Important

# Machine Learning Exercise Sheet 05

## Linear Classification

Exercise sheets consist of two parts: homework and in-class exercises. You solve the homework exercises on your own or with your registered group and upload it to Moodle for a possible grade bonus. The in-class exercises will be solved and explained during the tutorial. You do not have to upload any solutions of the in-class exercises.

## In-class Exercises

### Multi-Class Classification

**Problem 1:** Consider a generative classification model for $C$ classes defined by class probabilities $\mathrm{p}(y = c) = \boldsymbol{\pi}_c$ and general class-conditional densities $\mathrm{p}(\boldsymbol{x} \mid y = c, \boldsymbol{\theta}_c)$ where $\boldsymbol{x} \in \mathbb{R}^D$ is the input feature vector and $\boldsymbol{\theta} = \{\boldsymbol{\theta}_c\}_{c=1}^C$ are further model parameters. Suppose we are given a training set $\mathcal{D} = \{(\boldsymbol{x}^{(n)}, y^{(n)})\}_{n=1}^N$ where $y^{(n)}$ is a binary target vector of length $C$ that uses the 1-of-$C$ (one-hot) encoding scheme, so that it has components $y_c^{(n)} = \delta_{ck}$ if pattern $n$ is from class $y = k$. Assuming that the data points are i.i.d., show that the maximum-likelihood solution for the class probabilities $\boldsymbol{\pi}$ is given by

$$\pi_c = \frac{N_c}{N}$$

where $N_c$ is the number of data points assigned to class $c$.

Start!

The data likelihood given the parameters $\{\pi_c, \boldsymbol{\theta}_c\}_{c=1}^C$ is

$$\mathrm{p}(\mathcal{D}|\{\pi_c, \boldsymbol{\theta}_c\}_{c=1}^C) = \prod_{n=1}^{N} \prod_{c=1}^{C} (\mathrm{p}(\boldsymbol{x}^{(n)}|\boldsymbol{\theta}_c)\pi_c)^{y_c^{(n)}}$$

and so the data log-likelihood is given by

$$\log \mathrm{p}(\mathcal{D}|\{\pi_c, \boldsymbol{\theta}_c\}_{c=1}^C) = \sum_{n=1}^{N} \sum_{c=1}^{C} y_c^{(n)} \log \pi_c + \text{const w.r.t. } \pi_c.$$

In order to maximize the log likelihood with respect to $\pi_c$ we need to preserve the constraint $\sum_c \pi_c = 1$. For this we use the method of Lagrange multipliers where we introduce $\lambda$ as an unconstrained additional parameter and find a local extremum of the unconstrained function

$$\sum_{n=1}^{N} \sum_{c=1}^{C} y_c^{(n)} \log \pi_c - \lambda \left( \sum_{c=1}^{C} \pi_c - 1 \right).$$

instead. See wikipedia article on Lagrange multipliers for an intuition of why this works. This function is a sum of concave terms in $\pi_c$ as well as $\lambda$ and is therefore itself concave in these variables.

We can find the extremum by finding the root of the derivatives. Setting the derivative with respect to $\pi_c$ equal to zero, we obtain

$$\pi_c = \frac{1}{\lambda}\sum_{n=1}^{N} y_c^{(n)} = \frac{N_c}{\lambda}.$$

Setting the derivative with respect to $\lambda$ equal to zero, we obtain the original constraint

$$\sum_{c=1}^{C} \pi_c = 1$$

where we can now plug in the previous result $\pi_c = \frac{N_c}{\lambda}$ and obtain $\lambda = \sum_c N_c = N$. Plugging this in turn into the expression for $\pi_c$ we obtain

$$\pi_c = \frac{N_c}{N}$$

which we wanted to show.

## Linear Discriminant Analysis

**Problem 2:** Using the same classification model as in the previous question, now suppose that the class-conditional densities are given by Gaussian distributions with a *shared* covariance matrix, so that

$$\mathrm{p}(\boldsymbol{x} \mid y = c, \boldsymbol{\theta}) = \mathrm{p}(\boldsymbol{x} \mid \boldsymbol{\theta}_c) = \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}).$$

Show that the maximum likelihood estimate for the mean of the Gaussian distribution for class $c$ is given by

$$\boldsymbol{\mu}_c = \frac{1}{N_c} \sum_{\substack{n=1 \\ y^{(n)}=c}}^{N} \boldsymbol{x}^{(n)}$$

which represents the mean of the observations assigned to class $c$.

