reads as

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T S_B \boldsymbol{w}}{\boldsymbol{w}^T S_W \boldsymbol{w}}.$$

Finally, let us give a solution to this criterion. First, we compute the gradient of $J(\boldsymbol{w})$ and set it to zero

$$\nabla J(\boldsymbol{w}) = \frac{S_B \boldsymbol{w}}{\boldsymbol{w}^T S_W \boldsymbol{w}} - \frac{\boldsymbol{w}^T S_B \boldsymbol{w}}{(\boldsymbol{w}^T S_W \boldsymbol{w})^2} S_W \boldsymbol{w} = 0.$$

Multiplying the equation with $(\boldsymbol{w}^T S_W \boldsymbol{w})^2$ and rearranging the terms gives

$$(\boldsymbol{w}^T S_W \boldsymbol{w}) S_B \boldsymbol{w} = (\boldsymbol{w}^T S_B \boldsymbol{w}) S_W \boldsymbol{w},$$

Now, we observe that

$$S_B \boldsymbol{w} = (\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T \boldsymbol{w} = \alpha(\boldsymbol{m}_2 - \boldsymbol{m}_1),$$

for some $\alpha \in \mathbb{R}$. Hence

$$\underbrace{\alpha \frac{(\boldsymbol{w}^T S_W \boldsymbol{w})}{(\boldsymbol{w}^T S_B \boldsymbol{w})}}_{\beta} (\boldsymbol{m}_2 - \boldsymbol{m}_1) = S_W \boldsymbol{w}.$$

Again, since we are not interested in the length of $\boldsymbol{w}$ we can assume without loss of generality that $\beta = 1$. Hence Fisher's linear discriminant is given by

$$\boldsymbol{w} \propto S_W^{-1} (\boldsymbol{m}_2 - \boldsymbol{m}_1).$$

4.2 Generative model using Gaussian class conditionals

We consider the posterior distribution for 2-class classification based on Bayes' theorem:

$$p(C_1|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_1)p(C_1)}{p(\boldsymbol{x}|C_1)p(C_1) + p(\boldsymbol{x}|C_2)p(C_2)} = \frac{1}{1+\exp(-a)} = \sigma(a),$$

where $\sigma(a)$ is the logistic sigmoid function and

$$a = \log \frac{p(\boldsymbol{x}|C_1)p(C_1)}{p(\boldsymbol{x}|C_2)p(C_2)},$$

is the log-ratio. We shall assume that the class conditionals $p(\boldsymbol{x}|C_1)$ are given by Gaussian distributions, that is

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

Hence , the log-ratio becomes

$$\begin{aligned} a &= -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_2)^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2) + \log\frac{p(C_1)}{p(C_2)} \\ &= x^T \underbrace{\Sigma^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)}_{\boldsymbol{w}} \underbrace{-\frac{1}{2}\boldsymbol{\mu}_1^T\Sigma^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^T\Sigma^{-1}\boldsymbol{\mu}_2 + \log\frac{p(C_1)}{p(C_2)}}_{w_0}. \end{aligned}$$

Observe that the final sigmoid function is linear in the parameter vector $\boldsymbol{w}$.

4.3 Logistic regression

Assume, we have given a data set $(\Phi, \boldsymbol{t}) = \{\phi_n, t_n\}$, $n = 1, ..., N$, with $t_n \in \{0, 1\}$ and $\phi_n = \phi(\boldsymbol{x}_n)$. As usual, the likelihood function is given by

$$p(\boldsymbol{t}, \Phi|\boldsymbol{w}) = \prod_{n=1}^{N} y_n^{t_n}(1-y_n)^{1-t_n},$$

where $y_n = \sigma(\boldsymbol{w}^T\phi_n)$ is the predictive function and $\sigma(a) = 1/(1+\exp(-a)$ is the logistic sigmoid function. Taking the negative of the logarithm of the likelihood function yields the cross-entropy error function

$$E(\boldsymbol{w}) = -\log p(\boldsymbol{t}, \Phi|\boldsymbol{w}) = -\sum_{n=1}^{N} t_n \log y_n + (1-t_n)\log(1-y_n).$$

The cross-entropy error function represents a convex optimization problem. This is easily seen from the fact that it is a non-negatively weighted sum of terms which are given by the composition of the negative logarithm (which is a convex function) with a linear function.

In order to learn the parameter vector $\boldsymbol{w}$ on the data, we consider a gradient method. Therefore, we need to compute the gradient of the cross-entropy error function with respect to the parameter vector $\boldsymbol{w}$. First we note that

$$\sigma'(a) = \frac{\exp(-a)}{(1+\exp(-a))^2} = \sigma(a)\frac{\exp(-a)}{1+\exp(-a)} = \sigma(a)\frac{\exp(-a)+1-1}{1+\exp(-a)} = \sigma(a)(1-\sigma(a)).$$

Then, using $\nabla_{\boldsymbol{w}} y_n = \sigma'(\boldsymbol{w}^T\phi_n)\phi_n = y_n(1-y_n)\phi_n$,

$$\begin{aligned}
\nabla_{\boldsymbol{w}} E(\boldsymbol{w}) &= -\sum_{n=1}^{N} t_n \frac{1}{y_n}\sigma'(\boldsymbol{w}^T\phi_n)\phi_n + (1-t_n)\frac{1}{1-y_n}(-1)\sigma'(\boldsymbol{w}^T\phi_n)\phi_n \\
&= -\sum_{n=1}^{N} t_n \frac{1}{y_n} y_n(1-y_n)\phi_n + (1-t_n)\frac{1}{1-y_n}(-1)\, y_n(1-y_n)\phi_n \\
&= -\sum_{n=1}^{N} t_n(1-y_n)\phi_n + (1-t_n)(-1)\, y_n\phi_n \\
&= \sum_{n=1}^{N} t_n(y_n-1)\phi_n + (1-t_n)\, y_n\phi_n \\
&= \sum_{n=1}^{N} (y_n - t_n)\phi_n.
\end{aligned}$$

