

Applied Data Science

Machine Learning Lecture 3

**John Tsitsiklis
September 3, 2021**

Today's agenda — Classification

- The general problem
 - formulation
 - types of error
- (Gaussian) Model-based
 - linear discriminant analysis
 - quadratic discriminant analysis
- Regression methods
 - logistic regression
 - training and validation
- Other approaches
 - Nearest-neighbor methods

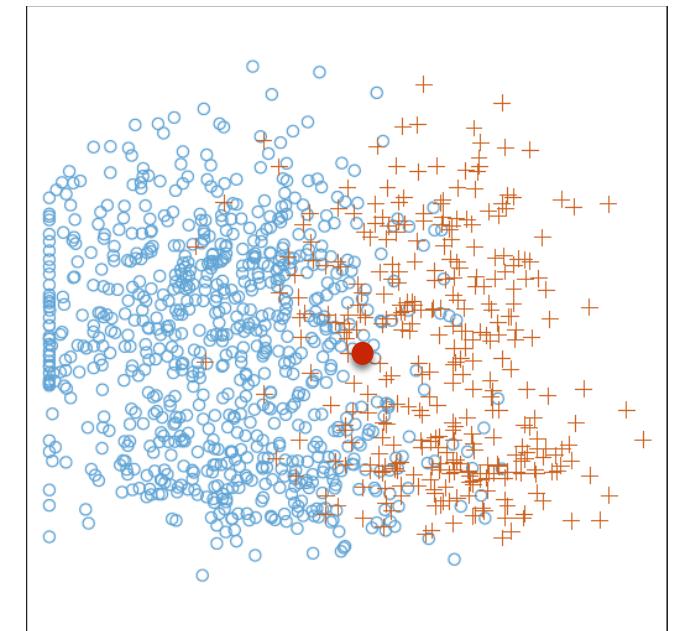
The general context — applications

X_1	Y_1
\vdots	\vdots
X_n	Y_n
<hr/>	
X	$Y?$

- Classify a data record into one of multiple categories, based on examples

$Y = 0, 1$ (binary)

$Y = 1, \dots, m$ (m -ary)



X : symptoms, test results

Y : cancer?

X : an email message

Y : spam?

X : image of a digit Y

Y : which digit is it? ($m = 10$)

X : image of an animal Y

Y : cat, dog, cow, ...

Artificial intelligence model detects asymptomatic Covid-19 infections through cellphone-recorded coughs

Results might provide a convenient screening tool for people who may not suspect they are infected.

Jennifer Chu | MIT News Office

October 29, 2020



COVID cough app Hyfe launches in UK and Ireland



Phil Taylor

March 9, 2021

A smartphone app that uses artificial intelligence to assess the sound of coughing has been launched in the UK, and according to its developer could be an early warning system for COVID-19 and other diseases of the lung.

Hyfe uses acoustic technology to track users' coughing habits, gauging factors like volume, frequency, amplitude and context, whilst running in the background on an iOS or Android device. It was originally developed as a tool for people living in heavily polluted cities who are at risk of respiratory conditions.

Example: prediction of loan default

Y	X_1	X_2	X_3
default	student	balance	income
No	No	729.5265	44361.63
No	Yes	817.1804	12106.13
No	No	1073.549	31767.14
No	No	529.2506	35704.49
No	No	785.6559	38463.5
No	Yes	919.5885	7491.559
No	No	825.5133	24905.23
No	Yes	808.6675	17600.45
No	No	1161.058	37468.53

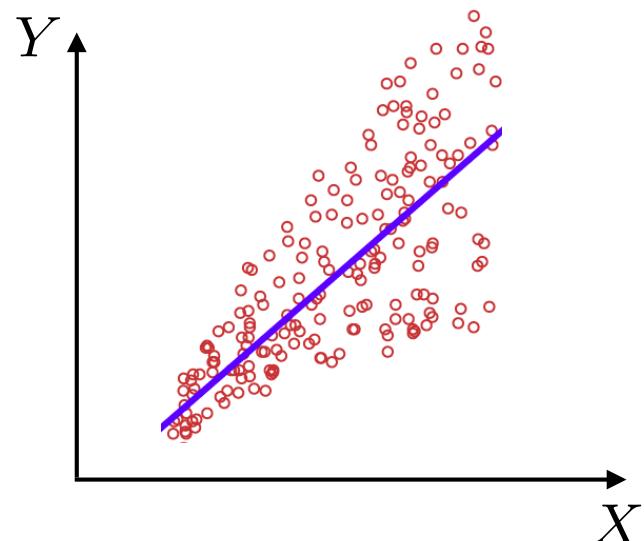
- 10,000 samples
 - 3.33% of the samples represent defaults
- encoding: “No” = 0, “Yes” = 1”

Predictors and classifiers

- In regression:

