# Linear Classification: Regression & LDA for Classification (and final remarks on the Lasso & Dantzig)

MATH 697 AM:ST

October 17th, 2017

(Today we wrap up “linear” regression, and start with classification)

# Warmup

## Cutting space with hyperplanes

Given two affine functions $\ell_1(x)$ and $\ell_2(x)$, they define two halfspaces

$$\{x \mid \ell_1(x) > \ell_2(x)\} \text{ and } \{x \mid \ell_1(x) < \ell_2(x)\}$$

which result from dividing in space by the hyperplane $\{x \mid \ell_1(x) = \ell_2(x)\}$.

# Warmup

## Cutting space with hyperplanes

Given three affine functions $\ell_1(x), \ell_2(x)$ and $\ell_3(x)$, we may consider the sets

$$\{x \mid \ell_1(x) > \ell_2(x) \text{ and } \ell_1(x) > \ell_3(x)\}$$
$$\{x \mid \ell_2(x) > \ell_1(x) \text{ and } \ell_2(x) > \ell_3(x)\}$$
$$\{x \mid \ell_3(x) > \ell_1(x) \text{ and } \ell_3(x) > \ell_2(x)\}$$

How complicated can these sets be? Are there any special configurations?

# Warmup

## Cutting space with hyperplanes

Given three affine functions $\ell_1(x), \ell_2(x)$ and $\ell_3(x)$, we may consider the sets

$$\{x \mid \ell_1(x) > \ell_2(x) \text{ and } \ell_1(x) > \ell_3(x)\}$$
$$\{x \mid \ell_2(x) > \ell_1(x) \text{ and } \ell_2(x) > \ell_3(x)\}$$
$$\{x \mid \ell_3(x) > \ell_1(x) \text{ and } \ell_3(x) > \ell_2(x)\}$$

How complicated can these sets be? Are there any special configurations?

In each case above, the resulting set* is a **convex** angle, being the **intersection** of two halfspaces.

*except for a few degenerate cases

# Warmup

## Cutting space with hyperplanes

If we start varying the functions $\ell_1, \ell_2, \ell_3$, how do these three sets change?

Well, by changing the $\ell_i$'s, can we obtain?

1. an empty set?
2. a bounded set?
3. repeated sets?

# Warmup

## Cutting space with hyperplanes

In general, if we have $K$ affine functions $\ell_1, \ldots, \ell_m$, what can be said about the sets

$$\{x \mid \ell_k(x) > \ell_j(x) \ \ \forall\, j \neq k\}$$

for $k = 1, 2, \ldots, m$?

# The (Grouped) Lasso

In this instance, we think of the predictors $\beta \in \mathbb{R}^p$ and their coefficients as divided in $K$ subgroups, so that

$$\beta = (\beta_1, \ldots, \beta_K),\ \beta \in \mathbb{R}^{p_k}$$
$$\text{where } p_1 + \ldots + p_K = p$$

Then, one seeks to minimize

$$|\mathbf{y} - \beta_0 \mathbf{1} - \sum_{k=1}^{K} \mathbf{X}_k \beta_k|_{\ell^2}^2 + \lambda \sum_{k=1}^{K} \sqrt{p_k} |\beta_k|_{\ell^1}$$

The minimizers for this problem then tend to be sparse both at the group level ($\beta_j = 0$) and at the individual level (the non-zero $\beta_j$'s tend to have a few coefficients equal to zero).

# The Lasso + Gauss

The Gauss–Lasso estimator aims to take advantage of the **variable selection** feature of the Lasso, while mitigating the Lasso's underlying bias.

Given a set of indices $\mathcal{I} \subset [1, \dots, p]$ let $\pi_{\mathcal{I}}$ be given by

$$\pi_{\mathcal{I}} : \mathbb{R}^p \mapsto \mathbb{R}^p,\ \pi_{\mathcal{I}}(x)_i = x_i \chi_{\mathcal{I}}(i)$$

i.e. $\pi$ is the orthogonal projection onto the space of vectors whose only nonzero coordinates are those whose indices are in $\mathcal{I}$.

# The Lasso + Gauss

Then, the Gauss–Lasso estimator $\hat{\beta}^{GL}$ is defined as follows: let

$$\mathrm{I}(\beta) := \{i \mid \hat{\beta}_i \neq 0\}$$

Then one takes the **projection**

$$\hat{\beta}^{\mathrm{GL}} = \pi_{\mathcal{I}(\hat{\beta}^{\mathrm{L}})}(\hat{\beta})$$

# The Dantzig Selector

$p >> N$

Let us consider one last time the problem of finding $\beta$ such that

$$y = X \cdot \beta + \text{ (as small an error as possible)}$$

in the situation where, the input space dimension, $p$, is much larger than the sample size, $N$.

As soon as $p > N$, we have that

$$\mathbf{X}^t\mathbf{X} \text{ cannot be invertible!}$$

and least squares starts running into problems.

In 2007, Candes and Tao proposed an estimator that tries to deal with this problem.

