CLASSIFICATION BASICS

Sergey Nikolenko

Harbour Space University

May 6, 2025

Random facts:

- on May 6, 1527, Spanish and German troops sacked Rome, and event that is widely considered to be the end of the Renaissance
- on May 6, 1682, Louis XIV moved his court to the Palace of Versailles
- on May 6, 1840, the Penny Black, the first postage stamp in history, became valid for use
- on May 6, 1889, the Eiffel Tower was officially opened to the public at the Universal Exposition in Paris
- on May 6, 1937, while landing at Lakehurst, New Jersey, on its first transatlantic crossing of the year, the German dirigible *Hindenburg* burst into flames and was destroyed, killing 36 of the 97 persons aboard
- on May 6, 1998, Steve Jobs of Apple Inc. unveiled the first iMac
- on May 6, 2004, the final episode of *Friends* was aired
- on May 6, 2023, in the first British coronation in seven decades, Charles III and Camilla were crowned king and queen, respectively

Intro to classification

- Now classification: assign vector $\mathbf{x}$ to one of K classes C_k.
- In the end, our entire space will be divided into these classes.
- So in fact we are looking for a *decision surface* (decision surface, decision boundary).

- How to encode? Binary task – very naturally, variable t, $t = 0$ corresponds to C_1, $t = 1$ corresponds to C_2.
- The estimate t can be interpreted as a probability (at least, we will try to make it possible).
- If several classes – convenient to use 1-of-K:

$$\mathbf{t} = (0, \dots, 0, 1, 0, \dots)^\top.$$

- This can also be interpreted as probabilities – or proportional to them.

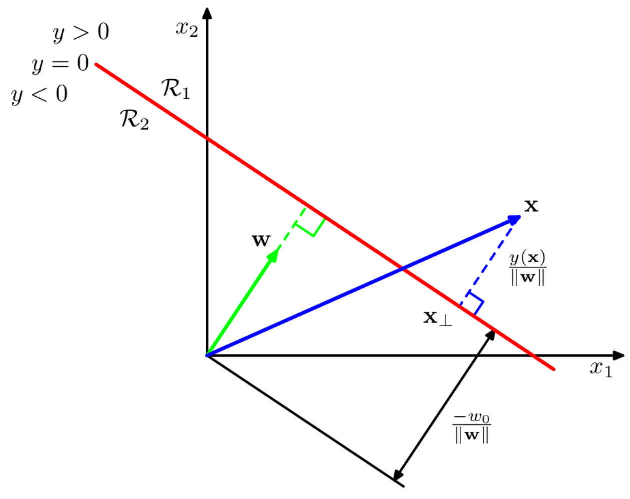
Decision hyperplane

- Let's start with geometry: consider a linear discriminant function

$$y(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + w_0.$$

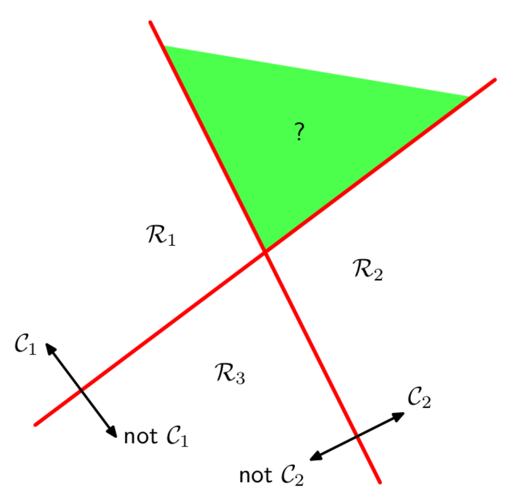
- This is a hyperplane, and $\mathbf{w}$ is normal to it.
- The distance from the origin to the hyperplane is $\frac{-w_0}{\|\mathbf{w}\|}$.
- $y(\mathbf{x})$ is related to the distance to the hyperplane: $d = \frac{y(\mathbf{x})}{\|\mathbf{w}\|}$.

Decision hyperplane

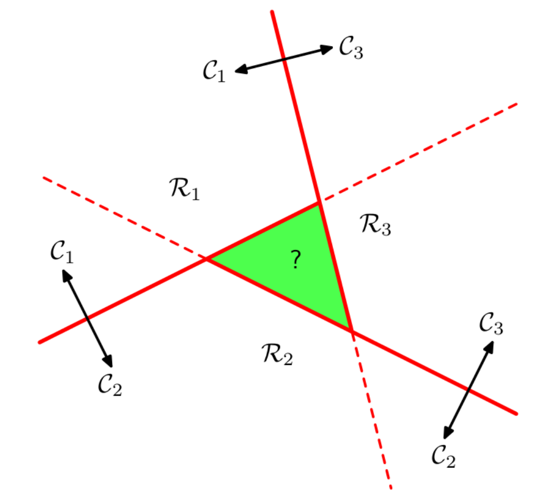


Multiple classes

- With multiple classes there's a complication.
- We can consider K surfaces of the "one versus all" type.
- We can also use $\binom{K}{2}$ surfaces of the "each versus each" type.
- But all of this doesn't seem very good.