predictor $\hat{Y} = g(\mathbf{X})$

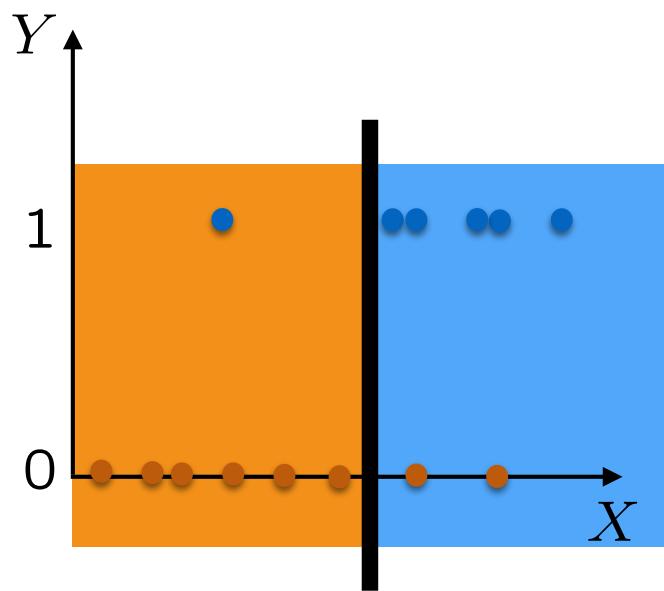
metric $\mathbb{E}[(\hat{Y} - Y)^2]$



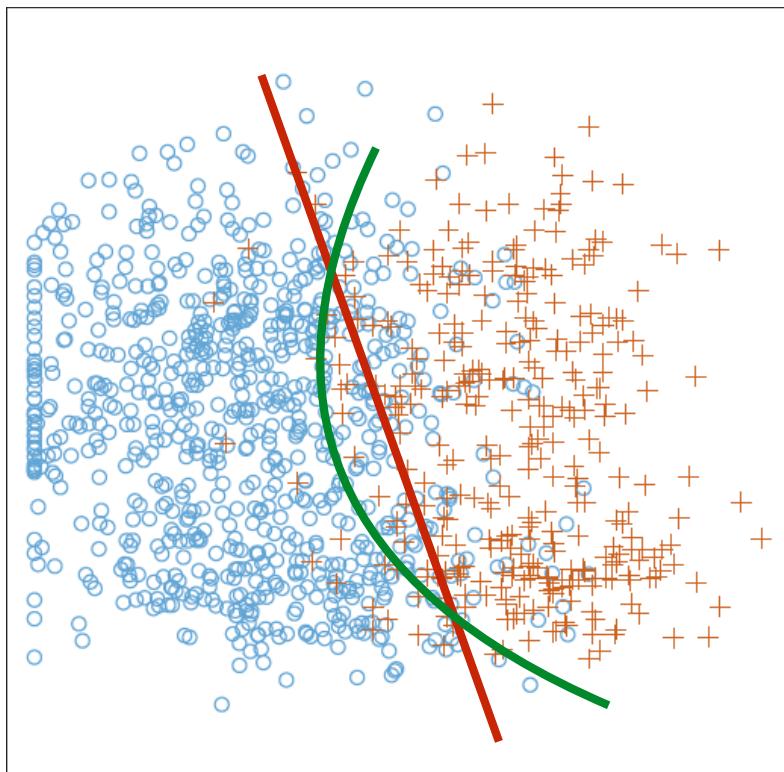
- In classification:

classifier $\hat{Y} = g(\mathbf{X}) \in \{1, \dots, m\}$

metric $\mathbb{P}(\text{mistake})$



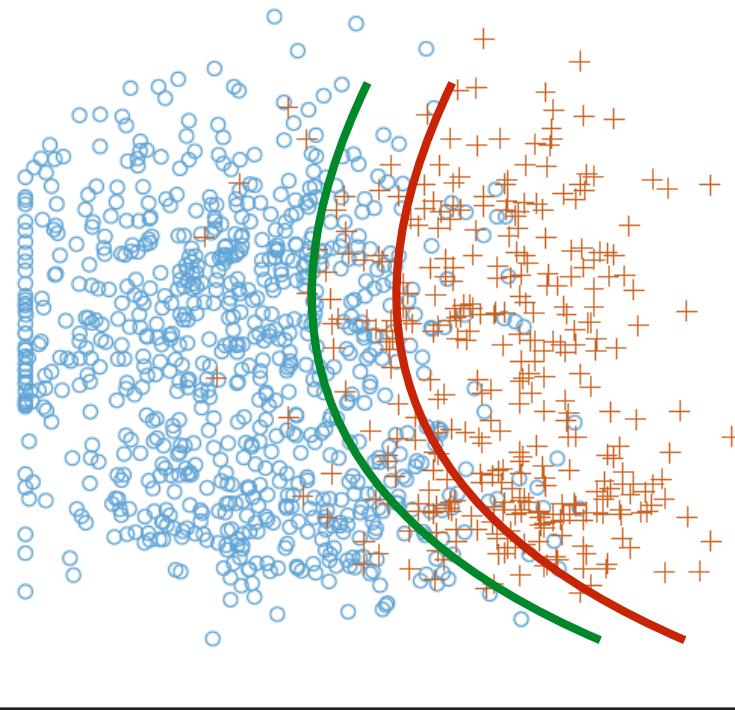
Types of classifiers



- **Linear classifiers:** compare $\theta^T \mathbf{X}$ to a threshold
 - learn “good” vector θ and threshold
- **Nonlinear classifiers:** compare $h(\mathbf{X})$ to a threshold
 - learn “good” function h and threshold

- Nonlinear classifier $\theta_1 X_1 + \theta_2 X_2 + \theta_{12} X_1 X_2$
is actually **linear** if we redefine $\mathbf{X} = (X_1, X_2, X_1 X_2)$
- **Feature-based linear classifier:** compare $\theta^T \phi(\mathbf{X})$ to a threshold

Error types and confusion matrix



- Binary classification, two error types:
truth is “blue”, decide $\hat{Y} = \text{“brown”}$
truth is “brown”, decide $\hat{Y} = \text{“blue”}$

