Freie Universität Berlin

Lecture: Machine Learning for Data Science

Winter semester 2021/22

Lectures 5: Classification (Naïve Bayes classifiers)

Prof. Dr. Eirini Ntoutsi

Recap

- Overfitting in general
- Overfitting in decision trees
- KNNs

Happiness check

- Any feedback on the course (lectures/tutorials)?
- How happy are you with the material/pace?
- How happy are you with the hybrid format
- Any wishes (e.g., further readings, programming exercises)?
- Other issues you would like to discuss

Outline

- A few more words on KNNs
- Generative vs Discriminative models
- Bayesian Classifiers
- Naïve Bayes classifiers
- Laplace correction
- Bayesian Belief Networks
- Things you should know from this lecture & reading material

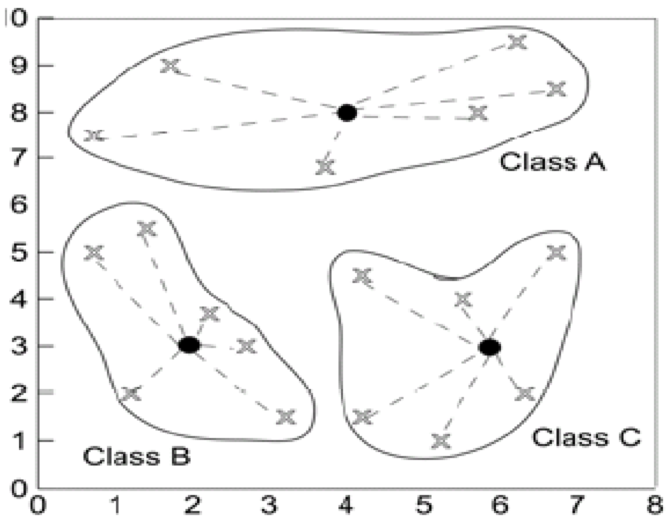
KNN classifiers: Inductive bias

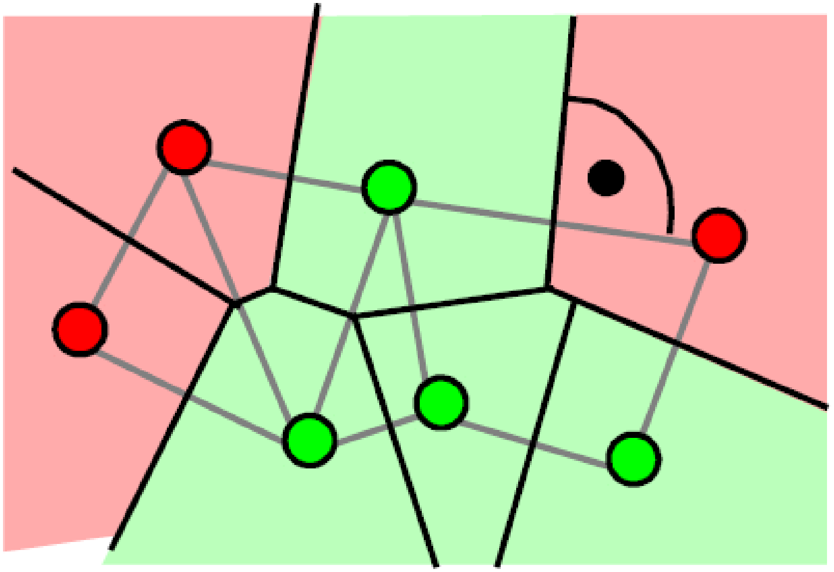
- Inductive bias: the set of assumptions that, together with the training data, deductively justify the classifications assigned by the learner to future instances.

- Inductive bias of KNN classifiers: What is the policy by which a KNN classifier generalizes from observed training examples to classify unseen instances?
  - Similar instances have similar class labels.
  - All attributes/dimensions contribute equally.
  - …

KNN classifiers: Decision boundaries

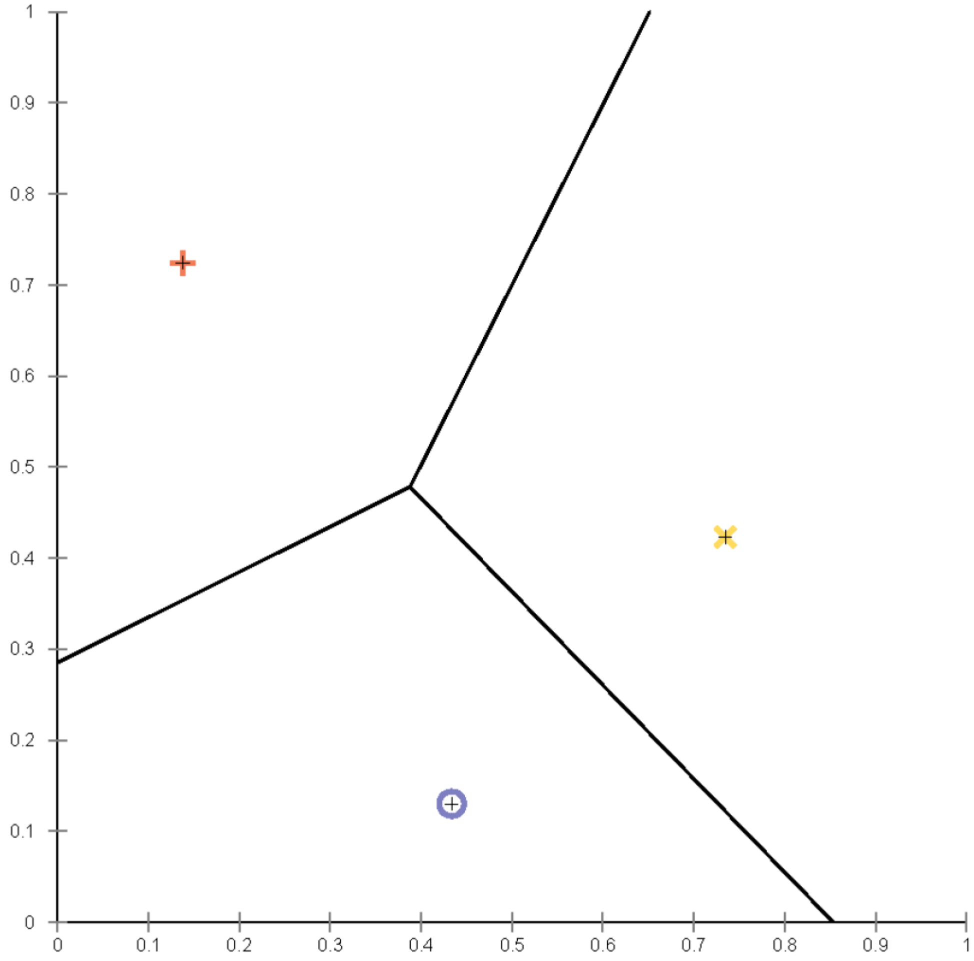
- Nearest-neighbor classifiers can produce arbitrarily shaped decision boundaries
- We can visualize class regions by Voronoi cells
  - Each cell is a region, a set of points that are closer to the training example than to any other training example in the dataset





