Lecture 15

Applications

Marco Chiarandini

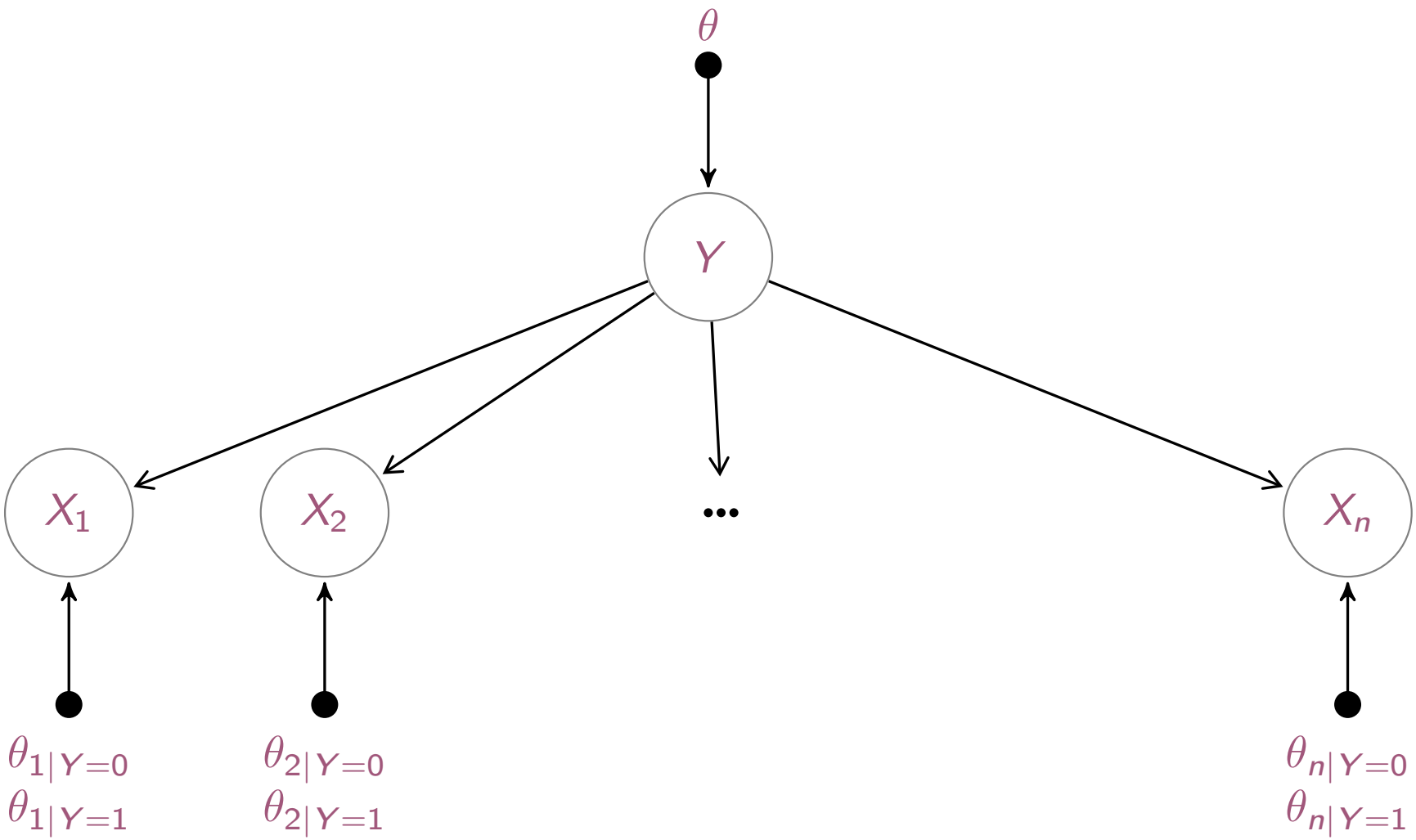
Department of Mathematics & Computer Science
University of Southern Denmark

Some slides are by Dan Klein at Berkeley

Course Overview

- ✔ Introduction
  - ✔ Artificial Intelligence
  - ✔ Intelligent Agents
- ✔ Search
  - ✔ Uninformed Search
  - ✔ Heuristic Search
- ✔ Uncertain knowledge and Reasoning
  - ✔ Probability and Bayesian approach
  - ✔ Bayesian Networks
  - ✔ Hidden Markov Chains
  - ✔ Kalman Filters
- ✔ Learning
  - ✔ Supervised
    Decision Trees, Neural Networks
    Learning Bayesian Networks
  - ✔ Unsupervised
    EM Algorithm
- ✔ Reinforcement Learning
- ▶ Games and Adversarial Search
  - ▶ Minimax search and Alpha-beta pruning
  - ▶ Multiagent search
- ▶ Knowledge representation and Reasoning
  - ▶ Propositional logic
  - ▶ First order logic
  - ▶ Inference
  - ▶ Plannning