Similarly, show that the maximum likelihood estimate for the shared covariance matrix is given by

$$\boldsymbol{\Sigma} = \sum_{c=1}^{C} \frac{N_c}{N} \boldsymbol{S}_c \quad \text{where} \quad \boldsymbol{S}_c = \frac{1}{N_c} \sum_{\substack{n=1 \\ y^{(n)}=c}}^{N} (\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)(\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)^{\mathrm{T}}.$$

Thus $\boldsymbol{\Sigma}$ is given by a weighted average of the sample covariances of the data associated with each class, in which the weighting coefficients $N_c/N$ are the prior probabilities of the classes.

We begin by writing out the data log-likelihood.

$$\begin{aligned} &\log \mathrm{p}(\mathcal{D} | \{\pi_c, \boldsymbol{\theta}_c\}_{c=1}^{C}) \\ &= \sum_{n=1}^{N} \sum_{c=1}^{C} y_c^{(n)} \log \pi_c \cdot \mathrm{p}(\boldsymbol{x}^{(n)} \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}) \end{aligned}$$

Then we plug in the definition of the multivariate Gaussian

$$= \sum_{n=1}^{N} \sum_{c=1}^{C} y_c^{(n)} \log \left( (2\pi)^{-\frac{D}{2}} \det(\boldsymbol{\Sigma})^{-\frac{1}{2}} \exp\left( -\frac{1}{2} (\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)^{\mathrm{T}} \Sigma^{-1} (\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c) \right) \right) + y^{(n)} \log \pi_c$$

and simplify.

$$= -\frac{1}{2} \sum_{n=1}^{N} \sum_{c=1}^{C} y_c^{(n)} \left( D \log 2\pi + \log\det(\boldsymbol{\Sigma}) + (\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)^{\mathrm{T}} \Sigma^{-1} (\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c) - 2 \log \pi_c \right)$$

This expression is concave in $\boldsymbol{\mu}_c$, so we can obtain the maximizer by finding the root of the derivative. With the help of the matrix cookbook, we identify the derivative with respect to $\boldsymbol{\mu}_c$ as

$$\sum_{n=1}^{N} y_c^{(n)} \Sigma^{-1} (\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)$$

which we can set to 0 and solve for $\boldsymbol{\mu}_c$ to obtain

$$\boldsymbol{\mu}_c = \frac{1}{\sum_{n=1}^{N} y_c^{(n)}} \sum_{n=1}^{N} y_c^{(n)} \boldsymbol{x}^{(n)} = \frac{1}{N_c} \sum_{\substack{n=1 \\ y^{(n)}=c}}^{N} \boldsymbol{x}^{(n)}.$$

To find the optimal $\boldsymbol{\Sigma}$, we need the trace trick

$$a = \mathrm{Tr}(a) \text{ for all } a \in \mathbb{R} \quad \text{and} \quad \mathrm{Tr}(\boldsymbol{ABC}) = \mathrm{Tr}(\boldsymbol{BCA}).$$

With this we can rewrite

$$(\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)^{\mathrm{T}} \Sigma^{-1} (\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c) = \mathrm{Tr}\left( \Sigma^{-1} (\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)(\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)^{\mathrm{T}} \right)$$

and use the matrix-trace derivative rule $\frac{\partial}{\partial \boldsymbol{A}} \mathrm{Tr}(\boldsymbol{AB}) = \boldsymbol{B}^{\mathrm{T}}$ to find the derivative of the data log-likelihood with respect to $\boldsymbol{\Sigma}$. Because the log-likelihood contains both $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}^{-1}$, we convert one into the other with $\log\det \boldsymbol{A} = -\log\det \boldsymbol{A}^{-1}$ to obtain

$$-\frac{1}{2} \sum_{n=1}^{N} \sum_{c=1}^{C} y_c^{(n)} \left( -\log\det \boldsymbol{\Sigma}^{-1} + \mathrm{Tr}\left( \Sigma^{-1} (\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)(\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)^{\mathrm{T}} \right) \right) + \text{const w.r.t. } \boldsymbol{\Sigma}.$$

Finally, we use rule (57) from the matrix cookbook $\frac{\partial \log|\det \boldsymbol{X}|}{\partial \boldsymbol{X}} = (\boldsymbol{X}^{-1})^{\mathrm{T}}$ and compute the derivative of the log-likelihood with respect to $\boldsymbol{\Sigma}^{-1}$ as

$$-\frac{1}{2} \sum_{n=1}^{N} \sum_{c=1}^{C} y_c^{(n)} \left( -\boldsymbol{\Sigma}^{\mathrm{T}} + (\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)(\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)^{\mathrm{T}} \right).$$

