

Lecture 6: Classification

Reading: Sections 4.3, 4.4

GU4241/GR5241 Statistical Machine Learning

Linxi Liu
February 2, 2018

Classification problems

Supervised learning with a **qualitative or categorical** response.

Just as common, if not more common than regression:

- ▶ *Medical diagnosis*: Given the symptoms a patient shows, predict which of 3 conditions they are attributed to.
- ▶ *Online banking*: Determine whether a transaction is fraudulent or not, on the basis of the IP address, client's history, etc.
- ▶ *Web searching*: Based on a user's history, location, and the string of a web search, predict which link a person is likely to click.
- ▶ *Online advertising*: Predict whether a user will click on an ad or not.

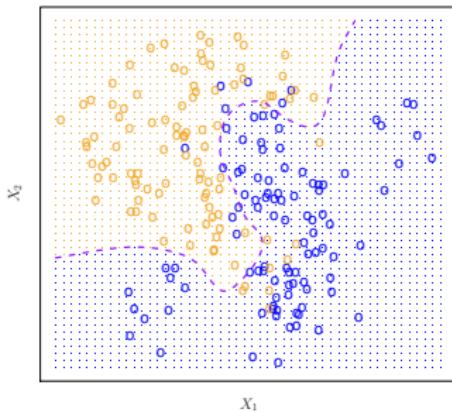
$$(X, Y) \quad P(Y=1 | X=x) + P(Y=0 | X=x) = 1$$

$\uparrow Y \in \{0, 1\}$

Prediction: based on input (background)
know which class does this belong to

$P(Y=1 | X=x) = \frac{P(X=x | Y=1) P(Y=1)}{P(X=x)}$

Classification problem



ISL Figure 2.13

Recall:

- ▶ $X = (X_1, X_2)$ are inputs.
- ▶ Color $Y \in \{\text{Yellow}, \text{Blue}\}$ is the output.
- ▶ (X, Y) have a joint distribution.
- ▶ Purple line is *Bayes boundary* — the best we could do if we knew the joint distribution of (X, Y)

Review: Bayes classifier

Suppose $P(Y | X)$ is known. Then, given an input x_0 , we predict the response

最大化他的后验预测分布

$$\hat{y}_0 = \operatorname{argmax}_y P(Y = y | X = x_0).$$

The Bayes classifier minimizes the expected 0-1 loss:

$$E \left[\frac{1}{m} \sum_{i=1}^m \mathbf{1}(\hat{y}_i \neq y_i) \right]$$

这是一种expected loss

This minimum 0-1 loss (the best we can hope for) is the **Bayes error rate**.

Example: Spam Filtering

Representing emails

$$P(X_1, \dots, X_n | Y=1) = P(X_1|Y=1) \cdot P(X_2|Y=1) \cdot \dots \cdot P(X_n|Y=1)$$

a collection of words

Assumption:
 x_1, \dots, x_n are independent.

- $Y = \{ \text{spam, email} \}$
- $X = \mathbb{R}^d$
- Each axis is labelled by one possible word.
- $d = \text{number of distinct words in vocabulary}$
- $x_j = \text{number of occurrences of word } j \text{ in email represented by } x$

For example, if axis j represents the term "the", $x_j = 3$ means that "the" occurs three times in the email x . This representation is called a **vector space model of text**.

Example dimensions

	george	you	your	hp	free	hpl	!	our	re	edu
spam	0.00	2.26	1.38	0.02	0.52	0.01	0.51	0.51	0.13	0.01
email	1.27	1.27	0.44	0.90	0.07	0.43	0.11	0.18	0.42	0.29

With Bayes equation

关键在于给定y的情况下x的条件概率

$$f(x) = \operatorname{argmax}_{y \in \{\text{spam, email}\}} P(y|x) = \operatorname{argmax}_{y \in \{\text{spam, email}\}} p(x|y)P(y)$$

Naive Bayes

Simplifying assumption

The classifier is called a **naive Bayes classifier** if it assumes

$$p(\mathbf{x}|y) = \prod_{j=1}^d p_j(x_i|y) ,$$

i.e. if it treats the individual dimensions of \mathbf{x} as conditionally **independent** given y .

In spam example

- ▶ Corresponds to the assumption that the number of occurrences of a word carries information about y .
- ▶ **Co-occurrences (how often do given combinations of words occur?) is neglected.** 因为视各个变量在给定 y 之后的条件分布是独立的。所以不能考察它们之间的交互作用。

Estimation

Class prior

The distribution $P(y)$ is easy to estimate from training data:

$$P(y) = \frac{\text{\#observations in class } y}{\text{\#observations}}$$

Class-conditional distributions

The class conditionals $p(x|y)$ usually require a modeling assumption. Under a given model:

- ▶ Separate the training data into classes.
- ▶ Estimate $p(x|y)$ on class y by maximum likelihood.

Strategy: estimate $P(Y | X)$

If we have a good estimate for the conditional probability $\hat{P}(Y | X)$, we can use the classifier:

$$\hat{y}_0 = \operatorname{argmax}_y \hat{P}(Y = y | X = x_0).$$

Suppose Y is a binary variable. Could we use a linear model?

$$P(Y = 1 | X) = \beta_0 + \beta_1 \underline{X_1} + \cdots + \beta_1 \underline{X_p}$$

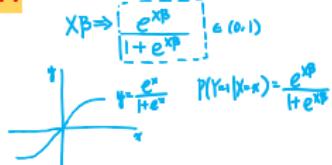
They can be any number.

can not guarantee $0 \leq P \leq 1$

Problems:

- ▶ This would allow probabilities < 0 and > 1 .
- ▶ Difficult to extend to more than 2 categories.