Naive Bayes Network



$1 + 2n$ parameters

$$\Pr(Y, X_1, X_2, \ldots, X_n) = \Pr(X_1, X_2, \ldots, X_n \mid Y) \Pr(Y) \qquad \text{(product rule)}$$

$$\Pr(X_1, X_2, \ldots, X_n \mid Y) = \prod_i \Pr(X_i \mid Y) \qquad \text{(conditional indenpendence)}$$

Estimation/learning: maximize joint likelihood

$$\max \mathcal{L} = \max \prod_j \Pr(Y_j) \prod_i \Pr(X_{ij} \mid Y_j)$$

Prediction:

$$\Pr(Y \mid X_1, X_2, \ldots, X_n) = \frac{\Pr(Y, X_1, X_2, \ldots, X_n)}{\Pr(X_1, X_2, \ldots, X_n)} = \alpha \Pr(Y, X_1, X_2, \ldots, X_n)$$

Maximum a posteriori probability prediction:

$$y = \operatorname{argmax}_Y \Pr(Y, X_1, X_2, \ldots, X_n)$$

$$\theta = \frac{\sum_j I(Y_j = 1)}{N}$$

$$\theta_{i|Y=1} = \frac{\sum_j I(Y_j = 1 \wedge X_{ij} = 1)}{\sum_j I(Y_j = 1)}$$

$$\theta_{i|Y=0} = \frac{\sum_j I(Y_j = 0 \wedge X_{ij} = 1)}{\sum_j I(Y_j = 0)}$$

- Naïve Bayes spam filter
- Data:
  - Collection of emails, labeled spam or ham
  - Note: someone has to hand label all this data!
  - Split into training, held-out, test sets
- Classifiers
  - Learn on the training set
  - (Tune it on a held-out set)
  - Test it on new emails

Dear Sir.

First, I must solicit your confidence in this transaction, this is by virture of its nature as being utterly confidencial and top secret. …

TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.

99 MILLION EMAIL ADDRESSES FOR ONLY $99

Ok, Iknow this is blatantly OT but I'm beginning to go insane. Had an old Dell Dimension XPS sitting in the corner and decided to put it to use, I know it was working pre being stuck in the corner, but when I plugged it in, hit the power nothing happened.

$P(C)$

```
ham : 0.66
spam: 0.33
```

$P(W|\text{spam})$

```
the :  0.0156
to  :  0.0153
and :  0.0115
of  :  0.0095
you :  0.0093
a   :  0.0086
with:  0.0080
from:  0.0075
...
```

$P(W|\text{ham})$

```
the :  0.0210
to  :  0.0133
of  :  0.0119
2002:  0.0110
with:  0.0108
from:  0.0107
and :  0.0105
a   :  0.0100
...
```

Where do these tables come from?

$$P(C, W_1, W_2 \ldots, W_n) = P(C)\Pi_i P(W_i|C)$$

$$P(C, W_1, W_2 \ldots, W_n) = \log P(C) \sum_i \log P(W_i|C)$$

Word	P(w|spam)	P(w|ham)	Tot Spam	Tot Ham
(prior)	0.33333	0.66666	-1.1	-0.4
Gary	0.00002	0.00021	-11.8	-8.9
would	0.00069	0.00084	-19.1	-16.0
you	0.00881	0.00304	-23.8	-21.8
like	0.00086	0.00083	-30.9	-28.9
to	0.01517	0.01339	-35.1	-33.2
lose	0.00008	0.00002	-44.5	-44.0
weight	0.00016	0.00002	-53.3	-55.0
while	0.00027	0.00027	-61.5	-63.2
you	0.00881	0.00304	-66.2	-69.0
sleep	0.00006	0.00001	-76.0	-80.5
P(spam | w) = 98.9				

Digit Recognition

- Input: images / pixel grids
- Output: a digit 0-9
- Setup:
  - Get a large collection of example images, each labeled with a digit
  - Note: someone has to hand label all this data!
  - Want to learn to predict labels of new, future digit images
- Features: The attributes used to make the digit decision
  - Pixels: (6,8)=ON
  - Shape Patterns: NumComponents, AspectRatio, NumLoops, ...