?
$\mathcal{R}_1$
$\mathcal{R}_2$
$\mathcal{R}_3$
$\mathcal{C}_1$
not $\mathcal{C}_1$
$\mathcal{C}_2$
not $\mathcal{C}_2$

Multiple classes



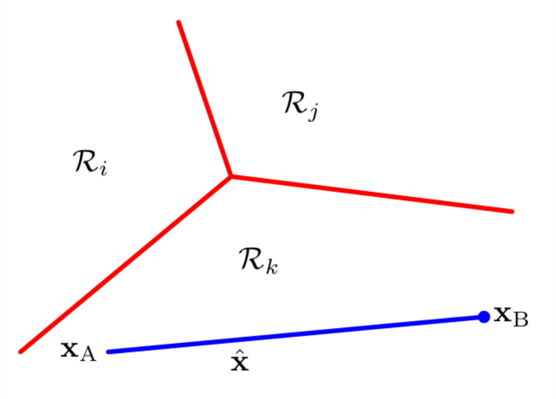
- It's better to consider a unified discriminant of K linear functions:

$$y_k(\mathbf{x}) = \mathbf{w}_k^\top \mathbf{x} + w_{k0}.$$

- Classify to C_k if $y_k(\mathbf{x})$ is maximal.
- Then the decision surface between C_k and C_j will be a hyperplane of the form $y_k(\mathbf{x}) = y_j(\mathbf{x})$:

$$\left(\mathbf{w}_k - \mathbf{w}_j\right)^\top \mathbf{x} + \left(w_{k0} - w_{j0}\right).$$

Multiple classes



Exercise. Prove that the regions corresponding to classes in this approach are always simply connected and convex.

- We can again use the least squares method: write $y_k(\mathbf{x}) = \mathbf{w}_k^\top \mathbf{x} + w_{k0}$ together (hiding the free term) as

$$\mathbf{y}(\mathbf{x}) = \mathbf{W}^\top \mathbf{x}.$$

- We can find $\mathbf{W}$ by optimizing the sum of squares; error function:

$$E_D(\mathbf{W}) = \frac{1}{2}\mathrm{Tr}\left[(\mathbf{XW}-\mathbf{T})^\top(\mathbf{XW}-\mathbf{T})\right].$$

- We take the derivative, solve...

Least squares method

- …we get the familiar

$$\mathbf{W} = \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{T} = \mathbf{X}^\dagger\mathbf{T},$$

  where $\mathbf{X}^\dagger$ is the Moore-Penrose pseudoinverse.
- Now we can find the discriminant function:

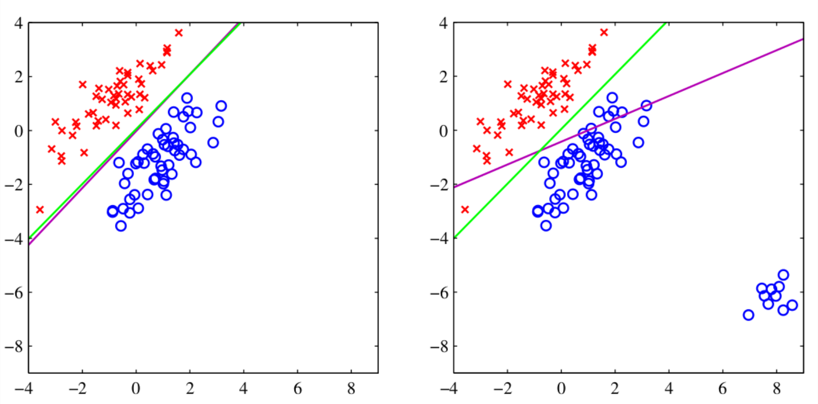
$$\mathbf{y}(\mathbf{x}) = \mathbf{W}^\top\mathbf{x} = \mathbf{T}^\top\left(\mathbf{X}^\dagger\right)^\top\mathbf{x}.$$

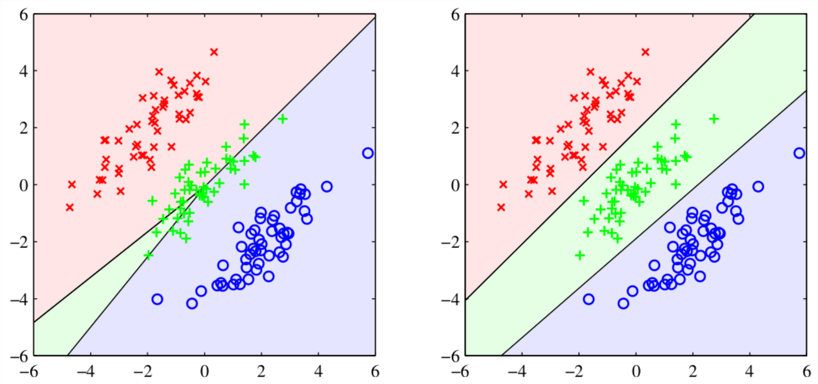
- This solution preserves linearity.

