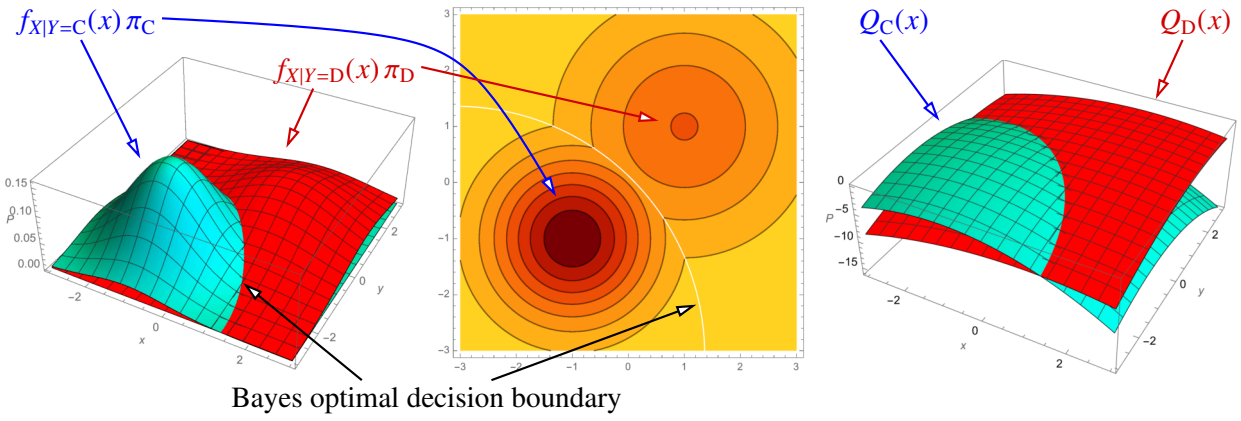
7 Gaussian Discriminant Analysis, including QDA and LDA

GAUSSIAN DISCRIMINANT ANALYSIS

Fundamental assumption: each class has a normal distribution [a Gaussian].

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

For each class C, suppose we know mean μ_C and variance σ_C^2, yielding PDF $f_{X|Y=C}(x)$,
and prior $\pi_C = P(Y = C)$.



qda3d.pdf, qdacontour.pdf, Q.pdf [Probability density functions for two classes.]

[This PDF is halfway between the univariate normal distribution and the standard multivariate normal distribution. It is multivariate: x and μ can be vectors, and I've plotted an example in a 2D feature space. But the variance σ^2 is just a scalar; for simplicity, we will avoid the covariance matrix until next lecture. That's why the isocontours are circles and not ellipses. I call this the *isotropic normal distribution*, because the variance is the same in every direction. Next lecture, we'll look at anisotropic Gaussians where the isosurfaces are ellipsoids. Here, the Bayes optimal decision boundary is an ellipse.]

Given x, Bayes decision rule $r^*(x)$ predicts class C that maximizes $f_{X|Y=C}(x)\,\pi_C$.
[Remember our last lecture's main principle: **pick the class with the biggest posterior probability!**]

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

$$Q_C(x) = \ln\left((\sqrt{2\pi})^d f_{X|Y=C}(x)\,\pi_C\right) = -\frac{\|x-\mu_C\|^2}{2\sigma_C^2} - d\ln\sigma_C + \ln\pi_C. \qquad [Q_C \text{ is quadratic in } x]$$

[In a 2-class problem, you can also incorporate an asymmetrical loss function by adding $\ln L(\text{not C}, \text{C})$ to $Q_C(x)$. In a multi-class problem, asymmetric loss is more difficult to account for, because the penalty for guessing wrong might depend on both the wrong guess and the true class.]

Quadratic Discriminant Analysis (QDA)

Suppose only 2 classes C, D. Then the Bayes classifier is

$$r^*(x) = \begin{cases} \text{C} & \text{if } Q_C(x) - Q_D(x) > 0, \\ \text{D} & \text{otherwise.} \end{cases} \qquad [\text{Picks the class with the biggest posterior probability}]$$

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

- In 1D, B.d.b. may have 1 or 2 points. [Solutions to a quadratic equation]
- In d-D, B.d.b. is a quadric. [In 2D, that's a conic section; see figure above]

[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 estimate the probability that your prediction is correct. Let's work that out.]