KNN classifiers: Decision boundaries

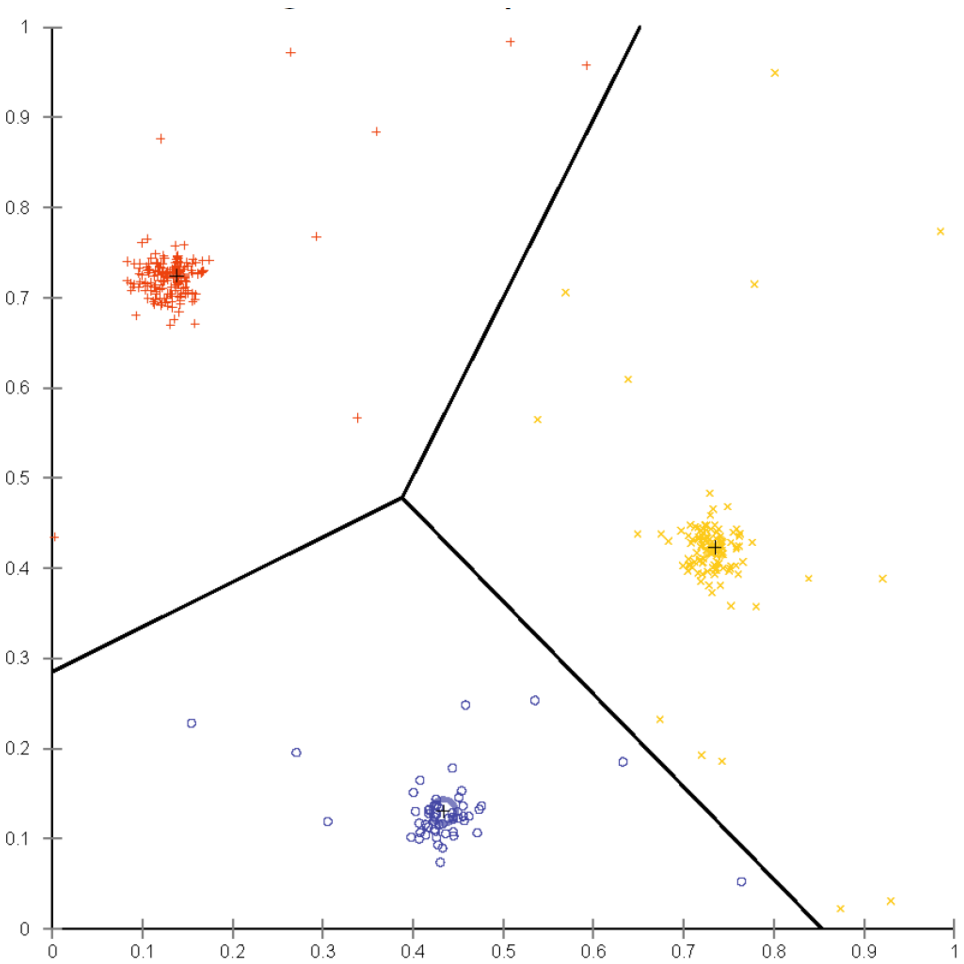
- Small training set → Simple decision boundaries



Slide by Arthur Zimek @SDU

KNN classifiers: Decision boundaries

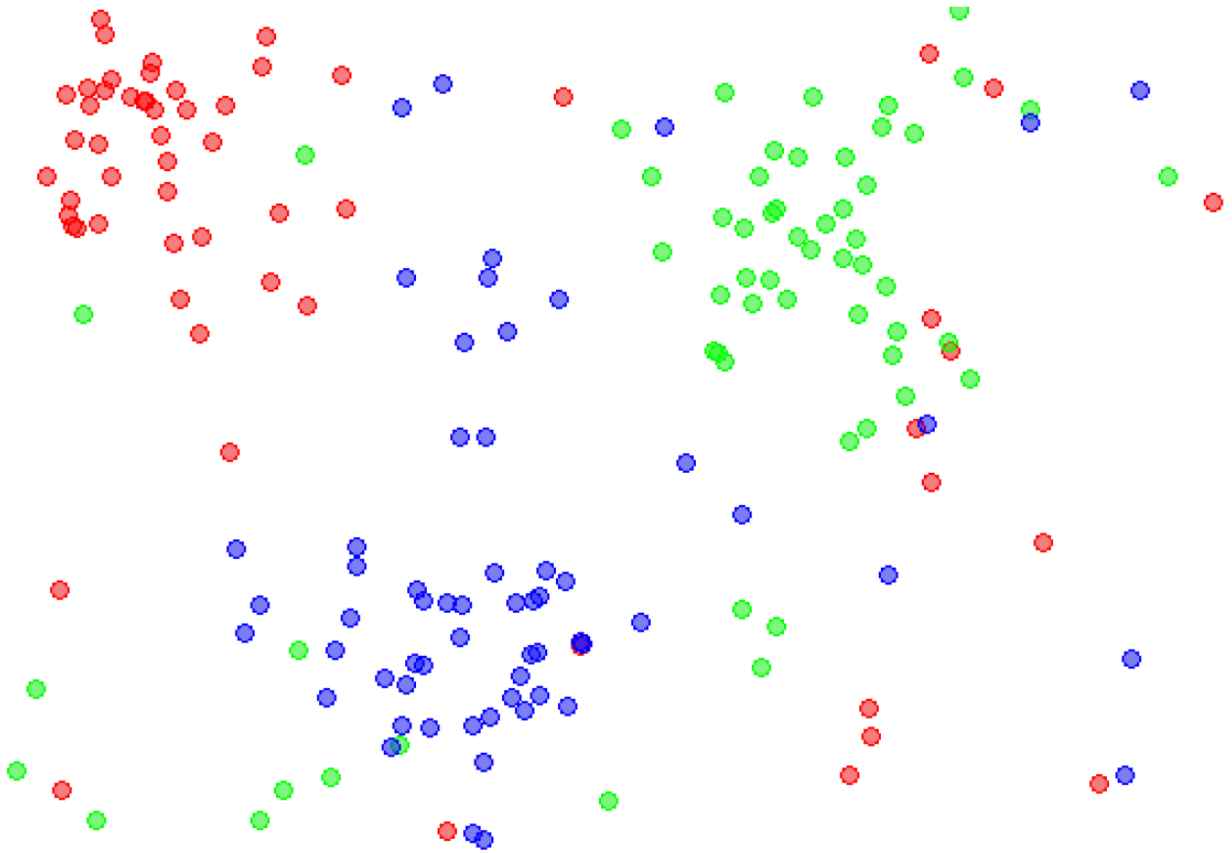
- Small training set → Simple decision boundaries