# The Dantzig Selector

$p >> N$

One considers a matrix of inputs $\mathbf{X}$ and a vector of scalar observations $\mathbf{y}$ (both random or not), then for a given parameter $s > 0$ one defines the Dantzig estimator,

$$\hat{\beta}^{\mathrm{D}} := \operatorname{argmin}\{\ |\beta|_{\ell^1}\ \ |\ \ |\mathbf{X}^t(\mathbf{X}\beta - \mathbf{y})|_\infty \leq s\}$$

This is, like the Lasso, given by a convex optimization problem with finitely many linear constraints.

# The Dantzig Selector

$p >> N$

Besides $p >> N$, the estimator $\hat{\beta}^{\mathrm{D}}$ works specially well at estimating linear models of the form

$$\mathbf{y} = \mathbf{X}\beta_0 + \bar{\varepsilon}$$

where $\mathbf{X}$ is allowed to be random and one assumes that $\bar{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_N)$ is a Gaussian vector (independent of $\mathbf{X}$) whose coordinates are mean zero, uncorrelated, and with common variance $\sigma^2$.

# The Dantzig Selector

The Dantzig estimator is in many ways similar to the Lasso.

**Example**

Assume that $p = N$ and that $\mathbf{X} = \mathbf{I}$, then, one can check that

$$\hat{\beta}^{\mathrm{D}} = \mathcal{S}_s(\mathbf{y})$$

Indeed, in this case the constraint

$$|\mathbf{X}^t(\mathbf{X}\beta - \mathbf{y})|_\infty \leq s$$

reduces to the constraints

$$|\beta_i - \mathbf{y}_i| \leq s, \ \ \text{for } i = 1, 2, \ldots$$

# The Dantzig Selector

The Dantzig estimator is in many ways similar to the Lasso.

**Example**

…which in turn become

$$|\beta_i - (\beta_0)_i - \varepsilon_i| \leq s, \quad \text{for } i = 1, 2, \ldots$$

these constraints are uncoupled between the different variables, so one simply minimizes each $|\beta_i|$ independently, yielding

$$\hat{\beta}_i^{\mathrm{D}} = \mathcal{S}_s(\mathrm{y}_i)$$

# The Dantzig Selector

How does the Dantzig selector behave in general?

Define the vector

$$\mathbf{y} = \mathbf{X}\beta_0 + \bar{\varepsilon}$$

for some $\beta_0 \in \mathbb{R}^p$, a $\mathbf{X}$ which is deterministic, and $\bar{\varepsilon}$ a Gaussian in $\mathbb{R}^N$ with i.i.d. entries, mean zero, and common variance $\sigma^2$.

Then, of course, we want to estimate $\beta_0$ from $\mathbf{y}$...

# The Dantzig Selector

Since $p >> N$, one should expect there to be lots of vectors $\beta'$ such that $\mathbf{X}\beta' = \mathbf{y}$.

One of the main observations of Candes and Tao is that for certain classes of matrices $\mathbf{X}$, there are only a few such $\beta'$ which are also **sparse**.

Given $k < p$ a vector is said to be $k$-sparse if it has **at most** $k$ **non-zero coefficients**.

# The Dantzig Selector

## UUP Matrices

Indeed, Candes and Tao noted a property shared by random matrices $\mathbf{X}$ which arise as the input matrices of a set of $N$ i.i.d. normal vectors. This is known as the Uniform Uncertainty Principle property.

Note that if the columns of $\mathbf{X}$ are orthonormal vectors, then

$$|\mathbf{X}\beta|_2 = |\beta|_2 \ \forall \ \beta \in \mathbb{R}^p$$

That is, $\mathbf{X}$ places $\mathbb{R}^p$ inside $\mathbb{R}^N$ via an isometry.

# The Dantzig Selector

## UUP Matrices

Of course, if $\mathbb{R}^p$ is placed inside $\mathbb{R}^N$ isometrically, we must necessarily have $p \leq N$. $\mathbf{X}$ cannot be an isometry if $p > N$!.

How can we relax this?

1. Ask that now, one has instead the inequalities

   $$(1-\delta)|\beta|_2 \leq |\mathbf{X}\beta|_2 \leq (1+\delta)|\beta|_2 \; \forall \; \beta \in \mathbb{R}^p$$

   for some $\delta \in (0,1)$ (typically $\delta \approx 0$).
2. The above is still too strong! Instead, ask

   $$(1-\delta)|\beta|_2 \leq |\mathbf{X}\beta|_2 \leq (1+\delta)|\beta|_2$$

   but **only for those $\beta$'s which are $k$-sparse.**

# The Dantzig Selector

## UUP Matrices

If a matrix $\mathbf{X}$ satisfies the last condition, it is said to have the Uniform Uncertainty Principle (UUP) of order $k$ with constant $\delta$.

This condition is also known as the Restricted Isometry Property (RIP).

# The Dantzig Selector

## UUP Matrices

**Theorem** (Candes, Tao 2007)

Assume that $\beta_0$ is $k$-sparse and $\mathbf{X}$ satisfies the UUP with this $k$ and some $\delta \in (0,1)$...

There is a constant $C$ (determined by $\mathbf{X}$ alone) such that, if we choose $s = \sqrt{2(1+a)\log(p)}$, then

$$\|\beta_0 - \beta^{\mathrm{D}}\|_{\ell^2} \leq C_1^2(2(1+a)\log(p))k\sigma^2$$

with probability larger than $1 - \left(\sqrt{\pi \log(p)}p^a\right)^{-1}$.

