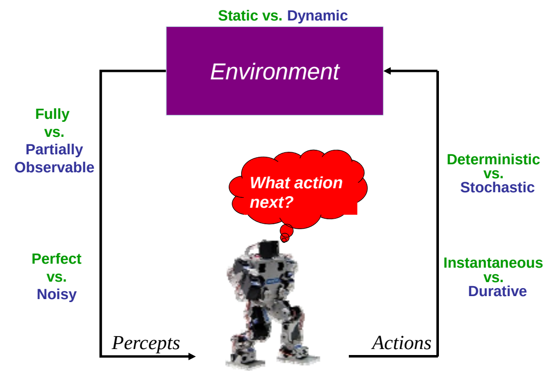
CSE 573: Artificial Intelligence
Spring 2012

Learning Bayesian Networks

Dan Weld

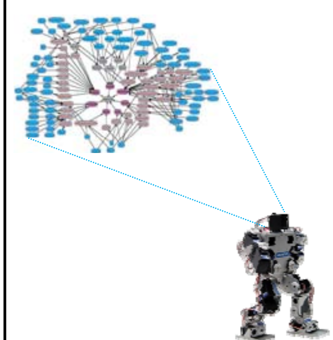
Slides adapted from Carlos Guestrin, Krzysztof Gajos, Dan Klein, Stuart Russell, Andrew Moore & Luke Zettlemoyer



Algorithms

Blind search
Heuristic search
Mini-max & Expectimax
MDPs (& POMDPS)
Reinforcement learning
State estimation

Knowledge Representation



HMMs
Bayesian networks
First-order logic
Description logic
Constraint networks
Markov logic networks
…

Learning

What is Machine Learning ?

Machine Learning

Study of algorithms that
- improve their <u>performance</u>
- at some <u>task</u>
- with <u>experience</u>

Exponential Growth in Data

Supremacy of Machine Learning

- Machine learning is preferred approach to
  - Speech recognition, Natural language processing
  - Web search – result ranking
  - Computer vision
  - Medical outcomes analysis
  - Robot control
  - Computational biology
  - Sensor networks
  - …
- This trend is accelerating
  - Improved machine learning algorithms
  - Improved data capture, networking, faster computers
  - Software too complex to write by hand
  - New sensors / IO devices
  - Demand for self-customization to user, environment

Space of ML Problems

Type of Supervision
(eg, Experience, Feedback)

What is Being Learned?

	Labeled Examples	Reward	Nothing
Discrete Function	Classification		Clustering
Continuous Function	Regression		
Policy	Apprenticeship Learning	Reinforcement Learning	

Classification

from data to discrete classes

Spam filtering

data

prediction

Weather prediction

Object detection

(Prof. H. Schneiderman)

Example training images
for each orientation

The classification pipeline

Machine Learning

Supervised Learning
Unsupervised Learning
Reinforcement Learning

Parametric
Non-parametric

Nearest neighbor
Kernel density estimati
Support vector machin

Machine Learning

Supervised Learning
Unsupervised Learning
Reinforcement Learning

Parametric
Non-parametric

Y Continuous
Y Discrete

Gaussians
Learned in closed form

Linear Functions
1. Learned in closed form
2. Using gradient descent

Decision Trees
Greedy search; pruning

Probability of class | features
1. Learn P(Y), P(X|Y); apply Bayes
2. Learn P(Y|X) w/ gradient descent

Non-probabilistic Linear Classifier
Learn w/ gradient descent

Regression

predicting a numeric value

Stock market

Weather prediction revisted

Clustering

discovering structure in data

Machine Learning

Supervised Learning

Unsupervised Learning

Reinforcement Learning

Parametric

Non-parametric

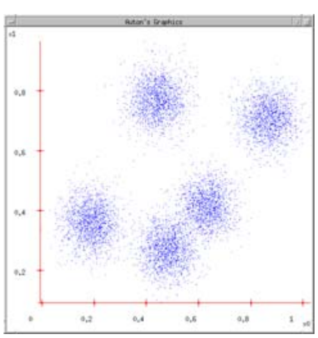
Agglomerative Clustering

K-means

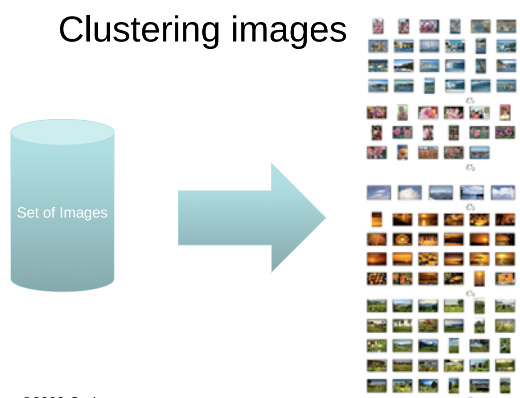
Expectation Maximization (EM)

Principle Component Analysis (PCA)

Clustering Data: Group similar things

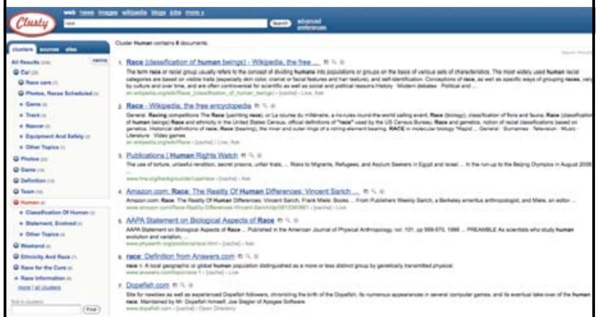


Clustering images



[Goldberger et al.]