Slide by Arthur Zimek @SDU

KNN classifiers: Decision boundaries

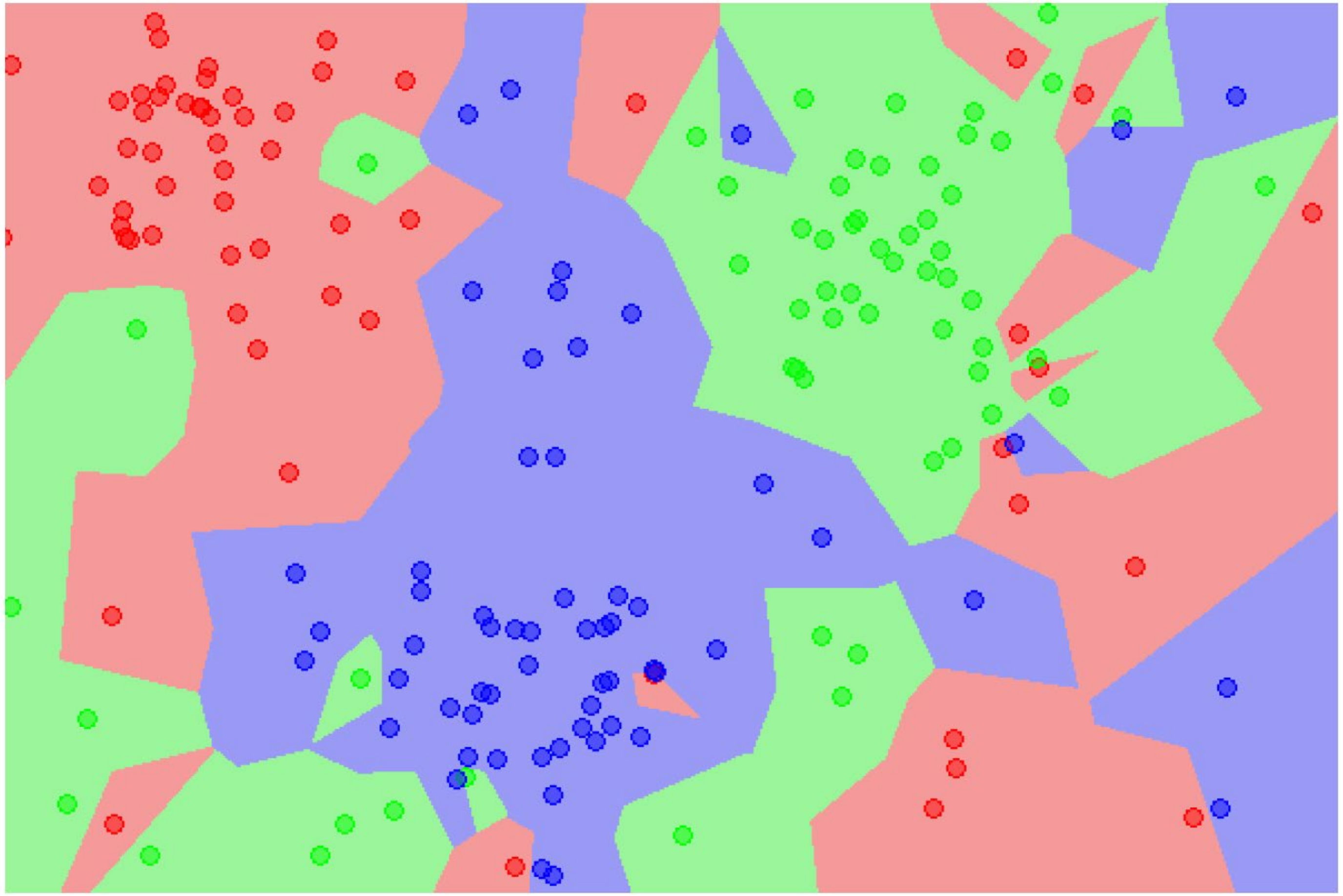
- Large training set → Potentially complex decision boundaries



Slide by Arthur Zimek @SDU

KNN classifiers: Decision boundaries

- Large training set → Potentially complex decision boundaries (k=1)

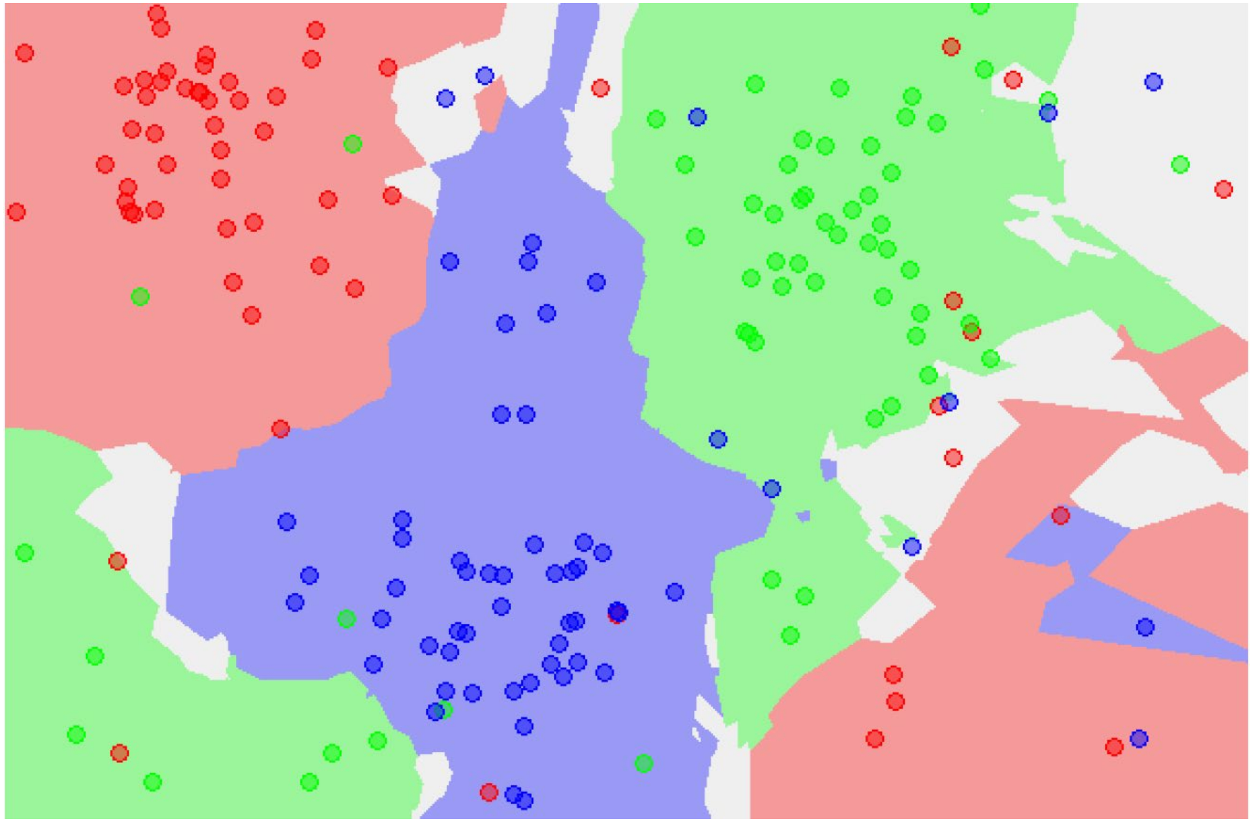


- Non-linear decision boundary
- Reflects the classes well

Slide by Arthur Zimek @SDU

KNN classifiers: Decision boundaries

- Large training set → Potentially complex decision boundaries (k=5)



Slide by Arthur Zimek @SDU

Outline

- A few more words on KNNs
- Generative vs Discriminative models
- Bayesian Classifiers
- Naïve Bayes classifier
- Laplace correction
- Bayesian Belief Networks
- Things you should know from this lecture & reading material