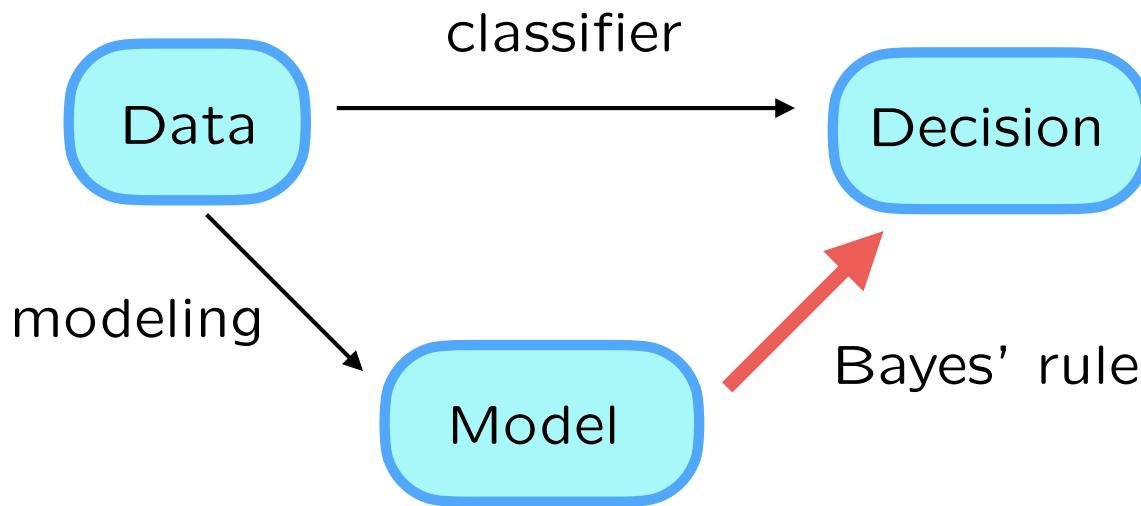
		True labels	
		blue	brown
Predicted labels	blue	✓	✗
	brown	✗	✓

- Interested in having few errors on new data records
- Aim at few errors on training set
- Tradeoff between the two error types
- Similar **confusion matrix**, for m -ary classification

		True labels	
		good	default
Predicted labels	good	9644	252
	default	23	81

(GAUSSIAN) MODEL-BASED APPROACH

The model-based approach



1. Exemplify the Bayes' rule, for Gaussian models
2. Gaussian modeling

Gaussian model

- $\mathbb{P}(Y = k) = \pi_k \quad (k = 1, \dots, m)$ “prior” probabilities

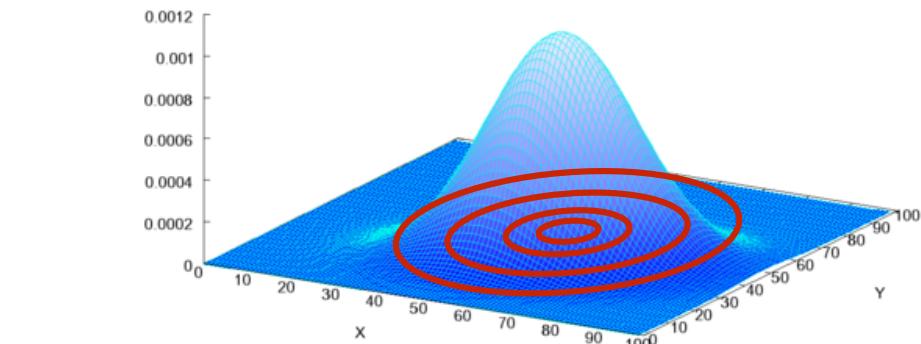
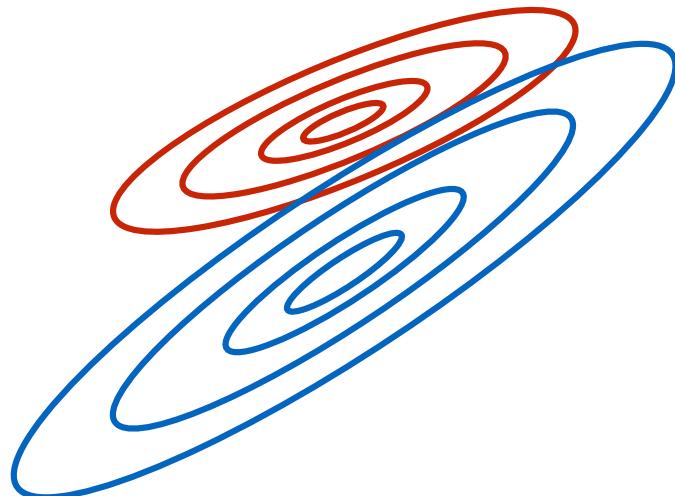
- $\mathbb{P}(\mathbf{X} | Y = k) \sim N(\boldsymbol{\mu}_k, C_k)$

$$\gamma_k \cdot \exp \left\{ -\frac{(X - \mu_k)^2}{2\sigma_k^2} \right\}$$

$$\boldsymbol{\mu}_k: \text{mean vector } \mathbb{E}[\mathbf{X} | Y = k] \quad \gamma_k \cdot \exp \left\{ -\frac{1}{2}(\mathbf{X} - \boldsymbol{\mu}_k)^T C_k^{-1} (\mathbf{X} - \boldsymbol{\mu}_k) \right\}$$