Exercise. Prove that in the 1-of-K coding scheme, predictions $y_k(\mathbf{x})$ for different classes with any $\mathbf{x}$ will sum to 1. Why will they still not be reasonable probability estimates?

- Problems with least squares:
  - outliers are poorly handled;
  - ”too correct” predictions add penalty.

Problems with least squares




6
4
2
0
−2
−4
−6
−6 −4 −2 0 2 4 6

Problems with least squares

- Why is that? Why do least squares work so poorly?

Problems with least squares

- Why is that? Why do least squares work so poorly?
- They assume a Gaussian distribution of error.
- But, of course, the distribution of binary vectors is far from Gaussian.

Fisher's Linear Discriminant

- Another view of classification: in the linear case, we want to project points into dimension 1 (onto the normal of the decision hyperplane) so that in this dimension 1 they are well separated.
- That is, classification is a method of radical dimensionality reduction.
- Let's look at classification from this perspective and try to achieve optimality in some sense.

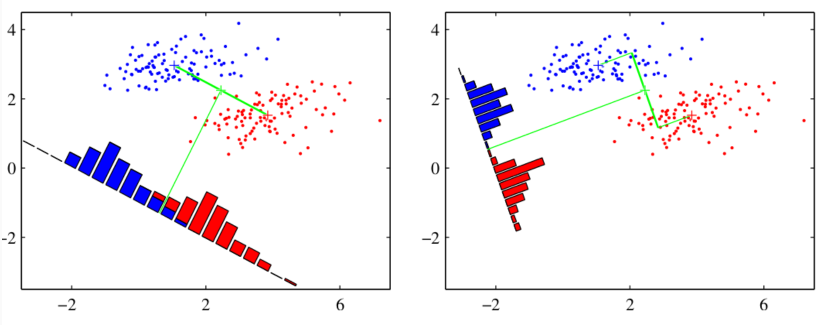
Fisher's Linear Discriminant

- Consider two classes C_1 and C_2 with N_1 and N_2 points.
- First idea – we need to find the middle perpendicular between the centers of the clusters

$$\mathbf{m}_1 = \frac{1}{N_1}\sum_{C_1}\mathbf{x}, \text{ and } \mathbf{m}_2 = \frac{1}{N_2}\sum_{C_2}\mathbf{x},$$

$$\text{i.e. maximize } \mathbf{w}^\top\left(\mathbf{m}_2 - \mathbf{m}_1\right).$$

- We need to add the constraint $\|\mathbf{w}\| = 1$, but it still doesn't work so well.



How is the left image worse than the right one?

- On the left, each cluster has greater variance.
- Idea: minimize class overlap by optimizing both the projection distance and the variance.
- Sample variances in the projection: for $y_n = \mathbf{w}^\top \mathbf{x}_n$

$$s_1 = \sum_{n \in C_1} (y_n - m_1)^2 \text{ and } s_1 = \sum_{n \in C_2} (y_n - m_2)^2 .$$

- Fisher's criterion:

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} = \frac{\mathbf{w}^\top \mathbf{S}_B \mathbf{w}}{\mathbf{w}^\top \mathbf{S}_W \mathbf{w}}, \text{ where}$$

$$\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^\top,$$

$$\mathbf{S}_W = \sum_{n \in C_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^\top + \sum_{n \in C_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^\top.$$

(between-class covariance and within-class covariance).

- Differentiating with respect to $\mathbf{w}$...

- ...we find that $J(\mathbf{w})$ is maximized when

$$\left(\mathbf{w}^\top \mathbf{S}_B \mathbf{w}\right) \mathbf{S}_W \mathbf{w} = \left(\mathbf{w}^\top \mathbf{S}_W \mathbf{w}\right) \mathbf{S}_B \mathbf{w}.$$

- Since $\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^\top$, $\mathbf{S}_B \mathbf{w}$ will still be in the direction of $\mathbf{m}_2 - \mathbf{m}_1$, and the length of $\mathbf{w}$ doesn't matter to us.
- Therefore we get

$$\mathbf{w} \propto \mathbf{S}_W^{-1} (\mathbf{m}_2 - \mathbf{m}_1).$$

- In the end, we've chosen the projection direction, and it remains only to separate the data in this projection.