# The Dantzig Selector

Some remarks:

1. Observe that in the absence of randomness, that is, $\sigma = 0$, then this theorem says we recover the **exact** solution.
2. For non-zero $\sigma$, by making $a$ larger, we get an exponential improvement in our estimate of the probability, with only an increase in the size of our error that is linear with respect to $a$.
3. The estimate's dependence in $p$ is logarithmic.
4. The constant $C_1$ can actually be computed explicitly in terms of the UUP constants for $\mathbf{X}$.

# Linear Regression: A Summary (Part I)

1. Standard least squares produce unbiased estimators, however, by going beyond strictly linear estimators, one usually can get lower error at the expense of non-zero bias.
2. Shrinkage methods, such as Ridge and Lasso, are good models that provide better generalization error even though they are unbiased.
3. The Lasso in particular is good at producing sparse solutions. However, they all have a bias towards zero, with non-zero components decaying in size (and this, is often an undesirable feature).

# Linear Regression: A Summary (Part II)

4. The Lasso acts via the shrinkage operator applied to the least squares solution when the matrix $\mathbf{X}$ is orthogonal.
5. In general, one can use many of the standard linear program algorithms to approximate the Lasso solution (coordinate descent, simplex method).
6. By using Lasso as a **subset selection** mechanism, one can often get rid of the decay of the non-zero components by running least squares on just the dimensions with non-zero components.
7. The Dantzig selector is a powerful estimator in cases where one has relatively high dimensional data in comparison to the number of data points.

# Linear Classification

# Classification

The topics in linear classification we discuss will be as follows

1. Regression with indicator matrices
2. Linear discriminants for Gaussian mixtures
3. Logistic regression
4. The perceptron and optimally separating hyperplanes

The methods will not be linear in the sense of a resulting linear equation, but linear in that the boundaries between two adjacent classes in the input space $\mathbb{R}^p$ will be a portion of a hyperplane, thus given by a linear relation.

# Classification: Statistical Decision Theory

Let us recall a few things:

If we are given a random variable $X$ (in $\mathbb{R}^p$), and $G : \mathbb{R}^p \mapsto \mathcal{G}$ then, for any $\hat{G}$ meant to estimate $G$, we have the expected prediction error

$$\mathrm{EPE} = \mathbb{E}[L(G, \hat{G}(x))]$$

where $L(g, g')$ is the loss function.

# Classification: Statistical Decision Theory

Let us recall a few things:

For classification, the most popular $L$ is the $0-1$ function,

$$L(g, g') = 1 \text{ if } g = g', = 0 \text{ otherwise.}$$

As we know, minimizing the EPE amounts to minimizing the **conditional** EPE,

$$\hat{G} = \operatorname*{argmin}_{k} \mathbb{E}[L(G, g) \mid X = x]$$

# Classification: Statistical Decision Theory

Let us recall a few things:

But,

$$\mathbb{E}[L(G, k \mid X = x] = 1 - \mathbb{P}[G = k \mid X = x]$$

This is how we derived the Bayes classifier

$$\hat{G}(x) = \underset{k}{\operatorname{argmax}}\, \mathbb{P}[G = k \mid X = x]$$

# Classification: Discriminant Functions

One way to perform classification is by deciding on **discriminant functions**, these are scalar functions

$$\delta_k(x) \quad k = 1, \ldots, K.$$

Such that the class $\mathcal{G}_k = \{x \mid G(x) = k\}$ is characterized by

$$\mathcal{G}_k = \{\delta_k(x) > \delta_\ell(x) \ \forall \ \ell \neq k\}$$

The set $\{x \mid \delta_k(x) = \delta_\ell(x)\}$ is called the decision boundary between $\mathcal{G}_k$ and $\mathcal{G}_\ell$, and it is in general a $(p-1)$-dimensional hypersurface.

# Classification: Discriminant Functions

Statistical decision theory provides an important example: if our data is given by a probability distribution, then we can use for $\delta_k(x)$ the posterior distributions

$$\delta_k(x) := \mathbb{P}(G = k \mid X = x)$$

As such, the Bayes classifier simply assigns to $x$ whichever class is most likely, conditioning on $X = x$.

# Classification: Discriminant Functions

## Example: Gaussian mixture

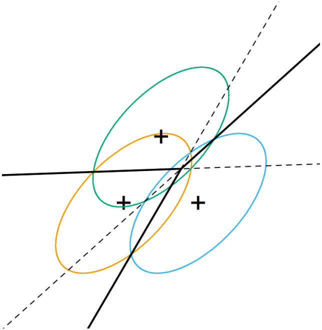

*Figure credit: Hastie-Tibshirani-Friedman*

Ellipses: region of 95% probability for each Gaussian.

Thick lines: decision boundaries.

Dotted lines: decision boundaries for two Gaussians.

# Indicator Matrices and good ol' Regression

An obvious thing to do, is to think of classification as a regression problem where the target function takes only the values $\{1, \ldots, K\}$.