Generative vs Discriminative models

- Thus far, we assumed that there is an underlying distribution *P(X,Y)* that generates our population/data but we don't have access to this distribution
  - Instead, we have access to training data $D=\{(\vec{x_1}, y_1), \ldots, (\vec{x_n}, y_n)\}$ coming from this distribution
- Discriminative models
  - Try to find a mapping/function/hypothesis *h(): X→Y* that separates the different classes
  - New instances are classified using *h()*
- Generative models
  - Try to build a model for each individual class
    - It learns *P(X|Y)* and *P(Y)*
  - New instances are tested against different models and the most likely model is the class (using Bayes' rule) → *P(Y|X)*

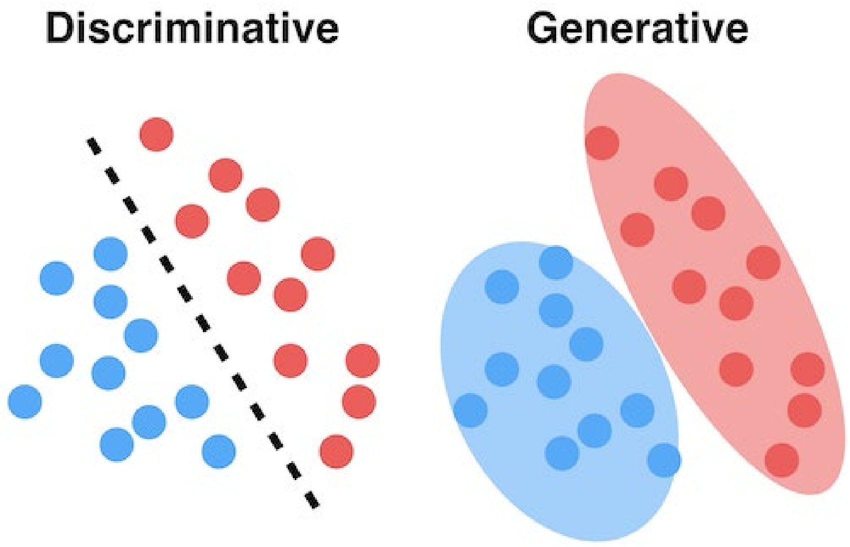


Image: link

Generative vs Discriminative models

- The discriminative approach has the advantage of directly optimizing the predictive performance instead of learning the underlying distribution
  - *"When solving a given problem, try to avoid a more general problem as an intermediate step."* Vapnik
- Usually it is harder to learn the underlying distribution than to learn an accurate predictor

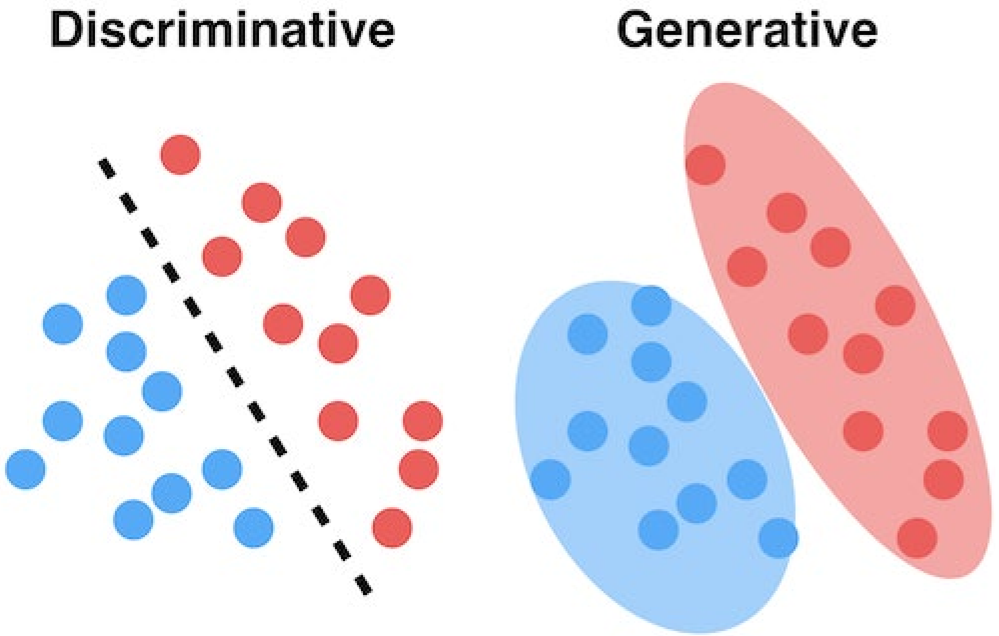


Image: [link]()

Outline

- Generative vs Discriminative models
- Bayesian Classifiers
- Naïve Bayes classifiers
- Laplace correction
- Bayesian Belief Networks
- Things you should know from this lecture & reading material

Bayesian classifiers

- A probabilistic framework for solving classification problems
- Predict class membership probabilities for an instance
  - The class of an instance is the most likely class for the instance (Maximum Likelihood classification)
- Based on Bayes' rule
- Bayesian classifiers
  - Naïve Bayes classifiers
    - Assume class-conditional independence among attributes
  - Bayesian Belief networks
    - Graphical models
    - Model dependencies among attributes
- A popular method for e.g.,: text classification, sentiment analysis

Bayes' theorem

- The probability of an event *C* given an observation *A*:

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

  - *P(C)*: prior
  - *P(A|C)*: likelihood
  - *P(A)*: evidence
  - *P(C|A)*: posterior

$$\boldsymbol{posterior} = \frac{\boldsymbol{prior} \times \boldsymbol{likelihood}}{\boldsymbol{evidence}}$$

Bayes' theorem

- The probability of an event *C* given an observation *A*:

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

posterior

prior

likelihood

- Example: Given that
  - A doctor knows that meningitis causes stiff neck 50% of the time
  - Prior probability of any patient having meningitis is *P(M)*=1/50,000
  - Prior probability of any patient having stiff neck is *P(S)*=1/20
- If a patient has stiff neck, what's the probability he/she has meningitis?

$$P(M \mid S) = ?$$

$$P(M \mid S) = \frac{P(S \mid M)P(M)}{P(S)} = ? \qquad \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

Bayes classification

- Let $C=\{c_1, c_2, \ldots, c_k\}$ be the class attribute.
- Let $X=(X_1, X_2, X_3, \ldots X_d)$ be a d-dimensional instance.
- Classification problem: What is the probability of a class $c \in C$ given instance X?