- Interestingly, Fisher's discriminant can also be derived from least squares.
- Let's choose for class C_1 the target value $\frac{N_1+N_2}{N_1}$, and for class C_2 take $-\frac{N_1+N_2}{N_2}$.

Exercise. Prove that with these target values, least squares gives Fisher's discriminant.

Fisher's Linear Discriminant

- And what about multiple classes? Consider $\mathbf{y} = \mathbf{W}^\top \mathbf{x}$, generalize the within-class scatter as

$$\mathbf{S}_W = \sum_{k=1}^{K} \mathbf{S}_k = \sum_{k=1}^{K} \sum_{n \in C_k} (\mathbf{x}_n - \mathbf{m}_k)(\mathbf{x}_n - \mathbf{m}_k)^\top .$$

- To generalize the between-class scatter, simply take the remainder of the total scatter

$$\mathbf{S}_T = \sum_{n} (\mathbf{x}_n - \mathbf{m})(\mathbf{x}_n - \mathbf{m})^\top ,$$

$$\mathbf{S}_B = \mathbf{S}_T - \mathbf{S}_W .$$

- The criterion can be generalized in different ways, for example:

$$J(\mathbf{W}) = \mathrm{Tr}\left[\mathbf{s}_W^{-1}\mathbf{s}_B\right],$$

where $\mathbf{s}$ are the covariances in the projection space on $\mathbf{y}$:

$$\mathbf{s}_W = \sum_{k=1}^{K}\sum_{n\in C_k}\left(\mathbf{y}_n - \mu_k\right)\left(\mathbf{y}_n - \mu_k\right)^\top,$$

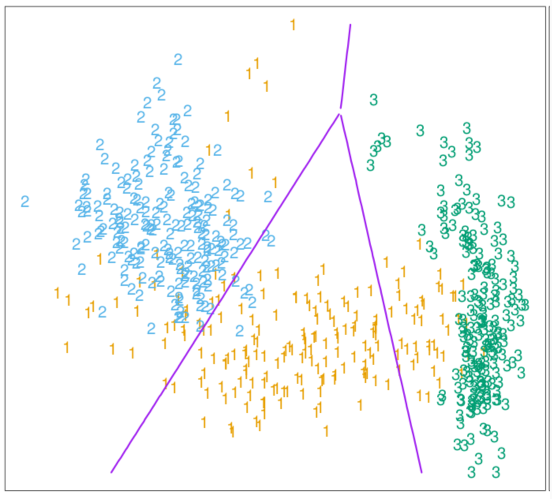
$$\mathbf{s}_B = \sum_{k=1}^{K} N_k\left(\mu_k - \mu\right)\left(\mu_k - \mu\right)^\top,$$

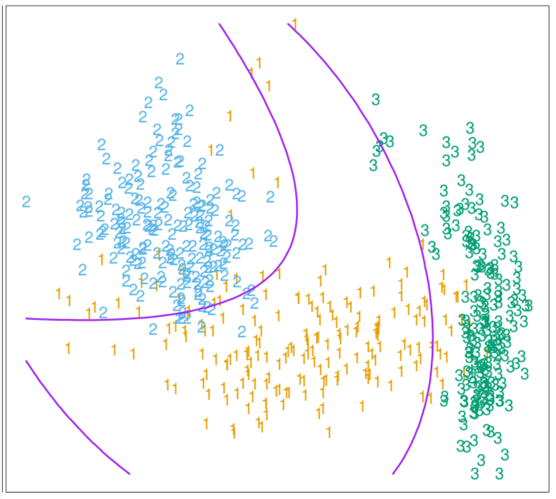
where $\mu_k = \frac{1}{N_k}\sum_{n\in C_k}\mathbf{y}_n$.

LDA and QDA

Nonlinear surfaces

- We have learned how to create separating hyperplanes.
- But what about nonlinear surfaces?
- We can make nonlinear from linear by increasing the dimensionality.





Generative models

- Now classification through generative models: let's assign a density $p(\mathbf{x} \mid C_k)$ to each class, find the prior distributions $p(C_k)$, and then find $p(C_k \mid \mathbf{x})$ using Bayes' theorem.
- For two classes:

$$p(C_1 \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_1)p(C_1)}{p(\mathbf{x} \mid C_1)p(C_1) + p(\mathbf{x} \mid C_2)p(C_2)}.$$

- Let's rewrite:

$$p(C_1 \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_1)p(C_1)}{p(\mathbf{x} \mid C_1)p(C_1) + p(\mathbf{x} \mid C_2)p(C_2)} = \frac{1}{1 + e^{-a}} = \sigma(a),$$

where

$$a = \ln \frac{p(\mathbf{x} \mid C_1)p(C_1)}{p(\mathbf{x} \mid C_2)p(C_2)}, \qquad \sigma(a) = \frac{1}{1 + e^{-a}}.$$

- $\sigma(a)$ – *logistic sigmoid*:

$$\sigma(a) = \frac{1}{1 + e^{-a}}$$

- $\sigma(-a) = 1 - \sigma(a)$.
- $a = \ln\left(\frac{\sigma}{1-\sigma}\right)$ – *logit function*.