We find the root with respect to $\boldsymbol{\Sigma}$ and find

$$\boldsymbol{\Sigma} = \frac{1}{\sum_{n=1}^{N} \sum_{c=1}^{C} y_c^{(n)}} \left( \sum_{n=1}^{N} \sum_{c=1}^{C} y_c^{(n)} (\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)(\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)^{\mathrm{T}} \right)^{\mathrm{T}} = \frac{1}{N} \sum_{c=1}^{C} \sum_{\substack{n=1 \\ y^{(n)}=c}}^{N} (\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)(\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_c)^{\mathrm{T}}$$

which we can immediately break apart into the representation in the instructions.

## Homework

### Linear classification

**Problem 3:** We want to create a generative binary classification model for classifying *non-negative* one-dimensional data. This means, that the labels are binary ($y \in \{0, 1\}$) and the samples are $x \in [0, \infty)$.

We assume uniform class probabilities

$$p(y = 0) = p(y = 1) = \frac{1}{2}.$$

As our samples $x$ are non-negative, we use exponential distributions (and not Gaussians) as class conditionals:

$$p(x \mid y = 0) = \mathrm{Expo}(x \mid \lambda_0) \qquad \text{and} \qquad p(x \mid y = 1) = \mathrm{Expo}(x \mid \lambda_1),$$

where $\lambda_0 \neq \lambda_1$. Assume, that the parameters $\lambda_0$ and $\lambda_1$ are known and fixed.

a) What is the name of the posterior distribution $p(y \mid x)$? You only need to provide the name of the distribution (e.g., “normal”, “gamma”, etc.), not estimate its parameters.

Bernoulli.

*Remark: $y$ can only take values in $\{0, 1\}$, so obviously Bernoulli is the only possible answer.*

b) What values of $x$ are classified as class 1? (As usual, we assume that the classification decision is $\hat{y} = \arg\max_k p(y = k \mid x)$)

Sample $x$ is classified as class 1 if $p(y = 1 \mid x) > p(y = 0 \mid x)$. This is the same as saying

$$\frac{p(y = 1 \mid x)}{p(y = 0 \mid x)} \overset{!}{>} 1 \qquad \text{or equivalently} \qquad \log \frac{p(y = 1 \mid x)}{p(y = 0 \mid x)} \overset{!}{>} 0.$$

We begin by simplifying the left hand side.

$$\begin{aligned}
\log \frac{p(y = 1 \mid x)}{p(y = 0 \mid x)} &= \log \frac{p(x \mid y = 1)\, p(y = 1)}{p(x \mid y = 0)\, p(y = 0)} \\
&= \log \frac{p(x \mid y = 1)}{p(x \mid y = 0)} \\
&= \log \frac{\lambda_1 \exp(-\lambda_1 x)}{\lambda_0 \exp(-\lambda_0 x)} \\
&= \log \frac{\lambda_1}{\lambda_0} + \lambda_0 x - \lambda_1 x = \log \frac{\lambda_1}{\lambda_0} + (\lambda_0 - \lambda_1) x
\end{aligned}$$

To figure out which $x$ are classified as class 1, we need to solve for $x$.

$$\log \frac{\lambda_1}{\lambda_0} + (\lambda_0 - \lambda_1) x > \log 1 \quad \Leftrightarrow \quad (\lambda_0 - \lambda_1) x > -\log \frac{\lambda_1}{\lambda_0} = \log \lambda_0 - \log \lambda_1$$

We have to be careful, because if $(\lambda_0 - \lambda_1) < 0$, dividing by it will flip the inequality sign. Hence the answer is

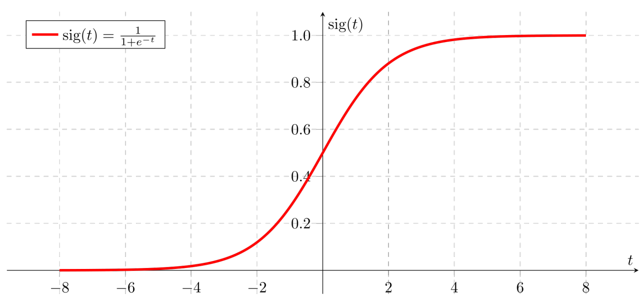
$$\begin{cases} x \in \left(\frac{\log \lambda_0 - \log \lambda_1}{\lambda_0 - \lambda_1}, \infty\right) & \text{if } \lambda_0 > \lambda_1 \\ x \in \left[0, \frac{\log \lambda_0 - \log \lambda_1}{\lambda_0 - \lambda_1}\right) & \text{otherwise.} \end{cases}$$

