

7 Gaussian Discriminant Analysis, including QDA and LDA

GAUSSIAN DISCRIMINANT ANALYSIS

Fundamental assumption: **each class comes from normal distribution** (Gaussian).

$$X \sim N(\mu, \sigma^2) : f(x) = \frac{1}{(\sqrt{2\pi}\sigma)^d} \exp\left(-\frac{\|x - \mu\|^2}{2\sigma^2}\right) \quad [\mu \text{ \& } x = \text{vectors}; \sigma = \text{scalar}; d = \text{dimension}]$$

For each class C , suppose we **estimate** mean μ_C , variance σ_C^2 , and **prior** $\pi_C = P(Y = C)$.

Given x , Bayes decision rule $r^*(x)$ **predicts class C that maximizes $f(X = x|Y = C)\pi_C$.**

$\ln \omega$ is monotonically increasing for $\omega > 0$, so it is equivalent to maximize

$$Q_C(x) = \ln\left((\sqrt{2\pi})^d f_C(x) \pi_C\right) = -\frac{\|x - \mu_C\|^2}{2\sigma_C^2} - d \ln \sigma_C + \ln \pi_C \quad \Rightarrow \text{quadratic in } x.$$

\uparrow quadratic in x . \uparrow normal PDF, estimates $f(X = x|Y = C)$

[In a 2-class problem, you can also incorporate an asymmetrical loss function the same way we incorporate the prior π_C . In a multi-class problem, it gets more difficult, because the penalty for guessing wrong might depend not just on the wrong guess, but also on the true class.]

Note: Posterior Probability
 $\frac{f(x|x|Y=C)\pi_C}{\sum_C f(x|x|Y=C)\pi_C} \leftarrow \text{ignore denominator (denominator same for every class)}$
 $\times (\text{optional Loss}(\cdot, \cdot)) P(x) \leftarrow \text{ignore denominator (denominator same for every class)}$
 and the $\frac{1}{\sum_C \pi_C}$ denom in $f(x)$ is also same for every class, so leave out.

Quadratic Discriminant Analysis (QDA)

Suppose only 2 classes C, D . Then

$$r^*(x) = \begin{cases} C & \text{if } Q_C(x) - Q_D(x) > 0, \\ D & \text{otherwise.} \end{cases}$$

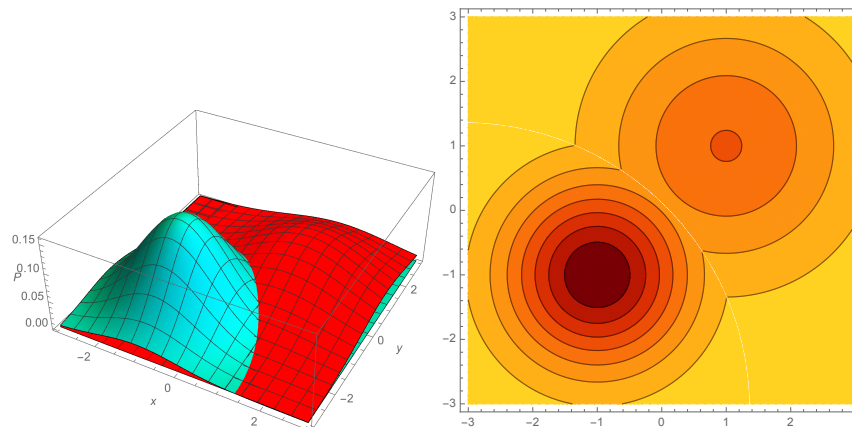
[Pick the class with the biggest posterior probability]

Decision fn is quadratic in x . **Bayes decision boundary is $Q_C(x) - Q_D(x) = 0$.**

- In 1D, B.d.b. may have 1 or 2 points.
- In d -D, B.d.b. is a quadric.

[Solutions to a quadratic equation]

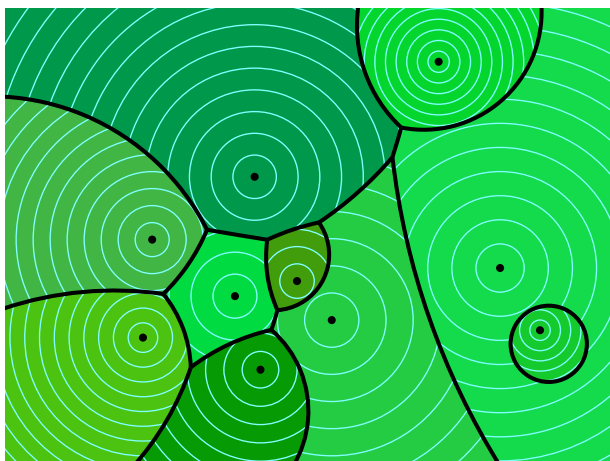
[In 2D, that's a conic section]



qda3d.pdf, qdacontour.pdf [The same example I showed during the previous lecture.]