Of course, linear regression will hardly ever return a function which takes these values exactly.

# Indicator Matrices and good ol' Regression

A better version of this idea involves doing $K$ different regressions instead of just one.

How so?

Instead of $G$, consider the indicator functions for the $K$ classes

$$\chi_k(x) = \begin{cases} 1 & \text{if } G(x) = k \\ 0 & \text{otherwise} \end{cases}$$

Then, run a linear regression for each function $\chi_k$, and then classify $x$ according to which of these $k$ estimators returned the largest number.

# Indicator Matrices and good ol' Regression

In other words, let $\hat{\chi}_k$ be the least squares estimator for $\chi_k$, then set

$$\hat{G}(x) = \operatorname*{argmax}_k \hat{\chi}_k(x)$$

In other words, the $\hat{\chi}_k$ are used as discriminant functions.

# Indicator Matrices and good ol' Regression

We are given data $(x_i, (Y_{ik})_i)$ where $x_i \in \mathbb{R}^p$ and

$$Y_{ik} = \begin{cases} 1 & \text{if } G(x_i) = k \\ 0 & \text{otherwise} \end{cases}$$

Denote by $\mathbf{Y}$ the $N \times K$ matrix made out of the $Y_{ik}$. This matrix is made out of 1's and 0's, and it has exactly one non-zero component in each row. This is known as the **indicator matrix**.

A column of $Y$ represents a measurement of $N$ data points, for one of the $k$-th indicator functions.

# Indicator Matrices and good ol' Regression

Applying least squares for each "indicator function" then, we obtain

$$\hat{\mathbf{Y}} = \mathbf{X}(\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\mathbf{Y}$$
$$\hat{\mathbf{B}} = (\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\mathbf{Y}$$

Note that these are matrices on the left, not vectors!

# Indicator Matrices and good ol' Regression

Given $x \in \mathbb{R}^p$, then

$$\hat{\chi}(x) = \hat{\mathbf{B}}^t (1, x) \in \mathbb{R}^K$$

The vector $\hat{\chi}(x)$ is usually far from $\chi(x) = (\chi_1(x), \ldots, \chi_K(x))$, the indicator vector.

# Indicator Matrices and good ol' Regression

Given $x \in \mathbb{R}^p$, then

$$\hat{\chi}(x) = \hat{\mathbf{B}}^t (1, x) \in \mathbb{R}^K$$

The vector $\hat{\chi}(x)$ is usually far from $\chi(x) = (\chi_1(x), \ldots, \chi_K(x))$, the indicator vector.

Still, the relative sizes of the components of $\hat{\chi}$ may be a good proxy for which is 1 and which is zero, so we set

$$\hat{G}(x) = \underset{k}{\operatorname{argmax}}\, \hat{\chi}_k(x)$$

# Indicator Matrices and good ol' Regression

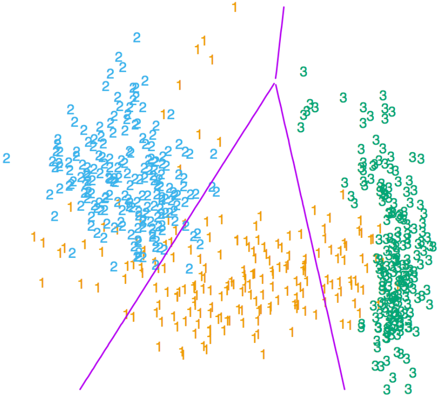

*Figure credit: Hastie-Tibshirani-Friedman*

# Indicator Matrices and good ol' Regression

Now, what are the potential issues in linear regression? There are the general issues of complexity, and the bias-variance trade off.

The following one is specific to classification via regression, although it relates to the more usual questions.

# Indicator Matrices and good ol' Regression

## Masking phenomenon

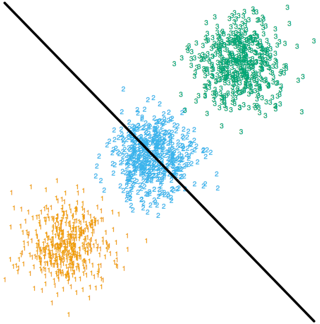

*Figure credit: Hastie-Tibshirani-Friedman*

**Masking:** An extreme situation is when one of the discriminant functions never dominates, and a class is completely hidden.

# Indicator Matrices and good ol' Regression

## Extra variables

One way to enrich linear regression for classification is incorporating extra variables, for instance, quadratic functions of the components of the vector, i.e.

$$X_a X_b \;\text{ where } a \leq b$$

As such, the $p$-dimensional vector

$$X = (X_1, \ldots, X_p) \in \mathbb{R}^p$$

is replaced with the $p + p(p+1)/2$ dimensional vector

$$(X_1, \ldots, X_p, X_1^2, X_1 X_2, \ldots, X_1 X_p, X_2^2, X_2 X_3, \ldots, X_p^2)$$

# Indicator Matrices and good ol' Regression

## Extra variables

This can be thought of as an embedding of our input space in higher dimensions

$$\mathbb{R}^p \mapsto \mathbb{R}^{\tilde{p}}, \ \tilde{p} > p$$

Then, we may perform linear regression in $\mathbb{R}^{\tilde{p}}$ and project our solution back to $\mathbb{R}^p$.

