Soft Computing

Lecture Notes on Machine Learning

Matteo Matteucci

matteucci@elet.polimi.it

Department of Electronics and Information

Politecnico di Milano

Supervised Learning

– Bayes Classifiers –

Maximum Likelihood vs. Maximum A Posteriory

According to the probability we want to maximize

- MLE (Maximum Likelihood Estimator):

$$\hat{Y} = \arg\max_{v_i} P(X_1, X_2, \ldots, X_m | Y = v_i)$$

- MAP (Maximum A Posteriori Estimator):

$$\hat{Y} = \arg\max_{v_i} P(Y = v_i | X_1, X_2, \ldots, X_m)$$

Maximum Likelihood vs. Maximum A Posteriory

According to the probability we want to maximize

- MLE (Maximum Likelihood Estimator):

$$\hat{Y} = \arg\max_{v_i} P(X_1, X_2, \ldots, X_m | Y = v_i)$$

- MAP (Maximum A Posteriori Estimator):

$$\hat{Y} = \arg\max_{v_i} P(Y = v_i | X_1, X_2, \ldots, X_m)$$

We can compute the second by applying the Bayes Theorem:

$$\begin{aligned} P(Y = v_i | X_1, X_2, \ldots, X_m) &= \frac{P(X_1, X_2, \ldots, X_m | Y = v_i) P(Y = v_i)}{P(X_1, X_2, \ldots, X_m)} \\ &= \frac{P(X_1, X_2, \ldots, X_m | Y = v_i) P(Y = v_i)}{\sum_{j=0}^{n_Y} P(X_1, X_2, \ldots, X_m | Y = v_j) P(Y = v_j)} \end{aligned}$$

Bayes Classifiers Unleashed

Using the MAP estimation, we get the Bayes Classifier:

- Learn the distribution over inputs for each value Y
  - This gives $P(X_1, X_2, \ldots, X_m | Y = v_i)$
- Estimate $P(Y = v_i)$ as fraction of records with $Y = v_i$
- For a new prediction:

$$\begin{aligned} \hat{Y} &= \arg\max_{v_i} P(Y = v_i | X_1, X_2, \ldots, X_m) \\ &= \arg\max_{v_i} P(X_1, X_2, \ldots, X_m | Y = v_i) P(Y = v_i) \end{aligned}$$

Bayes Classifiers Unleashed

Using the MAP estimation, we get the Bayes Classifier:

- Learn the distribution over inputs for each value Y
  - This gives $P(X_1, X_2, \ldots, X_m | Y = v_i)$
- Estimate $P(Y = v_i)$ as fraction of records with $Y = v_i$
- For a new prediction:

$$\begin{aligned} \hat{Y} &= \arg\max_{v_i} P(Y = v_i | X_1, X_2, \ldots, X_m) \\ &= \arg\max_{v_i} P(X_1, X_2, \ldots, X_m | Y = v_i) P(Y = v_i) \end{aligned}$$

You can plug any density estimator to get your flavor of Bayes Classifier:

- Joint Density Estimator
- Naïve Density Estimator
- ...

Unsupervised Learning

– Density Estimation –

The Joint Distribution

Given two random variables X and Y, the joint distribution of X and Y is the distribution of X and Y together: $P(X, Y)$.

The Joint Distribution

Given two random variables X and Y, the joint distribution of X and Y is the distribution of X and Y together: $P(X, Y)$.

How to make a joint distribution of M variables:

1. Make a truth table listing all combination of values
2. For each combination state/compute how probable it is
3. Check that all probabilities sum up to 1

The Joint Distribution

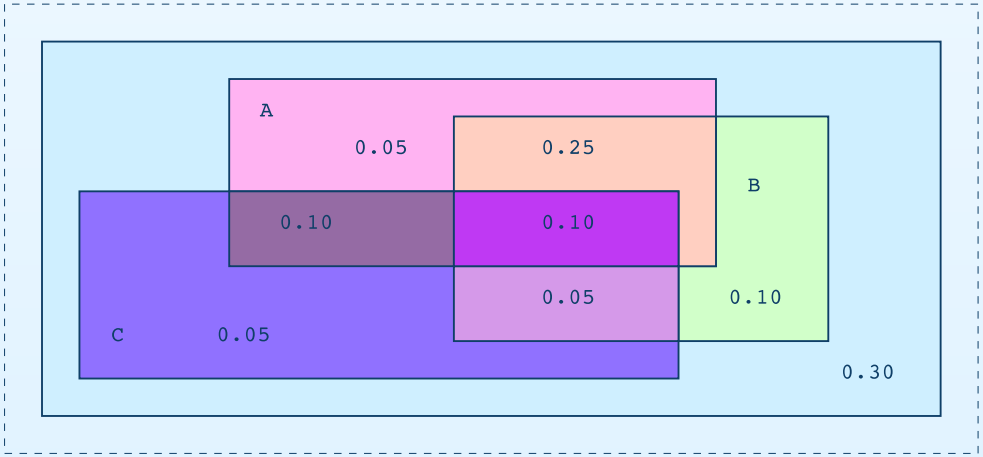
Given two random variables X and Y, the joint distribution of X and Y is the distribution of X and Y together: $P(X, Y)$.