[You're probably familiar with the Gaussian distribution where x and μ are scalars, but as I've written it, it applies equally well to a multi-dimensional feature space with isotropic Gaussians. Then x and μ are vectors, but the variance σ is still a scalar. Next lecture we'll look at anisotropic Gaussian distributions where the variance is different along different directions.]

[QDA works very naturally with more than 2 classes.]



multiplicative.pdf

[The feature space gets partitioned into regions. In two or more dimensions, you typically wind up with multiple decision boundaries that adjoin each other at joints. It looks like a sort of Voronoi diagram. In fact, it's a special kind of Voronoi diagram called a multiplicatively, additively weighted Voronoi diagram.]

[You might not be satisfied with just knowing how each point is classified. One of the great things about QDA is that you can also determine the probability that your classification is correct. Let's work that out.]

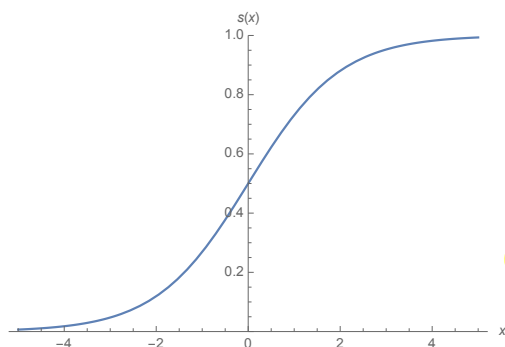
To recover posterior probabilities in 2-class case, use Bayes:

$$P(Y = C|X) = \frac{f(X|Y = C) \pi_C}{f(X|Y = C) \pi_C + f(X|Y = D) \pi_D}$$

$$\text{recall } e^{Q_C(x)} = (\sqrt{2\pi})^d f_C(x) \pi_C \quad [\text{by definition of } Q_C]$$

$$\begin{aligned} P(Y = C|X = x) &= \frac{e^{Q_C(x)}}{e^{Q_C(x)} + e^{Q_D(x)}} = \frac{1}{1 + e^{Q_D(x) - Q_C(x)}} \\ &= s(Q_C(x) - Q_D(x)), \quad \text{where} \end{aligned}$$

$$s(\gamma) = \frac{1}{1 + e^{-\gamma}} \quad \Leftarrow \text{logistic fn aka sigmoid fn} \quad [\text{recall } Q_C - Q_D \text{ is the decision fn}]$$



logistic.pdf

[The logistic function. Write beside it:] $s(0) = \frac{1}{2}$, $s(\infty) \rightarrow 1$, $s(-\infty) \rightarrow 0$, monotonically increasing.

[We interpret $s(0) = \frac{1}{2}$ as saying that on the decision boundary, there's a 50% chance of class C and a 50% chance of class D.]

Linear Discriminant Analysis (LDA)

[LDA is a variant of QDA with linear decision boundaries. It's **less likely to overfit** than QDA.]

Fundamental assumption: all the Gaussians **have same variance σ** .

[The equations simplify nicely in this case.]

$$Q_C(x) - Q_D(x) = \underbrace{\frac{(\mu_C - \mu_D) \cdot x}{\sigma^2}}_{w \cdot x} - \underbrace{\frac{\|\mu_C\|^2 - \|\mu_D\|^2}{2\sigma^2}}_{+\alpha} + \ln \pi_C - \ln \pi_D$$

[The **quadratic terms in Q_C and Q_D canceled each other out!**]