Exercise. Prove these properties.

Multiple classes

- In the case of multiple classes, we get

$$p(C_k \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_k)p(C_k)}{\sum_j p(\mathbf{x} \mid C_j)p(C_j)} = \frac{e^{a_k}}{\sum_j e^{a_j}}.$$

- Here $a_k = \ln p(\mathbf{x} \mid C_k)p(C_k)$.
- $\frac{e^{a_k}}{\sum_j e^{a_j}}$ – normalized exponential, or softmax function (smoothed maximum).

- Let's consider Gaussian distributions for classes:

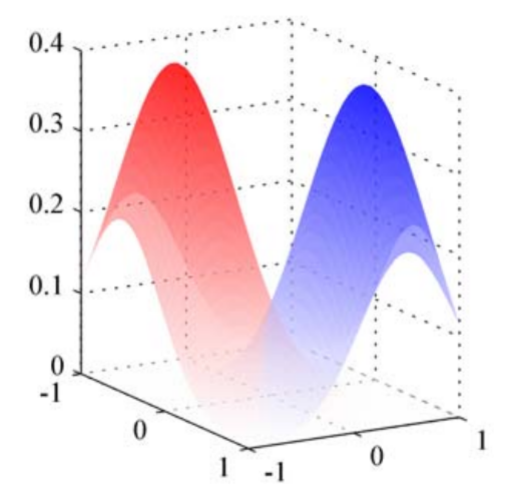
$$p(\mathbf{x} \mid C_k) = N(\mathbf{x} \mid \mu_k, \Sigma).$$

- First, let's assume that Σ is the same for all classes, and there are only two classes.
- Let's calculate the logistic sigmoid...

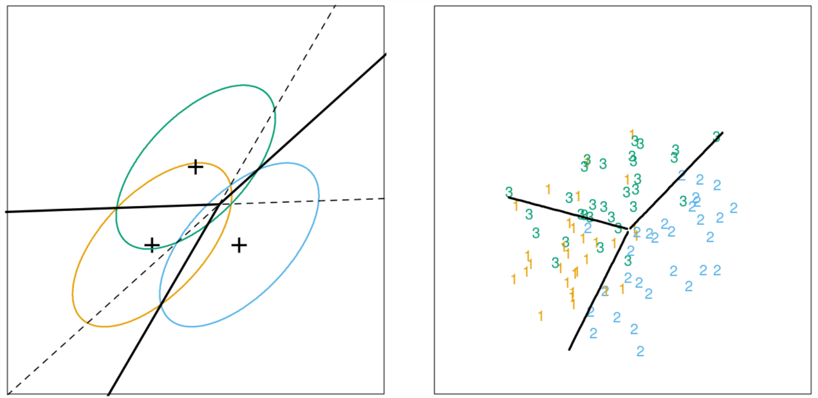
- …we get

$$\begin{aligned} p(C_1 \mid \mathbf{x}) &= \sigma(\mathbf{w}^\top \mathbf{x} + w_0), \text{ where} \\ \mathbf{w} &= \Sigma^{-1}\left(\mu_1 - \mu_2\right), \\ w_0 &= -\frac{1}{2}\mu_1^\top \Sigma^{-1} \mu_1 + \frac{1}{2}\mu_2^\top \Sigma^{-1}\mu_2 + \ln \frac{p(C_1)}{p(C_2)}. \end{aligned}$$

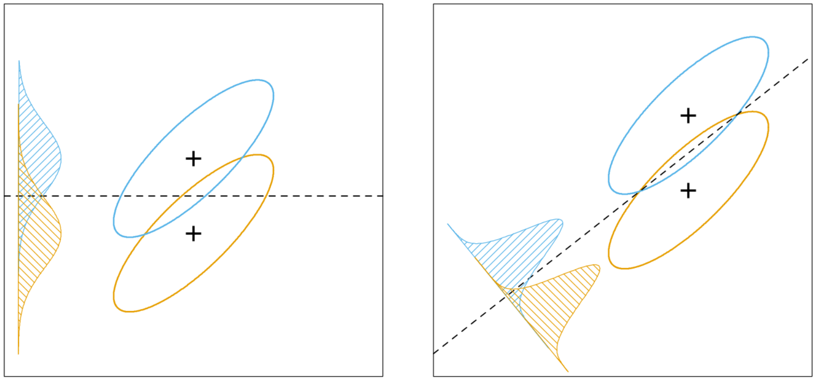
- So in the sigmoid's argument, we get a linear function of $\mathbf{x}$. Level surfaces – where $p(C_1 \mid \mathbf{x})$ is constant – are hyperplanes in the space of $\mathbf{x}$. The prior probabilities $p(C_k)$ simply shift these hyperplanes.