How to make a joint distribution of M variables:

1. Make a truth table listing all combination of values
2. For each combination state/compute how probable it is
3. Check that all probabilities sum up to 1

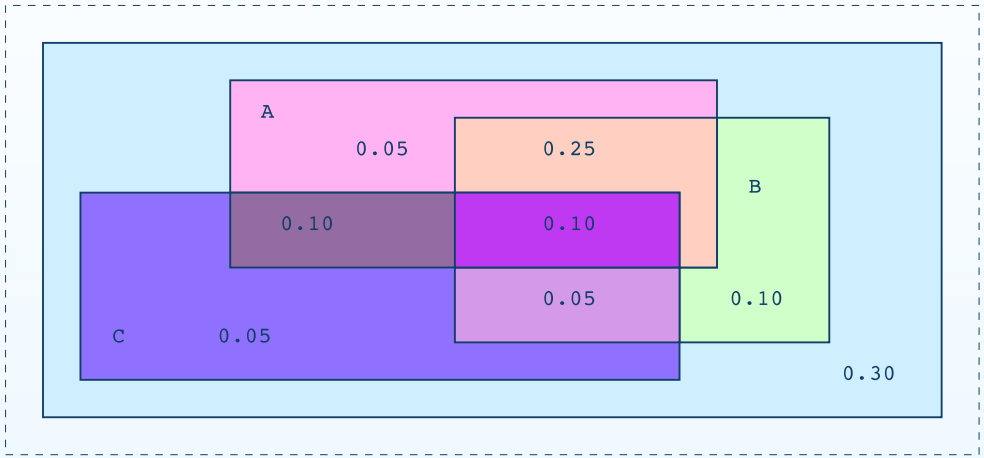
Example with 3 boolean variables A, B and C.

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10



Using the Joint Distribution (I)

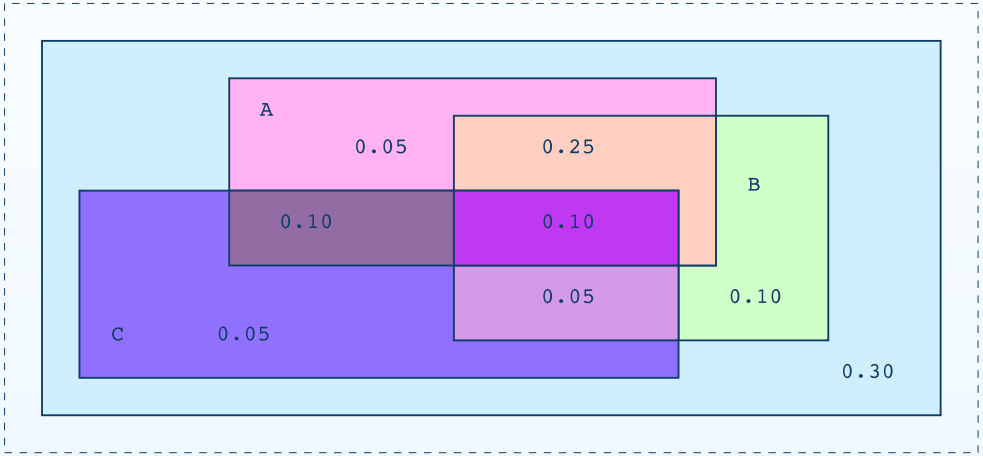
A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10



Compute **probability for logic expression**: $P(E) = \sum_{Row \sim E} P(Row)$.

Using the Joint Distribution (I)

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10

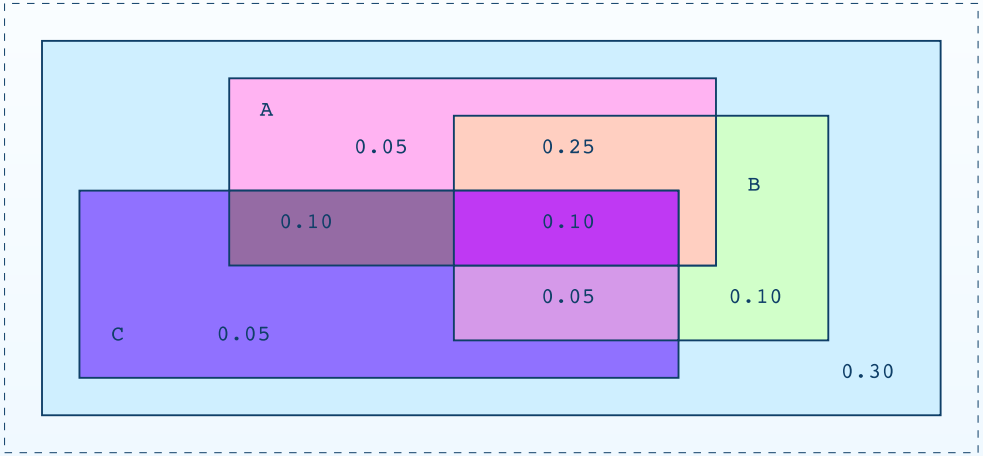


Compute **probability for logic expression**: $P(E) = \sum_{Row \sim E} P(Row)$.