C_k : covariance matrix

$$\mathbb{E}[(\mathbf{X} - \boldsymbol{\mu}_k)(\mathbf{X} - \boldsymbol{\mu}_k)^T | Y = k]$$



https://en.wikipedia.org/wiki/Multivariate_normal_distribution

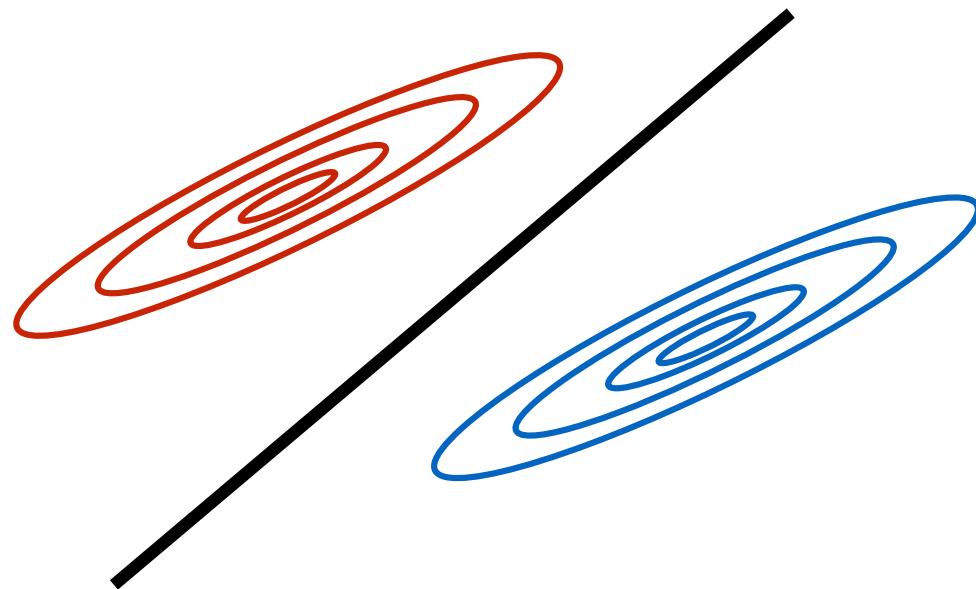
Bayes' rule

$$\mathbb{P}(Y = k | \mathbf{X}) = \frac{\pi_k \cdot \mathbb{P}(\mathbf{X} | Y = k)}{\mathbb{P}(\mathbf{X})} \quad (\text{posterior})$$

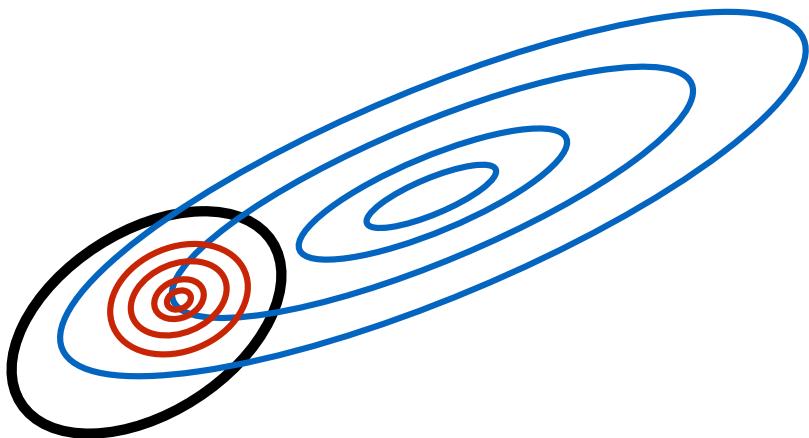
- Given value \mathbf{X} , choose k with largest posterior
 - minimizes $\mathbb{P}(\text{error})$ $\pi_k \cdot \gamma_k \cdot \exp \left\{ -\frac{1}{2}(\mathbf{X} - \boldsymbol{\mu}_k)^T C_k^{-1} (\mathbf{X} - \boldsymbol{\mu}_k) \right\}$
 - Compare $\log(\pi_k \cdot \gamma_k) - \frac{1}{2}(\mathbf{X} - \boldsymbol{\mu}_k)^T C_k^{-1} (\mathbf{X} - \boldsymbol{\mu}_k)$, for different classes k
 - boundary: when same value for some k, k' \rightarrow quadratic in \mathbf{X}
- Quadratic discriminant analysis (QDA)

- When all $C_k = C$ (same), second-order terms in \mathbf{X} are all the same
 - compare $\log(\pi_k \cdot \gamma_k) - (\boldsymbol{\mu}_k)^T C^{-1} \mathbf{X} - \frac{1}{2}(\boldsymbol{\mu}_k)^T C^{-1} \boldsymbol{\mu}_k$, for different k
 - boundary: when same value for some k, k' \rightarrow linear in \mathbf{X}
- Linear discriminant analysis (LDA)