Logistic regression



We model the joint probability as:

$$P(Y = 1 | X) = \frac{e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}},$$

$$P(Y = 0 | X) = \frac{1}{1 + e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}.$$

This is the same as using a linear model for the log odds:

$$\log \left[\frac{P(Y = 1 | X)}{P(Y = 0 | X)} \right] = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p.$$

Fitting logistic regression

The training data is a list of pairs $(y_1, x_1), (y_2, x_2), \dots, (y_n, x_n)$. In the linear model

$$\log \left[\frac{P(Y = 1 | X)}{P(Y = 0 | X)} \right] = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p,$$

we don't observe the left hand side.

We cannot use a least squares fit.

Fitting logistic regression

Solution:

The likelihood is the probability of the training data, for a fixed set of coefficients β_0, \dots, β_p :

$$\prod_{i=1}^n P(Y = y_i | X = x_i) \\ = \underbrace{\prod_{i:y_i=1} \frac{e^{\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}}}{1 + e^{\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}}} \underbrace{\text{Probability of responses } = 1}_{\text{Probability of responses } = 0}}$$

- ▶ Choose estimates $\hat{\beta}_0, \dots, \hat{\beta}_p$ which maximize the likelihood.
- ▶ Solved with numerical methods (e.g. Newton's algorithm).

Logistic regression in R

```
> glm.fit=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume ,  
    data=Smarket,family=binomial)  
> summary(glm.fit)
```