- $P(A) = 0.05 + 0.10 + 0.25 + 0.10 = 0.5$
- $P(A \wedge B) = 0.25 + 0.10 = 0.35$
- $P(\bar{A} \vee C) = 0.30 + 0.05 + 0.10 + 0.05 + 0.05 + 0.25 = 0.8$

Using the Joint Distribution (I)

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10



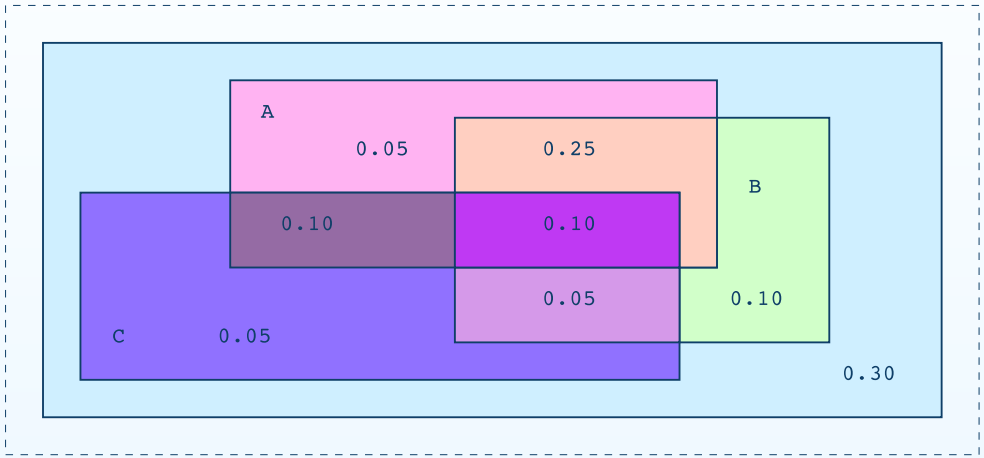
Compute **probability for logic expression**: $P(E) = \sum_{Row \sim E} P(Row)$.

- $P(A) = 0.05 + 0.10 + 0.25 + 0.10 = 0.5$
- $P(A \wedge B) = 0.25 + 0.10 = 0.35$
- $P(\bar{A} \vee C) = 0.30 + 0.05 + 0.10 + 0.05 + 0.05 + 0.25 = 0.8$

Can't we do something more useful?

Using the Joint Distribution (II)

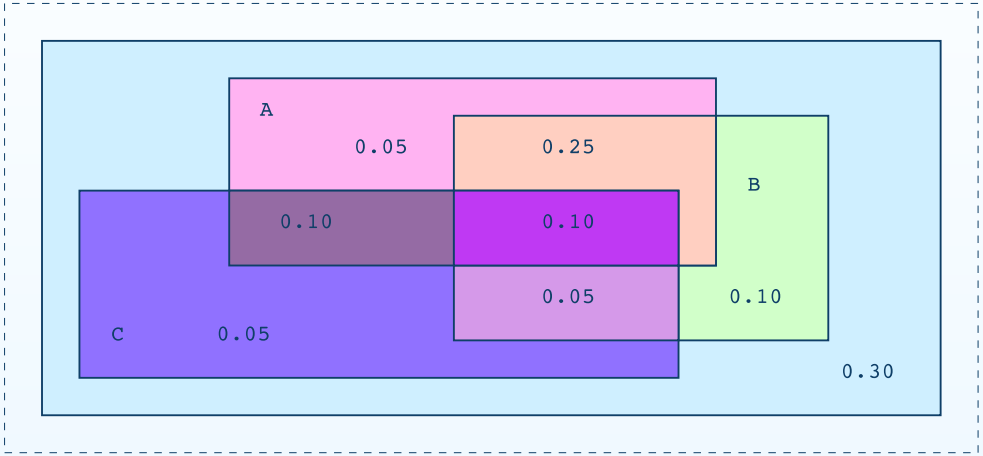
A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10



Use it for **making inference**: $P(E_1|E_2) = \frac{P(E_1 \wedge E_2)}{P(E_2)} = \frac{\sum_{Row \sim E_1 \wedge E_2} P(Row)}{\sum_{Row \sim E_2} P(Row)}$.