LDA versus QDA



different means
same covariances



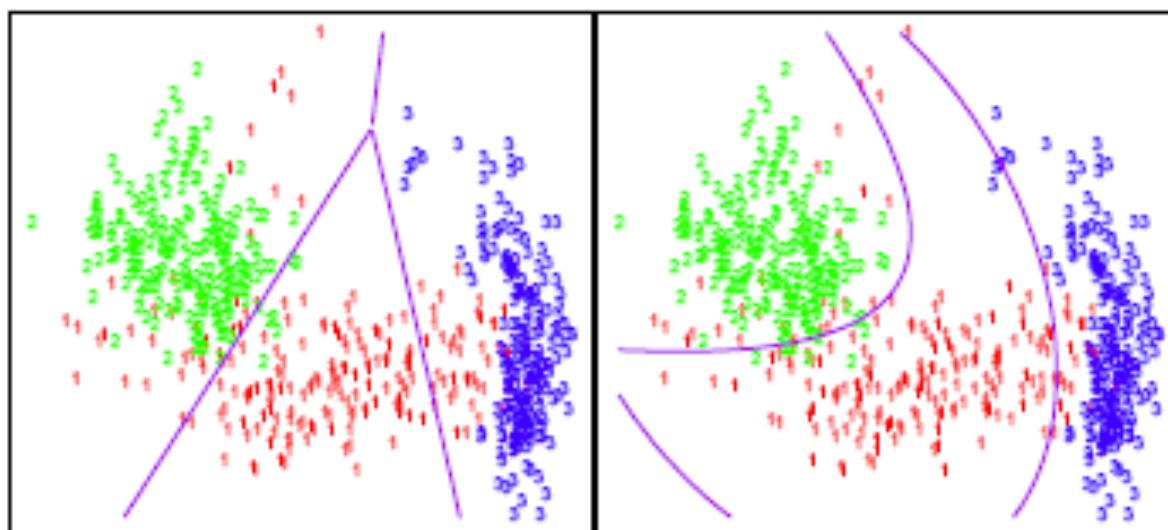
different means
different covariances

LDA versus QDA

Assume

same versus different

covariances



LDA

QDA

The modeling part

- $\mathbb{P}(Y = k) = \pi_k$
$$\hat{\pi}_k = \frac{\text{\# samples of class } k}{\text{total \# of samples}}$$
- $\mu_k = \mathbb{E}[\mathbf{X} | Y = k]$
$$\hat{\mu}_k = \begin{aligned} &\text{average value of } \mathbf{X} \\ &\text{over all samples of class } k \end{aligned}$$
- $C_k = \mathbb{E}[(\mathbf{X} - \mu_k)(\mathbf{X} - \mu_k)^T | Y = k]$
$$\hat{C}_k = \begin{aligned} &\text{average value of } (\mathbf{X} - \mu_k)(\mathbf{X} - \mu_k)^T, \\ &\text{over all samples of class } k \end{aligned}$$

“plugin estimates”

(replace expectations with averages)

Prediction default example

- Performance on the training set
- Recall that only 3.33% of the records correspond to defaults
 - **Always decide "no default:"** Miss-classification rate 3.33%
- Make predictions using only one component of \mathbf{X} ("balance")
 - LDA: 2.81%
 - QDA: 2.73%
- Make predictions using all of \mathbf{X}
 - LDA: 2.75%
 - QDA: 2.70%
- Should validate performance on a holdout set or through cross-validation

		True labels	
		good	default
Predicted labels	good	9644	252
	default	23	81

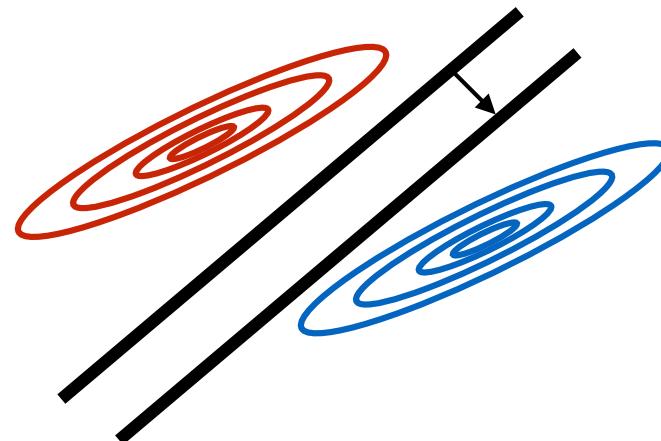
Unbalanced costs

- some error types may be more costly

$$c_{\text{def}} \gg c_{\text{lost}}$$

		True labels	
		good	default
Predicted labels	good	9644	252
	default	23	81

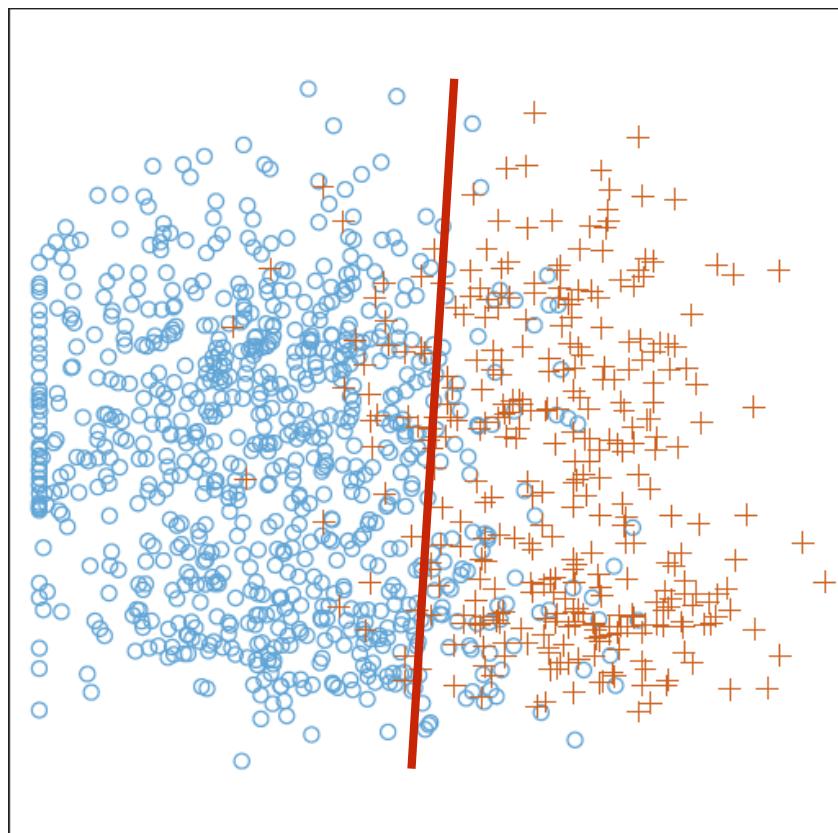
- c_{def} : cost if $\hat{Y} = \text{good}$ (give the loan) and $Y = \text{default}$
- c_{lost} : cost if $\hat{Y} = \text{default}$ and $Y = \text{good}$ (lost customer)
- Minimize probability of error versus expected cost:
compare $c_{\text{def}} \cdot \mathbb{P}(Y = \text{default} | \mathbf{X})$ to $c_{\text{lost}} \cdot \mathbb{P}(Y = \text{good} | \mathbf{X})$
- Shifts the thresholds