Clustering web search results



In Summary

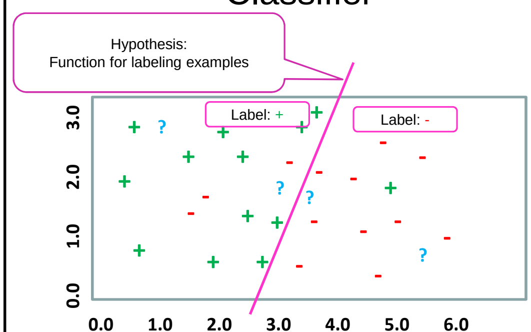
Type of Supervision
(eg, Experience, Feedback)

What is Being Learned?

	Labeled Examples	Reward	Nothing
Discrete Function	Classification		Clustering
Continuous Function	Regression		
Policy	Apprenticeship Learning	Reinforcement Learning	

Key Concepts

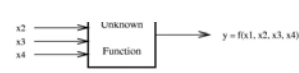
Classifier



Generalization

- Hypotheses must ***generalize*** to correctly classify instances not in the training data.
- Simply memorizing training examples is a consistent hypothesis ***that does not generalize***.

A Learning Problem



Example	x_1	x_2	x_3	x_4	y
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

Hypothesis Spaces

input features. We can't figure out which one is correct until we've seen every possible input-output pair. After 7 examples, we still have 2^9 possibilities.

x_1	x_2	x_3	x_4	y
0	0	0	0	?
0	0	0	1	?
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	?
1	0	0	0	?
1	0	0	1	1
1	0	1	0	?
1	0	1	1	?
1	1	0	0	0
1	1	0	1	?
1	1	1	0	?
1	1	1	1	?

Why is Learning Possible?

Experience alone never justifies any conclusion about any unseen instance.

Learning occurs when PREJUDICE meets DATA!

Learning a "Frobnitz"

Frobnitz

Not a Frobnitz

Bias

- The nice word for prejudice is "bias".
  - Different from "Bias" in statistics
- What kind of hypotheses will you ***consider***?
  - What is allowable ***range*** of functions you use when approximating?
- What kind of hypotheses do you ***prefer***?

Some Typical Biases

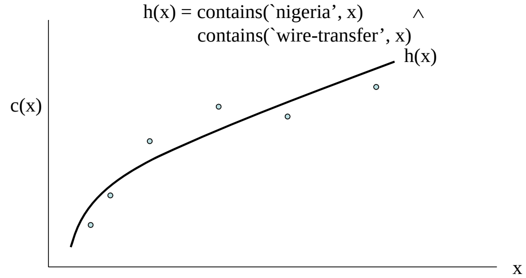
- Occam's razor
  "It is needless to do more when less will suffice"
  – William of Occam,
  died 1349 of the Black plague
- MDL – Minimum description length
- Concepts can be approximated by
- ... **conjunctions** of predicates
  ... by **linear** functions
  ... by **short** decision trees

Frobnitz?

ML = Function Approximation

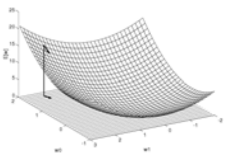
May not be any perfect fit
Classification ~ discrete functions

h(x) = contains(`nigeria', x) ^
contains(`wire-transfer', x)



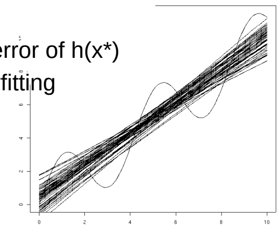
Learning as Optimization

- Preference Bias
- Loss Function
  - Minimize ***loss*** over training data (test data)
  - Loss(h,data) = error(h, data) + complexity(h)
  - Error + regularization
- Methods
  - Closed form
  - Greedy search
  - Gradient ascent



Bias / Variance Tradeoff

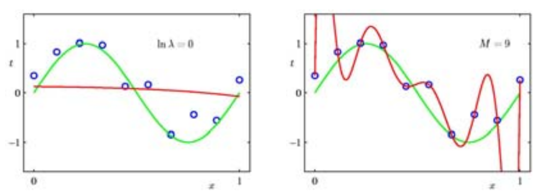
- Variance: **E[(h(x*) – h(x*))^2]**
  How much h(x*) varies between training sets
  Reducing variance risks underfitting
- Bias: **[h(x*) – f(x*)]**
  Describes the ***average*** error of h(x*)
  Reducing bias risks overfitting