Using the Joint Distribution (II)

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10

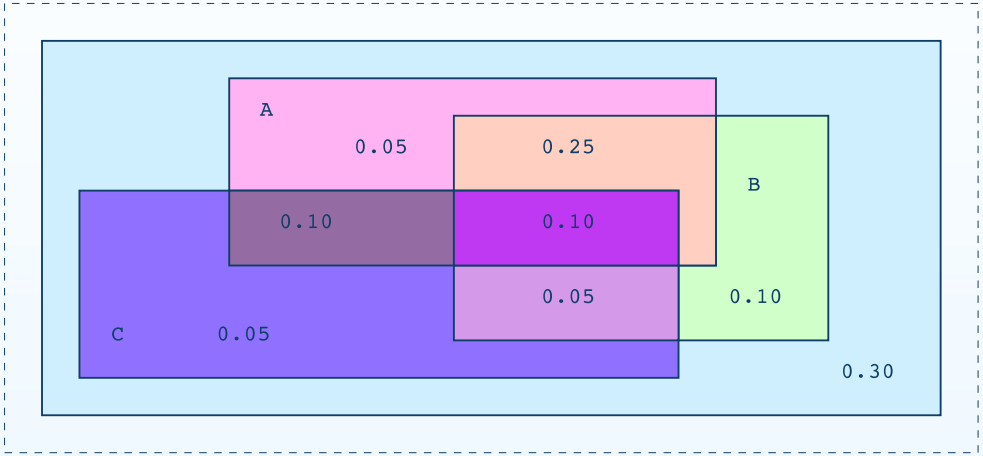


Use it for **making inference**: $P(E_1|E_2) = \frac{P(E_1 \wedge E_2)}{P(E_2)} = \frac{\sum_{Row \sim E_1 \wedge E_2} P(Row)}{\sum_{Row \sim E_2} P(Row)}$.

- $P(A|B) = (0.25 + 0.10)/(0.10 + 0.05 + 0.25 + 0.10) = 0.35/0.50 = 0.70$
- $P(C|A \wedge B) = (0.10)/(0.25 + 0.10) = 0.10/0.35 = 0.285$
- $P(\bar{A}|C) = (0.05 + 0.05)/(0.05 + 0.05 + 0.10 + 0.10) = 0.10/0.30 = 0.333$

Using the Joint Distribution (II)

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10



Use it for **making inference**: $P(E_1|E_2) = \frac{P(E_1 \wedge E_2)}{P(E_2)} = \frac{\sum_{Row \sim E_1 \wedge E_2} P(Row)}{\sum_{Row \sim E_2} P(Row)}$.

- $P(A|B) = (0.25 + 0.10)/(0.10 + 0.05 + 0.25 + 0.10) = 0.35/0.50 = 0.70$
- $P(C|A \wedge B) = (0.10)/(0.25 + 0.10) = 0.10/0.35 = 0.285$
- $P(\bar{A}|C) = (0.05 + 0.05)/(0.05 + 0.05 + 0.10 + 0.10) = 0.10/0.30 = 0.333$

Where do we get the Joint Density from?

The Joint Distribution Estimator

A Density Estimator learns a mapping from a set of attributes to a probability distribution over the attributes space

The Joint Distribution Estimator

A Density Estimator learns a mapping from a set of attributes to a probability distribution over the attributes space

Our Joint Distribution learner is our first example of something called Density Estimation

- Build a Joint Distribution table for your attributes in which the probabilities are unspecified
- The fill in each row with

$$\hat{P}(\text{row}) = \frac{\text{records matching row}}{\text{total number of records}}$$

The Joint Distribution Estimator

A Density Estimator learns a mapping from a set of attributes to a probability distribution over the attributes space

Our Joint Distribution learner is our first example of something called Density Estimation

- Build a Joint Distribution table for your attributes in which the probabilities are unspecified
- The fill in each row with

$$\hat{P}(\text{row}) = \frac{\text{records matching row}}{\text{total number of records}}$$

We will come back to its formal definition at the end of this lecture don't worry, but now . . .

How can we evaluate it?

Evaluating a Density Estimator

We can use **likelihood** for evaluating density estimation:

- Given a record $\mathbf{x}$, a density estimator M tells you how likely it is

$$\hat{P}(\mathbf{x}|M)$$

Evaluating a Density Estimator

We can use **likelihood** for evaluating density estimation:

- Given a record $\mathbf{x}$, a density estimator M tells you how likely it is

$$\hat{P}(\mathbf{x}|M)$$

- Given a dataset with R records, a density estimator can tell you how likely the dataset is under the assumption that all records were **independently** generated from it

$$\hat{P}(\text{dataset}) = \hat{P}(\mathbf{x}_1 \wedge \mathbf{x}_2 \wedge \ldots \wedge \mathbf{x}_R|M) = \prod_{k=1}^{R} \hat{P}(\mathbf{x}_k|M)$$

Evaluating a Density Estimator

We can use **likelihood** for evaluating density estimation:

- Given a record $\mathbf{x}$, a density estimator M tells you how likely it is

$$\hat{P}(\mathbf{x}|M)$$

- Given a dataset with R records, a density estimator can tell you how likely the dataset is under the assumption that all records were **independently** generated from it

$$\hat{P}(\text{dataset}) = \hat{P}(\mathbf{x}_1 \wedge \mathbf{x}_2 \wedge \ldots \wedge \mathbf{x}_R|M) = \prod_{k=1}^{R} \hat{P}(\mathbf{x}_k|M)$$

Since likelihood can get too small we usually use **log-likelihood**:

$$\log \hat{P}(\text{dataset}) = \log \prod_{k=1}^{R} \hat{P}(\mathbf{x}_k|M) = \sum_{k=1}^{R} \log \hat{P}(\mathbf{x}_k|M)$$