4.4 Computation of the Lipschitz constant

In order to implement a gradient-based minimization, we need to know the Lipschitz constant L of the gradient. For example, in Nesterov's algorithm the step-size is chosen as $t = \frac{1}{L}$. The Lipschitz constant of the gradient $\nabla E(\boldsymbol{w})$ is defined such that

$$\|\nabla E(\boldsymbol{w}) - \nabla E(\boldsymbol{w}')\| \leq L\|\boldsymbol{w} - \boldsymbol{w}'\|, \quad \forall \boldsymbol{w}, \boldsymbol{w}'$$

Moreover, for twice continuously differentiable functions (such as the loss function on logistic regression on has also that the operator norm (largest singular value) of the Hessian matrix is upper-bounded by that Lipschitz constant, that is

$$\|\nabla^2 E(\boldsymbol{w})\| \leq L.$$

It turns out that the Lipschitz constant of the gradient of logistic regression can be computed easily by upper bounding the operator norm of its Hessian matrix, which is given by

$$\nabla^2_{\boldsymbol{w}} E(\boldsymbol{w}) = \sum_{n=1}^{N} \phi_n \sigma'(\boldsymbol{w}^T\phi_n)\phi_n^T.$$

From this it follows that the operator norm is upper bounded by

$$\|\nabla^2_{\boldsymbol{w}} E(\boldsymbol{w})\| \leq c\sigma_{\max}\left(\sum_{n=1}^{N} \phi_n\phi_n^T\right) = L,$$

where $c = \max_t \sigma'(t)$ and $\sigma_{\max}(A)$ denotes the largest singular value of A. It remains to compute the constant c, which is given by the largest possible value of the function $\sigma'(t) = \sigma(t)(1-\sigma(t))$. It is easy to see that this function reaches its maximum at $t = 0$ with value $c = \frac{1}{4}$, hence

$$\|\nabla^2_{\boldsymbol{w}} E(\boldsymbol{w})\| \leq \frac{1}{4}\sigma_{\max}\left(\sum_{n=1}^{N} \phi_n \phi_n^T\right) = L.$$

4.5 Multiclass logistic regression

Next, we consider an extension to multi-class classification. The predictive functions $y_{n,k}$ are given by the softmax function

$$y_{n,k} = \frac{\exp(\boldsymbol{w}_k^T \phi_n)}{\sum_{j=1}^{K} \exp(\boldsymbol{w}_j^T \phi_n)}.$$

Observe that by construction $y_{n,k} > 0$ and $\sum_{k=1}^{K} y_{n,k} = 1$. For each training example ϕ_n, we associate a target vector $t_n \in \{0,1\}^K$, using the usual 1-of-K encoding. For example if $K = 5$ and t_n corresponds to class $C_k = 3$, then $t_n = (0,0,1,0,0)^T$. The cross-entropy error function is given by

$$E(\boldsymbol{w}_1, ..., \boldsymbol{w}_K) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{n,k} \log y_{n,k}.$$

In order to minimize the cross-entropy error function, we need to compute the gradients

$$\nabla_{\boldsymbol{w}_l} E(\boldsymbol{w}_1, ..., \boldsymbol{w}_K), \quad l = 1, ..., K.$$

First we compute the gradients

$$\nabla_{\boldsymbol{w}_l} y_{n,k} = \begin{cases} \phi_n \left(\frac{\exp(\boldsymbol{w}_l^T \phi_n)}{\sum_{j=1}^{K} \exp(\boldsymbol{w}_j^T \phi_n)} - \frac{\exp(\boldsymbol{w}_l^T \phi_n)^2}{\left(\sum_{j=1}^{K} \exp(\boldsymbol{w}_j^T \phi_n)\right)^2} \right) = \phi_n \left(y_{n,l} - y_{n,l}^2 \right) & \text{if} \quad l = k, \\ \phi_n \left(-\frac{\exp(\boldsymbol{w}_k^T \phi_n) \exp(\boldsymbol{w}_l^T \phi_n)}{\left(\sum_{j=1}^{K} \exp(\boldsymbol{w}_j^T \phi_n)\right)^2} \right) = \phi_n \left(-y_{n,l}\, y_{n,k} \right) & \text{if} \quad l \neq k. \end{cases}$$

Therefore, the gradients of the cross-entropy error function are given by

$$\begin{aligned} \nabla_{\boldsymbol{w}_l} E(\boldsymbol{w}_1, ..., \boldsymbol{w}_K) &= -\sum_{n=1}^{N}\sum_{k\neq l} t_{n,k} \frac{1}{y_{n,k}} (\phi_n(-y_{n,l} y_{n,k})) + t_{n,l} \frac{1}{y_{n,k}} \phi_n (y_{n,l} - y_{n,l}^2) \\ &= -\sum_{n=1}^{N} \phi_n \Big(\sum_{k\neq l} t_{n,k}(-y_{n,l}) + t_{n,l}(1 - y_{n,l}) \Big) \\ &= \sum_{n=1}^{N} \phi_n \Big(y_{n,l} \underbrace{\sum_{k\neq l} t_{n,k}}_{1-t_{n,l}} + t_{n,l}(y_{n,l} - 1) \Big) = \sum_{n=1}^{N} \phi_n \left(y_{n,l}(1 - t_{n,l}) + t_{n,l}(y_{n,l} - 1) \right) \\ &= \sum_{n=1}^{N} \phi_n \left(y_{n,l} - t_{n,l} \right). \end{aligned}$$

The Lipschitz constant of the gardient for the multi-class logistic regression can be estimated in a similar way as above.