**Problem 4:** Let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}$ be a linearly separable dataset for 2-class classification, i.e. there exists a vector $\boldsymbol{w}$ such that $\operatorname{sign}\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}\right)$ separates the classes. Show that the maximum likelihood parameter $\boldsymbol{w}$ of a logistic regression model has $\|\boldsymbol{w}\| \to \infty$. Assume that $\boldsymbol{w}$ contains the bias term.

How can we modify the training process to prefer a $\boldsymbol{w}$ of finite magnitude?

In logistic regression, we model the posterior distribution as

$$y_i \mid \boldsymbol{x} \sim \text{Bernoulli}(\sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i)) \quad \text{where} \quad \sigma(a) = \frac{1}{1 + \exp(-a)}.$$




We fit the logistic regression model by choosing the parameter $\boldsymbol{w}$ that maximizes the data log-likelihood or alternatively minimizes the negative log-likelihood which expands to

$$E(\boldsymbol{w}) = -\log \mathrm{p}\left(\boldsymbol{y} \mid \boldsymbol{w}, \boldsymbol{X}\right) = -\sum_{i=1}^{N} y_i \log \sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i) + (1 - y_i)\log(1 - \sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i)).$$

We assumed that the data-set is linearly separable, so by definition there is a $\tilde{\boldsymbol{w}}$ such that

$$\tilde{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{x}_i > 0 \text{ if } y_i = 1 \quad \text{and} \quad \tilde{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{x}_i < 0 \text{ if } y_i = 0.$$

Scaling this separator $\tilde{\boldsymbol{w}}$ by a factor $\lambda \gg 0$ makes the negative log-likelihood smaller and smaller. To see this, we compute the limit

$$\lim_{\lambda\to\infty} E(\lambda\tilde{\boldsymbol{w}}) = -\left(\sum_{\substack{i=1\\y_i=1}}^{N} \log \lim_{\lambda\to\infty} \sigma(\lambda\overbrace{\tilde{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{x}_i}^{>0}) + \sum_{\substack{i=1\\y_i=0}}^{N} \log\left(1 - \lim_{\lambda\to\infty} \sigma(\lambda\overbrace{\tilde{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{x}_i}^{<0})\right)\right) = 0$$

which equals the smallest achievable value ($E$ is the negative log of a probability, so $E(\boldsymbol{w}) \in [0, \infty)$ and thus $E(\boldsymbol{w}) \geq 0$).

We can see that $E$ is a convex function because log is concave and $\sigma$ is convex if $a < 0$ and concave if $a > 0$. So $\log \sigma(a)$ is concave if $a > 0$ and $\log(1 - \sigma(a))$ is concave if $a < 0$. It follows that $E$ is a convex function because $E$ is the negative sum of concave functions.

A convex function has a unique minimum *if* it attains its minimum value. We know that $E$ tends towards its minimum as $\lambda \to \infty$, so $E$ cannot have a finite minimizer and all its minima are only achieved in the limit. It follows that any solution to the loss minimization problem has infinite norm.

Because $E$ is convex and tends towards a limit of 0 in some directions, we can move the minimum into the space of finite vectors by adding any convex term that achieves its minimum such as $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}$ or similar forms of weight regularization.

**Problem 5:** Show that the softmax function is equivalent to a sigmoid in the 2-class case.

$$\begin{aligned}
\frac{\exp(\boldsymbol{w}_1^T\boldsymbol{x})}{\exp(\boldsymbol{w}_1^T\boldsymbol{x}) + \exp(\boldsymbol{w}_0^T\boldsymbol{x})} &= \frac{1}{1 + \exp(\boldsymbol{w}_0^T\boldsymbol{x}) / \exp(\boldsymbol{w}_1^T\boldsymbol{x})} \\
&= \frac{1}{1 + \exp(\boldsymbol{w}_0^T\boldsymbol{x} - \boldsymbol{w}_1^T\boldsymbol{x})} \\
&= \frac{1}{1 + \exp(-(\boldsymbol{w}_1 - \boldsymbol{w}_0)^T\boldsymbol{x})} \\
&= \sigma(\hat{\boldsymbol{w}}^T\boldsymbol{x})
\end{aligned}$$

where $\hat{\boldsymbol{w}} = \boldsymbol{w}_1 - \boldsymbol{w}_0$.