0.4
0.3
0.2
0.1
0
-1
0
1
-1
0
1

Decision hyperplane



By the way, this decision surface converges perfectly with Fisher's discriminant.



- With multiple classes, we get similarly:

$$\delta_k(\mathbf{x}) = \mathbf{x}^\top \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^\top \Sigma^{-1} \mu_k + \ln \pi_k,$$

  where $\pi_k = p(C_k)$.
- We get linear $\delta_k(\mathbf{x})$, and again the decision surfaces are linear (here decision surfaces occur where two maximum probabilities are equal).
- This method is called LDA – linear discriminant analysis.

- How to estimate the distributions $p(\mathbf{x} \mid C_k)$ if only data is given?
- We can use the maximum likelihood method.
- Let's consider the same example: two classes, Gaussians with the same covariance matrix, and we have $D = \{\mathbf{x}_n, t_n\}_{n=1}^{N}$, where $t_n = 1$ means C_1, $t_n = 0$ means C_2.
- Denote $p(C_1) = \pi$, $p(C_2) = 1 - \pi$.

Maximum likelihood method

- For one point in class C_1:

$$p(\mathbf{x}_n, C_1) = p(C_1)p(\mathbf{x}_n \mid C_1) = \pi N(\mathbf{x}_n \mid \mu_1, \Sigma).$$

- In class C_2:

$$p(\mathbf{x}_n, C_2) = p(C_2)p(\mathbf{x}_n \mid C_2) = (1-\pi)N(\mathbf{x}_n \mid \mu_2, \Sigma).$$

- Likelihood function:

$$p(\mathbf{t} \mid \pi, \mu_1, \mu_2, \Sigma) =$$
$$= \prod_{n=1}^{N} \left[\pi N(\mathbf{x}_n \mid \mu_1, \Sigma)\right]^{t_n} \left[(1-\pi)N(\mathbf{x}_n \mid \mu_2, \Sigma)\right]^{1-t_n}.$$

- We maximize the log-likelihood. First with respect to π, where only this remains

$$\sum_{n=1}^{N} \left[t_n \ln \pi + (1 - t_n) \ln(1 - \pi) \right],$$

and, taking the derivative, we get, quite unsurprisingly,

$$\hat{\pi} = \frac{N_1}{N_1 + N_2}.$$

- Now for μ_1; everything that depends on μ_1:

$$\sum_n t_n \ln N(\mathbf{x}_n \mid \mu_1, \Sigma) = -\frac{1}{2} \sum_n t_n \left(\mathbf{x}_n - \mu_1\right)^\top \Sigma^{-1} \left(\mathbf{x}_n - \mu_1\right) + C.$$

- Taking the derivative, we get, again quite unsurprisingly,

$$\hat{\mu}_1 = \frac{1}{N_1} \sum_{n=1}^{N} t_n \mathbf{x}_n.$$

- Similarly,

$$\hat{\mu}_2 = \frac{1}{N_2} \sum_{n=1}^{N} (1 - t_n) \mathbf{x}_n.$$

- For the covariance matrix, we'll need to work harder; the result will be

$$\hat{\Sigma} = \frac{N_1}{N_1 + N_2}\mathbf{S}_1 + \frac{N_2}{N_1 + N_2}\mathbf{S}_2, \text{ where}$$

$$\mathbf{S}_1 = \frac{1}{N_1}\sum_{n \in C_1}\left(\mathbf{x}_n - \mu_1\right)\left(\mathbf{x}_n - \mu_1\right)^\top,$$

$$\mathbf{S}_2 = \frac{1}{N_2}\sum_{n \in C_2}\left(\mathbf{x}_n - \mu_2\right)\left(\mathbf{x}_n - \mu_2\right)^\top.$$

- Also quite unsurprisingly: a weighted average of estimates for the two covariance matrices.

- This generalizes directly to the case of multiple classes.

Exercise. Do this.

- But with different covariance matrices, it will be different.
- Quadratic terms will not cancel out.
- Decision surfaces will become quadratic; QDA – quadratic discriminant analysis.