5 Kernel methods

5.1 Dual representation of regularized least squares

Let us consider the classical linear regression model based on the predictive linear function

$$y(\boldsymbol{x}) = \phi(\boldsymbol{x})^T \boldsymbol{w},$$

where $\phi(\boldsymbol{x}) \in \mathbb{R}^M$ is the feature vector and $\boldsymbol{w} \in \mathbb{R}^M$ is the parameter vector. The parameter vector $\boldsymbol{w}$ can be estimated from the training data consisting of input vectors $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$ and corresponding target vectors $\{t_1, ..., t_N\}$ by minimizing a regularized least squares error function

$$J(\boldsymbol{w}) = \frac{1}{2} \sum_{n=1}^{N} (\phi(\boldsymbol{x}_n)^T \boldsymbol{w} - t_n)^2 + \frac{\lambda}{2} \boldsymbol{w}^T \boldsymbol{w}.$$

Using a more compact notation, the regularized least squares model is equivalent to

$$J(\boldsymbol{w}) = \frac{1}{2} \|\Phi \boldsymbol{w} - \boldsymbol{t}\|_2^2 + \frac{\lambda}{2} \|\boldsymbol{w}\|_2^2,$$

where the rows of the matrix Φ are given by the features $\phi_n = \phi(\boldsymbol{x}_n)$ and the target vector $\boldsymbol{t} = (t_1, ..., t_N)^T$.

A severe computational problem arises, if the dimension M of the feature vectors ϕ_n is very high or even infinite dimensional. We now show that the regularized least squares model can be solved even in case ϕ_n is infinite dimensional based on a dual formulation and the kernel trick.

The dual formulation originates from the following fundamental transformation:

$$\frac{1}{2} \|\Phi \boldsymbol{w} - \boldsymbol{t}\|_2^2 = \max_{\boldsymbol{a}} (\Phi \boldsymbol{w} - \boldsymbol{t})^T \boldsymbol{a} - \frac{1}{2} \|\boldsymbol{a}\|^2,$$

where $\boldsymbol{a} \in \mathbb{R}^N$ is the so-called dual vector. The correctness of this transformation can be easily verified by computing the derivative of the right hand side with respect to $\boldsymbol{a}$, equating it with zero, solving for $\boldsymbol{a}$ and substituting it back. In convex analysis, this kind of transformation is known under the name Legendre-Fenchel transform and is of course not restricted to quadratic functions, only.

Next, we replace the first term in $J(\boldsymbol{w})$ with its Legendre-Fenchel transform to obtain the Lagrangian function $L(\boldsymbol{w}, \boldsymbol{a})$:

$$L(\boldsymbol{w}, \boldsymbol{a}) = (\Phi \boldsymbol{w} - \boldsymbol{t})^T \boldsymbol{a} - \frac{1}{2} \|\boldsymbol{a}\|^2 + \frac{\lambda}{2} \|\boldsymbol{w}\|_2^2.$$

The key observation is now, that $L(\boldsymbol{w}, \boldsymbol{a})$ can be minimized in closed form with respect to $\boldsymbol{w}$, indeed,

$$\nabla_{\boldsymbol{w}} L(\boldsymbol{w}, \boldsymbol{a}) = \Phi^T \boldsymbol{a} + \lambda \boldsymbol{w} = 0 \implies \boldsymbol{w} = -\frac{1}{\lambda} \Phi^T \boldsymbol{a},$$

which shows that the parameter vector $\boldsymbol{w}$ can be represented as a linear combination of the features ϕ_n. Substituting the expression for $\boldsymbol{w}$ back into the Lagrangian $L(\boldsymbol{w}, \boldsymbol{a})$ finally yields the dual problem

$$D(\boldsymbol{a}) = -\frac{1}{\lambda}(\Phi^T\boldsymbol{a})^T(\Phi^T\boldsymbol{a}) - \boldsymbol{t}^T\boldsymbol{a} - \frac{1}{2}\|\boldsymbol{a}\|^2 + \frac{1}{2\lambda}\|\Phi^T\boldsymbol{a}\|^2 = -\frac{1}{2\lambda}\|\Phi^T\boldsymbol{a}\|^2 - \boldsymbol{t}^T\boldsymbol{a} - \frac{1}{2}\|\boldsymbol{a}\|^2$$

Note, that while the original problem $J(\boldsymbol{w})$ was a minimization problem in $\boldsymbol{w}$, the dual problem $D(\boldsymbol{a})$ is now a maximization problem in $\boldsymbol{a}$ (note the maximization in the Legendre-Fenchel transform).