Call:

```
glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5  
+ Volume, family = binomial, data = Smarket)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.45	-1.20	1.07	1.15	1.33

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-0.12600	0.24074	-0.52	0.60
Lag1	-0.07307	0.05017	-1.46	0.15
Lag2	-0.04230	0.05009	-0.84	0.40
Lag3	0.01109	0.04994	0.22	0.82
Lag4	0.00936	0.04997	0.19	0.85
Lag5	0.01031	0.04951	0.21	0.83
Volume	0.13544	0.15836	0.86	0.39

Logistic regression in R

- ▶ We can estimate the Standard Error of each coefficient.
- ▶ The z -statistic is the equivalent of the t -statistic in linear regression:

$$z = \frac{\hat{\beta}_j}{\text{SE}(\hat{\beta}_j)}.$$

*only when n very large
can say sth. about the distribution*

- ▶ The p -values are test of the null hypothesis $\beta_j = 0$ (Wald's test).
- ▶ Other possible hypothesis tests: likelihood ratio test (chi-square distribution).

Example: Predicting credit card default

Predictors:

- ▶ student: 1 if student, 0 otherwise. (*categorical*)
- ▶ balance: credit card balance.
- ▶ income: person's income.

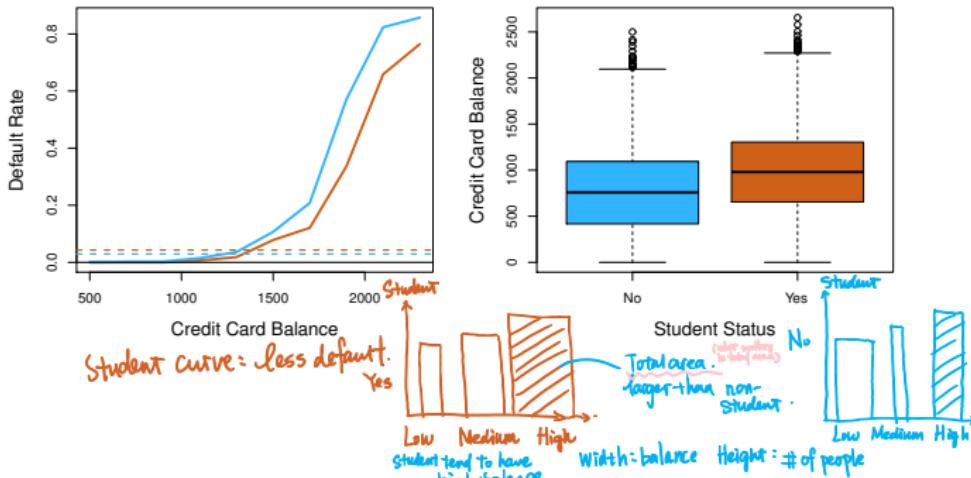
In this dataset, there is *confounding*, but little collinearity.

- ▶ Students tend to have higher balances. So, balance is explained by student, but not very well.
- ▶ People with a high balance are more likely to default.
- ▶ Among people with a given balance, students are less likely to default.

Example: Predicting credit card default

Predictors:

- ▶ student: 1 if student, 0 otherwise.
- ▶ balance: credit card balance.
- ▶ income: person's income.



Example: Predicting credit card default

Logistic regression using only balance:

	Coefficient	Std. error	Z-statistic	P-value
Intercept	-10.6513	0.3612	-29.5	<0.0001
balance	0.0055	0.0002	24.9	<0.0001

Logistic regression using only student:

	Coefficient	Std. error	Z-statistic	P-value
Intercept	-3.5041	0.0707	-49.55	<0.0001
student [Yes]	0.4049	0.1150	3.52	0.0004 *

student more likely to default

Logistic regression using all 3 predictors:

	Coefficient	Std. error	Z-statistic	P-value
Intercept	-10.8690	0.4923	-22.08	<0.0001
balance	0.0057	0.0002	24.74	<0.0001 *
income	0.0030	0.0082	0.37	0.7115
student [Yes]	-0.6468	0.2362	-2.74	0.0062

If use all the variables.
for student will get opposite conclusion: student less likely to default.

Extending logistic regression to more than 2 categories

Multinomial logistic regression:

Suppose Y takes values in $\{1, 2, \dots, K\}$, then we use a linear model for the log odds against a **baseline category** (e.g. 1):

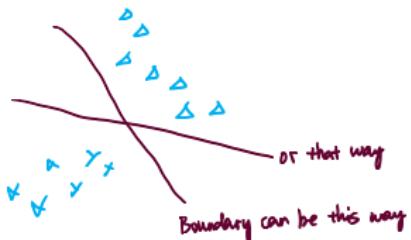