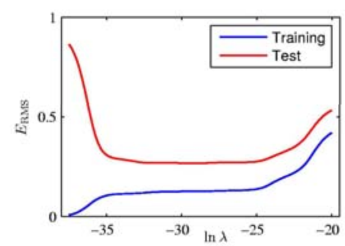
Note: **inductive bias** *vs* **estimator bias**

Slide from T Dietterich

Regularization

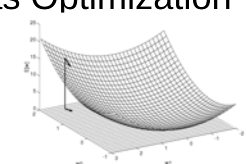


Regularization: E_{RMS} vs $\ln \lambda$



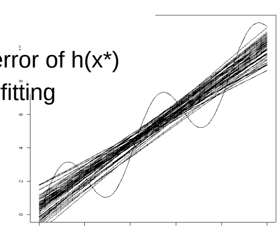
Learning as Optimization

- Methods
  - Closed form
  - Greedy search
  - Gradient ascent
- Loss Function
  - Minimize ***loss*** over training data (test data)
  - Loss(h,data) = error(h, data) + complexity(h)
  - Error + regularization



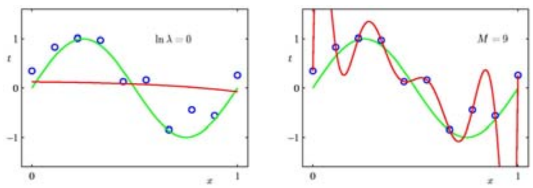
Bia / Variance Tradeoff

- Variance: **E[(h(x*) – h(x*))^2]**
  How much h(x*) varies between training sets
  Reducing variance risks underfitting
- Bias: **[h(x*) – f(x*)]**
  Describes the ***average*** error of h(x*)
  Reducing bias risks overfitting

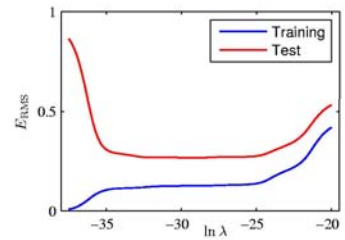


Slide from T Dietterich

Regularization



Regularization: E_{RMS} vs $\ln \lambda$



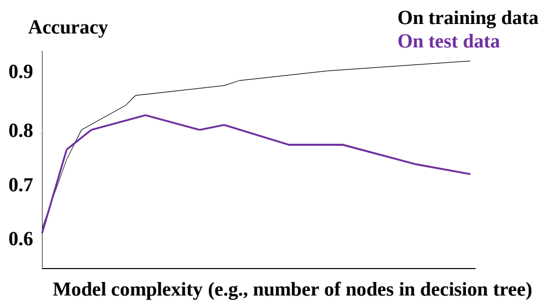
Overfitting

- Hypothesis H is *overfit* when ∃ H' and
  - H has ***smaller*** error on training examples, but
  - H has ***bigger*** error on test examples

Overfitting

- Hypothesis H is *overfit* when ∃ H' and
  - H has ***smaller*** error on training examples, but
  - H has ***bigger*** error on test examples
- Causes of overfitting
  - Training set is too small
  - Large number of features
- Big problem in machine learning
  - Solutions: bias, regularization
  - Validation set

Overfitting

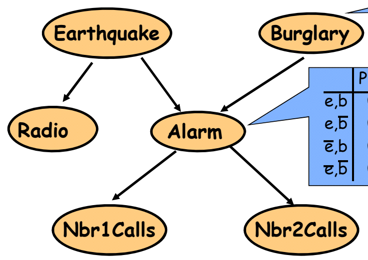


 49

Learning Bayes Nets

- Learning Parameters for a Bayesian Network
  - Fully observable
    - Maximum Likelihood (ML)
    - Maximum A Posteriori (MAP)
    - Bayesian
  - Hidden variables (EM algorithm)
- Learning Structure of Bayesian Networks

What's in a Bayes Net?

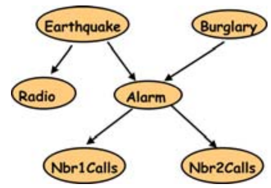


Pr(B=t)	Pr(B=f)
0.05	0.95

	Pr(A|E,B)
e,b	0.9 (0.1)
e,$\bar{b}$	0.2 (0.8)
$\bar{e}$,b	0.85 (0.15)
$\bar{e}$,$\bar{b}$	0.01 (0.99)

 51

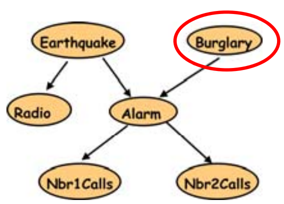
Parameter Estimation and Bayesian Networks



E	B	R	A	J	M
T	F	T	T	F	T
F	F	F	F	F	T
F	T	F	T	T	T
F	F	F	T	T	T
F	T	F	F	F	F
...					

We have:
- Bayes Net structure and observations
- We need: Bayes Net parameters

Parameter Estimation and Bayesian Networks



B
F
F
T
F
T

P(B) = ? = 0.4

P(¬B) = 1- P(B) = 0.6

Parameter Estimation and Bayesian Networks

E	B
T	F
F	F
F	T
F	F
F	T
...	