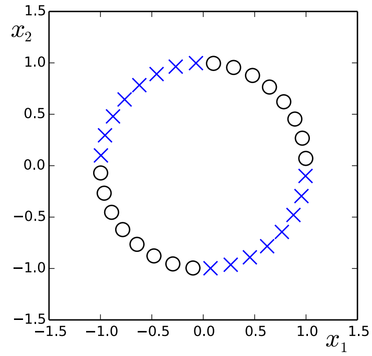
One conclusion we can draw from this is that if we have $C$ parameter vectors $\boldsymbol{w}_c$ for $C$ classes, the logistic regression model is unidentifiable. This means that adding a constant $\boldsymbol{\tau} \in \mathbb{R}^D$ to each vector $\boldsymbol{w}_c := \boldsymbol{w}_c + \boldsymbol{\tau}$ would lead to the same logistic regression model. We can fix this issue by adding a constraint $\boldsymbol{w}_1 = \boldsymbol{0}$, which is what is done implicitly when we use sigmoid (instead of 2-class softmax) in binary classification.

**Problem 6:** Show that the derivative of the sigmoid function $\sigma(a) = (1 + e^{-a})^{-1}$ can be written as

$$\frac{\partial \sigma(a)}{\partial a} = \sigma(a)\,(1 - \sigma(a))\,.$$

$$\frac{\partial \sigma(a)}{\partial a} = -\frac{1}{(1 + e^{-a})^2} \cdot e^{-a} \cdot (-1) = \frac{1}{1 + e^{-a}}\frac{e^{-a}}{1 + e^{-a}} = \sigma(a)\frac{1 + e^{-a} - 1}{1 + e^{-a}} = \sigma(a)\,(1 - \sigma(a))$$

**Problem 7:** Give a basis function $\phi(x_1, x_2)$ that makes the data in the example below linearly separable (crosses in one class, circles in the other).




One example is $\phi(\boldsymbol{x}) = \boldsymbol{x}_1\boldsymbol{x}_2$ which makes the data separable by the hyperplane $\boldsymbol{w} = (1)$ because the circles will be mapped to the positive real numbers while the crosses go to the negative numbers, i.e. $\boldsymbol{w}^\mathrm{T}\boldsymbol{x} > 0$ if $\boldsymbol{x}$ is a circle and $\boldsymbol{w}^\mathrm{T}\boldsymbol{x} < 0$ otherwise.

## Naive Bayes

**Problem 8:** In 2-class classification the decision boundary $\Gamma$ is the set of points where both classes are assigned equal probability,

$$\Gamma = \{\boldsymbol{x} \mid \mathrm{p}(y = 1 \mid \boldsymbol{x}) = \mathrm{p}(y = 0 \mid \boldsymbol{x})\}.$$

Show that Naive Bayes with Gaussian class likelihoods produces a quadratic decision boundary in the 2-class case, i.e. that $\Gamma$ can be written with a quadratic equation of $\boldsymbol{x}$,

$$\Gamma = \left\{\boldsymbol{x} \mid \boldsymbol{x}^\mathrm{T}\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}\boldsymbol{x} + c = 0\right\},$$

for some $\boldsymbol{A}$, $\boldsymbol{b}$ and $c$.

As a reminder, in Naive Bayes we assume class prior probabilities

$$\mathrm{p}(y = 0) = \boldsymbol{\pi}_0 \quad \text{and} \quad \mathrm{p}(y = 1) = \boldsymbol{\pi}_1$$

and class likelihoods

$$\mathrm{p}(\boldsymbol{x} \mid y = c) = \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$$

with per-class means $\boldsymbol{\mu}_c$ and *diagonal* (because of the feature independence) covariances $\boldsymbol{\Sigma}_c$.