Joint Distribution Summary

Now we have a way to learn a Joint Density estimator from data

- Joint Density estimators can do many **good** things
  - Can sort the records by probability, and thus spot weird records (e.g., anomaly/outliers detection)
  - Can do inference: $P(E_1|E_2)$ (e.g., Automatic Doctor, Help Desk)
  - Can be used for Bayes Classifiers (see later)

Joint Distribution Summary

Now we have a way to learn a Joint Density estimator from data

- Joint Density estimators can do many **good** things
  - Can sort the records by probability, and thus spot weird records (e.g., anomaly/outliers detection)
  - Can do inference: $P(E_1|E_2)$ (e.g., Automatic Doctor, Help Desk)
  - Can be used for Bayes Classifiers (see later)

- Joint Density estimators can **badly** overfit!
  - Joint Estimator just mirrors the training data
  - Suppose you see a **new dataset**, its likelihood is going to be:

$$\log \hat{P}(\text{new dataset}|M) = \sum_{k=1}^{R} \log \hat{P}(\mathbf{x}_k|M) = -\infty$$
$$if\ \exists k:\ \hat{P}(\mathbf{x}_k|M) = 0$$

Joint Distribution Summary

Now we have a way to learn a Joint Density estimator from data

- Joint Density estimators can do many **good** things
  - Can sort the records by probability, and thus spot weird records (e.g., anomaly/outliers detection)
  - Can do inference: $P(E_1|E_2)$ (e.g., Automatic Doctor, Help Desk)
  - Can be used for Bayes Classifiers (see later)

- Joint Density estimators can **badly** overfit!
  - Joint Estimator just mirrors the training data
  - Suppose you see a **new dataset**, its likelihood is going to be:

$$\log \hat{P}(\text{new dataset}|M) = \sum_{k=1}^{R} \log \hat{P}(\mathbf{x}_k|M) = -\infty$$
$$if\ \exists k:\ \hat{P}(\mathbf{x}_k|M) = 0$$

We need something which generalizes! $\rightarrow$ **Naïve Density Estimator**

Naïve Density Estimator

The naïve model assumes that each attribute is distributed independently of any of the other attributes.

- Let $\mathbf{x}[i]$ denote the i^{th} field of record $\mathbf{x}$.
- The Naïve Density Estimator says that:

$$\mathbf{x}[i] \perp \{\mathbf{x}[1], \mathbf{x}[2], \ldots, \mathbf{x}[i-1], \mathbf{x}[i+1], \ldots, \mathbf{x}[M]\}$$

Naïve Density Estimator

The naïve model assumes that each attribute is distributed independently of any of the other attributes.

- Let $\mathbf{x}[i]$ denote the i^{th} field of record $\mathbf{x}$.
- The Naïve Density Estimator says that:

$$\mathbf{x}[i] \perp \{\mathbf{x}[1], \mathbf{x}[2], \ldots, \mathbf{x}[i-1], \mathbf{x}[i+1], \ldots, \mathbf{x}[M]\}$$

It is important to recall every time we use a Naïve Density that:

- Attributes are equally important
- Knowing the value of one attribute says nothing about the value of another
- Independence assumption is almost never correct . . .
- . . . this scheme works well in practice!

Naïve Density Estimator: An Example

From a Naïve Distribution you can compute the Joint Distribution:

- Suppose A, B, C, D independently distributed, $P(A \wedge \bar{B} \wedge C \wedge \bar{D}) = ?$

Naïve Density Estimator: An Example

From a Naïve Distribution you can compute the Joint Distribution:

- Suppose A, B, C, D independently distributed, $P(A \wedge \bar{B} \wedge C \wedge \bar{D}) = ?$

$$
\begin{aligned}
P(A \wedge \bar{B} \wedge C \wedge \bar{D}) &= P(A|\bar{B} \wedge C \wedge \bar{D})P(\bar{B} \wedge C \wedge \bar{D}) \\
&= P(A)P(\bar{B} \wedge C \wedge \bar{D}) \\
&= P(A)P(\bar{B}|C \wedge \bar{D})P(C \wedge \bar{D}) \\
&= P(A)P(\bar{B})P(C \wedge \bar{D}) \\
&= P(A)P(\bar{B})P(C|\bar{D})P(\bar{D}) = P(A)P(\bar{B})P(C)P(\bar{D})
\end{aligned}
$$

Naïve Density Estimator: An Example

From a Naïve Distribution you can compute the Joint Distribution:

- Suppose A, B, C, D independently distributed, $P(A \wedge \bar{B} \wedge C \wedge \bar{D}) = ?$

$$\begin{aligned} P(A \wedge \bar{B} \wedge C \wedge \bar{D}) &= P(A|\bar{B} \wedge C \wedge \bar{D})P(\bar{B} \wedge C \wedge \bar{D}) \\ &= P(A)P(\bar{B} \wedge C \wedge \bar{D}) \\ &= P(A)P(\bar{B}|C \wedge \bar{D})P(C \wedge \bar{D}) \\ &= P(A)P(\bar{B})P(C \wedge \bar{D}) \\ &= P(A)P(\bar{B})P(C|\bar{D})P(\bar{D}) = P(A)P(\bar{B})P(C)P(\bar{D}) \end{aligned}$$

Example: suppose to randomly shake a green dice and a red dice

- Dataset 1: A = red value, B = green value
- Dataset 2: A = red value, B = sum of values
- Dataset 3: A = sum of values, B = difference of values

Which of these datasets violates the naïve assumption?