# Indicator Matrices and good ol' Regression

## Extra variables

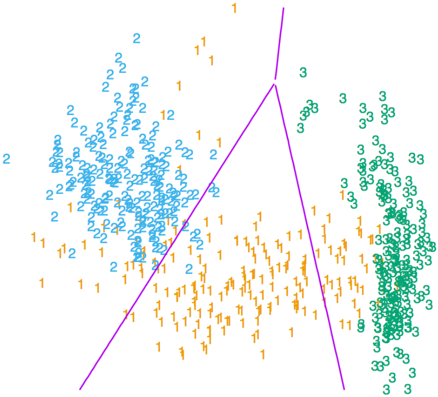

*Figure credit: Hastie-Tibshirani-Friedman*

Using affine functions to estimate the indicator functions

# Indicator Matrices and good ol' Regression

## Extra variables

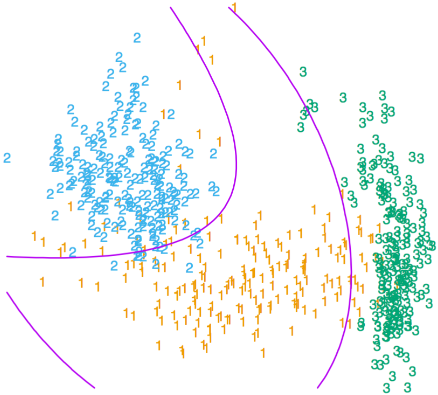

*Figure credit: Hastie-Tibshirani-Friedman*

Using quadratic polynomials to estimate the indicator functions

# Linear Classification: LDA and basics of Logistic regression

MATH 697 AM:ST

October 19th, 2017

# Indicator Matrices and good ol' Regression

(Last class)

Now, what are the potential issues in linear regression? There are the general issues of complexity, and the bias-variance trade off.

The following one is specific to classification via regression, although it relates to the more usual questions.

# Indicator Matrices and good ol' Regression

## Masking phenomenon
(Last class)

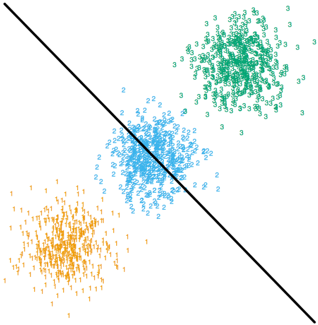

*Figure credit: Hastie-Tibshirani-Friedman*

**Masking:** An extreme situation is when one of the discriminant functions never dominates, and a class is completely hidden.

# Indicator Matrices and good ol' Regression

## Extra variables
## (Last class)

One way to enrich linear regression for classification is incorporating extra variables, for instance, quadratic functions of the components of the vector, i.e.

$$X_a X_b \;\text{ where } a \leq b$$

As such, the $p$-dimensional vector

$$X = (X_1, \ldots, X_p) \in \mathbb{R}^p$$

is replaced with the $p + p(p+1)/2$ dimensional vector

$$(X_1, \ldots, X_p, X_1^2, X_1 X_2, \ldots, X_1 X_p, X_2^2, X_2 X_3, \ldots, X_p^2)$$

# Indicator Matrices and good ol' Regression

## Extra variables
## (Last class)

This can be thought of as an embedding of our input space in higher dimensions

$$\mathbb{R}^p \mapsto \mathbb{R}^{\tilde{p}},\ \tilde{p} > p$$

Then, we may perform linear regression in $\mathbb{R}^{\tilde{p}}$ and project our solution back to $\mathbb{R}^p$.

# Indicator Matrices and good ol' Regression

Extra variables
(Last class)

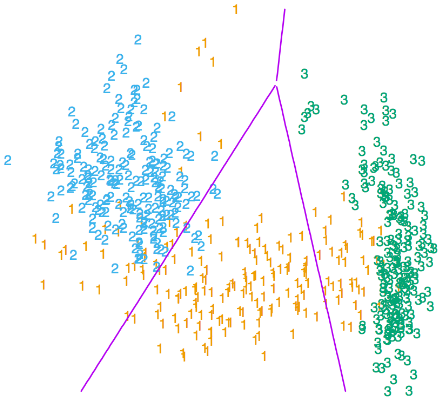

*Figure credit: Hastie-Tibshirani-Friedman*

Using affine functions to estimate the indicator functions

# Indicator Matrices and good ol' Regression

Extra variables
(Last class)

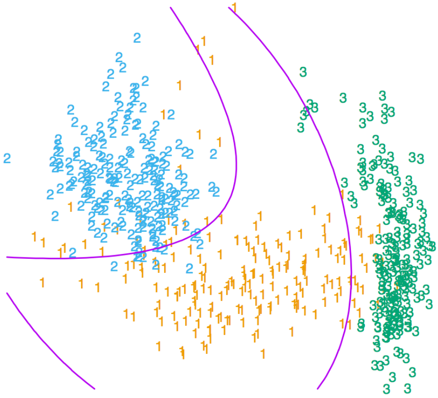

*Figure credit: Hastie-Tibshirani-Friedman*

Using quadratic polynomials to estimate the indicator functions

# Warmup: Gradient Flows

A differential equation is best thought of as given by a vector field $v : \mathbb{R}^p \mapsto \mathbb{R}^p$

$$\dot{x} = v(x), \;\; x(0) = x_0.$$

When $v$ is such that for some scalar scalar function $f$

$$v(x) = -\nabla f(x)$$

then we say the differential equation is a **gradient flow**.