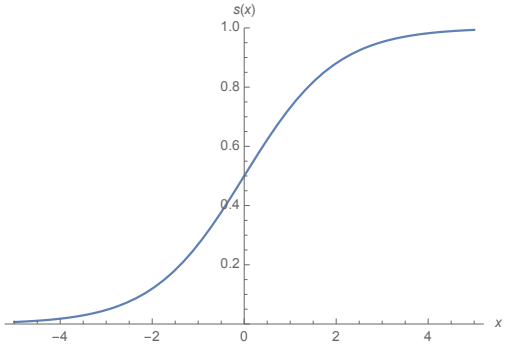
To recover posterior probabilities in 2-class case, use Bayes.

$$P(Y = \mathrm{C}|X) = \frac{f_{X|Y=\mathrm{C}}\,\pi_\mathrm{C}}{f_{X|Y=\mathrm{C}}\,\pi_\mathrm{C} + f_{X|Y=\mathrm{D}}\,\pi_\mathrm{D}}$$

recall $e^{Q_\mathrm{C}(x)} = (\sqrt{2\pi})^d\, f_{X|Y=\mathrm{C}}(x)\,\pi_\mathrm{C}$ [by definition of Q_C]

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

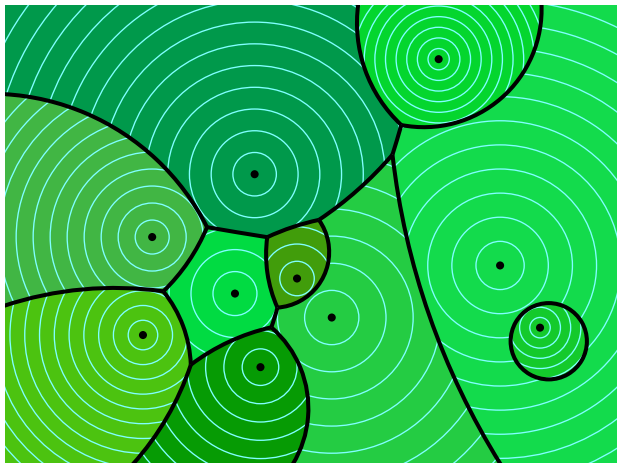
$$s(\gamma) = \frac{1}{1 + e^{-\gamma}} \quad \Leftarrow \underline{\text{logistic fn}} \text{ aka } \underline{\text{sigmoid fn}}$$ [recall $Q_C - Q_D$ is the decision fn]



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

Multi-class QDA: [QDA works very naturally with more than 2 classes.]



multiplicative.pdf [Multi-class QDA partitions the feature space 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.]

Linear Discriminant Analysis (LDA)

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

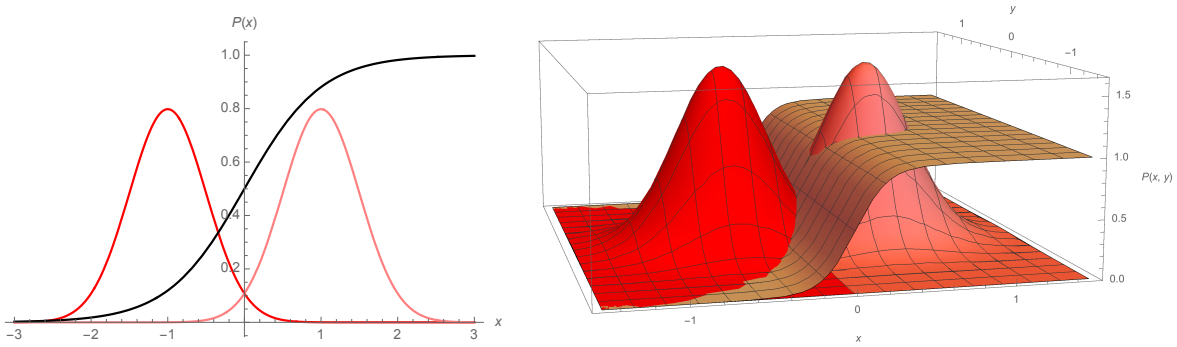
Fundamental assumption: all the Gaussians have same variance σ^2.
[The equations simplify nicely in this case.]