Learning a Naïve Density Estimator

Suppose $\mathbf{x}[1], \mathbf{x}[2], \ldots, \mathbf{x}[M]$ are independently distributed

- Once we have the Naïve Distribution, we can construct any row of the implied Joint Distribution on demand

$$P(\mathbf{x}[1] = u_1, \mathbf{x}[2] = u_2, \ldots, \mathbf{x}[M] = u_M) = \prod_{k=1}^{M} P(\mathbf{x}[k] = u_k)$$

- We can do any inference!

Learning a Naïve Density Estimator

Suppose $\mathbf{x}[1], \mathbf{x}[2], \ldots, \mathbf{x}[M]$ are independently distributed

- Once we have the Naïve Distribution, we can construct any row of the implied Joint Distribution on demand

$$P(\mathbf{x}[1] = u_1, \mathbf{x}[2] = u_2, \ldots, \mathbf{x}[M] = u_M) = \prod_{k=1}^{M} P(\mathbf{x}[k] = u_k)$$

- We can do any inference!

But how do we learn a Naïve Density Estimator?

$$\hat{P}(\mathbf{x}[i] = u) = \frac{\text{number of record for which } \mathbf{x}[i] = u}{\text{total number of records}}$$

Learning a Naïve Density Estimator

Suppose $\mathbf{x}[1], \mathbf{x}[2], \ldots, \mathbf{x}[M]$ are independently distributed

- Once we have the Naïve Distribution, we can construct any row of the implied Joint Distribution on demand

$$P(\mathbf{x}[1] = u_1, \mathbf{x}[2] = u_2, \ldots, \mathbf{x}[M] = u_M) = \prod_{k=1}^{M} P(\mathbf{x}[k] = u_k)$$

- We can do any inference!

But how do we learn a Naïve Density Estimator?

$$\hat{P}(\mathbf{x}[i] = u) = \frac{\text{number of record for which } \mathbf{x}[i] = u}{\text{total number of records}}$$

Please wait a few minute, I'll get the reason for this too!!

Joint Density vs. Naïve Density

What we got so far? Let's try to summarize things up:

- Joint Distribution Estimator
  - Can model anything
  - Given 100 records and more than 6 Boolean attributes will perform poorly
  - Can easily overfit the data

Joint Density vs. Naïve Density

What we got so far? Let's try to summarize things up:

- Joint Distribution Estimator
  - Can model anything
  - Given 100 records and more than 6 Boolean attributes will perform poorly
  - Can easily overfit the data

- Naïve Distribution Estimator
  - Can model only very boring distributions
  - Given 100 records and 10,000 multivalued attributes will be fine
  - Quite robust to overfitting

Joint Density vs. Naïve Density

What we got so far? Let's try to summarize things up:

- Joint Distribution Estimator
  - Can model anything
  - Given 100 records and more than 6 Boolean attributes will perform poorly
  - Can easily overfit the data

- Naïve Distribution Estimator
  - Can model only very boring distributions
  - Given 100 records and 10,000 multivalued attributes will be fine
  - Quite robust to overfitting

So far we have two simple density estimators, in other lectures we'll see vastly more impressive ones (Mixture Models, Bayesian Networks, . . .).

Joint Density vs. Naïve Density

What we got so far? Let's try to summarize things up:

- Joint Distribution Estimator
  - Can model anything
  - Given 100 records and more than 6 Boolean attributes will perform poorly
  - Can easily overfit the data

- Naïve Distribution Estimator
  - Can model only very boring distributions
  - Given 100 records and 10,000 multivalued attributes will be fine
  - Quite robust to overfitting

So far we have two simple density estimators, in other lectures we'll see vastly more impressive ones (Mixture Models, Bayesian Networks, . . .).

But first, why should we care about density estimation?

Joint Density Bayes Classifier

In the case of the Joint Density Bayes Classifier

$$\hat{Y} = \arg\max_{v_i} P(X_1, X_2, \ldots, X_m | Y = v_i) P(Y = v_i)$$

This degenerates to a very simple rule:

$\hat{Y} =$ most common Y among records having $X_1 = u_1, X_2 = u_2, \ldots, X_m = u_m$

Joint Density Bayes Classifier

In the case of the Joint Density Bayes Classifier

$$\hat{Y} = \arg\max_{v_i} P(X_1, X_2, \ldots, X_m | Y = v_i) P(Y = v_i)$$

This degenerates to a very simple rule:

$\hat{Y}$ = most common Y among records having $X_1 = u_1, X_2 = u_2, \ldots, X_m = u_m$

Important Note:

If no records have the exact set of inputs $X_1 = u_1, X_2 = u_2, \ldots, X_m = u_m$, then $P(X_1, X_2, \ldots, X_m | Y = v_i) = 0$ for all values of Y.