# Warmup: Gradient Flows

If $x(t)$ solves $\dot{x} = -\nabla f(x)$ then

$$\frac{d}{dt} f(x(t)) = -|(\nabla f)(x(t))|^2$$

So, as $x(t)$ moves the functional $f$ is decrasing strictly, until $x$ reaches an equilibrium point.

If $f$ is **a convex function**, then $x(t)$ moves towards a global equilibrium.

# Warmup: Gradient Flows

## Smooth convex functions

In fact, if $f$ is uniformly strictly convex (it's second derivative is bounded away from zero) then $x(t)$ converges exponentially fast to the minimum.

**Theorem**
Suppose $f$ is a twice differentiable function such that $D^2 f(x) \geq \lambda \mathbb{I}$ for all $x$, and let $x_\infty$ denote it's unique (why is it unique?) global minimum. Then, if $\dot{x} = -\nabla f$, we have

$$|x(t) - x_\infty| \leq e^{-\lambda t} |x(0) - x_\infty|$$

# Warmup: Gradient ~~Flow~~ Descent

Often, specially in the CS and Statistics literature, the term **Gradient descent** is used to refer to a discretization of a gradient flow.

Roughly speaking, one considers a sequence of points $x_k$, where $x_0$ is fixed initialized in some way or another, and then one recursively writes

$$x_{k+1} = x_k - h\nabla f(x_k)$$

where $h > 0$ is a fixed time step. One typically stops this recursive procedure when $|\nabla f(x_k)|$ becomes smaller than some predetermined threshold.

# Warmup: Gradient ~~Flow~~ Descent

(more minima = more problems)

When the functional $f$ is not necessarily convex, and it has multiple local minima, gradient descent can fail rather spectacularly.

Most importantly, as $k \to \infty$ (or $t \to \infty$ for the continuum case) the point $x_k$ will approach one critical point of $f$, but which of the possibly multiple critical points we converge to will depend on the initialization $x_0$.

This is an ever present issue in learning algorithms where one is minimizing a non-convex objective functional depending on a family of weights via gradient descent.

# Classification for Gaussian mixtures

Consider a model where the points $\{x_1, \ldots, x_N\}$ are being drawn in an i.i.d. manner from some underlying probability distribution.

Denoting by $X$ a random variable distributed by this probability distribution, the Bayesian classifier for a given input $x \in \mathbb{R}^p$ is

$$\hat{G}(x) = \underset{k}{\operatorname{argmax}}\, \mathbb{P}(G = k \mid X = x)$$

# Classification for Gaussian mixtures

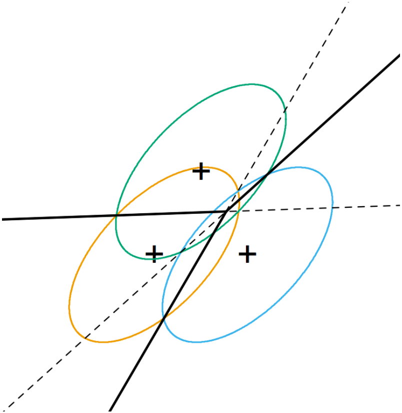

*Figure credit: Hastie-Tibshirani-Friedman*

# Classification for Gaussian mixtures

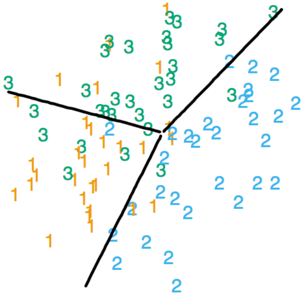

*Figure credit: Hastie-Tibshirani-Friedman*

# Classification for Gaussian mixtures

When the points arise from a mixture of Gaussians, one can determine $\hat{G}$ entirely from linear operations.

For a Gaussian mixture, we have (via Bayes theorem)

$$\mathbb{P}(G = k \mid X = x) = \frac{f_k(x)\pi_k}{\sum_{\ell=1}^{K} f_\ell(x)\pi_\ell}$$

Therefore

$$\frac{\mathbb{P}(G = k \mid X = x)}{\mathbb{P}(G = \ell \mid X = x)} = \frac{f_k(x)\pi_k}{f_\ell(x)\pi_\ell}$$

# Classification for Gaussian mixtures

Since,

$$f_k(x) = (2\pi)^{-\frac{p}{2}} |\Sigma_k|^{-\frac{1}{2}} e^{-\frac{1}{2}(\Sigma_k^{-1}(x-\mu_k), x-\mu_k)} \pi_k$$

taking the logarithm is advantageous, so

$$\begin{aligned} &\log\left(\frac{\mathbb{P}(G = k \mid X = x)}{\mathbb{P}(G = \ell \mid X = x)}\right) \\ &= \log\left(\frac{|\Sigma_k|^{-\frac{1}{2}} e^{-\frac{1}{2}(\Sigma_k^{-1}(x-\mu_k), x-\mu_k)} \pi_k}{|\Sigma_\ell|^{-\frac{1}{2}} e^{-\frac{1}{2}(\Sigma_k^{-1}(x-\mu_\ell), x-\mu_\ell)} \pi_\ell}\right) \end{aligned}$$

# Classification for Gaussian mixtures

What if all the $\Sigma_k$ are all equal to some $\Sigma$?.