LEARNING/REGRESSION METHODS FOR CLASSIFICATION

An approach that does not work

- Restrict to special family of classifiers (e.g., linear)
 - find classifier with **fewest missclassified samples**
 - computationally intractable

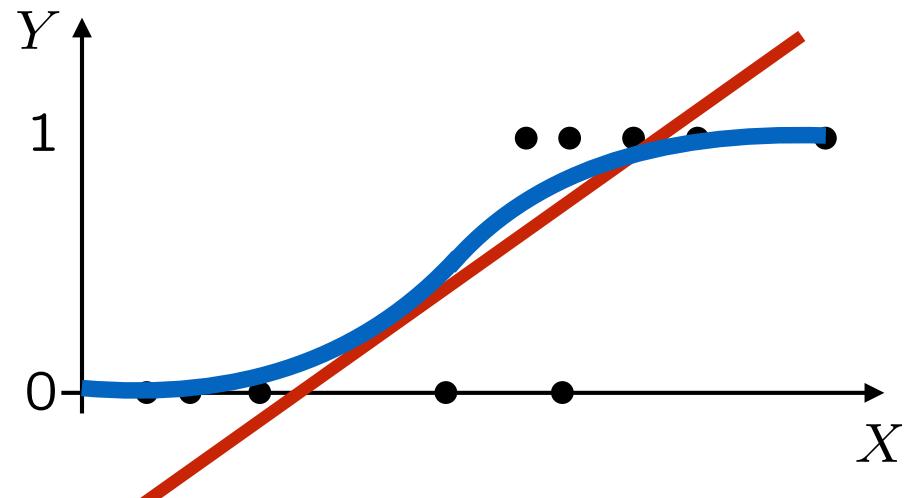


An attempt to use regression

- Regress Y on \mathbf{X}

$$\min_{\theta} \sum_i (Y_i - \theta^T \mathbf{X}_i)^2$$

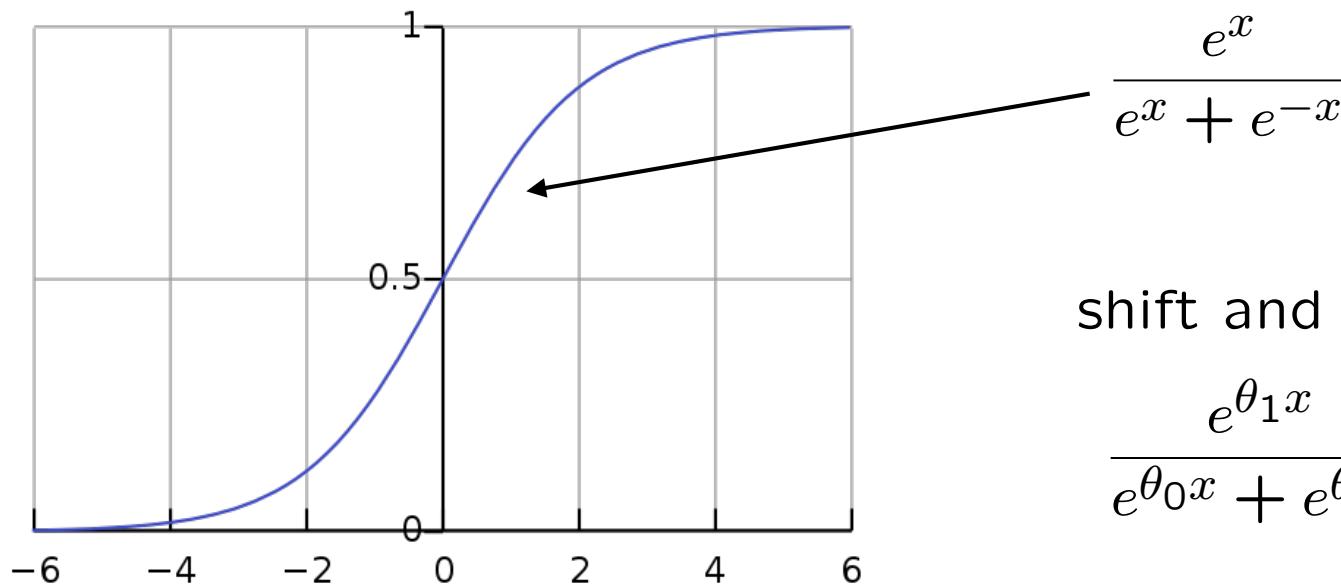
- predict $\hat{Y} = 1$ if $\hat{\theta}^T \mathbf{X} \geq 0.5$



- error criterion is not meaningful

- **A better idea:** Fit a curve that ranges between 0 and 1 (must be nonlinear)
- interpret as (estimated) probability $\mathbb{P}(Y = 1 | \mathbf{X})$

The logistic model



shift and scale

$$\frac{e^{\theta_1 x}}{e^{\theta_0 x} + e^{\theta_1 x}}$$

- Assume model of the form $\mathbb{P}(Y = k | \mathbf{X}) = \frac{\exp\{\boldsymbol{\theta}_k^T \mathbf{X}\}}{\sum_s \exp\{\boldsymbol{\theta}_s^T \mathbf{X}\}}$
- Use the data to find “best-fitting” vectors $\hat{\boldsymbol{\theta}}_1, \dots, \hat{\boldsymbol{\theta}}_m$
- Decision rule: choose class k with biggest $\hat{\boldsymbol{\theta}}_k^T \mathbf{X}$ linear!
- But we do not have training data for these **probabilities**