$$\log \left[\frac{P(Y = 2 | X)}{P(Y = 1 | X)} \right] = \beta_{0,2} + \beta_{1,2}X_1 + \dots + \beta_{p,2}X_p,$$

.....

$$\log \left[\frac{P(Y = K | X)}{P(Y = 1 | X)} \right] = \beta_{0,K} + \beta_{1,K}X_1 + \dots + \beta_{p,K}X_p.$$

Some issues with logistic regression

- ▶ The coefficients become unstable when there is collinearity. Furthermore, this affects the convergence of the fitting algorithm.
- ▶ When the classes are well separated, the coefficients become unstable. This is always the case when $p \geq n - 1$.



Main strategy in Chapter 4

Find an estimate $\hat{P}(Y | X)$. Then, given an input x_0 , we predict the response as in a Bayes classifier:

$$\hat{y}_0 = \operatorname{argmax}_y \hat{P}(Y = y | X = x_0).$$

Linear Discriminant Analysis (LDA)

Strategy: Instead of estimating $P(Y | X)$, we will estimate:

1. $\hat{P}(X | Y)$: Given the response, what is the distribution of the inputs.
2. $\hat{P}(Y)$: How likely are each of the categories.

Then, we use **Bayes rule** to obtain the estimate:

$$\hat{P}(Y = k | X = x) = \frac{\hat{P}(X = x | Y = k)\hat{P}(Y = k)}{\hat{P}(X = x)}$$

likelihood matters.
determine which dist.
to u

↑
doesn't matter.

Linear Discriminant Analysis (LDA)

Strategy: Instead of estimating $P(Y | X)$, we will estimate:

1. $\hat{P}(X | Y)$: Given the response, what is the distribution of the inputs.
2. $\hat{P}(Y)$: How likely are each of the categories.

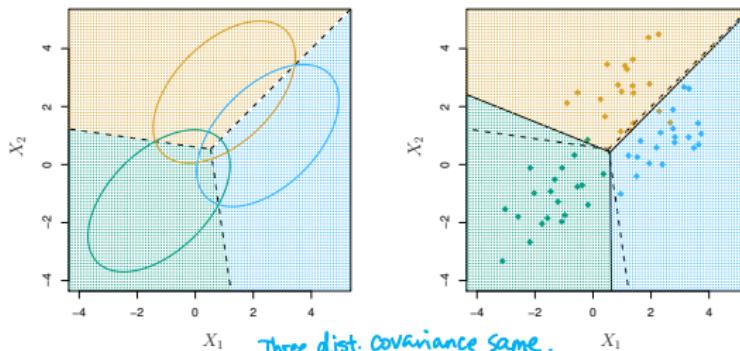
Then, we use *Bayes rule* to obtain the estimate:

$$\hat{P}(Y = k | X = x) = \frac{\hat{P}(X = x | Y = k)\hat{P}(Y = k)}{\sum_j \hat{P}(X = x | Y = j)\hat{P}(Y = j)}$$

Linear Discriminant Analysis (LDA)

Strategy: Instead of estimating $P(Y | X)$, we will estimate:

1. We model $\hat{P}(X = x | Y = k) = \hat{f}_k(x)$ as a **Multivariate Normal Distribution**: with the same variance



2. $\hat{P}(Y = k) = \hat{\pi}_k$ is estimated by the fraction of training samples of class k .

LDA has linear decision boundaries

Suppose that:

- ▶ We know $P(Y = k) = \pi_k$ exactly.
- ▶ $P(X = x|Y = k)$ is Multivariate Normal with density:

$$f_k(x) = \frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma^{-1} (x-\mu_k)}$$

μ_k : Mean of the inputs for category k .

Σ : Covariance matrix (**common to all categories**).

Then, what is the Bayes classifier?

LDA has linear decision boundaries

By Bayes rule, the probability of category k , given the input x is:

$$P(Y = k \mid X = x) = \frac{f_k(x)\pi_k}{P(X = x)}$$

The denominator does not depend on the response k , so we can write it as a constant:

$$P(Y = k \mid X = x) = C \times f_k(x)\pi_k$$

Now, expanding $f_k(x)$:

$$P(Y = k \mid X = x) = \frac{C\pi_k}{(2\pi)^{p/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma^{-1}(x-\mu_k)}$$

LDA has linear decision boundaries

$$P(Y = k \mid X = x) = \frac{C\pi_k}{(2\pi)^{p/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma^{-1}(x-\mu_k)}$$

Now, let us absorb everything that does not depend on k into a constant C' :

$$P(Y = k \mid X = x) = C'\pi_k e^{-\frac{1}{2}(x-\mu_k)^T \Sigma^{-1}(x-\mu_k)}$$

and take the logarithm of both sides:

$$\log P(Y = k \mid X = x) = \log C' + \log \pi_k - \frac{1}{2}(x - \mu_k)^T \Sigma^{-1}(x - \mu_k).$$

This is the same for every category, k .

LDA has linear decision boundaries

$$P(Y = k \mid X = x) = \frac{C\pi_k}{(2\pi)^{p/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma^{-1}(x-\mu_k)}$$

Now, let us absorb everything that does not depend on k into a constant C' :

$$P(Y = k \mid X = x) = C' \pi_k e^{-\frac{1}{2}(x-\mu_k)^T \Sigma^{-1}(x-\mu_k)}$$

and take the logarithm of both sides:

$$\log P(Y = k \mid X = x) = \underbrace{\log C' + \log \pi_k}_{\text{maximize this blue part.}} - \frac{1}{2}(x - \mu_k)^T \Sigma^{-1}(x - \mu_k).$$

This is the same for every category, k .

So we want to find the maximum of this over k .

LDA has linear decision boundaries

Goal, maximize the following over k :

$$\begin{aligned} & \log \pi_k - \frac{1}{2}(x - \mu_k)^T \Sigma^{-1}(x - \mu_k). \\ &= \log \pi_k - \frac{1}{2} [x^T \Sigma^{-1} x + \mu_k^T \Sigma^{-1} \mu_k] + x^T \Sigma^{-1} \mu_k \\ &= C'' + \log \pi_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + x^T \Sigma^{-1} \mu_k \end{aligned}$$

We define the objective:

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

At an input x , we predict the response with the highest $\delta_k(x)$.

LDA has linear decision boundaries

What is the decision boundary? It is the set of points in which 2 classes do just as well:

$$\delta_k(x) = \delta_\ell(x)$$

$$\log \pi_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + x^T \Sigma^{-1} \mu_k = \log \pi_\ell - \frac{1}{2} \mu_\ell^T \Sigma^{-1} \mu_\ell + x^T \Sigma^{-1} \mu_\ell$$

if x is fixed *Want to max:*

$$\begin{aligned} & \log \pi_k - \frac{1}{2} (x - \mu_k)^T \Sigma^{-1} (x - \mu_k) \\ &= \log \pi_k - \frac{1}{2} (x^T \Sigma^{-1} x - 2\mu_k^T \Sigma^{-1} x + \mu_k^T \Sigma^{-1} \mu_k) \\ &= C + \underbrace{\log \pi_k + \mu_k^T \Sigma^{-1} x - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k}_{S(x)} \\ &= \log \pi_k + \mu_k^T \Sigma^{-1} x - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k \\ &= \log \pi_k + \mu_k^T \Sigma^{-1} x - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k \\ &\quad \text{linear in } x + \text{(decision boundary is linear)} \end{aligned}$$