X_1	X_2	X_3	….	X_d	**?**

  - The event C to be predicted is the class of the instance
  - The observation is the instance values X

- The class of the instance is the class value with the highest probability: $\underset{c \in C}{\operatorname{argmax}} P(c|X)$
  - $P(c_1|X)$: posterior for c_1
  - $P(c_2|X)$: posterior for c_2
  - …
  - $P(c_k|X)$: posterior for c_k

Bayes classification

- Consider each attribute and class label as random variables
- Given an instance $X=(X_1, X_2, A_3,....X_d)$
  - Goal is to predict class label $c \in C$
  - Specifically, we want to find the value $c \in C$ that maximizes $P(c|X)$, i.e., $\operatorname*{argmax}_{c \in C} P(c|X)$

$$c = \operatorname*{argmax}_{c \in C} P(c|X)$$

Bayes' rule

$$c = \operatorname*{argmax}_{c \in C} \frac{P(c) * P(X|c)}{P(X)}$$

instances are identically distributed (i.i.d. assumption)

$$c = \operatorname*{argmax}_{c \in C} * \; P(c) * P(X|c)$$

max a posteriori = the most likely class

prior

likelihood

Bayes classification

- How can we estimate:

$$c = \underset{c \in C}{\operatorname{argmax}}\, P(c) * P(X|c)$$

  - Short answer: from the data

- How to compute the class priors *P(c)*:
  - Count the relative frequencies of the classes in the training set to estimate their priors
  - Example: What is P(Yes)?
    - $P(Yes) = \frac{3}{10}$
  - Example: What is P(No)?
    - $P(No) = \frac{7}{10}$

Tid	**Refund**	**Marital Status**	**Taxable Income**	**Evade**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

Bayes classification

- How can we estimate:

$$c = \underset{c \in C}{\operatorname{argmax}} P(c) * P(X|c)$$

  - Short answer: from the data

- How to compute instance likelihood *P(X|c)*:
  - What is the probability of an instance *X* given a class *c*?
  - $X=(X_1,X_2 \ldots X_d)$, so, $P(X|c)=P(X_1 \cap X_2 \cap \ldots \cap X_d|c)$
    - i.e., the probability of an instance given the class is equal to the probability of a set of features given the class

- So: $c = \underset{c \in C}{\operatorname{argmax}} P(c) * P(X_1 \cap X_2 \cap \cdots \cap X_d|c)$

Tid	**Refund**	**Marital Status**	**Taxable Income**	**Evade**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

Simplification: For single-dimensional data (d=1)

- How to compute $P(X|c) = P(X_1 \cap X_2 \cap \cdots \cap X_d|c)$?
- Let us assume X is univariate (so only 1 dimension X_1, i.e., d=1)

$$c = \underset{c \in C}{\operatorname{argmax}}\, P(c) * P(X_1|c)$$

- We can compute $P(X_1|c)$ from the data
- Depending on the data type for X_1 we distinguish between
  - Probability estimation for categorical attributes (e.g., hair color)
  - Probability estimation for continuous attributes (e.g., income)

Probability estimation for categorical attributes

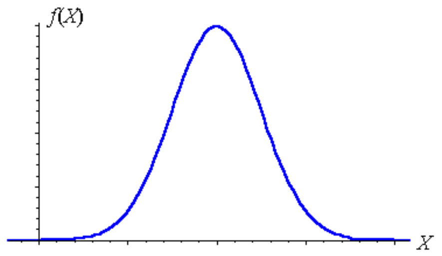
- Let X_1 be a categorical attribute, what is $P(X_1|c)$?
  - E.g., Marital status={Yes, No}, Color={green, blue, red}
- Based on the relative frequency of value X_1 in class c in the training data
  - $P(X_1|c) = \frac{n_{1c}}{n_c}$
  - n_c:#instances in class c
  - n_{1c}:#instances in class c having value X_1
- Example: What is P(Status=Married|No)?
  - $= \frac{4}{7}$
- Example: What is P(Refund=No|Yes)?
  - $= \frac{3}{3}$

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Probability estimation for continuous attributes

- Let X_1 be a continuous attribute, what is $P(X_1|c)$?
  - E.g., Income
- Idea 1: Discretization
  - Discretize the attribute → categorical attribute case
- Idea 2: Probability density estimation
  - Assume the attribute follows a known distribution
    - For example, Gaussian
  - Use data to estimate parameters of distribution
    - For example, mean and standard deviation
  - Once probability distribution is known, can be used to estimate the conditional probability $P(X_1|c)$

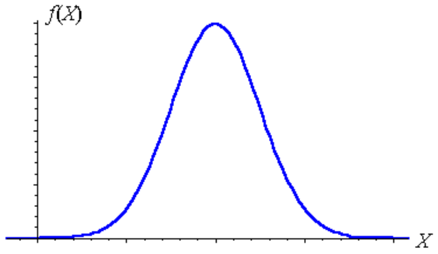
Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Probability estimation for continuous attributes (Gaussian NBs)

- Example: Assume income follows the Gaussian distribution



- Use data to estimate the parameters of the distribution for each class, i.e., mean and standard deviation (See lecture 2)
  - For class No: mean income = 110
  - For class No: variance σ^2=2975
- Once probability distribution is known, can be used to estimate the conditional probability *$P(X_1|c)$*
  - E.g., what is the probability of income value 120 in class No?

$$P(Income = 120 \mid No) = \frac{1}{\sqrt{2\pi 2975}} e^{-\frac{(120-110)^2}{2(2975)}} = 0.0072$$

Tid	**Refund**	**Marital Status**	**Taxable Income**	**Evade**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Multidimensional data

- So far, we consider only one attribute for probability estimation
- For multi-dimensional data , we need to estimate the combined probabilities of specific attribute values:

$$P(X|c) = P(X_1 \cap X_2 \cap \cdots \cap X_d|c)$$

- Example:
  - P(Refund=yes ∩Marital status=single|No)
    - =1/10
  - P(Refund=no ∩Marital status=married|No)=
    - 3/10
  - …
- If we have *d* attributes each taking *k* values, we have k^d different combinations
- Typically, there are not enough training instances available to reliably estimate probabilities.

Tid	**Refund**	**Marital Status**	**Taxable Income**	**Evade**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

Multidimensional data: Example

ID	shape	color	class
1	round	orange	A
2	round	green	A
3	round	yellow	A
4	square	green	A
5	oval	white	B

$$\Pr(\text{shape=round} \cap \text{color=orange}|A) = \frac{1}{4}$$
$$\Pr(\text{shape=round} \cap \text{color=green}|A) = \frac{1}{4}$$
$$\Pr(\text{shape=round} \cap \text{color=yellow}|A) = \frac{1}{4}$$
$$\Pr(\text{shape=round} \cap \text{color=white}|A) = \frac{0}{4}$$

$$\Pr(\text{shape=oval} \cap \text{color=orange}|A) = \frac{0}{4}$$
$$\Pr(\text{shape=oval} \cap \text{color=green}|A) = \frac{0}{4}$$
$$\Pr(\text{shape=oval} \cap \text{color=yellow}|A) = \frac{0}{4}$$
$$\Pr(\text{shape=oval} \cap \text{color=white}|A) = \frac{0}{4}$$
$$\Pr(\text{shape=square} \cap \text{color=orange}|A) = \frac{0}{4}$$
$$\Pr(\text{shape=square} \cap \text{color=green}|A) = \frac{1}{4}$$
$$\Pr(\text{shape=square} \cap \text{color=yellow}|A) = \frac{0}{4}$$
$$\Pr(\text{shape=square} \cap \text{color=white}|A) = \frac{0}{4}$$
$$\Pr(\text{shape=round} \cap \text{color=orange}|B) = \frac{0}{1}$$

- The probability estimates are unreliable, because the sample size is too small for each instance.