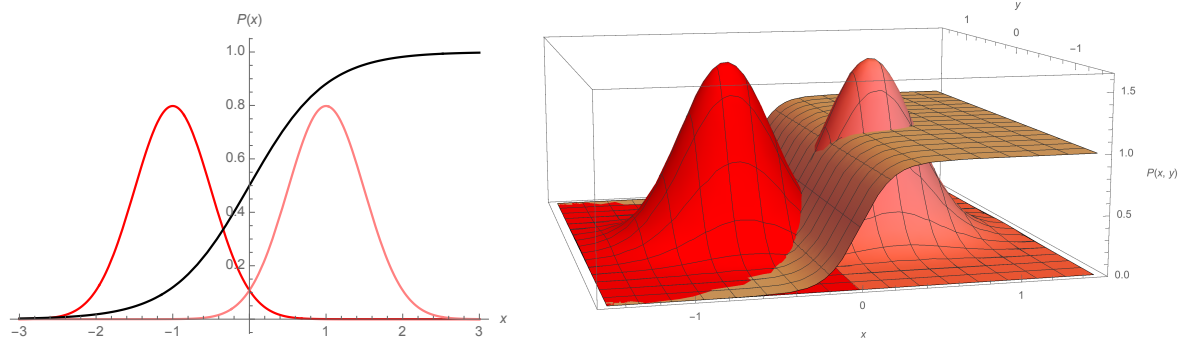
Now it's a linear classifier! Choose C that maximizes linear discriminant fn

$$\frac{\mu_C \cdot x}{\sigma^2} - \frac{\|\mu_C\|^2}{2\sigma^2} + \ln \pi_C \quad \text{[this works for any number of classes]}$$

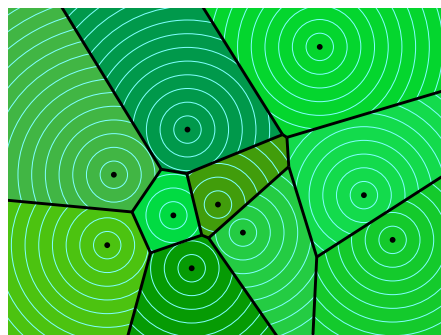
In 2-class case: decision boundary is **$w \cdot x + \alpha = 0$**

posterior is $P(Y = C|X = x) = s(w \cdot x + \alpha)$

[The effect of " $w \cdot x + \alpha$ " is to scale and translate the logistic fn in x -space. It's a linear transformation.]



lda1d.pdf, lda2d.pdf [Two Gaussians (red) and the logistic function (black). The logistic function is the right Gaussian divided by the sum of the Gaussians. Observe that even when the Gaussians are 2D, the logistic function still looks 1D.]



voronoi.pdf [When you have many classes, their LDA decision boundaries form a classical Voronoi diagram if the priors π_C are equal. All the Gaussians have the same width.]

$$\text{If } \pi_C = \pi_D = \frac{1}{2} \Rightarrow (\mu_C - \mu_D) \cdot x - (\mu_C - \mu_D) \cdot \left(\frac{\mu_C + \mu_D}{2} \right) = 0$$

This **is the centroid method!**

MAXIMUM LIKELIHOOD ESTIMATION OF PARAMETERS (Ronald Fisher, circa 1912)

[To use Gaussian discriminant analysis, we must first fit Gaussians to the sample points and estimate the class prior probabilities. We'll do priors first—they're easier, because they involve a discrete distribution. Then we'll fit the Gaussians—they're less intuitive, because they're continuous distributions.]

Let's flip biased coins! Heads with probability p ; tails w/prob. $1 - p$.

10 flips, 8 heads, 2 tails. [Let me ask you a weird question.] What is the most likely value of p ?

of heads is $X \sim \mathcal{B}(n, p)$, binomial distribution:

$$P[X = x] = \binom{n}{x} p^x (1 - p)^{n-x} \quad \text{[this is the probability of getting exactly } x \text{ heads in } n \text{ coin flips]}$$

Our example: $n = 10$,

$$P[X = 8] = 45p^8 (1 - p)^2 \stackrel{\text{def}}{=} \mathcal{L}(p)$$

Probability of 8 heads in 10 flips:

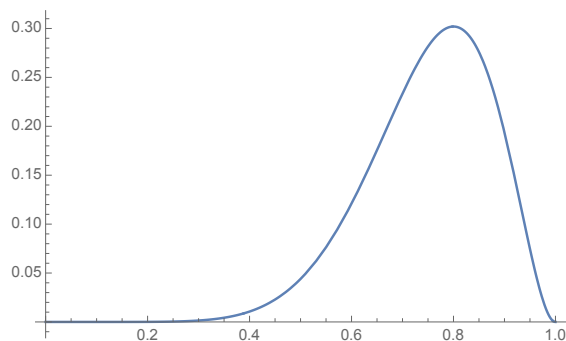
written as a fn $\mathcal{L}(p)$ of distribution parameter(s), this is the likelihood fn.