A
T
F
T
T
F

P(A|E,B) = ?
P(A|E,¬B) = ?
P(A|¬E,B) = ?
P(A|¬E,¬B) = ?

Parameter Estimation and Bayesian Networks

Coin Flip

C_1: $P(H|C_1) = 0.1$ C_2: $P(H|C_2) = 0.5$ C_3: $P(H|C_3) = 0.9$

Which coin will I use?

$P(C_1) = 1/3$ $P(C_2) = 1/3$ $P(C_3) = 1/3$

Prior: Probability of a hypothesis before we make any observations

Coin Flip

C_1 C_2 C_3

$P(H|C_1) = 0.1$ $P(H|C_2) = 0.5$ $P(H|C_3) = 0.9$

Which coin will I use?

$P(C_1) = 1/3$ $P(C_2) = 1/3$ $P(C_3) = 1/3$

Uniform Prior: All hypothesis are equally likely before we make any observations

Experiment 1: Heads

Which coin <u>did</u> I use?

$P(C_1|H) = ?$ $P(C_2|H) = ?$ $P(C_3|H) = ?$

$$P(C_1|H) = \frac{P(H|C_1)P(C_1)}{P(H)} \qquad P(H) = \sum_{i=1}^{3} P(H|C_i)P(C_i)$$

C_1 C_2 C_3

$P(H|C_1)=0.1$ $P(H|C_2) = 0.5$ $P(H|C_3) = 0.9$

$P(C_1)=1/3$ $P(C_2) = 1/3$ $P(C_3) = 1/3$

Experiment 1: Heads

Which coin <u>did</u> I use?

$P(C_1|H) = 0.066$ $P(C_2|H) = 0.333$ $P(C_3|H) = 0.6$

Posterior: Probability of a hypothesis given data

C_1 C_2 C_3

$P(H|C_1) = 0.1$ $P(H|C_2) = 0.5$ $P(H|C_3) = 0.9$

$P(C_1) = 1/3$ $P(C_2) = 1/3$ $P(C_3) = 1/3$

Terminology

- **Prior**:
  - Probability of a hypothesis before we see any data
- **Uniform Prior**:
  - A prior that makes all hypothesis equally likely
- **Posterior**:
  - Probability of a hypothesis after we saw some data
- **Likelihood**:
  - Probability of data given hypothesis

Experiment 2: Tails

Now, Which coin did I use?

$P(C_1|HT) = ?$ $P(C_2|HT) = ?$ $P(C_3|HT) = ?$

$$P(C_1|HT) = \alpha P(HT|C_1)P(C_1) = \alpha P(H|C_1)P(T|C_1)P(C_1)$$

C_1 C_2 C_3

$P(H|C_1) = 0.1$ $P(H|C_2) = 0.5$ $P(H|C_3) = 0.9$

$P(C_1) = 1/3$ $P(C_2) = 1/3$ $P(C_3) = 1/3$

Experiment 2: Tails

Now, Which coin did I use?

$P(C_1|HT) = 0.21$ $P(C_2|HT) = 0.58$ $P(C_3|HT) = 0.21$

$$P(C_1|HT) = \alpha P(HT|C_1)P(C_1) = \alpha P(H|C_1)P(T|C_1)P(C_1)$$

C_1 C_2 C_3

$P(H|C_1) = 0.1$ $P(H|C_2) = 0.5$ $P(H|C_3) = 0.9$

$P(C_1) = 1/3$ $P(C_2) = 1/3$ $P(C_3) = 1/3$

Experiment 2: Tails

Which coin did I use?

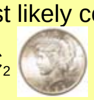
$P(C_1|HT) = 0.21$ $P(C_2|HT) = 0.58$ $P(C_3|HT) = 0.21$

C_1	C_2	C_3
$P(H|C_1) = 0.1$	$P(H|C_2) = 0.5$	$P(H|C_3) = 0.9$
$P(C_1) = 1/3$	$P(C_2) = 1/3$	$P(C_3) = 1/3$

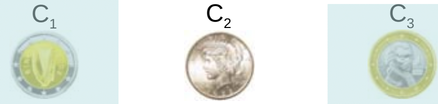
Your Estimate?

What is the probability of heads after two experiments?

Most likely coin: C_2



Best estimate for P(H): $P(H|C_2) = 0.5$

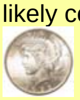


C_1	C_2	C_3
$P(H|C_1) = 0.1$	$P(H|C_2) = 0.5$	$P(H|C_3) = 0.9$
$P(C_1) = 1/3$	$P(C_2) = 1/3$	$P(C_3) = 1/3$

Your Estimate?