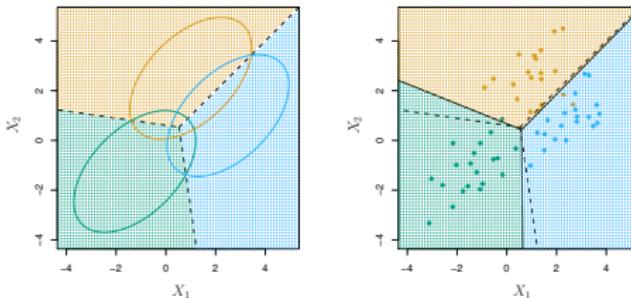
LDA has linear decision boundaries

What is the decision boundary? It is the set of points in which 2 classes do just as well:

$$\delta_k(x) = \delta_\ell(x)$$

$$\log \pi_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \textcolor{red}{x}^T \Sigma^{-1} \mu_k = \log \pi_\ell - \frac{1}{2} \mu_\ell^T \Sigma^{-1} \mu_\ell + \textcolor{red}{x}^T \Sigma^{-1} \mu_\ell$$

This is a linear equation in $\textcolor{red}{x}$.



Estimating π_k

$$\hat{\pi}_k = \frac{\#\{i ; y_i = k\}}{n}$$

In English, the fraction of training samples of class k .

$\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n$.

$$\frac{\hat{\mu}_k}{n-k} \sum_i (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T$$

Estimating the parameters of $f_k(x)$

Estimate the center of each class μ_k :

$$\hat{\mu}_k = \frac{1}{\#\{i ; y_i = k\}} \sum_{i ; y_i = k} x_i$$

Estimate the common covariance matrix Σ :

- One predictor ($p = 1$):

所有的数据都放进来一起计算

$$\hat{\sigma}^2 = \frac{1}{n - K} \sum_{k=1}^K \sum_{i ; y_i = k} (x_i - \hat{\mu}_k)^2.$$

- Many predictors ($p > 1$): Compute the vectors of deviations $(x_1 - \hat{\mu}_{y_1}), (x_2 - \hat{\mu}_{y_2}), \dots, (x_n - \hat{\mu}_{y_n})$ and use an unbiased estimate of its covariance matrix, Σ .

`\sigma^2 = 1/(n-K) \sum^K_{k=1} \sum_{i : y_i=k} t(X_k - U_k) * % (X_k - U_k)`

LDA prediction

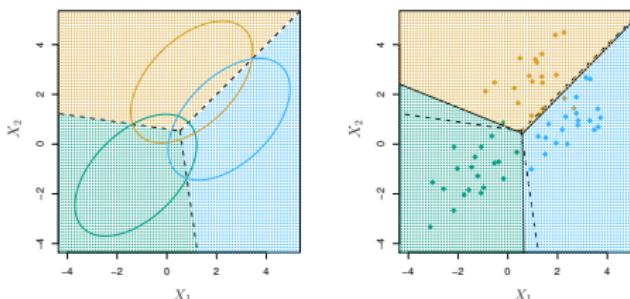
For an input x , predict the class with the largest:

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

The decision boundaries are defined by:

$$\log \hat{\pi}_k - \frac{1}{2} \hat{\mu}_k^T \hat{\Sigma}^{-1} \hat{\mu}_k + x^T \hat{\Sigma}^{-1} \hat{\mu}_k = \log \hat{\pi}_\ell - \frac{1}{2} \hat{\mu}_\ell^T \hat{\Sigma}^{-1} \hat{\mu}_\ell + x^T \hat{\Sigma}^{-1} \hat{\mu}_\ell$$

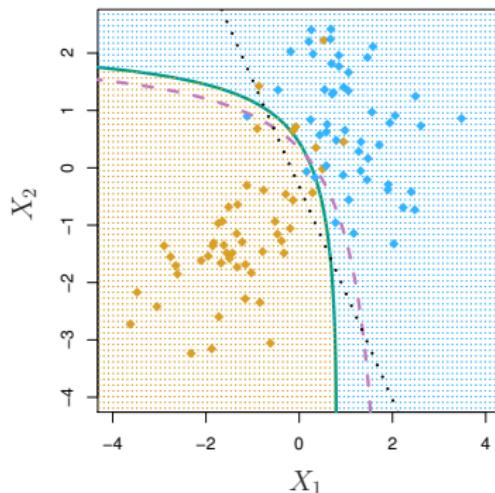
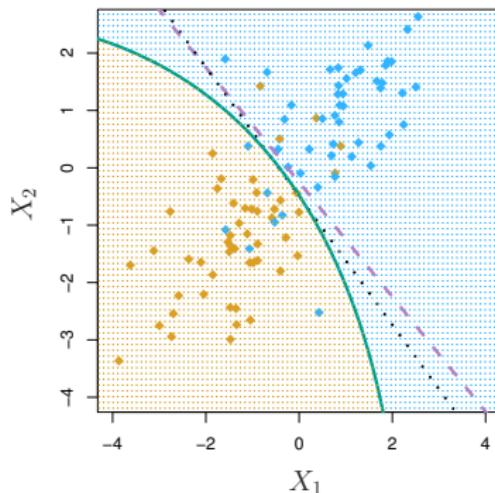
Solid lines in:



Quadratic discriminant analysis (QDA)

The assumption that the inputs of every class have the same covariance Σ can be quite restrictive:

Purple: decision boundary



Quadratic discriminant analysis (QDA)

In quadratic discriminant analysis we estimate a mean $\hat{\mu}_k$ and a covariance matrix $\hat{\Sigma}_k$ for each class separately.

Given an input, it is easy to derive an objective function:

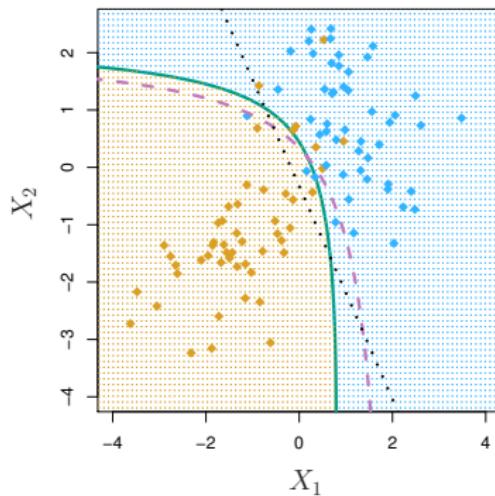
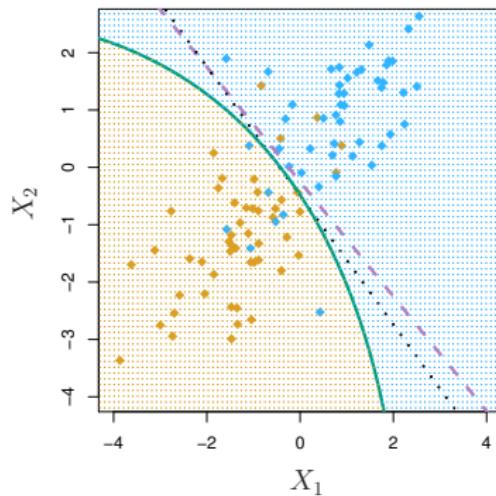
$$\delta_k(x) = \log \pi_k - \frac{1}{2} \mu_k^T \Sigma_k^{-1} \mu_k + \underbrace{x^T \Sigma_k^{-1} \mu_k}_{\text{Add this two parts into } \delta(x) \text{ [LDA].}} - \frac{1}{2} x^T \Sigma_k^{-1} x - \underbrace{\frac{1}{2} \log |\Sigma_k|}$$

