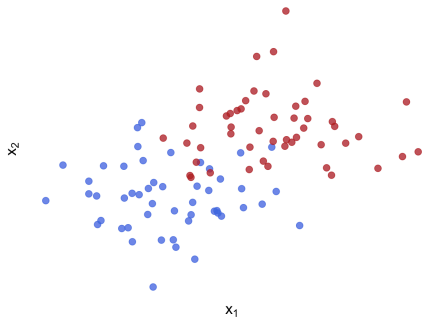
Classification II

Lecture 5

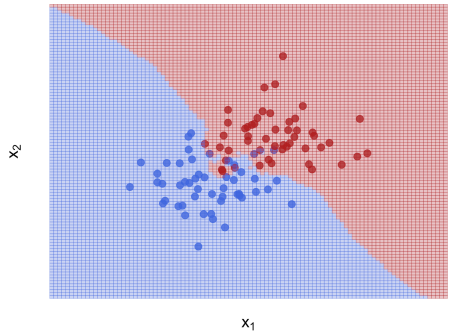
Termeh Shafie

K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN)



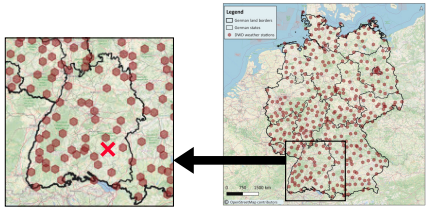
K-Nearest Neighbors (KNN)



K-Nearest Neighbors (KNN)

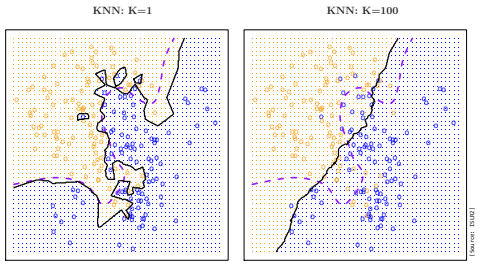
KNN regression tries predicting values of output variable by using a local average

KNN classification attempts predicting the class to which the output variable belong by computing the local probability



The Parameter K

- Smaller K $\Longrightarrow$ Less smooth and more sensitive to noise (more flexible)
- Larger K $\Longrightarrow$ More smooth (less flexible)



The Hyperparameter K

- Smaller K $\Longrightarrow$ Less smooth and more sensitive to noise (more flexible)
- Larger K $\Longrightarrow$ More smooth (less flexible)

how do we choose K?

- choose yourself
- let the data decide (hyperparameter tuning: more on this later...)

The Influence of the Similarity Function

The choice of the distance/similarity function is also important:

- Performance only be good if the distance function encodes "relevant information"
  Example: you want to classify mushrooms as "edible" or "not edible" and as distance function between mushrooms you use the difference in weight...
- Not so obvious sometimes how to define a good distance or similarity function
  Example: you want to classify the genre of songs but how do you compute a similarity between different songs?

Generative vs. Discriminative Classification Methods

Generative	Discriminant
• Learns a "recipe" for each class • Calculate probability based on recipe of a new point belonging to each class • Example: Naive Bayes, LDA, QDA	• Focus directly on distinguishing classes and not the data generating process • Draw the best possible boundary to separate the classes based on the data • Example: Logistic regression, KNN
↓	↓
Model $P(Y)$ and $P(X \mid Y)$, derive $P(Y \mid X)$	Model $\Pr(Y \mid X)$ directly

Naive Bayes

The Bayes in Naive Bayes

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

$$P(\text{category} \mid x_1, x_2, x_3) = \frac{P(x_1, x_2, x_3 \mid \text{category})P(\text{category})}{P(x_1, x_2, x_3)}$$

The Bayes in Naive Bayes

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

for computational efficiency

$$P(\text{heart attack} \mid \text{S, FH, HBP}) = \frac{P(\text{S, FH, HBP} \mid \text{heart attack})P(\text{heart attack})}{\cancel{P(\text{S, FH, HBP})}}$$

$$P(\overline{\text{heart attack}} \mid \text{S, FH, HBP}) = \frac{P(\text{S, FH, HBP} \mid \overline{\text{heart attack}})P(\overline{\text{heart attack}})}{\cancel{P(\text{S, FH, HBP})}}$$

The Bayes in Naive Bayes

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

$P(\text{heart attack} \mid \text{S, FH, HBP}) \propto P(\text{S, FH, HBP} \mid \text{heart attack})P(\text{heart attack})$

$P(\overline{\text{heart attack}} \mid \text{S, FH, HBP}) \propto P(\text{S, FH, HBP} \mid \overline{\text{heart attack}})P(\overline{\text{heart attack}})$

The Naive in Naive Bayes

features are conditionally independent given the class label

$$P(\text{S, FH, HBP}) = P(\text{S}) \cdot P(\text{FH}) \cdot P(\text{HBP})$$

Naive Bayes

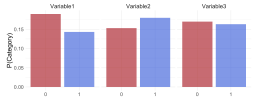
spam	dear	lunch	viagra	money
0	0.25	0.46	0.01	0.14
1	0.32	0.05	0.53	0.67

$$P(\text{category} \mid x_1, x_2, \ldots, x_p) \propto \prod_{i=1}^{p} P(x_i \mid \text{category}) \cdot P(\text{category})$$

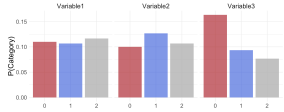
 [1,1,0,0]

Naive Bayes

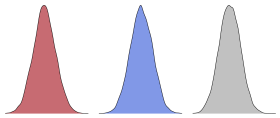
Bernoulli Naive Bayes



Categorical Naive Bayes



Gaussian Naive Bayes



Bayes Classifier vs. Naive Bayes

Warning: The Bayes classifier should not be confused with a Naive Bayes classifier!