Slide by Arthur Zimek @SDU

Outline

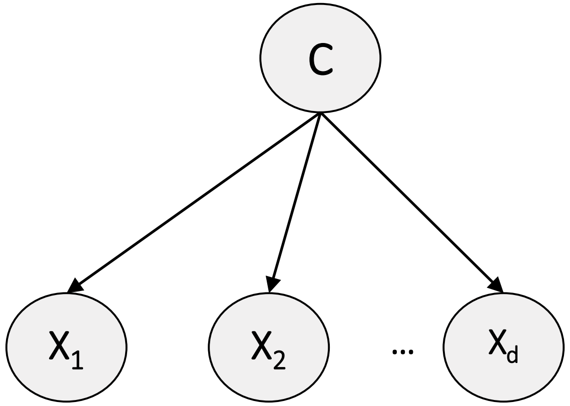
- Generative vs Discriminative models
- Bayesian Classifiers
- Naïve Bayes classifier
- Laplace correction
- Bayesian Belief Networks
- Things you should know from this lecture & reading material

Naïve Bayes classifier

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- How to estimate instance likelihood $P(X|c)=P(X \cap {}_1X_2 \cap \ldots \cap X_d|c)$?
- Assume independence among attributes X_i when class is given

$$P(X_1 \cap X_2 \cap \ldots \cap X_d|c) = \underbrace{\prod P(X_i|c) = P(X_1|c)(P(X_2|c) \ldots P(Xd|c)}_{\text{Strong class conditional independence assumption!!!}}$$



- The class value is the hidden factor that explains all the dependencies between the attributes.
- In other words, once the class is observed an attribute does not give any information about other attributes

What class conditional independence means?
E.g., in the context of sentiment analysis/ spam filtering?

Naïve Bayes classifier

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Assume independence among attributes X_i when class is given:

$$P(X_1 \cap X_2 \cap \ldots \cap X_d / c) = \underbrace{\prod (X_i|c) = P(X_1|c)(P(X_2|c) \ldots P(Xd|c)}_{\text{Strong class conditional independence assumption!!!}}$$

- Methodology
  - Using the independence assumption estimate $P(X/c)$ for all classes $c \in C$ based on the training data
  - The instance X is finally classified into:

$$c = \underset{c \in C}{\operatorname{argmax}}\, P(X|c) * P(c)$$

Naive Bayes classifier: Example 1

Training set

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Test instance *X*

Outlook	Temperature	Humidity	Wind	Play
Sunny	Cool	High	Strong	?

$$\mathrm{P(yes \mid X)} = \frac{\mathrm{P(X \mid yes)P(yes)}}{P(X)} = \frac{\mathrm{P(O="sunny" \mid yes)P(T="}cool\mathrm{" \mid yes)P(H="high" \mid yes)P(W="strong" \mid yes)P(yes)}}{P(X)}$$

$$\mathrm{P(O="sunny" \mid yes)} = \frac{2}{9} \qquad \mathrm{P(T="}cool\mathrm{" \mid yes)} = \frac{3}{9} \qquad \mathrm{P(H="}high\mathrm{" \mid yes)} = \frac{3}{9} \qquad \mathrm{P(W="}strong\mathrm{" \mid yes)} = \frac{3}{9}$$

$$\mathrm{P}(yes) = \frac{9}{14}$$

$$\mathrm{P(no \mid X)} = \frac{\mathrm{P(X \mid no)P(no)}}{P(X)} = \frac{\mathrm{P(O="sunny" \mid no)P(T="}cool\mathrm{" \mid no)P(H="high" \mid no)P(W="strong" \mid no)P(no)}}{P(X)}$$

Naive Bayes classifier: Example 2

Training set

Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals
dolphin	yes	no	yes	no	mammals
eagle	no	yes	no	yes	non-mammals

$$P(X \mid M) = \frac{6}{7} \times \frac{6}{7} \times \frac{2}{7} \times \frac{2}{7} = 0.06$$

$$P(X \mid N) = \frac{1}{13} \times \frac{10}{13} \times \frac{3}{13} \times \frac{4}{13} = 0.0042$$

$$P(M \mid X) = P(X \mid M)P(M) = 0.06 \times \frac{7}{20} = 0.021$$

$$P(N \mid X) = P(X \mid N)P(N) = 0.004 \times \frac{13}{20} = 0.0027$$

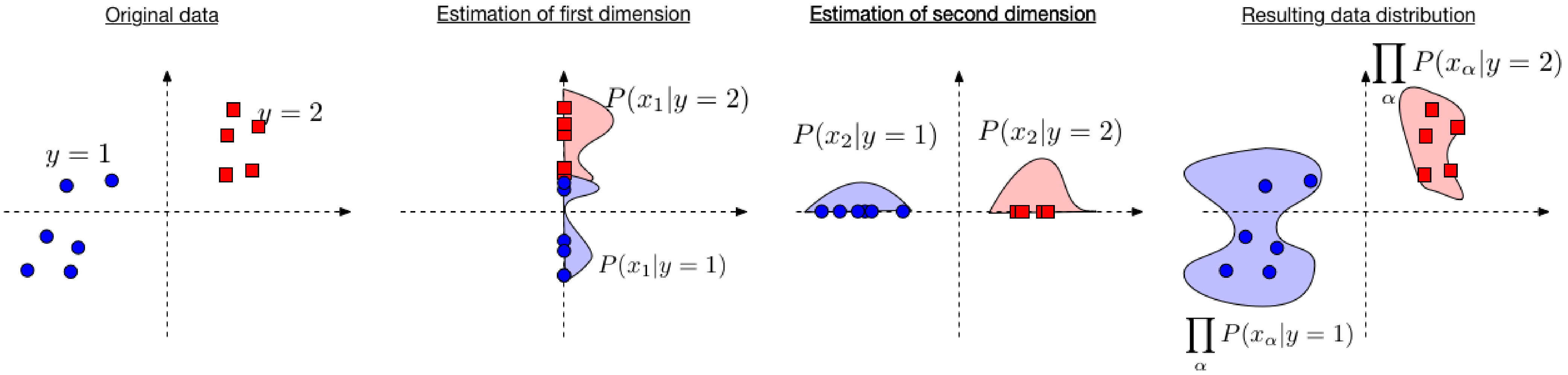
Test instance *X*

Give Birth	Can Fly	Live in Water	Have Legs	Class
yes	no	yes	no	?

P(M|X) > P(N|X) ➔ Mammals

Naive Bayes classifier: Example 3

- We estimate *P(X|c)* independently in each dimension
- We multiple these probabilities under the conditional independence assumption



Source: [link]

Outline

- Generative vs Discriminative models
- Bayesian Classifiers
- Naïve Bayes classifier
- Laplace correction
- Bayesian Belief Networks
- Things you should know from this lecture & reading material

NB predictions

- Consider the following example

Traning set

- perfect location
- expensive breakfast
- expensive breakfast, fair rooms
- perfect location, fair price
- fair parking facilities

NB model

Word-class distribution

perfect: (2,0)
expensive: (0,2)
fair: (2,1)

Class distribution

+: 3
-: 2

- Predict the class of the following instances:
  - *X*="Perfect breakfast"
  - *X*="Perfect weather"