The dual problem can also be rewritten as the following equivalent (up to constants) minimization problem

$$\min_{\boldsymbol{a}} \frac{1}{2}\boldsymbol{a}^T \underbrace{\Phi\Phi^T}_{K} \boldsymbol{a} + \frac{\lambda}{2}\|\boldsymbol{a} + \boldsymbol{t}\|^2,$$

where $K \in \mathbb{R}^{N\times N}$ is the kernel matrix with $K_{n,m} = \phi(\boldsymbol{x}_n)^T\phi(\boldsymbol{x}_m) = k(\boldsymbol{x}_n, \boldsymbol{x}_m)$, where $k(\cdot,\cdot)$ is the kernel function. The optimal solution of the dual problem is given by

$$K\boldsymbol{a} + \lambda(\boldsymbol{a} + \boldsymbol{t}) = 0 \implies \boldsymbol{a} = -(I + K/\lambda)^{-1}\boldsymbol{t}.$$

Substituting this expression back into the expression for the parameter vector $\boldsymbol{w}$ yields

$$\boldsymbol{w} = -\frac{1}{\lambda}\Phi^T\boldsymbol{a} = \frac{1}{\lambda}\Phi^T(I + K/\lambda)^{-1}\boldsymbol{t} = \Phi^T(\lambda I + K)^{-1}\boldsymbol{t}.$$

Finally the predictive function $y(\boldsymbol{x})$ is computed as

$$y(\boldsymbol{x}) = \phi(\boldsymbol{x})^T\boldsymbol{w} = \phi(\boldsymbol{x})^T\Phi^T(\lambda I + K)^{-1}\boldsymbol{t} = \boldsymbol{k}(\boldsymbol{x})^T(\lambda I + K)^{-1}\boldsymbol{t},$$

where $\boldsymbol{k}(\boldsymbol{x}) = (k(\boldsymbol{x}, \boldsymbol{x}_1), ..., k(\boldsymbol{x}, \boldsymbol{x}_N))^T$. From the last equation is is evident that the feature vectors $\phi(\boldsymbol{x})$ never have to be constructed explicitly as all the inner products of the form $\phi(\boldsymbol{x})^T\phi(\boldsymbol{x}')$ can be replaced by its equivalent kernel $k(\boldsymbol{x}, \boldsymbol{x}')$. Hence the linear regression problem is also feasible in case of very large (possibly infinite) feature dimension M. Of course, this comes with the cost of inverting the matrix $(\lambda I + K)$, which is of size $N \times N$.

5.2 The Nadaraya-Watson model

Assume, we have given training data $\{\boldsymbol{x}_n, t_n\}$, $n = 1, 2, ..., N$. The idea of the Nadaraya-Watson model is to use a kernel density estimate (KDE) to express the joint probability

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

where $f(\boldsymbol{x}, t)$ is a kernel function and for each training example $(\boldsymbol{x}_n, t_n)$, there will be one such function centered around $(\boldsymbol{x}_n, t_n)$. Note that by definition of the kernel function, the joint probability is properly normalized.

Recall that the regression function $y(\boldsymbol{x})$ is computed as

$$y(\boldsymbol{x}) = \mathbb{E}[t|\boldsymbol{x}] = \int_{-\infty}^{\infty} tp(t|\boldsymbol{x})\,\mathrm{d}t = \int_{-\infty}^{\infty} t\frac{p(t,\boldsymbol{x})}{p(\boldsymbol{x})}\,\mathrm{d}t = \frac{\int_{-\infty}^{\infty} tp(t,\boldsymbol{x})\,\mathrm{d}t}{\int_{-\infty}^{\infty} p(t,\boldsymbol{x})\,\mathrm{d}t} = \frac{\sum_{n=1}^{N}\int_{-\infty}^{\infty} tf(\boldsymbol{x}-\boldsymbol{x}_n, t-t_n)\,\mathrm{d}t}{\sum_{n=1}^{N}\int_{-\infty}^{\infty} f(\boldsymbol{x}-\boldsymbol{x}_n, t-t_n)\,\mathrm{d}t}.$$

In order to simplify the last expression, we shall assume that the kernel function $f(\boldsymbol{x}, t)$ has zero mean in its second argument, that is

$$\int_{-\infty}^{\infty} tf(\boldsymbol{x}, t)\,\mathrm{d}t = 0,$$

for all $\boldsymbol{x}$. From this it follows that

$$\int_{-\infty}^{\infty} tf(\boldsymbol{x}-\boldsymbol{x}_n, t-t_n)\,\mathrm{d}t = \int_{-\infty}^{\infty} (s+t_n)f(\boldsymbol{x}-\boldsymbol{x}_n, s)\,\mathrm{d}s = \underbrace{\int_{-\infty}^{\infty} sf(\boldsymbol{x}-\boldsymbol{x}_n, s)\,\mathrm{d}s}_{=0} + t_n \underbrace{\int_{-\infty}^{\infty} f(\boldsymbol{x}-\boldsymbol{x}_n, s)\,\mathrm{d}s}_{g(\boldsymbol{x}-\boldsymbol{x}_n)}.$$

Combining the expressions, the regression function can be rewritten in the form of the Nadaraya-Watson model:

$$y(\boldsymbol{x}) = \frac{\sum_{n=1}^{N} t_n g(\boldsymbol{x}-\boldsymbol{x}_n)}{\sum_{n=1}^{N} g(\boldsymbol{x}-\boldsymbol{x}_n)} = \sum_{n=1}^{N} t_n k(\boldsymbol{x}, \boldsymbol{x}_n),$$

where

$$k(\boldsymbol{x}, \boldsymbol{x}_n) = \frac{g(\boldsymbol{x}-\boldsymbol{x}_n)}{\sum_{n=1}^{N} g(\boldsymbol{x}-\boldsymbol{x}_n)}, \quad g(\boldsymbol{x}) = \int_{-\infty}^{\infty} f(\boldsymbol{x}, t)\,\mathrm{d}t.$$