Maximum Likelihood Estimate: The best hypothesis that fits observed data assuming uniform prior

Most likely coin: C_2



Best estimate for P(H): $P(H|C_2) = 0.5$

C_2

$P(H|C_2) = 0.5$
$P(C_2) = 1/3$

Using Prior Knowledge

- Should we always use a ***Uniform Prior*** ?
- Background knowledge:
  - Heads => we have to buy Dan chocolate
  - Dan ***likes*** chocolate…
  - => Dan is more likely to use a coin biased in his favor

C_1	C_2	C_3
$P(H|C_1) = 0.1$	$P(H|C_2) = 0.5$	$P(H|C_3) = 0.9$

Using Prior Knowledge

We can encode it in the prior:

$P(C_1) = 0.05$ $P(C_2) = 0.25$ $P(C_3) = 0.70$

C_1	C_2	C_3
$P(H|C_1) = 0.1$	$P(H|C_2) = 0.5$	$P(H|C_3) = 0.9$

Experiment 1: Heads

Which coin did I use?

$P(C_1|H) = ?$ $P(C_2|H) = ?$ $P(C_3|H) = ?$

$$P(C_1|H) = \alpha P(H|C_1)P(C_1)$$

C_1	C_2	C_3
$P(H|C_1) = 0.1$	$P(H|C_2) = 0.5$	$P(H|C_3) = 0.9$
$P(C_1) = 0.05$	$P(C_2) = 0.25$	$P(C_3) = 0.70$

Experiment 1: Heads

Which coin did I use?

$P(C_1|H) = 0.006$ $P(C_2|H) = 0.165$ $P(C_3|H) = 0.829$

Compare with ML posterior after Exp 1:

$P(C_1|H) = 0.066$ $P(C_2|H) = 0.333$ $P(C_3|H) = 0.600$

C_1	C_2	C_3
$P(H|C_1) = 0.1$	$P(H|C_2) = 0.5$	$P(H|C_3) = 0.9$
$P(C_1) = 0.05$	$P(C_2) = 0.25$	$P(C_3) = 0.70$

Experiment 2: Tails

Which coin did I use?

$P(C_1|HT) = ?$ $P(C_2|HT) = ?$ $P(C_3|HT) = ?$

$$P(C_1|HT) = \alpha P(HT|C_1)P(C_1) = \alpha P(H|C_1)P(T|C_1)P(C_1)$$

C_1	C_2	C_3
$P(H|C_1) = 0.1$	$P(H|C_2) = 0.5$	$P(H|C_3) = 0.9$
$P(C_1) = 0.05$	$P(C_2) = 0.25$	$P(C_3) = 0.70$

Experiment 2: Tails

Which coin did I use?

$P(C_1|HT) = 0.035$ $P(C_2|HT) = 0.481$ $P(C_3|HT) = 0.485$

$$P(C_1|HT) = \alpha P(HT|C_1)P(C_1) = \alpha P(H|C_1)P(T|C_1)P(C_1)$$

C_1	C_2	C_3
$P(H|C_1) = 0.1$	$P(H|C_2) = 0.5$	$P(H|C_3) = 0.9$
$P(C_1) = 0.05$	$P(C_2) = 0.25$	$P(C_3) = 0.70$

Experiment 2: Tails

Which coin did I use?

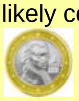
$P(C_1|HT) = 0.035$ $P(C_2|HT)=0.481$ $P(C_3|HT) = 0.485$

C_1	C_2	C_3
$P(H|C_1) = 0.1$	$P(H|C_2) = 0.5$	$P(H|C_3) = 0.9$
$P(C_1) = 0.05$	$P(C_2) = 0.25$	$P(C_3) = 0.70$

Your Estimate?

What is the probability of heads after two experiments?

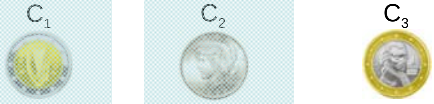
Most likely coin: C_3



Best estimate for P(H)

$P(H|C_3) = 0.9$

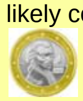
C_1	C_2	C_3
$P(H|C_1) = 0.1$	$P(H|C_2) = 0.5$	$P(H|C_3) = 0.9$
$P(C_1) = 0.05$	$P(C_2) = 0.25$	$P(C_3) = 0.70$



Your Estimate?

Maximum A Posteriori (MAP) Estimate:
The best hypothesis that fits observed data assuming a non-uniform prior

Most likely coin: C_3



Best estimate for P(H)

$P(H|C_3) = 0.9$

C_3

$P(H|C_3) = 0.9$

$P(C_3) = 0.70$

Did We Do The Right Thing?

$P(C_1|HT)$=0.035 $P(C_2|HT)$=0.481 $P(C_3|HT)$=0.485

C_1 C_2 C_3

$P(H|C_1) = 0.1$ $P(H|C_2) = 0.5$ $P(H|C_3) = 0.9$

Did We Do The Right Thing?