This objective is now quadratic in x and so are the decision boundaries.

$$\sigma^2_K = \sum_{i:y_i=k} t(X_k - U_k)^* (X_k - U_k)$$

Quadratic discriminant analysis (QDA)

- ▶ Bayes boundary (---)
- ▶ LDA (· · · · ·)
- ▶ QDA (—).



Evaluating a classification method

We have talked about the 0-1 loss:

$$\frac{1}{m} \sum_{i=1}^m \mathbf{1}(y_i \neq \hat{y}_i).$$

It is possible to make the wrong prediction for some classes more often than others. The 0-1 loss doesn't tell you anything about this.

A much more informative summary of the error is a **confusion matrix**:

		<i>Predicted class</i>		Total
<i>True class</i>	– or Null	+ or Non-null		
	– or Null	True Neg. (TN)	False Pos. (FP)	N
	+ or Non-null	False Neg. (FN)	True Pos. (TP)	P
Total		N*	P*	

Example. Predicting default

Used LDA to predict credit card default in a dataset of 10K people.

Predicted "yes" if $P(\text{default} = \text{yes} | X) > 0.5$.

实操中这个是可以被实现的。运用LDA，可以计算出 $p(y|x)$ ，那么只要关注 $P(y=1|x)$ 是可以计算出具体的概率的。

want to avoid
False Negatives
(predict = -
True = +)

		True default status		
		No	Yes	Total
Predicted default status	No	9,644	252	9,896
	Yes	23	81	104
Total		9,667	333	10,000

negative or positive 是值判断是正还是负

True or False 是说这样的判断是不是对的

FN: 真的被判断为假的；FP就是：假的判断为真

- ▶ The error rate among people who do **not** default (**false positive rate**) is very low. FP: 放在题目中就是本来是会有违约风险的人，被判定成不会违约 ==> 坏账率高
FN: 放在题目中就是本来没有违约风险的人，被判定成会有违约风险 ==> 业务流式
- ▶ However, the rate of **false negatives** is 76%. #fn/#n
- ▶ It is possible that false negatives are a bigger source of concern!
- ▶ One possible solution: Change the **threshold**.

Example. Predicting default

Changing the threshold to 0.2 makes it easier to classify to “yes”.

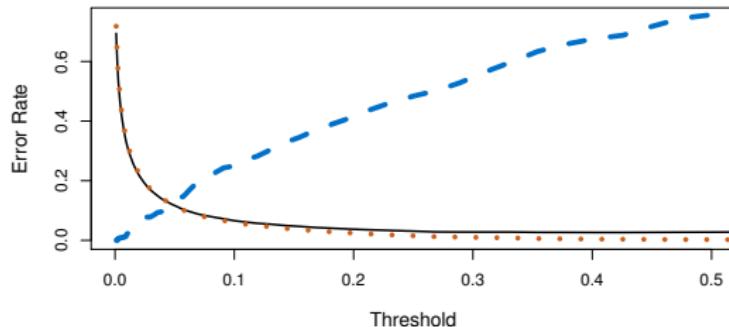
Predicted “yes” if $P(\text{default} = \text{yes}|X) > 0.2$.

		True default status		Total
		No	Yes	
Predicted default status	No	9,432	138	9,570
	Yes	235	195	430
Total	9,667	333	10,000	

Note that the rate of false positives became higher! That is the price to pay for fewer false negatives.

Example. Predicting default

Let's visualize the dependence of the error on the threshold:



$$FNR = FN / (FN + TP)$$

$$FPR = FP / (FP + TN)$$

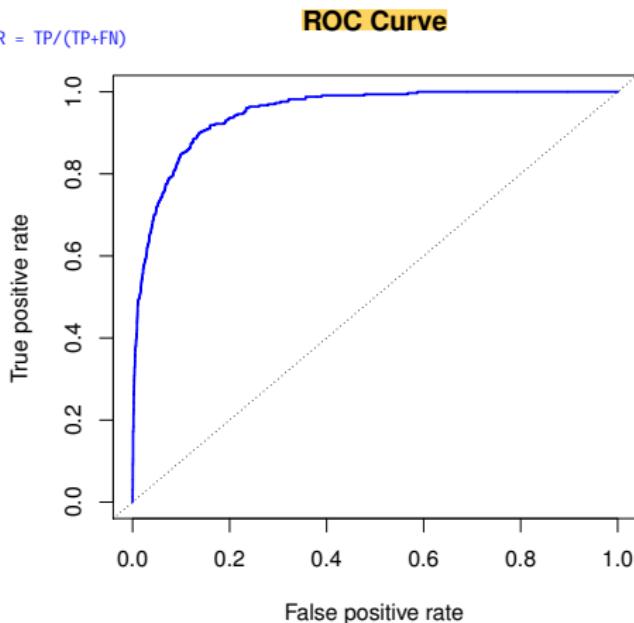
$$\text{total error rate} = (TN + TP) / (TP + TN + FP + FN)$$

首先：FNR与FPR 此消彼长是一定的。
其次：total error rate和谁一起涨是不一定

- ▶ —— False negative rate (error for defaulting customers)
- ▶ False positive rate (error for non-defaulting customers)
- ▶ — 0-1 loss or total error rate.

Example. The ROC curve

TPR = $TP/(TP+FN)$



- ▶ Displays the performance of the method for any choice of threshold.