Training the logistic model: maximum likelihood

- Assume the \mathbf{X}_i are somehow set (random or otherwise)
- Assume the model structure
$$\mathbb{P}(Y = k \mid \mathbf{X}) = \frac{\exp\{\boldsymbol{\theta}_k^T \mathbf{X}\}}{\sum_s \exp\{\boldsymbol{\theta}_s^T \mathbf{X}\}}$$
(+ independence)
- Likelihood function of the observed data $L(\text{data}; \boldsymbol{\theta}) = \prod_i \mathbb{P}(Y_i \mid \mathbf{X}_i)$
- Example data: $(\mathbf{X}_1, 1), (\mathbf{X}_2, 0), (\mathbf{X}_3, 1)$
$$L(\text{data}; \boldsymbol{\theta}) = \mathbb{P}(Y = 1 \mid \mathbf{X}_1) \cdot \mathbb{P}(Y = 0 \mid \mathbf{X}_2) \cdot \mathbb{P}(Y = 1 \mid \mathbf{X}_3)$$
- $\log L(\text{data}; \boldsymbol{\theta}) = \sum_{i \text{ of class } 0} \log \mathbb{P}(Y = 0 \mid \mathbf{X}_i) + \sum_{i \text{ of class } 1} \log \mathbb{P}(Y = 1 \mid \mathbf{X}_i)$
- “Messy, but:
 - convex
 - gradients are easy to calculate
 - efficient algorithms

Comments, caveats

- Much from linear regression applies here too
- Good predictions do not imply causality
- Can use features $\exp\{\boldsymbol{\theta}_k^T \mathbf{X}\} \longrightarrow \exp\{\boldsymbol{\theta}_k^T \phi(\mathbf{X})\}$
- Can add regularization terms
 - $\log L(\text{data}; \boldsymbol{\theta}) - \gamma \cdot \|\boldsymbol{\theta}\|^2$
or sum of absolute values
 - tune hyperparameter γ “experimentally”
- Evaluating performance on the training set is not enough
 - use separate validation set
- Confidence intervals, various tests,...

Results

- Recall that only 3.33% of the records correspond to defaults
 - **Always decide "no default:"** Miss-classification rate 3.33%
- Make predictions using only one component of \mathbf{X} ("balance")
 - LDA: 2.81%
 - QDA: 2.73%
 - Logistic: 2.73%
- Make predictions using all of \mathbf{X}
 - LDA: 2.75%
 - QDA: 2.70%
 - Logistic: 3.33%
 - Logistic with absolute value regularization: 2.66%

Unbalanced data sets and/or costs

$$\log L(\text{data}; \theta) = \sum_{i \text{ of class 0}} \log \mathbb{P}(Y = 0 | \mathbf{X}_i) + w \sum_{i \text{ of class 1}} \log \mathbb{P}(Y = 1 | \mathbf{X}_i)$$

- Solution pays less attention to classes with few samples
 - always declaring “class 0” does pretty well
- Problem is exacerbated if errors on defaulting customers are much more **costly**
- Solution: increase weight of “red” sum
 - equivalent to adding to the data set replicas of the “class 1” samples

defaulting customers in the data set

too few

Predicted labels

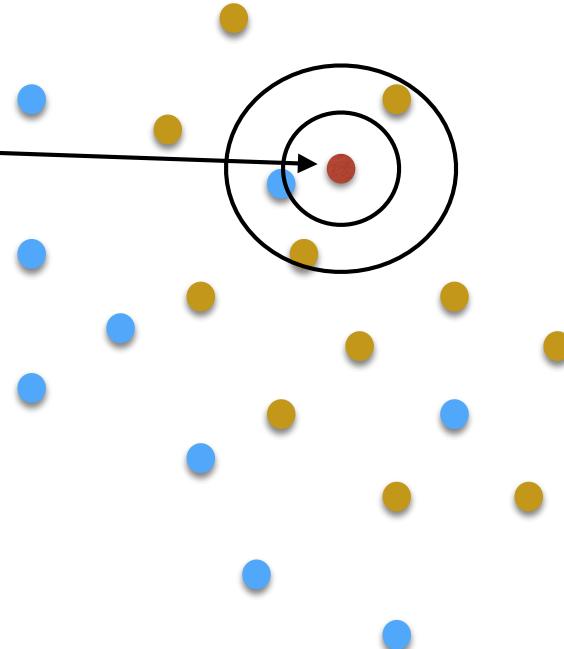
good
default

True labels	
good	default
good	9644
default	23
	252
	81

NEAREST-NEIGHBOR CLASSIFIERS

The k -NN classifier

- The nearest-neighbor classifier:
 - given new \mathbf{X}
 - find closest \mathbf{X}_i in the data set
(with smallest $\|\mathbf{X} - \mathbf{X}_i\|$)
 - prediction: the label Y_i of \mathbf{X}_i
- The k -NN classifier:
 - given new \mathbf{X}
 - find k closest \mathbf{X}_i in the data set
 - prediction: majority vote of the k labels