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

[The quadratic terms in Q_C and Q_D canceled each other out!]
Now it's a linear classifier!

- decision boundary is $w \cdot x + \alpha = 0$
- posterior is $P(Y = C|X = x) = s(w \cdot x + \alpha)$

[The effect of "$s(w \cdot x + \alpha)$" is to scale and translate the logistic fn in x-space.]

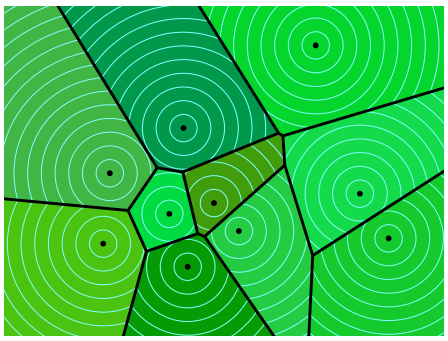


lda1d.pdf, lda2d.pdf [Two Gaussians (red) and the logistic function (black). The logistic function is the right Gaussian divided by the sum of the Gaussians. Observe that even when the Gaussians are 2D, the logistic function still looks 1D.]

Special case: if $\pi_C = \pi_D = \frac{1}{2} \quad \Rightarrow \quad (\mu_C - \mu_D) \cdot x - (\mu_C - \mu_D) \cdot \left(\frac{\mu_C + \mu_D}{2} \right) = 0.$

This is the centroid method!

Multi-class LDA: choose C that maximizes linear discriminant fn $\frac{\mu_C \cdot x}{\sigma^2} - \frac{\|\mu_C\|^2}{2\sigma^2} + \ln \pi_C$.



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.]

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. $1 - p$. [But we don't know 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:

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

Prob. of $x = 8$ heads in $n = 10$ flips is

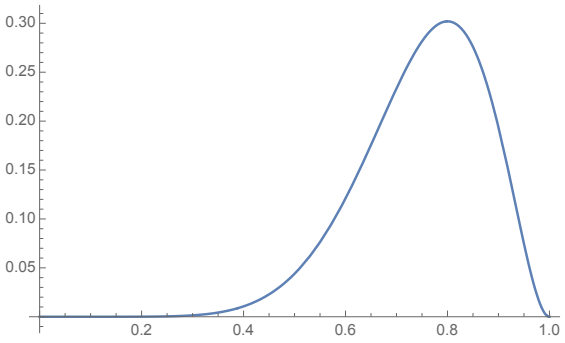
$$P[X = 8] = 45p^8 (1-p)^2 \qquad \stackrel{\text{def}}{=} \mathcal{L}(p)$$

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

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 by finding critical point of $\mathcal{L}$:

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}p} = 360p^7(1-p)^2 - 90p^8(1-p) = 0$$

$$\Rightarrow \quad 4(1-p) - p = 0 \quad \Rightarrow \quad p = 0.8$$

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

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

[Here's how this applies to prior probabilities.]
Suppose our training set is n points, with x in class C. Then our estimated prior for class C is $\hat{\pi}_\mathrm{C} = x/n$.

Likelihood of a Gaussian

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

[Now we want to fit a normal distribution to data, instead of a binomial distribution. If you draw 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 drawing these points [in the specified order] is

$$\mathcal{L}(\mu, \sigma; X_1, \ldots, X_n) = f(X_1)\, f(X_2) \cdots f(X_n). \qquad \text{[How do we maximize this?]}$$

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

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

Set $\nabla_\mu \ell = 0, \dfrac{\partial \ell}{\partial \sigma} = 0$ [Find the critical point of ℓ]

$$\nabla_\mu \ell = \sum_{i=1}^{n} \frac{X_i - \mu}{\sigma^2} = 0 \quad \Rightarrow \quad \hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} X_i \qquad \text{[The hats ˆ mean "estimated"]}$$

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

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

Takeaway: 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:

$$\hat{\pi}_C = \frac{n_C}{\sum_D n_D} \quad \Leftarrow \quad \text{total sample points in all classes} \qquad [\hat{\pi}_C \text{ is the coin flip parameter}]$$

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

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

[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.]