The zero frequency problem

- Naïve Bayesian prediction requires each conditional probability $P(X_i|c)$ be non-zero. Otherwise, the predicted probability will be zero

$$c = \underset{c \in C}{\operatorname{argmax}} P(X|c) * P(c)$$

- Solution: Correction/Smoothing:

$$\text{Original}: P(X_i \mid c) = \frac{n_{ic}}{n_c}$$

$$\text{Laplace}: P(X_i \mid c) = \frac{n_{ic} + 1}{n_c + k}$$

k: number of classes

The problem of 0-probabilities: example

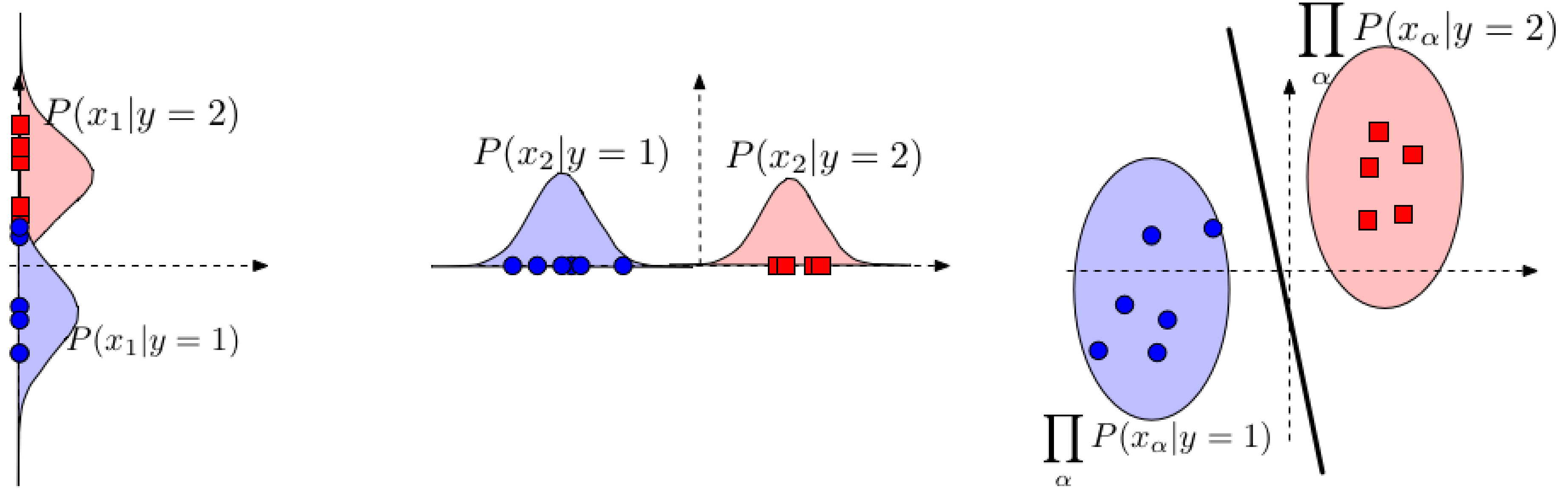
- Suppose a dataset with 1000 tuples:
  - income=low (0)
  - income= medium (990)
  - income = high (10)
- Use Laplacian correction (or Laplacian estimator): add 1 to each class value
  - Prob(income = low) = 1/1003
  - Prob(income = medium) = 991/1003
  - Prob(income = high) = 11/1003
- Result
  - The probabilities are never 0
  - The "corrected" prob. estimates are close to their "uncorrected" counterparts

Inductive bias of NBs

- The assumption of independence can be seen as the bias inherent to the Naïve Bayes classifier.
- Relying on the bias, the classifier may have a tendency to be wrong (if the assumption does not hold).
- An unbiased probabilistic classifier is not practical due to a notorious lack of training examples.
  - In any practical scenario, it would overfit.
- As an example: For a binary classification problem, with 30 binary predictive attributes it would require more than 2 billion instances just to see each combination once (which is not sufficient for reliable probability estimates)
- So bias is necessary to make generalization feasible.

Decision boundary

- Naive Bayes leads to a linear decision boundary in many common cases
- Illustrated here is the case of Gaussian NB where standard deviation is the same across all classes



Source: [Link]

Naïve Bayes (NB) classifiers: overview

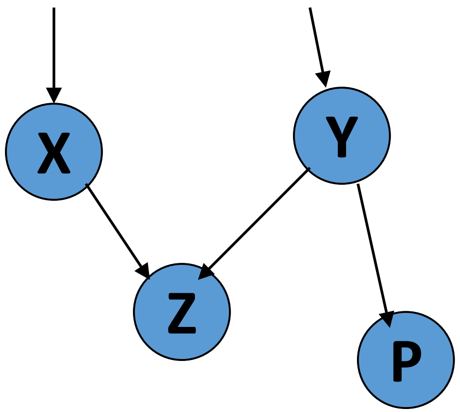
- (+) Easy to implement
- (+) It works surprisingly well in practice, although the independence assumption is too strong.
  - It does not require precise estimations of the probabilities
  - It is enough if the max probability belongs to the correct class
- (+) Robust to irrelevant attributes
- (+) Handles missing values by ignoring the value during probability estimate calculations
- (+) Robust to noise
- (+) Incremental
- (-) Strong independence assumption
- (-) Practically, there exist dependencies among variables
  - Such dependencies cannot be modeled by NB classifiers
  - Use other techniques such as Bayesian Belief Networks (BBN)

Outline

- Generative vs Discriminative models
- Bayesian Classifiers
- Naïve Bayes classifier
- Laplace correction
- Bayesian Belief Networks
- Things you should know from this lecture & reading material

Bayesian Belief Networks (BBN)

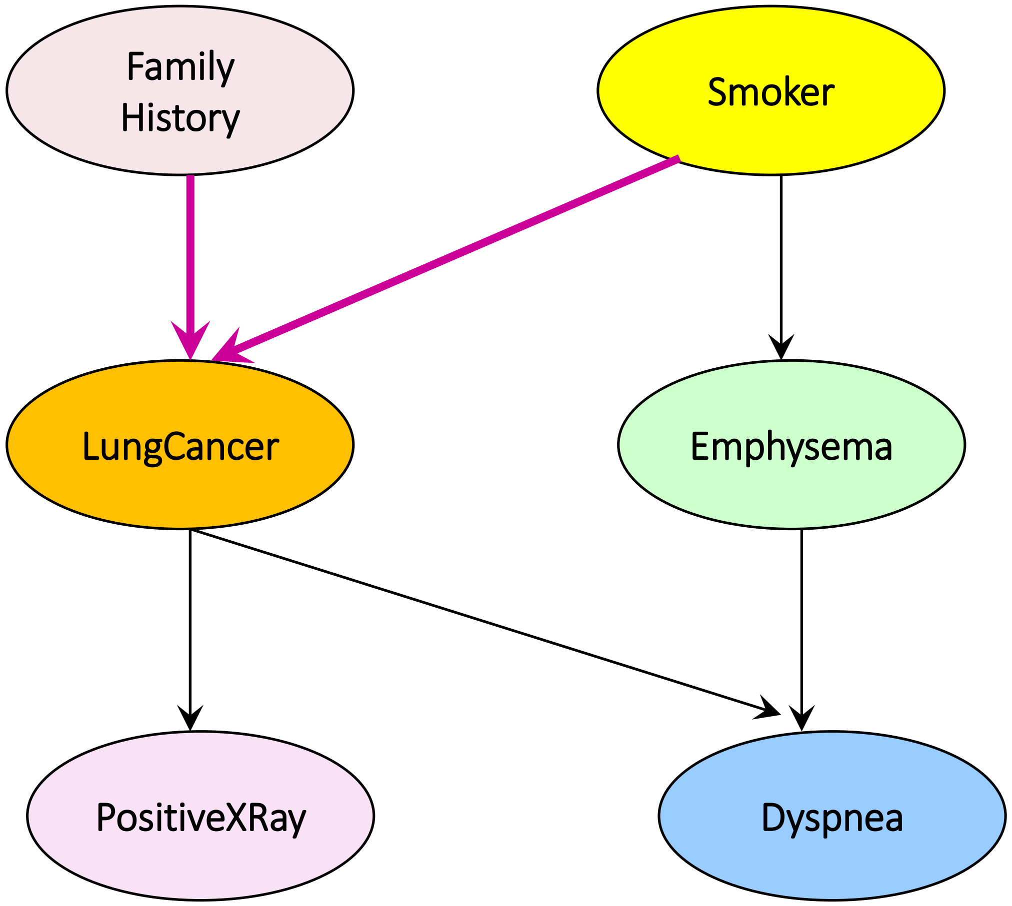
- Bayesian belief networks allow class conditional independence to be defined between subsets of variables instead of all variables (as in NB).
- They provide a graphical model of causal relationships, on which learning can be performed
- A belief network is defined by two components:
  - A directed acyclic graph of nodes encoding the dependence relationships among a set of variables.
  - A set of conditional probability tables (CPT) that associates each node to its immediate parent nodes.