$P(C_1|HT)$ =0.035 $P(C_2|HT)$=0.481 $P(C_3|HT)$=0.485

C_2 and C_3 are almost equally likely

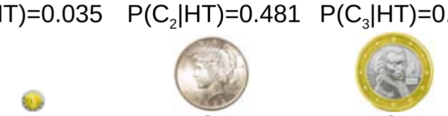
C_1 C_2 C_3

$P(H|C_1) = 0.1$ $P(H|C_2) = 0.5$ $P(H|C_3) = 0.9$

A Better Estimate

Recall: $P(H) = \sum_{i=1}^{3} P(H|C_i)P(C_i)$ = 0.680

$P(C_1|HT)$=0.035 $P(C_2|HT)$=0.481 $P(C_3|HT)$=0.485



C_1 C_2 C_3

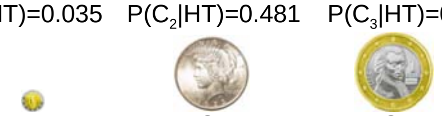
$P(H|C_1) = 0.1$ $P(H|C_2) = 0.5$ $P(H|C_3) = 0.9$

Bayesian Estimate

Bayesian Estimate: Minimizes prediction error, given data assuming an arbitrary prior

$$P(H) = \sum_{i=1}^{3} P(H|C_i)P(C_i) = 0.680$$

$P(C_1|HT)$=0.035 $P(C_2|HT)$=0.481 $P(C_3|HT)$=0.485



C_1 C_2 C_3

$P(H|C_1) = 0.1$ $P(H|C_2) = 0.5$ $P(H|C_3) = 0.9$

Comparison

After more experiments: HTHHHHHHHHH

ML (Maximum Likelihood):
- P(H) = 0.5
- after 10 experiments: P(H) = 0.9

MAP (Maximum A Posteriori):
- P(H) = 0.9
- after 10 experiments: P(H) = 0.9

Bayesian:
- P(H) = 0.68
- after 10 experiments: P(H) = 0.9

Summary

	Prior	Hypothesis
Maximum Likelihood Estimate	Uniform	The most likely
Maximum A Posteriori Estimate	Any	The most likely
Bayesian Estimate	Any	Weighted combination

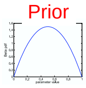
Easy to compute (Maximum Likelihood Estimate)

Still easy to compute
Incorporates prior knowledge (Maximum A Posteriori Estimate)

Minimizes error
Great when data is scarce
Potentially much harder to compute (Bayesian Estimate)

Bayesian Learning

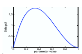
Use Bayes rule:



Data Likelihood → Prior

Posterior

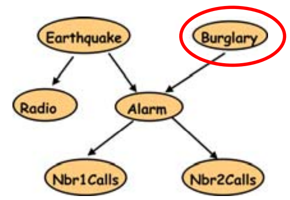
$$P(Y \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid Y)\, P(Y)}{P(\mathbf{X})}$$



Normalization

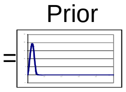
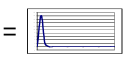
Or equivalently: $P(Y \mid \mathbf{X}) \propto P(\mathbf{X} \mid Y)\, P(Y)$

Parameter Estimation and Bayesian Networks



B
F
F
T
F
T

Prior

P(B) =  + data = 

Now compute either MAP or Bayesian estimate

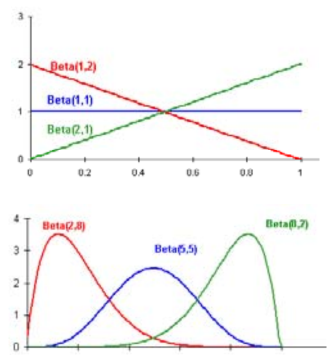
What Prior to Use?

- Prev, you ***knew***: it was one of only three coins

  - Now more complicated…
- The following are two common priors
- Binary variable Beta
  - Posterior distribution is binomial
  - Easy to compute posterior
- Discrete variable Dirichlet
  - Posterior distribution is multinomial
  - Easy to compute posterior

83

Beta Distribution



Beta Distribution

- Example: Flip coin with *Beta* distribution as prior over p [prob(heads)]
  1. Parameterized by two positive numbers: a, b
  2. Mode of distribution (E[p]) is *a/(a+b)*
  3. Specify our prior belief for *p = a/(a+b)*
  4. Specify confidence in this belief with high initial values for *a* and *b*
- Updating our prior belief based on data
  - incrementing *a* for every *heads* outcome
  - incrementing *b* for every tails outcome
- So after *h* heads out of *n* flips, our posterior distribution says P(*head*)=(*a*+*h*)/(*a*+*b*+*n*)

One Prior: Beta Distribution

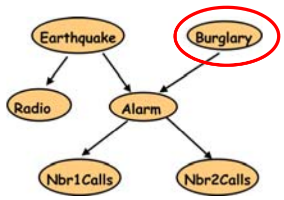
$$\beta_{a,b}(x) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1},$$

$$0 \le x \le 1 \text{ and } a, b > 0$$

$$\text{Here } \Gamma(y) = \int_0^\infty x^{y-1} e^{-x} dx$$

For any positive integer *y*, $\Gamma(y) = (y-1)!$

Parameter Estimation and Bayesian Networks



B
F
F
T
F
T

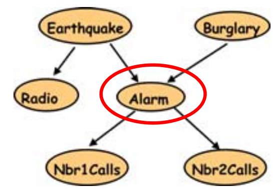
Prior

P(B|data) = Beta(1,4) "+ data" = (3,7)

B	¬B
.3	.7

Prior P(B)= 1/(1+4) = 20% with equivalent sample size 5

Parameter Estimation and Bayesian Networks



E	B
T	F
F	F
F	T
F	F
F	T
...	