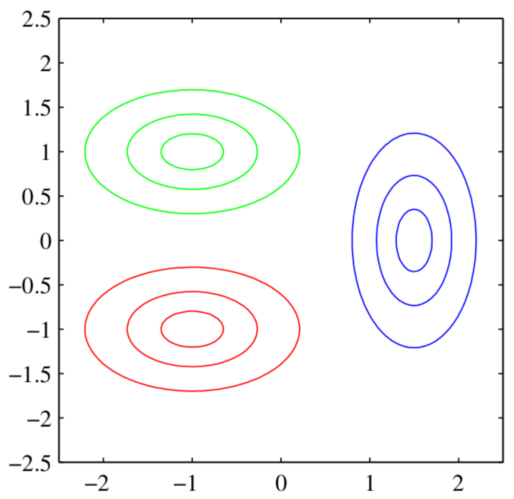
Different covariance matrices

- In QDA, we get

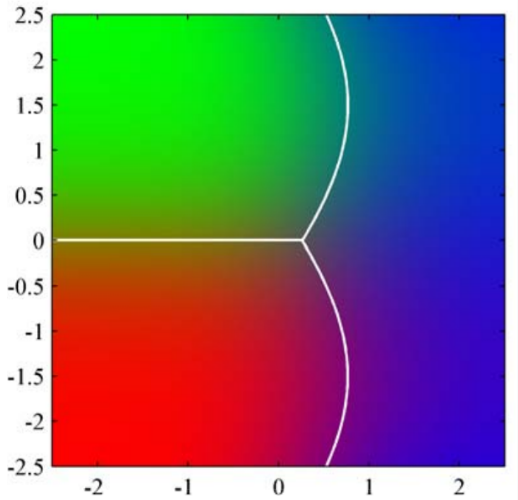
$$\delta_k(\mathbf{x}) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}\left(\mathbf{x}-\mu_k\right)^{-1}\Sigma_k^{-1}\left(\mathbf{x}-\mu_k\right) + \log\pi_k.$$

- The decision surface between C_i and C_j is $\{\mathbf{x} \mid \delta_i(\mathbf{x}) = \delta_j(\mathbf{x})\}$.
- Maximum likelihood estimates are the same, except we need to estimate covariance matrices separately.

Different covariance matrices

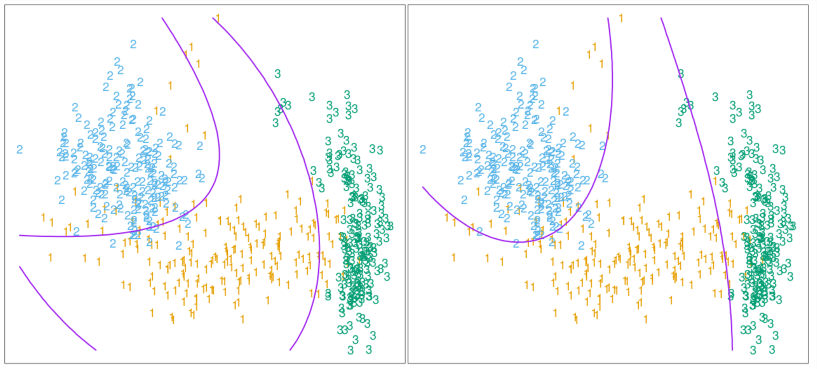


Different covariance matrices



LDA vs. QDA

The difference between LDA with quadratic terms and QDA is usually small.



LDA vs. QDA

- LDA and QDA work well in practice, better than they should!
- Number of parameters:
  - LDA has $(K-1)(d+1)$ parameters: $d+1$ for each difference of the form $\delta_k(\mathbf{x}) - \delta_K(\mathbf{x})$;
  - QDA has $(K-1)(d(d+3)/2+1)$ parameters, but it looks much better than its age.

LDA vs. QDA

- Why do they work well?
- Most likely because linear and quadratic estimates are quite stable: even if the bias is relatively large (as it will be if the data is not actually generated by Gaussians), the variance will be small.

RDA

- A compromise between LDA and QDA – regularized discriminant analysis, RDA.
- Let's shrink the covariances of each class toward the common covariance matrix:

$$\hat{\Sigma}_k(\alpha) = \alpha\hat{\Sigma}_k + (1-\alpha)\hat{\Sigma},$$

where $\hat{\Sigma}_k$ is the estimate from QDA, $\hat{\Sigma}$ is the estimate from LDA.

- Or shrink toward the identity matrix:

$$\hat{\Sigma}_k(\gamma) = \gamma\hat{\Sigma}_k + (1-\gamma)\hat{\sigma}^2\mathbf{I}.$$

- Suppose that the dimension d is greater than the number of classes K.
- Then the class centroids $\hat{\mu}_k$ lie in a subspace of dimension $\leq K - 1$.
- And when we determine the nearest centroid, we only need to calculate distances in this subspace.
- Thus, we can reduce the rank of the problem.

Rank reduction in LDA

- Where exactly to project? Not necessarily the subspace spanned by the centroids will be optimal.
- We've already covered this: for dimension 1, this is Fisher's linear discriminant.
- And that's what it is: the optimal subspace will be where the between-class variance is maximized relative to the within-class variance.