- Nodes: random variables
- Links: dependency between variables
- X, Y are the parents of Z; Y is the parent of P.

Conditional independence in BBN

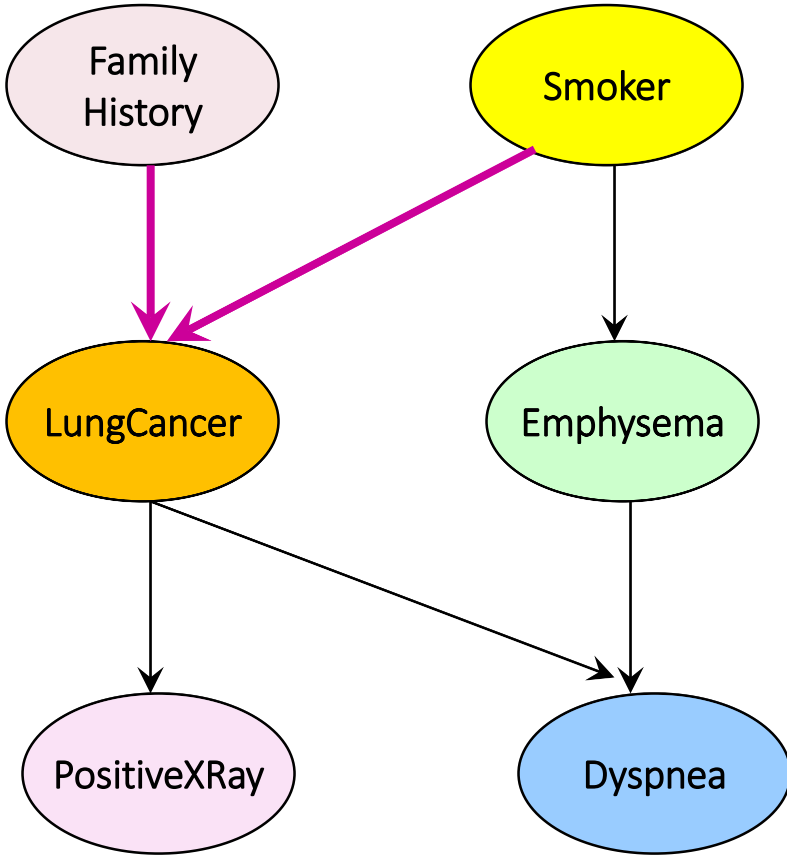
- A node in a Bayesian network is conditional independent of its non-descendants, if its parents are known.
- In our example
  - having lung cancer is influenced by a person's family history and on whether or not the person is a smoker
  - PositiveXRay is independent of "family history" and "smoker" attributes once we know that the person has LungCancer.



Bayesian Belief Networks

- Each variable X_i is associated with a conditional probability table (CPT)
- CPT of X_i specifies the conditional distribution $P(X_i \mid Parents(X_i))$
- Let the conditional probability table (CPT) for variable LungCancer (LC):

	(FH, S)	(FH, ~S)	(~FH, S)	(~FH, ~S)
LC	0.8	0.5	0.7	0.1
~LC	0.2	0.5	0.3	0.9



- Let a new instance $X = (X_1, X_2, \ldots, X_d)$.
- The probability of X is given by:

$$P(X) = \prod_{i=1}^{d} P(X_i \mid Parents(X_i))$$

- Key challenge: How to get the CPTs

Outline

- Bayesian Classifiers
- Naïve Bayes classifier
- Laplace correction
- Bayesian Belief Networks
- Things you should know from this lecture & reading material

Overview and Reading

- Overview
  - Bayes rule
  - Bayesian classifiers
  - Naïve Bayes classifiers
  - Laplace correction

- Reading
  - Chapter 24: Generative Models, Understanding Machine Learning book by Shai Shalev-Schwartz and Shai Ben-David
  - Chapter 5, Bayesian classifiers, Tan et al book

Hands on experience

- Build a sentiment classifier for Twitter data (is a tweet positive or negative?)
  - Many datasets available
    - E.g., [Sentiment140 dataset]
- Classify newsgroup messages (20 newsgroups from politics to atheism)
  - Dataset: [20 newsgroup dataset]
- Spam filter (is tweet spam or not?)
  - Dataset: [HSPAM]

- If features do not represent categories but counts, we use multinomial distribution →Multinomial Naïve Bayes

Acknowledgements

- The slides are based on
  - KDD I lecture at LMU Munich (Johannes Aßfalg, Christian Böhm, Karsten Borgwardt, Martin Ester, Eshref Januzaj, Karin Kailing, Peer Kröger, Eirini Ntoutsi, Jörg Sander, Matthias Schubert, Arthur Zimek, Andreas Züfle)
  - Introduction to Data Mining book slides at http://www-users.cs.umn.edu/~kumar/dmbook/
  - Pedro Domingos Machine Lecture course slides at the University of Washington
  - Machine Learning book by T. Mitchel slides at http://www.cs.cmu.edu/~tom/mlbook-chapter-slides.html
  - (DTs) J. Fürnkranz slides from TU Darmstadt (https://www.ke.tu-darmstadt.de/lehre/archiv/ws0809/mldm/)
  - Thank you to all TAs contributing to their improvement, namely Vasileios Iosifidis, Damianos Melidis, Tai Le Quy, Han Tran.

Thank you

Questions/Feedback/Wishes?

Acknowledgements

- The slides are based on
  - KDD I lecture at LMU Munich (Johannes Aßfalg, Christian Böhm, Karsten Borgwardt, Martin Ester, Eshref Januzaj, Karin Kailing, Peer Kröger, Eirini Ntoutsi, Jörg Sander, Matthias Schubert, Arthur Zimek, Andreas Züfle)
  - Introduction to Data Mining book slides at http://www-users.cs.umn.edu/~kumar/dmbook/
  - Pedro Domingos Machine Lecture course slides at the University of Washington
  - Machine Learning book by T. Mitchel slides at http://www.cs.cmu.edu/~tom/mlbook-chapter-slides.html
  - Arthur Zimek DMML lecture at SDU
  - Thank you to all TAs contributing to their improvement, namely Vasileios Iosifidis, Damianos Melidis, Tai Le Quy, Han Tran.