0


1


2

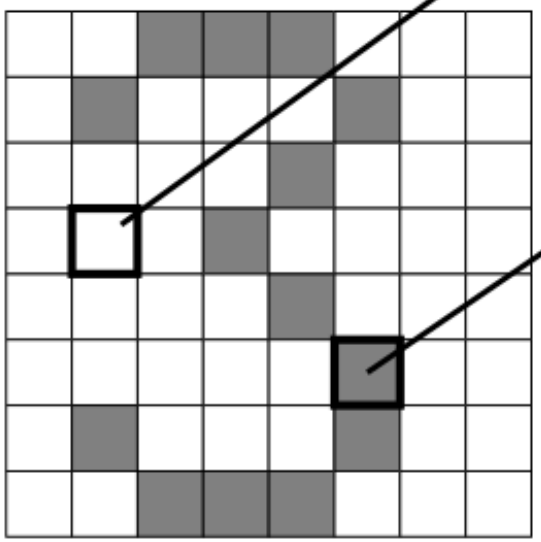

1


??

CPT

$P(Y)$

1	0.1
2	0.1
3	0.1
4	0.1
5	0.1
6	0.1
7	0.1
8	0.1
9	0.1
0	0.1



$P(F_{3,1} = on|Y)$

1	0.01
2	0.05
3	0.05
4	0.30
5	0.80
6	0.90
7	0.05
8	0.60
9	0.50
0	0.80

$P(F_{5,5} = on|Y)$

1	0.05
2	0.01
3	0.90
4	0.80
5	0.90
6	0.90
7	0.25
8	0.85
9	0.60
0	0.80

Overfitting

$P(\text{features}, C = 2)$

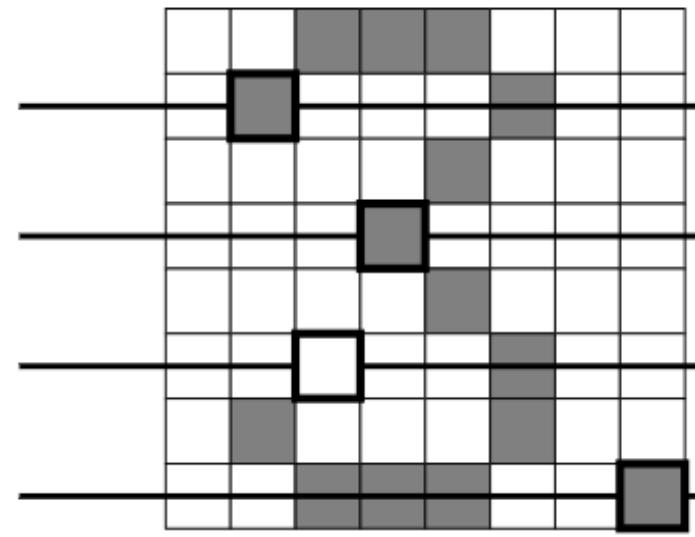
$P(C = 2) = 0.1$

$P(\text{on}|C = 2) = 0.8$

$P(\text{on}|C = 2) = 0.1$

$P(\text{off}|C = 2) = 0.1$

$P(\text{on}|C = 2) = 0.01$



$P(\text{features}, C = 3)$

$P(C = 3) = 0.1$

$P(\text{on}|C = 3) = 0.8$

$P(\text{on}|C = 3) = 0.9$

$P(\text{off}|C = 3) = 0.7$

$P(\text{on}|C = 3) = 0.0$

Laplace Smoothing

Assume k samples for each value of the joint distribution

$$\theta = \frac{\sum_j I(Y_j = 1) + k}{N + k|Y|}$$

$$\theta_{i|Y=1} = \frac{\sum_j I(Y_j = 1 \wedge X_{ij} = 1) + k}{\sum_j I(Y_j = 1) + k|X_i|}$$

$$\theta_{i|Y=0} = \frac{\sum_j I(Y_j = 0 \wedge X_{ij} = 1) + k}{\sum_j I(Y_j = 1) + k|X_i|}$$

Tuning

- Now we've got two kinds of unknowns
  - Parameters: the probabilities P(X|Y), P(Y)
  - Hyperparameters, like the amount of smoothing to do: k
- Where to learn?
  - Learn parameters from training data
  - Tune hyperparameters on different data

For each value of the hyperparameters, train and test on the held-out data
Choose the best value and do a final test on the test data