Because $\mathrm{p}(y = 1 \mid \boldsymbol{x}) + \mathrm{p}(y = 0 \mid \boldsymbol{x}) = 1$ and we want them to be equal, we can assume that $\mathrm{p}(y = 0 \mid \boldsymbol{x}) > 0$ and rewrite the defining equation as

$$\frac{\mathrm{p}(y = 1 \mid \boldsymbol{x})}{\mathrm{p}(y = 0 \mid \boldsymbol{x})} = 1.$$

Now apply the logarithm to both sides and simplify.

$$
\begin{aligned}
\log \frac{\mathrm{p}(y=1 \mid \boldsymbol{x})}{\mathrm{p}(y=0 \mid \boldsymbol{x})} &= \log \left( \frac{\mathrm{p}(\boldsymbol{x} \mid y=1)\,\mathrm{p}(y=1)}{\mathrm{p}(\boldsymbol{x})} \cdot \frac{\mathrm{p}(\boldsymbol{x})}{\mathrm{p}(\boldsymbol{x} \mid y=0)\,\mathrm{p}(y=0)} \right) \\
&= \log\left(\mathrm{p}(\boldsymbol{x} \mid y=1)\,\mathrm{p}(y=1)\right) - \log\left(\mathrm{p}(\boldsymbol{x} \mid y=0)\,\mathrm{p}(y=0)\right) \\
&= \log \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_1, \boldsymbol{S}_1) - \log \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_0, \boldsymbol{S}_0) + \log \frac{\pi_1}{\pi_0} \\
&= -\frac{1}{2}\log(2\pi)^D |\boldsymbol{S}_1| - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^T \boldsymbol{S}_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1) \\
&\quad + \frac{1}{2}\log(2\pi)^D |\boldsymbol{S}_0| + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_0)^T \boldsymbol{S}_0^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_0) + \log\frac{\pi_1}{\pi_0} \\
&= -\frac{1}{2}\boldsymbol{x}^T \boldsymbol{S}_1^{-1}\boldsymbol{x} + \boldsymbol{x}^T \boldsymbol{S}_1^{-1}\boldsymbol{\mu}_1 - \frac{1}{2}\boldsymbol{\mu}_1^T \boldsymbol{S}_1^{-1}\boldsymbol{\mu}_1 \\
&\quad + \frac{1}{2}\boldsymbol{x}^T \boldsymbol{S}_0^{-1}\boldsymbol{x} - \boldsymbol{x}^T \boldsymbol{S}_0^{-1}\boldsymbol{\mu}_0 + \frac{1}{2}\boldsymbol{\mu}_0^T \boldsymbol{S}_0^{-1}\boldsymbol{\mu}_0 + \frac{1}{2}\log\frac{|\boldsymbol{S}_0|}{|\boldsymbol{S}_1|} + \log\frac{\pi_1}{\pi_0} \\
&= \frac{1}{2}\boldsymbol{x}^T[\boldsymbol{S}_0^{-1} - \boldsymbol{S}_1^{-1}]\boldsymbol{x} + \boldsymbol{x}^T[\boldsymbol{S}_1^{-1}\boldsymbol{\mu}_1 - \boldsymbol{S}_0^{-1}\boldsymbol{\mu}_0] \\
&\quad - \frac{1}{2}\boldsymbol{\mu}_1^T \boldsymbol{S}_1^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_0^T \boldsymbol{S}_0^{-1}\boldsymbol{\mu}_0 + \log\frac{\pi_1}{\pi_0} + \frac{1}{2}\log\frac{|\boldsymbol{S}_0|}{|\boldsymbol{S}_1|}
\end{aligned}
$$

This shows that $\Gamma$ is quadratic and can alternatively be written as

$$\Gamma = \left\{ \boldsymbol{x} \mid \boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}\boldsymbol{x} + c = 0 \right\}$$

where

$$
\begin{aligned}
\boldsymbol{A} &= \frac{1}{2}[\boldsymbol{S}_0^{-1} - \boldsymbol{S}_1^{-1}] \qquad \boldsymbol{b} = \boldsymbol{S}_1^{-1}\boldsymbol{\mu}_1 - \boldsymbol{S}_0^{-1}\boldsymbol{\mu}_0 \\
c &= -\frac{1}{2}\boldsymbol{\mu}_1^T \boldsymbol{S}_1^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_0^T \boldsymbol{S}_0^{-1}\boldsymbol{\mu}_0 + \log\frac{\pi_1}{\pi_0} + \frac{1}{2}\log\frac{|\boldsymbol{S}_0|}{|\boldsymbol{S}_1|}.
\end{aligned}
$$

If both classes had the same covariance matrix ($\boldsymbol{S}_0 = \boldsymbol{S}_1$), $\boldsymbol{A}$ would be the zero matrix and we would receive a linear decision boundary as we did in the lecture (also, $\log \frac{|\boldsymbol{S}_0|}{|\boldsymbol{S}_1|} = 0$).