A
T
F
T
T
F

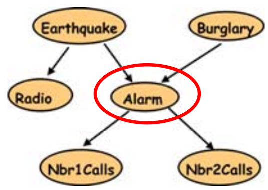
P(A|E,B) = ? Prior

P(A|E,¬B) = ?

P(A|¬E,B) = Beta(2,3)

P(A|¬E,¬B) = ?

Parameter Estimation and Bayesian Networks



E	B
T	F
F	F
F	T
F	F
F	T
...	

A
T
F
T
T
F

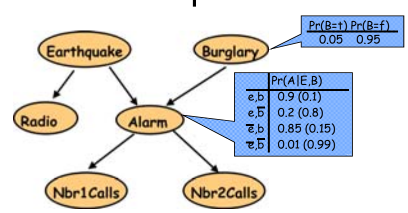
P(A|E,B) = ? Prior

P(A|E,¬B) = ?

P(A|¬E,B) = Beta(2,3) + data= Beta(3,4)

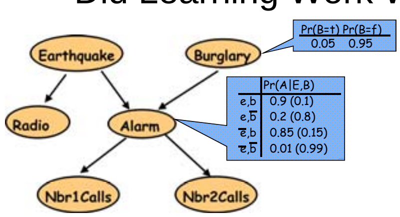
P(A|¬E,¬B) = ?

Output of Learning



E	B	R	A	J	M
T	F	T	T	F	T
F	F	F	F	F	T
F	T	F	T	T	T
F	F	F	T	T	T
F	T	F	F	F	F
...					

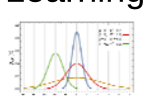
Did Learning Work Well?

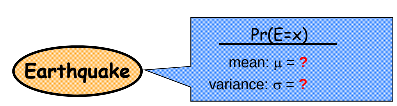


E	B	R	A	J	M
T	F	T	T	F	T
F	F	F	F	F	T
F	T	F	T	T	T
F	F	F	T	T	T
F	T	F	F	F	F
...					

Can easily calculate P(data) for learned parameters

Learning with Continuous Variables





$$\hat{\mu}_{MLE} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\hat{\sigma}^2_{MLE} = \frac{1}{N}\sum_{i=1}^{N} (x_i - \hat{\mu})^2$$

Using Bayes Nets for Classification

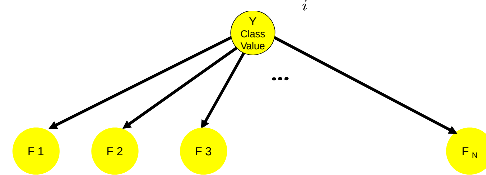
- One method of classification:
  - Use a probabilistic model!
  - Features are observed random variables F_i
  - Y is the query variable
  - Use probabilistic inference to compute most likely Y

$$y = \text{argmax}_y \; P(y|f_1 \ldots f_n)$$

- You already know how to do this inference

A Popular Structure: Naïve Bayes

$$P(\mathsf{Y}, \mathsf{F}_1 \ldots \mathsf{F}_n) = P(\mathsf{Y}) \prod_i P(\mathsf{F}_i|\mathsf{Y})$$



Assume that features are conditionally independent given class variable
Works surprisingly well for ***classification*** (predicting the right class)
But forces probabilities towards 0 and 1

Naïve Bayes

- Naïve Bayes assumption:
  - Features are independent given class:

$$P(X_1, X_2|Y) = P(X_1|X_2, Y)P(X_2|Y)$$
$$= P(X_1|Y)P(X_2|Y)$$

  - More generally:

$$P(X_1 \ldots X_n|Y) = \prod_i P(X_i|Y)$$

- How many parameters?
  - Suppose **X** is composed of *n* binary features

A Spam Filter

- 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.

Naïve Bayes for Text

- Bag-of-Words Naïve Bayes:
  - Predict unknown class label (spam vs. ham)
  - Assume evidence features (e.g. the words) are independent
  - Warning: subtly different assumptions than before!
- Generative model

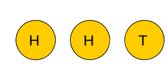
$$P(C, W_1 \ldots W_n) = P(C) \prod_i P(W_i|C)$$

Word at position i, not i^{th} word in the dictionary!

- Tied distributions and bag-of-words
  - Usually, each variable gets its own conditional probability distribution P(F|Y)
  - In a bag-of-words model
    - Each position is identically distributed
    - All positions share the same conditional probs P(W|C)
    - Why make this assumption?