In that case we just have to guess Y's value!

Naïve Bayes Classifier

In the case of the Naïve Bayes Classifier

$$\hat{Y} = \arg\max_{v_i} P(X_1, X_2, \ldots, X_m | Y = v_i) P(Y = v_i)$$

Can be simplified in:

$$\hat{Y} = \arg\max_{v_i} P(Y = v_i) \prod_{j=0}^{m} P(X_j = u_j | Y = v_i)$$

Naïve Bayes Classifier

In the case of the Naïve Bayes Classifier

$$\hat{Y} = \arg\max_{v_i} P(X_1, X_2, \ldots, X_m | Y = v_i) P(Y = v_i)$$

Can be simplified in:

$$\hat{Y} = \arg\max_{v_i} P(Y = v_i) \prod_{j=0}^{m} P(X_j = u_j | Y = v_i)$$

Technical Hint:

When we have 10,000 input attributes the product will underflow in floating point math, so we should use logs:

$$\hat{Y} = \arg\max_{v_i} \left(\log P(Y = v_i) + \sum_{j=0}^{m} \log P(X_j = u_j | Y = v_i) \right)$$

The Example: “Is this a nice day to play golf?”

Outlook	Temp	Humid.	Windy	Play
sunny	85	85	false	No
sunny	80	90	true	No
overcast	83	78	false	Yes
rain	70	96	false	Yes
rain	68	80	false	Yes
rain	65	70	true	No
overcast	64	65	true	Yes
sunny	72	95	false	No
sunny	69	70	false	Yes
rain	75	80	false	Yes
sunny	75	70	true	Yes
overcast	72	90	true	Yes
overcast	81	75	false	Yes
rain	71	80	true	No

The Example: “Is this a nice day to play golf?”

Outlook	Temp	Humid.	Windy	Play
sunny	85	85	false	No
sunny	80	90	true	No
overcast	83	78	false	Yes
rain	70	96	false	Yes
rain	68	80	false	Yes
rain	65	70	true	No
overcast	64	65	true	Yes
sunny	72	95	false	No
sunny	69	70	false	Yes
rain	75	80	false	Yes
sunny	75	70	true	Yes
overcast	72	90	true	Yes
overcast	81	75	false	Yes
rain	71	80	true	No

Attribute	Value	Play	Don’t
Outlook	sunny	2 (2/9)	3 (3/5)
	overcast	4 (4/9)	0 (0)
	rain	3 (3/9)	2 (2/5)
Temp.	hight	2 (2/9)	2 (2/5)
	mild	4 (4/9)	2 (2/5)
	cool	3 (3/9)	1 (1/5)
Humid.	high	3 (3/9)	4 (4/5)
	normal	6 (6/9)	1 (1/5)
Windy	true	3 (3/9)	3 (3/5)
	false	6 (6/9)	2 (2/5)

The Example: “Is this a nice day to play golf?”

Outlook	Temp	Humid.	Windy	Play
sunny	85	85	false	No
sunny	80	90	true	No
overcast	83	78	false	Yes
rain	70	96	false	Yes
rain	68	80	false	Yes
rain	65	70	true	No
overcast	64	65	true	Yes
sunny	72	95	false	No
sunny	69	70	false	Yes
rain	75	80	false	Yes
sunny	75	70	true	Yes
overcast	72	90	true	Yes
overcast	81	75	false	Yes
rain	71	80	true	No

Attribute	Value	Play	Don’t
Outlook	sunny	2 (2/9)	3 (3/5)
	overcast	4 (4/9)	0 (0)
	rain	3 (3/9)	2 (2/5)
Temp.	hight	2 (2/9)	2 (2/5)
	mild	4 (4/9)	2 (2/5)
	cool	3 (3/9)	1 (1/5)
Humid.	high	3 (3/9)	4 (4/5)
	normal	6 (6/9)	1 (1/5)
Windy	true	3 (3/9)	3 (3/5)
	false	6 (6/9)	2 (2/5)

- Play = 9 (9/14)
- Don’t Play = 5 (5/14)

The Example: “A brand new day”

You wake up and gain some new *Evidence* about the day:

Outlook	Temp	Humid.	Windy	Play
`sunny`	`cool`	`high`	`true`	???

The Example: “A brand new day”

You wake up and gain some new *Evidence* about the day:

Outlook	Temp	Humid.	Windy	Play
sunny	cool	high	true	???

Should you play golf or not? Let’s ask it to our Naïve Bayes Classifier!

$$
\begin{aligned}
P(Evidence|yes) &= P(sunny|yes)P(cool|yes)P(high|yes)P(true|yes) \\
&= 2/9 \cdot 3/9 \cdot 3/9 \cdot 3/9 = 0.00823 \\
P(Evidence|no) &= P(sunny|no)P(cool|no)P(high|no)P(true|no) \\
&= 3/5 \cdot 1/5 \cdot 4/5 \cdot 3/5 = 0.0576 \\
P(Play = yes|Evidence) &= \frac{P(Evidence|yes)P(yes)}{\sum_{class} P(Evidence|class)P(class)} = 0.205 \\
P(Play = no|Evidence) &= \frac{P(Evidence|no)P(no)}{\sum_{class} P(Evidence|class)P(class)} = 0.795
\end{aligned}
$$

Missing Values (still “A brand new day”)

You wake up and gain some new “partial” *Evidence* about the day:

Outlook	Temp	Humid.	Windy	Play
???	`cool`	`high`	`true`	???