Well, we have

$$
\begin{aligned}
&\log\left(\frac{\mathbb{P}(G = k \mid X = x)}{\mathbb{P}(G = \ell \mid X = x)}\right) \\
&= \log\left(\frac{\pi_k}{\pi_\ell}\right) + \log\left(\frac{e^{-\frac{1}{2}(\Sigma^{-1}(x-\mu_k),x-\mu_k)}}{e^{-\frac{1}{2}(\Sigma^{-1}(x-\mu_\ell),x-\mu_\ell)}}\right)
\end{aligned}
$$

# Classification for Gaussian mixtures

We compute

$$
\begin{aligned}
&-(\Sigma^{-1}(x-\mu_k), x-\mu_k) + (\Sigma^{-1}(x-\mu_\ell), x-\mu_\ell) \\
&= 2(\Sigma^{-1}(\mu_k-\mu_\ell), x) - (\Sigma^{-1}\mu_k, \mu_k) + (\Sigma^{-1}\mu_\ell, \mu_\ell) \\
&= 2(\Sigma^{-1}(\mu_k-\mu_\ell), x) + (\Sigma^{-1}(\mu_\ell+\mu_k), \mu_\ell-\mu_k)
\end{aligned}
$$

# Classification for Gaussian mixtures

and conclude that

$$\log\left(\frac{\mathbb{P}(G = k \mid X = x)}{\mathbb{P}(G = \ell \mid X = x)}\right)$$

is equal to

$$\log\left(\frac{\pi_k}{\pi_\ell}\right) + \frac{1}{2}(\Sigma^{-1}(\mu_k + \mu_\ell), \mu_k - \mu_\ell) + (\Sigma^{-1}(\mu_k - \mu_\ell), x)$$

# Classification for Gaussian mixtures

Further simplification shows that

$$\log\left(\frac{\mathbb{P}(G=k \mid X=x)}{\mathbb{P}(G=\ell \mid X=x)}\right) = \delta_k(x) - \delta_\ell(x)$$

where the discriminant function $\delta_k$ is given by

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

# Classification for Gaussian mixtures

## Practical considerations

Given a training set $\{x_1, \ldots, x_N\}$ with labels $g_i \in \{1, \ldots, K\}$, we have the estimators

$$
\begin{aligned}
\hat{\pi}_k &= \frac{N_k}{N}, \quad N_k := \#\{i : g_i = k\} \\
\hat{\mu}_k &= \frac{1}{N_k} \sum_{i:g_i=k} x_i \\
\hat{\Sigma}_{ab} &= \frac{1}{N-K} \sum_{k=1}^{K} \sum_{i:g_i=k} (x_i - \hat{\mu}_k)_a (x_i - \hat{\mu}_k)_b
\end{aligned}
$$

# Classification for Gaussian mixtures

## Practical considerations

and we have the “estimated” discriminant functions, which are linear

$$\hat{\delta}_k(x) = \log(\hat{\pi}_k) + \frac{1}{2}(\hat{\Sigma}^{-1}\hat{\mu}_k, \mu_k) + (\hat{\Sigma}^{-1}\hat{\mu}_k, x)$$

Then, the classifier is

$$\hat{G}(x) = \underset{k \in K}{\operatorname{argmax}} \, \hat{\delta}_k(x).$$

# Classification for Gaussian mixtures

Let us go back to the case where we allow $\Sigma_k \neq \Sigma_\ell$, and consider

$$\log\left(\frac{\mathbb{P}(G = k \mid X = x)}{\mathbb{P}(G = \ell \mid X = x)}\right)$$

and we have that is equal to

$$\log\left(\frac{\pi_k}{\pi_\ell}\right) - \frac{1}{2}\log\left(\frac{|\Sigma_k|}{|\Sigma_\ell|}\right) - \frac{1}{2}(\Sigma_k^{-1}(x - \mu_k), x - \mu_k)$$
$$+ \frac{1}{2}(\Sigma_\ell^{-1}(x - \mu_\ell), x - \mu_\ell)$$

there aren't as many cancelations as in the case of a single $\Sigma$...