Maximum likelihood estimation (MLE): A method of estimating the parameters of a statistical model by picking the params that maximize [the likelihood function] \mathcal{L} .

... is one method of density estimation: estimating a PDF [probability density function] from data.

[Let's phrase it as an optimization problem.]

Find p that maximizes $\mathcal{L}(p)$.



binomlikelihood.pdf [Graph of $\mathcal{L}(p)$ for this example.]

Solve this example by setting derivative = 0:

$$\begin{aligned} \frac{d\mathcal{L}}{dp} &= 360p^7(1 - p)^2 - 90p^8(1 - p) = 0 \\ \Rightarrow 4(1 - p) - p &= 0 \Rightarrow p = 0.8 \end{aligned}$$

[It shouldn't seem surprising that a coin that comes up heads 80% of the time is the coin most likely to produce 8 heads in 10 flips.]

[Note: $\frac{d^2\mathcal{L}}{dp^2} \doteq -18.9 < 0$ at $p = 0.8$, confirming it's a maximum.]

[Here's how this applies to prior probabilities. Suppose our data set is 10 sample points, and 8 of them are of class C and 2 are not. Then our estimated prior for class C will be $\pi_C = 0.8$.]

Likelihood of a Gaussian

Given sample points X_1, X_2, \dots, X_n , find best-fit Gaussian.

[Now we want a normal distribution instead of a binomial distribution. If you generate a random point from a normal distribution, what is the probability that it will be exactly at X_1 ?]

[Zero. So it might seem like we have a problem here. With a continuous distribution, the probability of generating any particular point is zero. But we're just going to ignore that and do "likelihood" anyway.]

Likelihood of generating these points is

$$\mathcal{L}(\mu, \sigma; X_1, \dots, X_n) = f(X_1) f(X_2) \cdots f(X_n).$$

[How do we maximize this?]

The log likelihood $\ell(\cdot)$ is the \ln of the likelihood $\mathcal{L}(\cdot)$.

Maximizing likelihood \Leftrightarrow maximizing log likelihood.

$$\begin{aligned} \ell(\mu, \sigma; X_1, \dots, X_n) &= \ln f(X_1) + \ln f(X_2) + \dots + \ln f(X_n) \\ &= \sum_{i=1}^n \underbrace{\left(-\frac{\|X_i - \mu\|^2}{2\sigma^2} - d \ln \sqrt{2\pi} - d \ln \sigma \right)}_{\text{ln of normal PDF}} \end{aligned}$$

Want to set $\nabla_{\mu} \ell = 0$, $\frac{\partial \ell}{\partial \sigma} = 0$

$$\nabla_{\mu} \ell = \sum_{i=1}^n \frac{X_i - \mu}{\sigma^2} = 0 \quad \Rightarrow \quad \hat{\mu} = \frac{1}{n} \sum_{i=1}^n X_i$$

[The hats ^ mean "estimated"]

$$\frac{\partial \ell}{\partial \sigma} = \sum_{i=1}^n \frac{\|X_i - \mu\|^2 - d\sigma^2}{\sigma^3} = 0 \quad \Rightarrow \quad \hat{\sigma}^2 = \frac{1}{dn} \sum_{i=1}^n \|X_i - \mu\|^2$$

We don't know μ exactly, so substitute $\hat{\mu}$ for μ to compute $\hat{\sigma}$.

I.e., we use sample mean & variance of pts in class C to estimate mean & variance of Gaussian for class C.

For QDA: estimate conditional mean $\hat{\mu}_C$ & conditional variance $\hat{\sigma}_C^2$ of **each class C separately** [as above] & estimate the priors:

$$\hat{\pi}_C = \frac{n_C}{\sum_D n_D} \quad \Leftarrow \quad \text{total sample points in all classes}$$

[$\hat{\pi}_C$ is the coin flip parameter]

For LDA: same means & priors; **one variance for all classes**:

$$\hat{\sigma}^2 = \frac{1}{dn} \sum_C \sum_{\{i: y_i = C\}} \|X_i - \hat{\mu}_C\|^2 \quad \Leftarrow \quad \text{pooled within-class variance}$$

[Notice that although LDA is computing one variance for all the data, each sample point contributes with respect to its own class's mean. This gives a very different result than if you simply use the global mean! It's usually smaller than the global variance. We say "within-class" because we use each point's distance from its class's mean, but "pooled" because we then pool all the classes together.]