Estimation: Laplace Smoothing

- Laplace's estimate:
  - pretend you saw every outcome once more than you actually did



$$P_{LAP}(x) = \frac{c(x) + 1}{\sum_x [c(x) + 1]}$$
$$= \frac{c(x) + 1}{N + |X|}$$

$P_{ML}(X) =$

$P_{LAP}(X) =$

Can derive this as a MAP estimate with *Dirichlet priors* (Bayesian justification)

NB with Bag of Words for text classification

- Learning phase:
  - Prior P(Y)
    - Count how many documents from each topic (prior)
  - $P(X_i|Y)$
    - For each of m topics, count how many times you saw word X_i in documents of this topic (+ k for prior)
    - Divide by number of times you saw the word (+ k×|words|)
- Test phase:
  - For each document
    - Use naïve Bayes decision rule

$$h_{NB}(\mathbf{x}) = \arg\max_y P(y) \prod_{i=1}^{LengthDoc} P(x_i|y)$$

Probabilities: Important Detail!

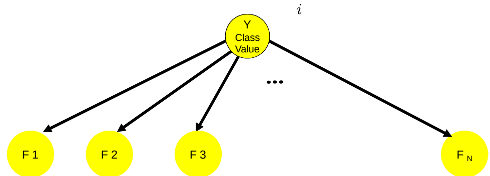
- $P(spam \mid X_1 \ldots X_n) = \prod_i P(spam \mid X_i)$

Any more potential problems here?

- We are multiplying lots of small numbers
  Danger of underflow!
  - $0.5^{57} = 7$ E -18
- Solution? Use logs and add!
  - $p_1 * p_2 = e^{\log(p1)+\log(p2)}$
  - Always keep in log form

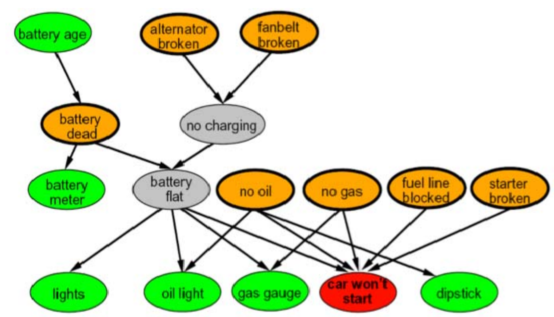
Naïve Bayes

$$P(Y, F_1 \ldots F_n) = P(Y) \prod_i P(F_i|Y)$$



Assume that features are conditionally independent given class variable
Works surprisingly well for classification (predicting the right class)
But forces probabilities towards 0 and 1

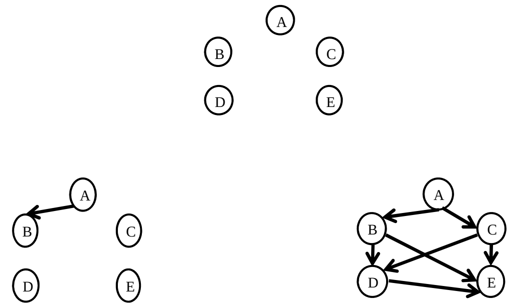
Example Bayes' Net: Car



What if we ***don't*** know structure?

Learning The Structure of Bayesian Networks

- Search thru the space…
  - of possible network structures!
  - (for now still assume can observe all values)
- For each structure, learn parameters
  - As just shown…
- Pick the one that fits observed data best
  - Calculate P(data)

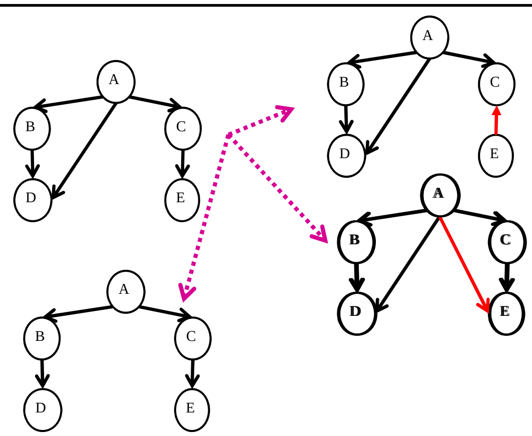


Two problems:
- Fully connected will be most probable
- Exponential number of structures

Learning The Structure of Bayesian Networks

- Search thru the space…
  - of possible network structures!
- For each structure, learn parameters
  - As just shown…
- Pick the one that fits observed data best
  - Calculate P(data)

Two problems:
- Fully connected will be most probable
  - Add penalty term (regularization) ∝ model complexity
- Exponential number of structures
  - Local search

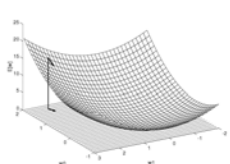


Score Functions

- Bayesian Information Criteion (BIC)
  - P(D | BN) – penalty
  - Penalty = ½ (# parameters) Log (# data points)
- MAP score
  - P(BN | D) = P(D | BN) P(BN)
  - P(BN) must decay exponentially with # of parameters for this to work well

Learning as Optimization

- Preference Bias
- Loss Function
  - Minimize ***loss*** over training data (test data)
  - Loss(h,data) = error(h, data) + complexity(h)
  - Error + regularization
- Methods
  - Closed form
  - Greedy search
  - Gradient ascent



Topics

- Learning Parameters for a Bayesian Network
  - Fully observable
    - Maximum Likelihood (ML),
    - Maximum A Posteriori (MAP)
    - Bayesian
  - Hidden variables (EM algorithm)
- Learning Structure of Bayesian Networks