Note that by construction $\sum_{n=1}^{N} k(\boldsymbol{x}, \boldsymbol{x}_n) = 1$. The model not only gives a conditional expectation but also defines a full conditional distribution:

$$p(t|\boldsymbol{x}) = \frac{p(t,\boldsymbol{x})}{p(\boldsymbol{x})} = \frac{p(t,\boldsymbol{x})}{\int_{-\infty}^{\infty} p(t,\boldsymbol{x})\,\mathrm{d}t} = \frac{\sum_{n=1}^{N} f(\boldsymbol{x}-\boldsymbol{x}_n, t-t_n)}{\sum_{n=1}^{N}\int_{-\infty}^{\infty} f(\boldsymbol{x}-\boldsymbol{x}_n, t-t_n)\,\mathrm{d}t},$$

from which other expectations can be computed.

5.3 Gaussian process for regression

We consider a standard linear model $y_n = y(\boldsymbol{x}_n) = \boldsymbol{w}^T\phi(\boldsymbol{x})$ and assume the following additive noise model:

$$t_n = y_n + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \beta^{-1}).$$

From the noise model we can directly state the conditional density

$$p(t_n|y_n) = \mathcal{N}(t_n|y_n, \beta^{-1}).$$

We further let $\boldsymbol{t} = (t_1, ..., t_N)^T$ and $\boldsymbol{y} = (y_1, ..., y_N)^T$. The joint density function of $\boldsymbol{t}$ conditioned on $\boldsymbol{y}$ is given by

$$p(\boldsymbol{t}|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{t}|\boldsymbol{y}, \beta^{-1}I_N),$$

where I_N is the $N \times N$ identity matrix.

According to the definition of a Gaussian process we consider a Gaussian prior on $\boldsymbol{w}$ given by

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|0, \alpha^{-1}I),$$

which in turn induces a Gaussian prior on the functions $\boldsymbol{y}$ given by

$$p(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}|0, K), \quad K = \alpha^{-1}\Phi\Phi^T,$$

with $\Phi = (\phi(\boldsymbol{x}_1)^T, ..., \phi(\boldsymbol{x}_N)^T)^T$.

From the product rule, the joint and marginal probability distributions are given by

$$p(\boldsymbol{y}, t) = p(t|\boldsymbol{y})p(\boldsymbol{y}), \quad p(\boldsymbol{t}) = \int p(\boldsymbol{y}, t)\, \mathrm{d}\boldsymbol{y} = \int p(t|\boldsymbol{y})p(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y}.$$

Since both the prior and the conditional distributions are Gaussian, the marginal distribution can be computed explicitly using the following marginalization formulas:

$$\boxed{p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{y}, \Lambda^{-1}),\ p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{y}|A\boldsymbol{x} + \boldsymbol{b}, L^{-1}) \implies p(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}|A\boldsymbol{\mu} + \boldsymbol{b}, L^{-1} + AA^{-1}A^T).}$$

Applying this formula to our setting gives the following expression for the marginal distribution:

$$p(\boldsymbol{t}) = \mathcal{N}(\boldsymbol{t}|0, C), \quad C = \beta^{-1}I_N + K.$$

Having computed the matrix C, we now wish to predict a new target value t_{N+1} for a new input vector $\boldsymbol{x}_{N+1}$. That is, we would like to compute the predictive distribution $p(t_{n+1}|\boldsymbol{t})$. In order to compute the predictive distribution we first have to compute the joint distribution $p(\boldsymbol{t}, t_{N+1})$ and then compute the conditional distribution $p(t_{N+1}|\boldsymbol{t})$ using the formulas for partitioned Gaussian distributions.

From the previous computations, we see that the joint distribution is given by

$$p(\boldsymbol{t}, t_{N+1}) = \mathcal{N}(\boldsymbol{t}_{N+1}|0, C_{N+1}), \quad C_{N+1} = \begin{pmatrix} C & \boldsymbol{k} \\ \boldsymbol{k}^T & c \end{pmatrix},$$

where $\boldsymbol{t}_{n+1} = (\boldsymbol{t}, t_{N+1})^T$, $\boldsymbol{k} = (k(\boldsymbol{x}_1, \boldsymbol{x}_{N+1}), ..., k(\boldsymbol{x}_N, \boldsymbol{x}_{N+1}))^T$ and $c = k(\boldsymbol{x}_{N+1}, \boldsymbol{x}_{N+1}) + \beta^{-1}$.

We next recall the formula of the conditional of a partitioned Gaussian distribution:

$$\boxed{\begin{aligned} &p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\mu, \Sigma), \quad \boldsymbol{x} = (\boldsymbol{x}_a, \boldsymbol{x}_b)^T, \quad \boldsymbol{\mu} = (\boldsymbol{\mu}_a, \boldsymbol{\mu}_b)^T, \quad \Sigma = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix} \\ &\implies p(\boldsymbol{x}_a|\boldsymbol{x}_b) = \mathcal{N}(\boldsymbol{x}_a|\boldsymbol{\mu}_a + \Sigma_{ab}\Sigma_{bb}^{-1}(\boldsymbol{x}_b - \boldsymbol{\mu}_b), \Sigma_{a|b}, \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba}). \end{aligned}}$$