- ▶ The **area under the curve** (AUC) measures the **quality** of the classifier:
 - ▶ 0.5 is the AUC for a random classifier
 - ▶ The closer AUC is to 1, the better.

Thinking about the loss function is important

Most of the **regression** methods we've studied aim to minimize the RSS, while **classification** methods aim to minimize the 0-1 loss.

In classification, we often care about certain kinds of error more than others; i.e. the natural loss function is not the 0-1 loss.

Even if we use a method which minimizes a certain kind of training error, we can *tune* it to optimize our true loss function.

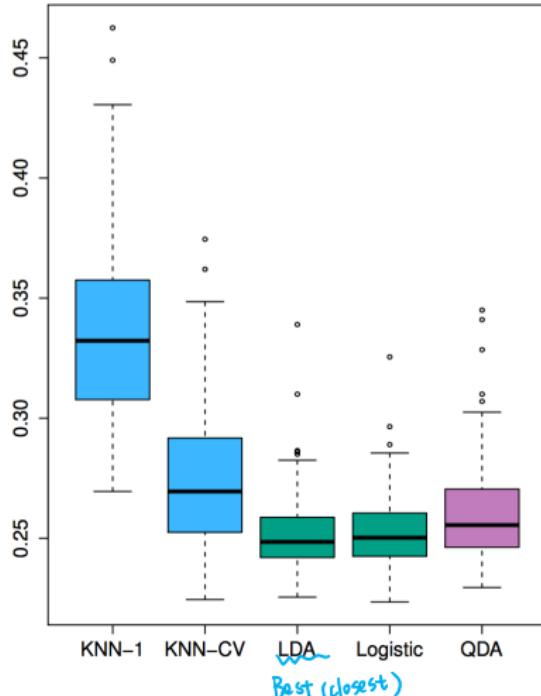
- ▶ e.g. Find the threshold that brings the False negative rate below an acceptable level.

Comparing classification methods through simulation

1. Simulate data from several different known distributions with 2 predictors and a binary response variable.
2. Compare the test error (0-1 loss) for the following methods:
 - ▶ KNN-1
 - ▶ KNN-CV (“optimal” KNN)
 - ▶ Logistic regression
 - ▶ Linear discriminant analysis (LDA)
 - ▶ Quadratic discriminant analysis (QDA)

Scenario 1

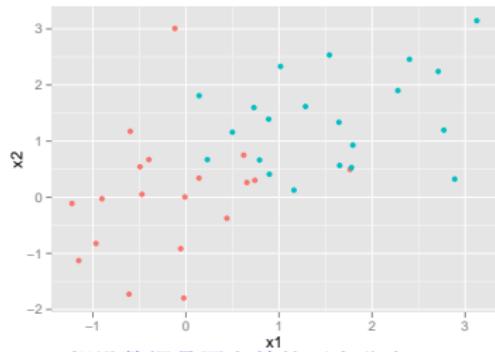
SCENARIO 1



KNN-1 适用于数据关系十分复杂，难以用线性表示的时候

数据看上去线性可分的程度比较高

- ▶ X_1, X_2 standard normal.
- ▶ No correlation in either class.

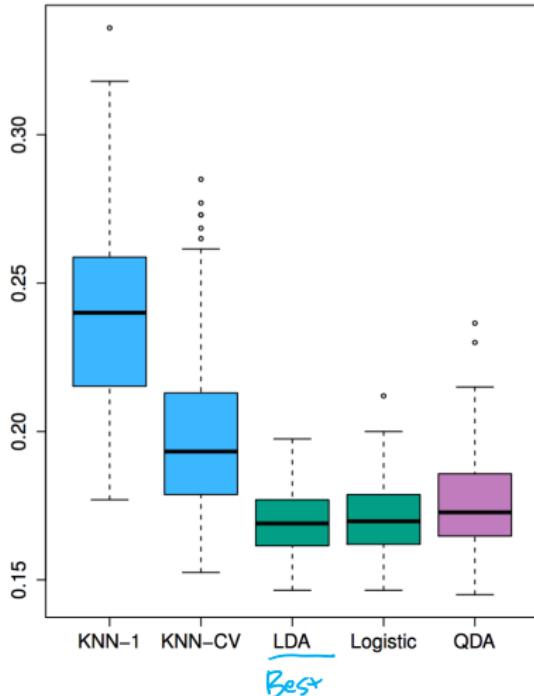


LDA假设数据是同方差的正态分布

logistics模型假设的是数据是线性可分的

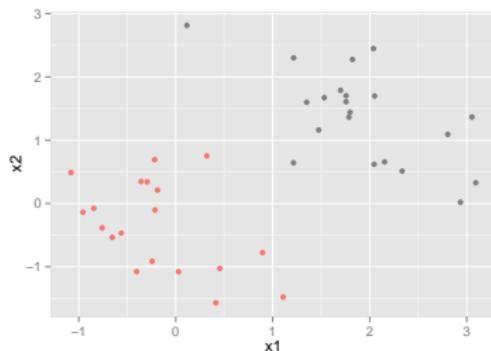
Scenario 2

SCENARIO 2



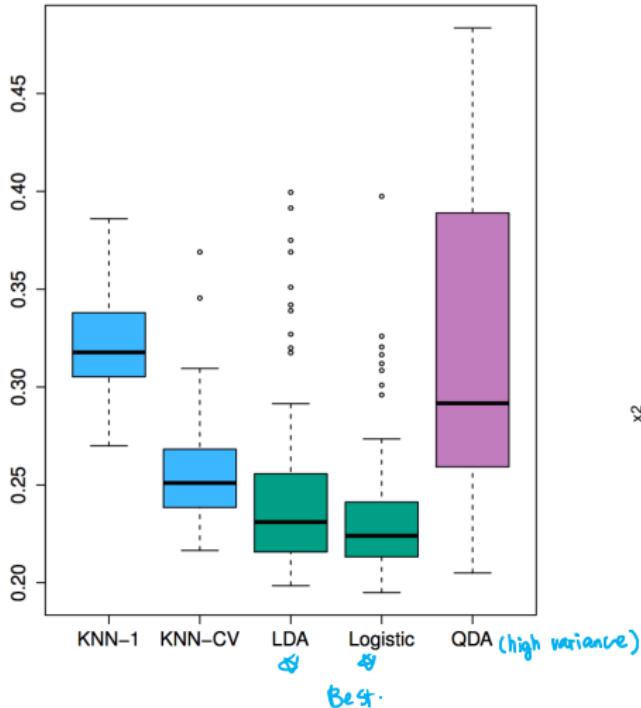
同样线性可分程度比较高

- ▶ X_1, X_2 standard normal.
- ▶ Correlation is -0.5 in both classes.



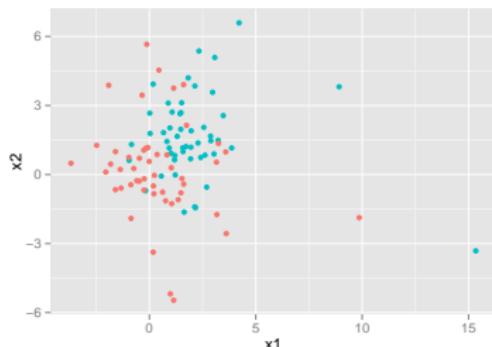
Scenario 3

SCENARIO 3



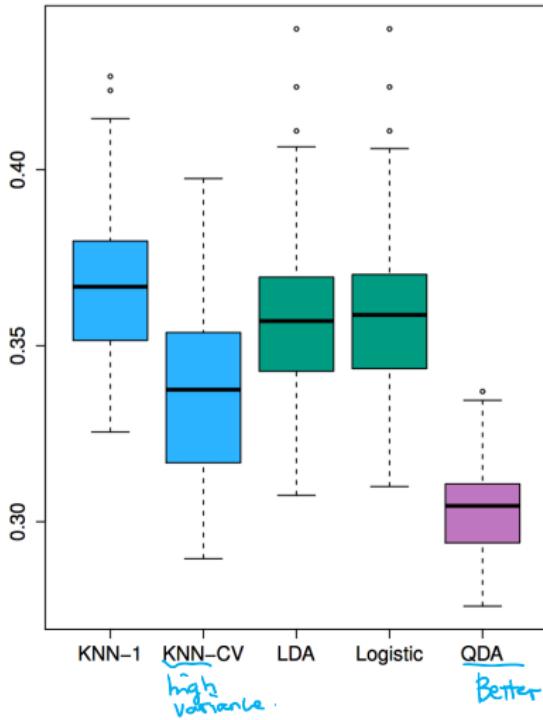
数据本身的方差较大

- X_1, X_2 Student t random variables.
heavier tails
- No correlation in either class.



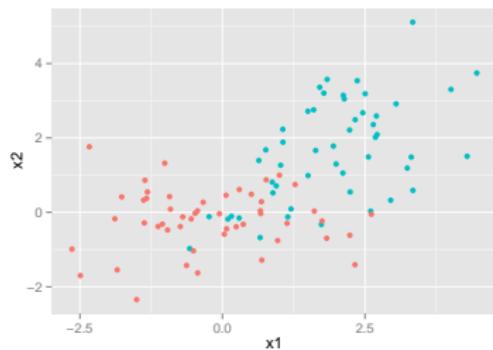
Scenario 4

SCENARIO 4



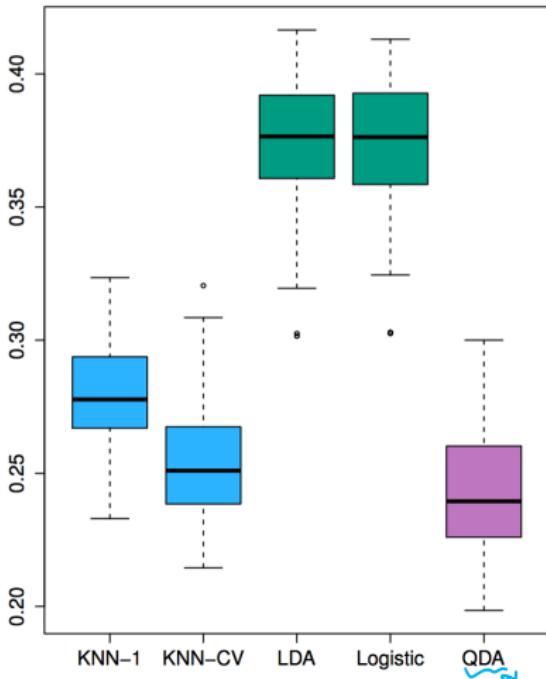
不同的方差

- X_1, X_2 standard normal.
- First class has correlation 0.5, second class has correlation -0.5.



Scenario 5

SCENARIO 5



- ▶ X_1, X_2 uncorrelated, standard normal.
- ▶ Response Y was sampled from:

Satisfy assumption of QDA

$$P(Y = 1|X) = \frac{e^{\beta_0 + \beta_1(X_1^2) + \beta_2(X_2^2) + \beta_3(X_1 X_2)}}{1 + e^{\beta_0 + \beta_1(X_1^2) + \beta_2(X_2^2) + \beta_3(X_1 X_2)}}.$$

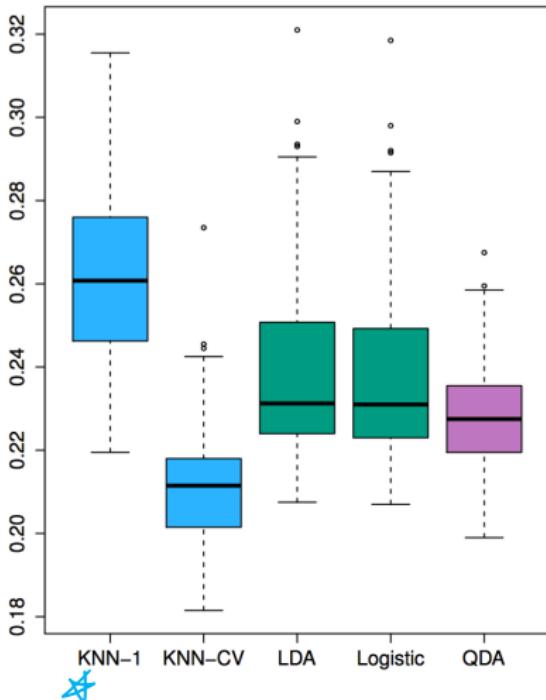
数据的形式是先行不可分的

- ▶ The true decision boundary is quadratic.

二次项形式

Scenario 6

SCENARIO 6



- ▶ X_1, X_2 uncorrelated, standard normal.
- ▶ Response Y was sampled from:

$$P(Y = 1|X) = \frac{e^{f_{\text{nonlinear}}(X_1, X_2)}}{1 + e^{f_{\text{nonlinear}}(X_1, X_2)}}.$$

形式越复杂
KNN的又是越明显
但是KNN在高纬度的情况下表现不佳

- ▶ The true decision boundary is very rough.