Logistic regression

- For classification problems we want to classify a vector $\mathbf{x}$ to one of K classes C_k.
- Suppose that class C_k has density $p(\mathbf{x} \mid C_k)$, find prior distributions $p(C_k)$, and then compute $p(C_k \mid \mathbf{x})$ by Bayes' theorem.
- For two classes:

$$p(C_1 \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_1)p(C_1)}{p(\mathbf{x} \mid C_1)p(C_1) + p(\mathbf{x} \mid C_2)p(C_2)}.$$

- We rewrite:

$$p(C_1 \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_1)p(C_1)}{p(\mathbf{x} \mid C_1)p(C_1) + p(\mathbf{x} \mid C_2)p(C_2)} = \frac{1}{1+e^{-a}} = \sigma(a),$$

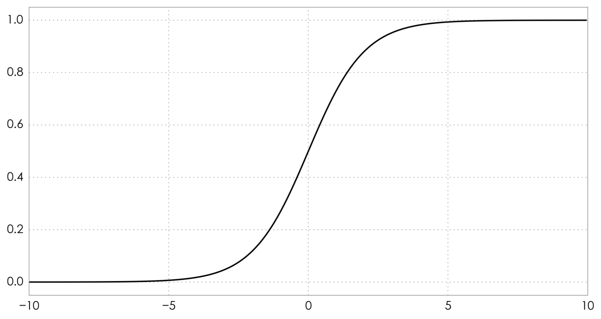
where

$$a = \ln \frac{p(\mathbf{x} \mid C_1)p(C_1)}{p(\mathbf{x} \mid C_2)p(C_2)}, \qquad \sigma(a) = \frac{1}{1+e^{-a}}.$$

- $\sigma(a)$ is the *logistic sigmoid*:

$$\sigma(a) = \frac{1}{1 + e^{-a}}$$

- $\sigma(-a) = 1 - \sigma(a)$.
- $a = \ln\left(\frac{\sigma}{1-\sigma}\right)$ – *logit function*.



Classification problems

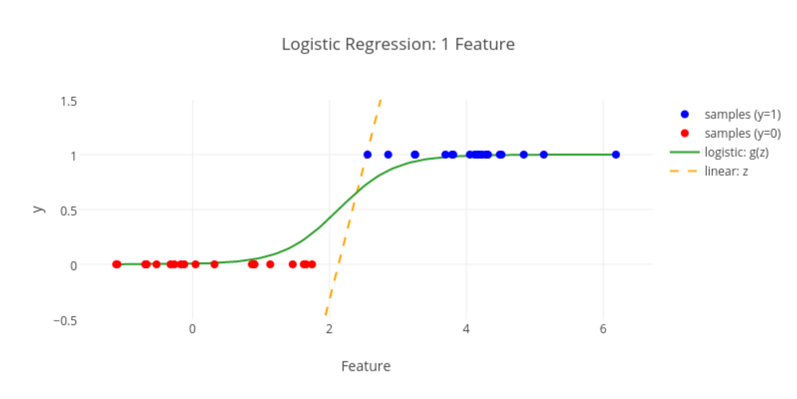
- This, in particular, leads to *logistic regression*: we optimize $\mathbf{w}$ directly.
- For a dataset $\{\phi_n, t_n\}$, $t_n \in \{0, 1\}$, $\phi_n = \phi(\mathbf{x}_n)$:

$$p(\mathbf{t} \mid \mathbf{w}) = \prod_{n=1}^{N} y_n^{t_n} (1 - y_n)^{1-t_n}, \quad y_n = p(C_1 \mid \phi_n).$$

- We find maximal likelihood parameters, minimizing $-\ln p(\mathbf{t} \mid \mathbf{w})$:

$$E(\mathbf{w}) = -\ln p(\mathbf{t} \mid \mathbf{w}) = -\sum_{n=1}^{N} \left[t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\right].$$

- And we get a sigmoid that optimally separates the data and that even tries to model probabilities:



Several classes

- In case of several classes we get

$$p(C_k \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_k)p(C_k)}{\sum_j p(\mathbf{x} \mid C_j)p(C_j)} = \frac{e^{a_k}}{\sum_j e^{a_j}}.$$

- Here $a_k = \ln p(\mathbf{x} \mid C_k)p(C_k)$.
- $\frac{e^{a_k}}{\sum_j e^{a_j}}$ is the normalized exponent (softmax).
- Conclusion: for a classification problem it makes sense to minimize the cross-entropy $\sum_{n=1}^{N} [t_n \ln y_n + (1 - t_n) \ln(1 - y_n)]$ and softmax (rather than classification error, which is problematic).
- One question remains: how do we optimize all this?
- How do we optimize complicated functions in general?

Thank you for your attention!