- **Bayes optimal classifier** (or **Bayes classifier**) is a theoretical construct, not a practical classification method
- It is defined as the classifier that has the smallest test error rate and **assumes we know** $P(\textbf{category} \mid \textbf{predictors})$
- It's what we would ideally use if we knew the true data generating process

A probabilistic model-based approach to using Bayes classifier is:

1. Estimate the true distribution of test set from the training set
2. Use the Bayes optimal classifier for the estimated distribution

Discriminant Analysis

Linear Discriminant Analysis (LDA)

- Model the distribution of predictors in each category separately
- Use **Bayes theorem** to flip things around and obtain $P(\text{category} \mid \text{predictors})$
- Naive Bayes: features are ~~**conditionally independent given the class label**~~
- Now: **model the joint distribution of features given the class label**
  - assume distribution of the features within each category is normally distributed
  - assume covariances of the MVN distributions are equal for both classes
  - use the Bayes optimal classifier

Linear Discriminant Analysis (LDA) with 1 Predictor

- $f_k(x)$ is normal with following density in one dimension:

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right)$$

where μ_k and σ_k^2 are mean and variance of kth class and assume variances are equal

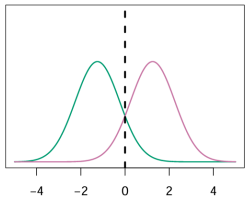
- Plug this into Bayes theorem

$$P_k(x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^K \pi_l f_l(x)} = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_k)^2\right)}{\sum_{l=1}^K \pi_l \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_l)^2\right)}$$

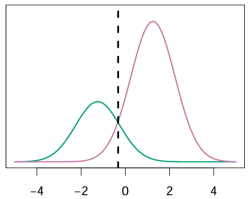
- The Bayes classifier assigns an observation to where the above is the largest which is equivalent to the largest: $\delta_k(x) = x\frac{\mu_k}{\sigma^2} + \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$
- This is **the linear discriminant classifier**

Linear Discriminant Analysis (LDA)

π_1=.5, π_2=.5

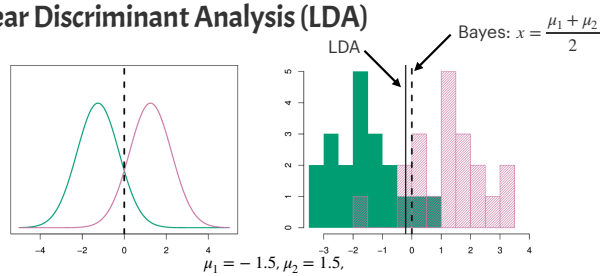


π_1=.3, π_2=.7



- the dashed lined represents the Bayes decision boundary (Bayes Classifier)
- we classify a new point to which density is highest
- when priors are different, take them into account and compare $\pi_k f_k(x)$
- on the right, we favor the pink class the decision boundary has shifted to the left

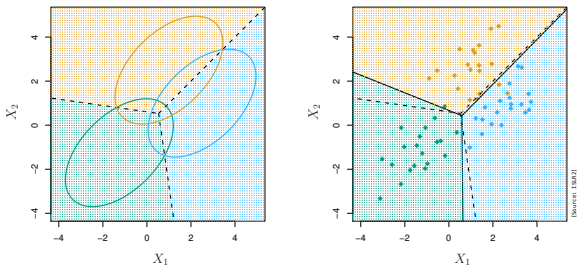
Linear Discriminant Analysis (LDA)



$$\mu_1 = -1.5, \mu_2 = 1.5,$$
$$\sigma_1^2 = \sigma_2^2 = 1, \pi_1 = \pi_2 = 0.5$$

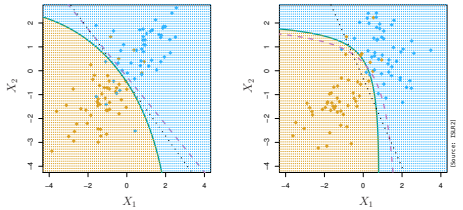
- typically we don't know these parameters
- in that case, we estimate them and plug them into the rule

LDA with three classes



Quadratic Discriminant Analysis (QDA)

does not assume a common covariance across classes for these MVNs



- LDA uses a linear decision rule (less flexible)
  - constrains such that it uses same covariance matrix for each class
- QDA uses a quadratic decision rule (more flexible)
  - allows each class k to have a different covariance matrix

KNN vs. Logistic Regression vs. LDA vs. QDA

- **KNN:** good when complex boundaries and n is sufficiently large
- **Logistic regression** and **LDA**: good when linear boundaries or p is big relative to n
  - LDA extends better to multi-class problems
  - LDA is more stable during estimation
  - Logistic regression is more robust to outliers
- **QDA**: good when quadratic (or moderately complex) boundaries and n is moderately big

Measuring Classification Performance

- Measure of classification performance is

  error rate = fraction of points that are classified incorrectly
- The training error rate is

$$\text{training error} = \frac{1}{n}\sum_{i=1}^{n} I(\hat{y}_i \neq y_i)$$

- The (expected) test error rate is given by $\mathbf{E}\left(I(\hat{Y}_0) \neq Y_0\right)$
- We have to construct $\hat{f}$ to minimize the test error rate
  $\Longrightarrow$ we need a loss function $L(\hat{y}, y)$ for penalizing errors in $\hat{y} = \hat{f}(x)$ when truth is y
- Strongly contingent on application

This Week's Practical

Continuing various classification methods and evaluating performance…