Applying this formulas to our setting, we obtain as a result that

$$p(t_{N+1}|\boldsymbol{t}) = \mathcal{N}(t_{N+1}|\underbrace{\boldsymbol{k}^T C_N^{-1}\boldsymbol{t}}_{m(\boldsymbol{x})}, \underbrace{c - \boldsymbol{k}^T C_N^{-1}\boldsymbol{k}}_{\sigma^2(\boldsymbol{x})}).$$

5.4 The support vector machine

Let us consider the classical two-class classification problem using a linear model

$$y(\boldsymbol{x}) = \boldsymbol{w}^T \phi(\boldsymbol{x}) + b,$$

where $\boldsymbol{w} \in \mathbb{R}^M$ is the normal vector of the hyperplane, $b \in \mathbb{R}$ is the bias parameter and $\phi(\boldsymbol{x}) \in \mathbb{R}^M$ is the feature transform. We assume that we have given data $\boldsymbol{x}_n \in \mathbb{R}^D$ together with class labels $t_n \in \{-1, +1\}$. The classification of a new input vector $\boldsymbol{x}$ is performed according to $\operatorname{sgn}(y(\boldsymbol{x}))$.

We shall assume here that the dataset is linearly separable, that is it can be separated by a hyperplane $(\boldsymbol{w}, b)$ and hence there exists $(\boldsymbol{w}, b)$ such that $t_n y(\boldsymbol{x}_n) > 0$ for all $n = 1, ..., N$.

Recall the the unsigned distance of a vector $\phi(\boldsymbol{x}_n)$ to the hyperplane is given by $|y(\boldsymbol{x}_n)|/\|\boldsymbol{w}\| = t_n y(\boldsymbol{x}_n)/\|\boldsymbol{w}\|$. The margin is defined as the distance of the closest vector that is

$$\text{margin} = \min_n \frac{t_n y(\boldsymbol{x}_n)}{\|\boldsymbol{w}\|} = \frac{1}{\|\boldsymbol{w}\|} \min_n t_n(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b).$$

Without loss of generality, we shall assume that the minimum is achieved with a value 1. If not, we can easily rescale $(\boldsymbol{w}, b)$ such that the minimum has value 1 and the unsigned distance $t_n y(\boldsymbol{x}_n)/\|\boldsymbol{w}\|$ remains unchanged. Moreover all points will satisfy

$$t_n(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b) \geq 1, \;\; n = 1, ..., N,$$

and equality is attained for the closest point(s).

The central idea of the support vector machine (SVM) is now to find the configuration $(\boldsymbol{w}, b)$ that maximizes the margin. This can be achieved my solving

$$\max_{\boldsymbol{w}, b} \frac{1}{\|\boldsymbol{w}\|}, \quad \text{s.t. } t_n(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b) \geq 1, \;\; n = 1, ..., N.$$

This problem is clearly equivalent to the minimization problem

$$\min_{\boldsymbol{w}, b} \frac{1}{2}\|\boldsymbol{w}\|^2, \quad \text{s.t. } t_n(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b) \geq 1, \;\; n = 1, ..., N.$$

The last formula is the objective function of the SVM. Observe that it can also be written as

$$\min_{\boldsymbol{w}, b} \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{n=1}^{N} \delta_{\geq 1}(t_n(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b)),$$

where $\delta_{\geq 1}(t) = 0$ if $t \geq 1$ and $\delta_{\geq 1}(t) = \infty$ else. Observe that in this form, the SVM looks very similar to the standard regularized least squares model for two-class classification, with the only difference that the quadratic function $\frac{1}{2}(\cdot)^2$ has been replaced by the $\delta_{\geq 1}(\cdot)$ function.

5.5 Dual formulation of the SVM

In the primal formulation, the SVM is hard (if not impossible) to solve in case the dimension of the feature space (and hence the dimension of the parameter vector $\boldsymbol{w}$) is large. In this section we derive the dual formulation which gives us the possibility to solve the SVM in a space whose dimension is equivalent to the size of the training set.

Consider we have given an inequality-constrained optimization problem of the form

$$\min f(\boldsymbol{x}), \quad \text{s.t.} \quad g_n(\boldsymbol{x}) \leq 0, \ n = 1...N.$$

Moreover, we consider the Lagrangian function

$$L(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \sum_{n=1}^{N} g_n(\boldsymbol{x})\lambda_n,$$

where $\lambda_n \geq 0$ are the Lagrange multipliers. Note that the Lagrangian function is a minimization problem in $\boldsymbol{x}$ but it is a maximization problem in $\boldsymbol{\lambda}$. Indeed, if $g_n(\boldsymbol{x}) > 0$, then the Lagrangian can be maximized by sending $\lambda_n \to \infty$. In turn, the Lagrangian can be minimized by modifying $\boldsymbol{x}$ such that $g_n(\boldsymbol{x}) < 0$. Then, as $g_n(\boldsymbol{x})$ is now negative, the Lagrangian can be maximized by sending $\lambda_n \to 0$. This behavior can be seen as a game between $\boldsymbol{x}$ and $\boldsymbol{\lambda}$ and any optimal solution (for both players) will satisfy $g_n(\boldsymbol{x}) \leq 0$. In case $g_n(\boldsymbol{x}) < 0$ we have $\lambda_n = 0$ and in case $g_n(\boldsymbol{x}) = 0$ we have $\lambda_n \geq 0$. In summary, an optimal solution pair $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ of the the Lagrangian function must satisfy the Karush-Kuhn-Tucker (KKT) conditions:

$$\begin{aligned}
\nabla_{\boldsymbol{x}} L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) &= 0, \\
g_n(\boldsymbol{x}^*) &\leq 0, \quad n = 1, ..., N, \\
\lambda_n^* &\geq 0, \quad n = 1, ..., N, \\
\lambda_n^* g_n(\boldsymbol{x}^*) &= 0, \quad n = 1, ..., N.
\end{aligned}$$

The last condition expresses the fact that either $g_n(\boldsymbol{x}) < 0$ and $\lambda_n = 0$, or $g_n(\boldsymbol{x}) = 0$ and $\lambda_n > 0$, or the rather rare case that $g_n(\boldsymbol{x}) = 0$ and $\lambda_n = 0$.