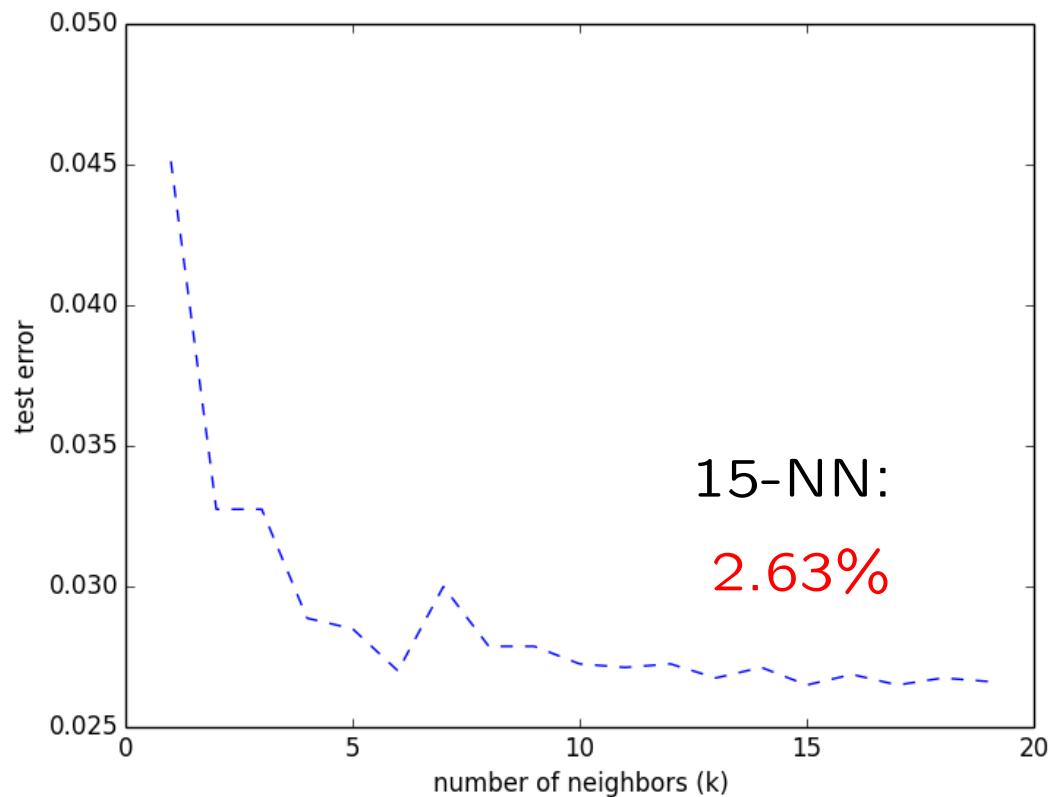
Comments on k -NN classifiers

- Simple, no training required
- Performance assessment:

	Error on training set	Error on validation set (20% of data set aside)
NN ($k = 1$):	0%	4.34%
always predict no default:	3.33%	3.33%
5-NN:	2.1%	2.99%

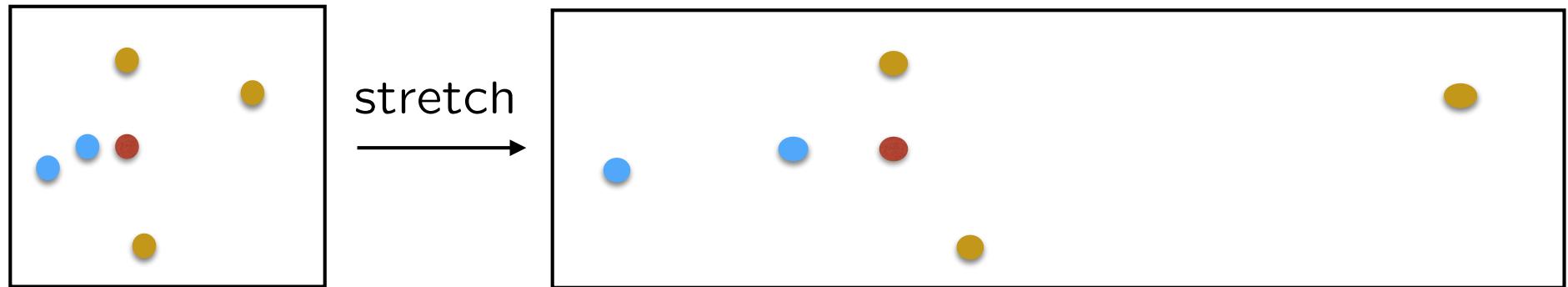
Selecting k

- Default data
 - set aside 20% of the data (validation set)
 - repeat 100 times



- This is an example of **hyperparameter search** to choose between models/algorithms

Scaling matters



- Scale so that the components of \mathbf{X} have comparable ranges
- equivalently: use a weighted distance metric,
e.g. $\|\mathbf{x}\| = \sqrt{x_1^2 + \mathbf{w} \cdot \mathbf{x}_2^2}$

Some other classification algorithms

- **Weighted NN**
 - k -NN: majority vote between k neighbors
 - Consider weighting the votes, as a function of the distance
 - every point could vote, far away points would have little weight
 - Similar idea applies to standard regression problems
 - estimate $\mathbb{E}[Y | \mathbf{X}]$ by weighing the Y_i 's of nearby \mathbf{X}_i **(kernel regression)**
- **Support vector machines**
- **Decision trees**
- **Neural networks**

Interpretability of models; explainability of predictions

- A generic issue in machine learning
- In regression, coefficients θ allow us to interpret the “logic” behind particular predictions

$$\widehat{\text{Sales}} = 2.94 + 0.046 \cdot (\text{TV}) + 0.19 \cdot (\text{Radio}) + 0.001 \cdot (\text{NewsP})$$

- Similarly in logistic regression $\mathbb{P}(Y = k | \mathbf{X}) = \frac{\exp\{\boldsymbol{\theta}_k^T \mathbf{X}\}}{\sum_s \exp\{\boldsymbol{\theta}_s^T \mathbf{X}\}}$
- Nearest neighbor methods do not yield an interpretable model
 - but allow us to explain the predictions

“Beth’s profile is similar to that of Barbara, Beatrice, and Bo...”
- A challenge for more complex machine learning models

Wrapping up

- Saw a number of methods
 - for prediction and “model-building”
- Encountered the main general methodologies
 - empirical risk minimization
 - maximum likelihood
 - plugin estimates
 - Bayesian
- “Classical” methods (regression, LDA, QDA, NN, etc.) form the starting point for more complex methods
- General toolbox for performance assessment (validation, bootstrap)
- Need care to avoid misinterpretations