# Classification for Gaussian mixtures

…but still, we have

$$\log\left(\frac{\mathbb{P}(G=k \mid X=x)}{\mathbb{P}(G=\ell \mid X=x)}\right)=\delta_k(x)-\delta_\ell(x)$$

where this time, $\delta_k(x)$ are quadratic functions

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

# Classification for Gaussian mixtures

As before, when using this algorithm in practice, we take

$$\hat{\delta}_k(x) := \log(\hat{\pi}_k) - \frac{1}{2}\log(|\hat{\Sigma}_k|) - \frac{1}{2}(\hat{\Sigma}_k^{-1}(x - \hat{\mu}_k), x - \hat{\mu}_k)$$

where $\hat{\pi}_k$, $\hat{\mu}_k$ and $\hat{\Sigma}_k^{-1}$ are the standard estimators, and then

$$\hat{G}(x) = \operatorname*{argmax}_k \hat{\delta}_k(x).$$

# Classification for Gaussian mixtures

## Examples

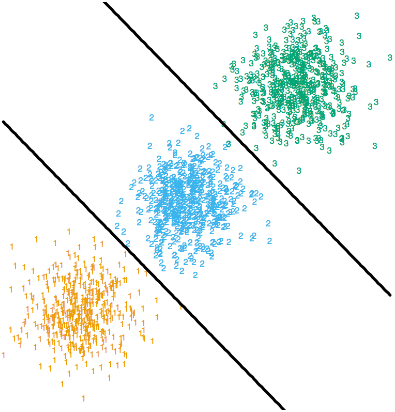

*Figure credit: Hastie-Tibshirani-Friedman*

Linear Discriminant Analysis Vs. masking.

# Classification for Gaussian mixtures

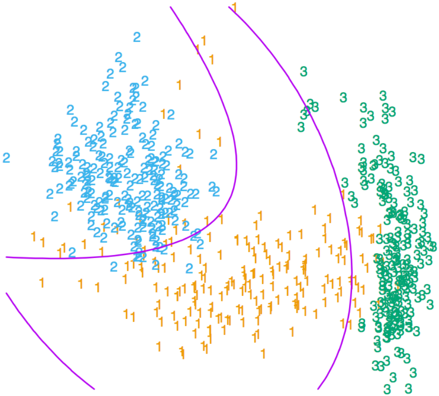

*Figure credit: Hastie-Tibshirani-Friedman*

(for comparison) Regression with quadratic functions

# Classification for Gaussian mixtures

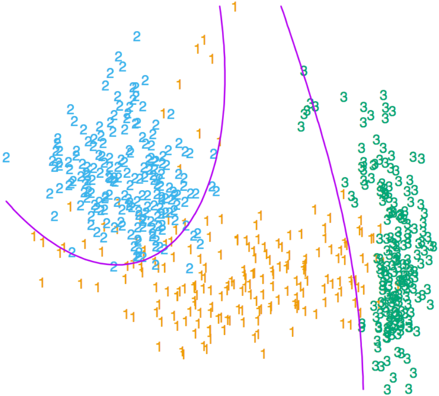

*Figure credit: Hastie-Tibshirani-Friedman*

(for comparison) and the result via QDA

# Logistic regression

The regression on the indicator approach to the indicator matrix has a serious drawback: the discriminant functions given by posterior probabilities

$$\mathbb{P}(G = k \mid X = x)$$

are not going to be even close to linear, so our intent to approximate them via linear functions is somewhat misguided. Worse still, the discriminant functions $\delta_k(x)$ obtained in this way won't satisfy

$$0 \leq \delta_k \text{ and } \sum_{k=1}^{K} \delta_k(x) = 1$$

# Logistic regression

This justifies in part the considereration of a logistic transformation, and assuming that

$$\mathbb{P}(G = k \mid X = x) = \frac{e^{L_k(x)}}{1 + \sum_{\ell=1}^{K-1} e^{L_\ell(x)}}$$

and

$$\mathbb{P}(G = K \mid X = x) = \frac{1}{1 + \sum_{j=1}^{K-1} e^{L_j(x)}}$$

where the $L_k$'s are $K - 1$ affine functions, written as

$$L_k(x) = \beta_{k,0} + (\beta_k, x)$$

# Logistic regression

Observe that the last probability can be written as

$$\mathbb{P}(G = K \mid X = x) = \frac{e^{L_K(x)}}{1 + \sum\limits_{j=1}^{K-1} e^{L_j(x)}}$$

if we take $L_K(x)$ to be trivial affine function, $L_K(x) \equiv 0$. As such, going forward we think of having actually $K$ affine functions, where

$$L_k(x) = \beta_{K,0} + (\beta_K, x),\ \beta_{K,0} = 0,\ \beta_K = 0.$$

# Logistic regression

Then, with this convention $\beta_{K,0} = 0, \beta_K = 0$ we actually have more succint expression

$$\mathbb{P}(G = k \mid X = x) = \frac{e^{L_K(x)}}{\sum_{j=1}^{K} e^{L_j(x)}}$$

for all $k \in \{1, \ldots, K\}$ and all $x$.

We will analyze this model in greater detail next class, along with Rosenblatt's hyperplane algorithm.

# The seventh week, in one slide

1. We discussed (briefly) the Lasso combined with Least squares to do subset selection while minimizing bias.
2. The Dantzig selector is quite robust at recovering sparse vectors in high dimensional problems.
3. One can think of classification in terms of discriminant functions: $x$ is labeled as $k$ if $\delta_k(x) > \delta_j(x)$ for all other $j$.
4. Least squares applied to an indicator matrix produces linear discriminants, however, masking issues may appear.
5. LDA is a better alternative for linear classification, specially for data given by Gaussians.