The dual formulation is then found by minimizing the Lagrangian function with respect to $\boldsymbol{x}$. The resulting problem is then a maximization problem in $\boldsymbol{\lambda}$ and it is is referred to as the Lagrange-dual problem.

Now, let us apply this procedure to the SVM: First, we rewrite it as

$$\min_{\boldsymbol{w}, b} \frac{1}{2}\|\boldsymbol{w}\|^2, \quad \text{s.t. } 1 - t_n(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b) \leq 0, \ n = 1, ..., N.$$

The Lagrangian function is given by

$$L(\boldsymbol{w}, b, \boldsymbol{a}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{n=1}^{N}(1 - t_n(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b))a_n,$$

where $\boldsymbol{a} = (a_1, ..., a_N)^T$ with $a_n \geq 0$ are the Lagrange multipliers.

Computing the derivatives with respect to $\boldsymbol{w}$ and b yields

$$\nabla \boldsymbol{w} L(\boldsymbol{w}, b, \boldsymbol{a}) = w - \sum_{n=1}^{N} a_n t_n \phi(\boldsymbol{x}_n) = 0 \implies w = \sum_{n=1}^{N} a_n t_n \phi(\boldsymbol{x}_n),$$

and

$$\nabla_b L(\boldsymbol{w}, b, \boldsymbol{a}) = -\sum_{n=1}^{N} t_n a_n = 0.$$

Substituting the first equation back into the Lagrangian and keeping the second equation as a side constraint, we readily obtain the Lagrange dual problem

$$D(\boldsymbol{a}) = \frac{1}{2} \| \underbrace{\sum_{n=1}^{N} a_n t_n \phi(\boldsymbol{x}_n)}_{=\boldsymbol{w}} \|^2 + \sum_{n=1}^{N} \left(1 - t_n \left(\underbrace{\sum_{m=1}^{N} a_m t_m \phi(\boldsymbol{x}_m)}_{=\boldsymbol{w}} \right)^T \phi(\boldsymbol{x}_n) \right) a_n.$$

After some simplifications, we obtain the final dual SVM problem as

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

subject to the constraints

$$0 \leq a_n, \quad n = 1, ..., N, \quad \sum_{n=1}^{N} a_n t_n = 0,$$

and where we have used the usual kernel trick $\phi(\boldsymbol{x}_n)^T \phi(\boldsymbol{x}_m) = k(\boldsymbol{x}_n, \boldsymbol{x}_m)$.

For a new input vector $\boldsymbol{x}$, the classifying function $y(\boldsymbol{x})$ is computed from the dual vector $\boldsymbol{a}$ as

$$y(\boldsymbol{x}) = \boldsymbol{w}^T \phi(\boldsymbol{x}) + b = \left(\sum_{n=1}^{N} a_n t_n \phi(\boldsymbol{x}_n) \right)^T \phi(\boldsymbol{x}) + b = \sum_{n=1}^{N} a_n t_n k(\boldsymbol{x}_n, \boldsymbol{x}) + b.$$

Observe that only dual variables $a_n > 0$ must be considered in this formulation, hence if many of the a_n are equal to zero, the classification of a new input vector $\boldsymbol{x}$ will be very efficient. Those points for which $a_n > 0$ are called the support vectors and they define the margin. We also know that for support vectors it holds by definition,

$$t_n y(\boldsymbol{x}_n) = t_n \left(\sum_{m=1}^{N} a_m t_m k(\boldsymbol{x}_n, \boldsymbol{x}_m) + b \right) = 1. \tag{5.1}$$

For convenience, we define the set of support vectors $\mathcal{S} = \{n : t_n y(\boldsymbol{x}_n) = 1\}$ such that the classifying function can be written in the more efficient form

$$y(\boldsymbol{x}) = \sum_{n \in \mathcal{S}} a_n t_n k(\boldsymbol{x}_n, \boldsymbol{x}) + b.$$

In order to compute the bias parameter b, we divide (5.1) by $t_n = t_n^{-1}$ to obtain

$$b = t_n - \sum_{m \in \mathcal{S}} a_m t_m k(\boldsymbol{x}_n, \boldsymbol{x}_m).$$

For numerical stability it will be better to compute the bias b from all support vectors and average, that is

$$\bar{b} = \frac{1}{|\mathcal{S}|} \sum_{n \in \mathcal{S}} \left(t_n - \sum_{m \in \mathcal{S}} a_m t_m k(\boldsymbol{x}_n, \boldsymbol{x}_m) \right).$$

5.6 Dual formulation of the soft-margin SVM

The primal formulation of the soft-margin SVM in standard form is given by

$$\min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \frac{1}{2} \|\boldsymbol{w}\|^2 + C \sum_{n=1}^{N} \xi_n \quad \text{s.t. } 1 - \xi_n - t_n(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b) \leq 0, \ -\xi_n \leq 0, \ n = 1, ..., N,$$