Missing Values (still “A brand new day”)

You wake up and gain some new “partial” *Evidence* about the day:

Outlook	Temp	Humid.	Windy	Play
???	cool	high	true	???

$$
\begin{aligned}
P(Evidence|yes) &= \sum_o P(overlook|yes)P(cool|yes)P(high|yes)P(true|yes) \\
&= 3/9 \cdot 3/9 \cdot 3/9 = 0.037 \\
P(Evidence|no) &= \sum_o P(overlook|yes)P(cool|no)P(high|no)P(true|no) \\
&= 1/5 \cdot 4/5 \cdot 3/5 = 0.096 \\
P(yes|Evidence) &= \frac{P(Evidence|yes)P(yes)}{\sum_{class} P(Evidence|class)P(class)} = 0.41 \\
P(no|Evidence) &= \frac{P(Evidence|no)P(no)}{\sum_{class} P(Evidence|class)P(class)} = 0.59
\end{aligned}
$$

The “Zero Frequency” Problem

What if an attribute value doesn’t occur with every class value (e.g. `Outlook = overcast` for class `no`)?

- Probability will be zero!
- No matter how likely the other values are, also a-posteriori probability will be zero!

$$P(Outlook = overcast|no) = 0 \quad \rightarrow \quad P(no|Evidence) = 0$$

The “Zero Frequency” Problem

What if an attribute value doesn’t occur with every class value (e.g. `Outlook = overcast` for class `no`)?

- Probability will be zero!
- No matter how likely the other values are, also a-posteriori probability will be zero!

$$P(Outlook = overcast|no) = 0 \quad \rightarrow \quad P(no|Evidence) = 0$$

The solution is related to something called “smoothing prior”:

- Add 1 to the count for every attribute value-class combination

The “Zero Frequency” Problem

What if an attribute value doesn’t occur with every class value (e.g. `Outlook = overcast` for class `no`)?

- Probability will be zero!
- No matter how likely the other values are, also a-posteriori probability will be zero!

$$P(Outlook = overcast|no) = 0 \quad \rightarrow \quad P(no|Evidence) = 0$$

The solution is related to something called “smoothing prior”:

- Add 1 to the count for every attribute value-class combination

This simple approach (Laplace estimator) solves the problem and stabilize probability estimates!

The “Zero Frequency” Problem

What if an attribute value doesn’t occur with every class value (e.g. `Outlook = overcast` for class `no`)?

- Probability will be zero!
- No matter how likely the other values are, also a-posteriori probability will be zero!

$$P(Outlook = overcast|no) = 0 \quad \rightarrow \quad P(no|Evidence) = 0$$

The solution is related to something called “smoothing prior”:

- Add 1 to the count for every attribute value-class combination

This simple approach (Laplace estimator) solves the problem and stabilize probability estimates!

We can do also fancy things!

M-Estimate Probability

We can use **M-estimate probability** to estimate $P(Attribute|Class)$:

$$P(A|C) = \frac{n_A + mp}{n + m}$$

- n_A number of examples with class C and attribute A
- n number of examples with class C
- $p = 1/k$ whre k are the possible values of attribute A
- m is a costant value

M-Estimate Probability

We can use **M-estimate probability** to estimate $P(Attribute|Class)$:

$$P(A|C) = \frac{n_A + mp}{n + m}$$

- n_A number of examples with class C and attribute A
- n number of examples with class C
- $p = 1/k$ whre k are the possible values of attribute A
- m is a costant value

Example: For the `Outlook` attribute we get:

$$\texttt{sunny} = \frac{2+m/3}{9+m}, \texttt{overcast} = \frac{4+m/3}{9+m}, \texttt{rain} = \frac{3+m/3}{9+m}.$$

M-Estimate Probability

We can use **M-estimate probability** to estimate $P(Attribute|Class)$:

$$P(A|C) = \frac{n_A + mp}{n + m}$$

- n_A number of examples with class C and attribute A
- n number of examples with class C
- $p = 1/k$ whre k are the possible values of attribute A
- m is a costant value

Example: For the `Outlook` attribute we get:

$$\texttt{sunny} = \frac{2+m/3}{9+m}, \texttt{overcast} = \frac{4+m/3}{9+m}, \texttt{rain} = \frac{3+m/3}{9+m}.$$

We can also use weights $p_1, p_2, \ldots, p_k$ summing up to 1!

$$\texttt{sunny} = \frac{2+m*p_1}{9+m}, \texttt{overcast} = \frac{4+m*p_2}{9+m}, \texttt{rain} = \frac{3+m*p_3}{9+m}.$$