From that, the Lagrangian function is computed as

$$L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{a}, \boldsymbol{b}) = \frac{1}{2} \|\boldsymbol{w}\|^2 + C \sum_{n=1}^{N} \xi_n + \sum_{n=1}^{N} (1 - \xi_n - t_n(\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b)) a_n + \sum_{n=1}^{N} (-\xi_n) b_n,$$

with Lagrange multipliers $a_n \geq 0$ and $b_n \geq 0$. The main difference to the standard SVM is that we also have to minimize the Lagrangian with respect to $\boldsymbol{\xi}$. The derivative with respect to ξ_n is given by

$$\nabla_{\xi_n} L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{a}, \boldsymbol{b}) = C - a_n - b_n = 0, \quad n = 1, ..., N.$$

From this equation it follows that

$$b_n = C - a_n \geq 0 \implies 0 \leq a_n \leq C.$$

Therefore, the only difference to the standard SVM is that the dual variables a_n are bounded from above with the parameter C.

5.7 SVM versus logistic regression

Let us consider the soft-margin SVM in its primal formulation with fixed $\boldsymbol{w}$ and b. The partial minimization problem with respect to a slack variable ξ_n can be written as the following function

$$h(t_n y_n) = \min_{\xi_n} \xi_n, \quad \text{s.t.} \quad \xi_n \geq 0, \ \xi_n \geq 1 - t_n y_n.$$

It is easy to see that the function $h(t_n y_n)$ is given by

$$h(t_n y_n) = \max\{0, 1 - t_n y_n\},$$

which is called the hinge-function. Therefore, the soft-margin SVM can also be written in the form

$$\min_{\boldsymbol{w},b} \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \sum_{n=1}^{N} h(t_n y_n),$$

where we have set $\lambda = C^{-1}$.

Let us now compare this formulation to the logistic regression model. In previous sections we derived the logistic regression model using target labels $t \in \{0, 1\}$ but in order to compare this model we need to reformulate it in terms of target labels $t_n \in \{-1, +1\}$.

Recall that in the 2-class classification problem, the posterior probability was written as

$$p(t_n = +1|y_n) = \sigma(y_n),$$

where $\sigma(y) = 1/(1 + \exp(-y))$ is the logistic sigmoid function. For class label -1 we have

$$p(t_n = -1|y_n) = 1 - \sigma(y_n) = \sigma(-y_n).$$

In summary, we see that

$$p(t_n|y_n) = \sigma(t_n y_n).$$

In turn the log-likelihood function is defined as

$$-\log p(t_1, ..., t_N|\boldsymbol{w}) = -\sum_{n=1}^{N} \log(1/(1 + \exp(-t_n y_n))) = \sum_{n=1}^{N} l(t_n y_n),$$

with $l(y) = \log(1+\exp(-y))$. Finally, adding a quadratic regularization term to the log-likelihood function yields the regularized logistic regression model

$$\min_{\boldsymbol{w},b} \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \sum_{n=1}^{N} l(t_n y_n).$$

Observe that the only difference lies in the function $h(y)$ versus $l(y)$. Plotting both functions reveals that $l(y)$ is just a smoothed version of $h(y)$.

5.8 Dual form of SVM regression

Let us consider the objective function of SVM regression

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{n=1}^{N} \iota_\varepsilon(\boldsymbol{w}^T\phi(\boldsymbol{x}_n) + b - t_n),$$

where

$$\iota_\varepsilon(z) = \max(0, |z| - \varepsilon)$$

is the ε-insensitive function. Instead of the Lagrange-dual formulation, we consider here the Fenchel-Rockafellar-dual formulation. It is based on the convex-conjugate function, which is defined for any function $f(x)$ as

$$f^*(y) = \max_x \; x^T y - f(x).$$

Moreover, for convex functions it holds that

$$f(x) = f^{**}(x) = \max_y \; x^T y - f^*(y),$$

Observe that we have already used the definition of the convex conjugate in the derivation of least-squares kernel regression. A simple computation shows that the convex conjugate of the function $C\iota_\varepsilon$, is given by

$$(C\iota_\varepsilon)^*(y) = \varepsilon|y| + \delta_{[-C,C]}(y),$$

where $\delta_{[-C,C]}(y) = 0$ if $y \in [-C, C]$ and $\delta_{[-C,C]}(y) = \infty$, else, which is equivalent to the constraint $y \in [-C, C]$. Now using that

$$C\iota_\varepsilon(x) = \max_y \; xy - \iota_\varepsilon^*(y) = \max_{y \in [-C,C]} xy - \varepsilon|y|,$$

the SVM regression model can be rewritten as

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{n=1}^{N} \max_{a_n \in [-C,C]} (\boldsymbol{w}^T \phi(\boldsymbol{x}_n) + b - t_n)a_n - \varepsilon|a_n|,$$

where $\boldsymbol{a} = (a_1, ..., a_N)$ are the dual variables. Minimizing with respect to $\boldsymbol{w}, b$ and substituting back the equations and after some simplifications we obtain the dual of the SVM regression model

$$\begin{aligned} \max_{\boldsymbol{a}} \; & -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m k(\boldsymbol{x}_n, \boldsymbol{x}_m) - \varepsilon \sum_{n=1}^{N} |a_n| + \sum_{n=1}^{N} t_n a_n \\ \text{s.t.} \quad & \sum_{n=1}^{N} a_n = 0, \quad -C \le a_n \le C, \quad n = 1, ..., N. \end{aligned}$$