Data Mining & Machine Learning

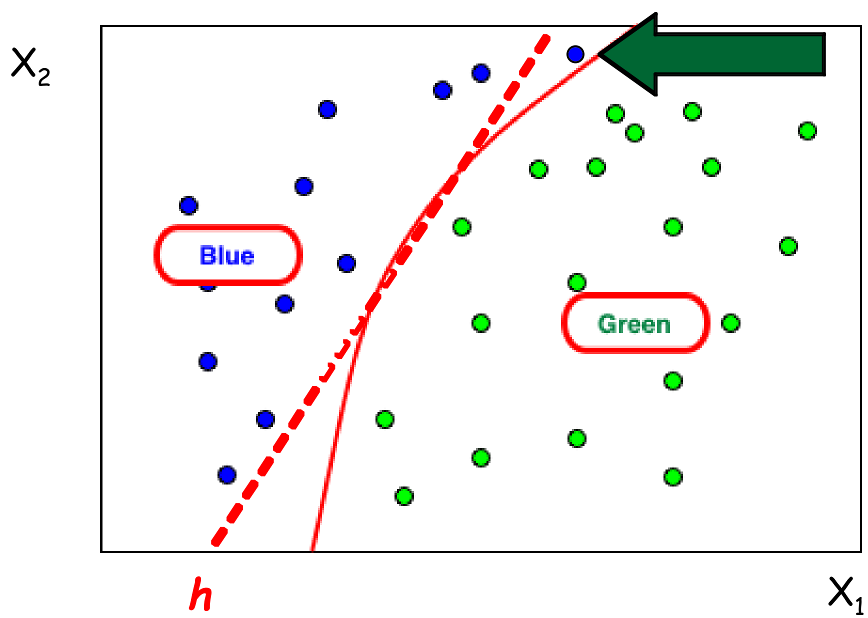
CS37300
Purdue University

September 15, 2017

Classification

- In its simplest form, a classification model defines a decision boundary (h) and labels for each side of the boundary

- Input: $\mathbf{x}=\{x_1,x_2,\ldots,x_n\}$ is a set of attributes, function f assigns a label y to input $\mathbf{x}$, where y is a discrete variable with a finite number of values



Discriminative classification

- Model the decision boundary directly
- Direct mapping from inputs **x** to class label y
- No attempt to model probability distributions
- May seek a discriminant function $f(\mathbf{x};\theta)$ that maximizes measure of separation between classes
- Examples:
  - Perceptrons, nearest neighbor classifiers, support vector machines, decision trees

Probabilistic classification

- Model the underlying probability distributions
  - Posterior class probabilities: $p(y|\mathbf{x})$
  - Class-conditional and class prior: $p(\mathbf{x}|y)$ and $p(y)$
- Maps from inputs $\mathbf{x}$ to class label y indirectly through posterior class distribution $p(y|\mathbf{x})$
- Examples:
  - Naive Bayes classifier, logistic regression, probability estimation trees

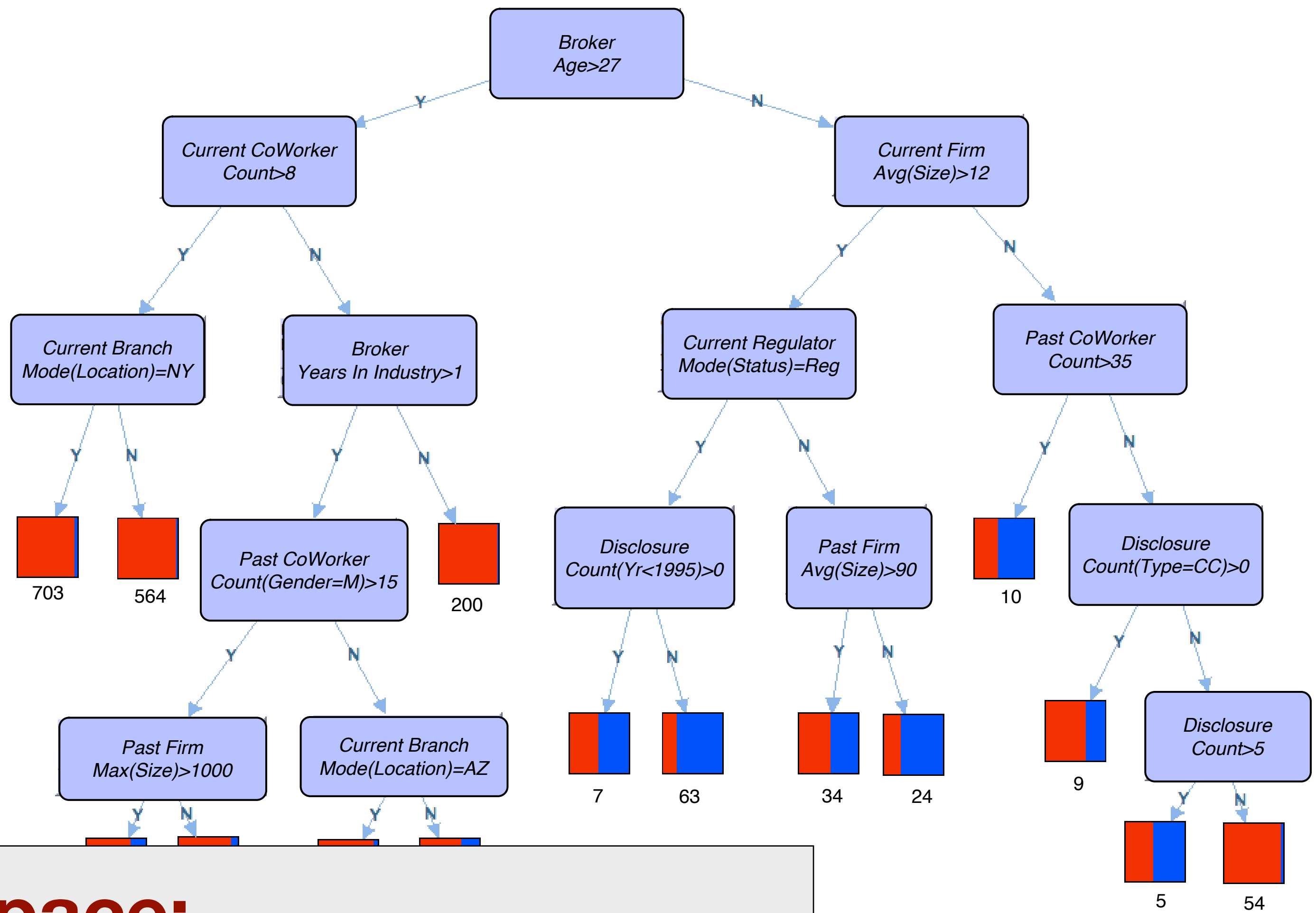
Knowledge representation

Knowledge representation

- *Underlying structure of the model or patterns that we seek from the data*
    - Defines space of possible models for algorithm to search over

- Model: high-level global description of dataset
    - “All models are wrong, some models are useful”
      G. Box and N. Draper (1987)
    - Choice of model family determines space of parameters and structure
    - Estimate model parameters and possibly model structure from training data

Classification tree



Model space:
all possible decision trees

Model space

- How large is the space?
- Can we search exhaustively?
- Simplifying assumptions
  - Binary tree
  - Fixed depth
  - 10 binary attributes

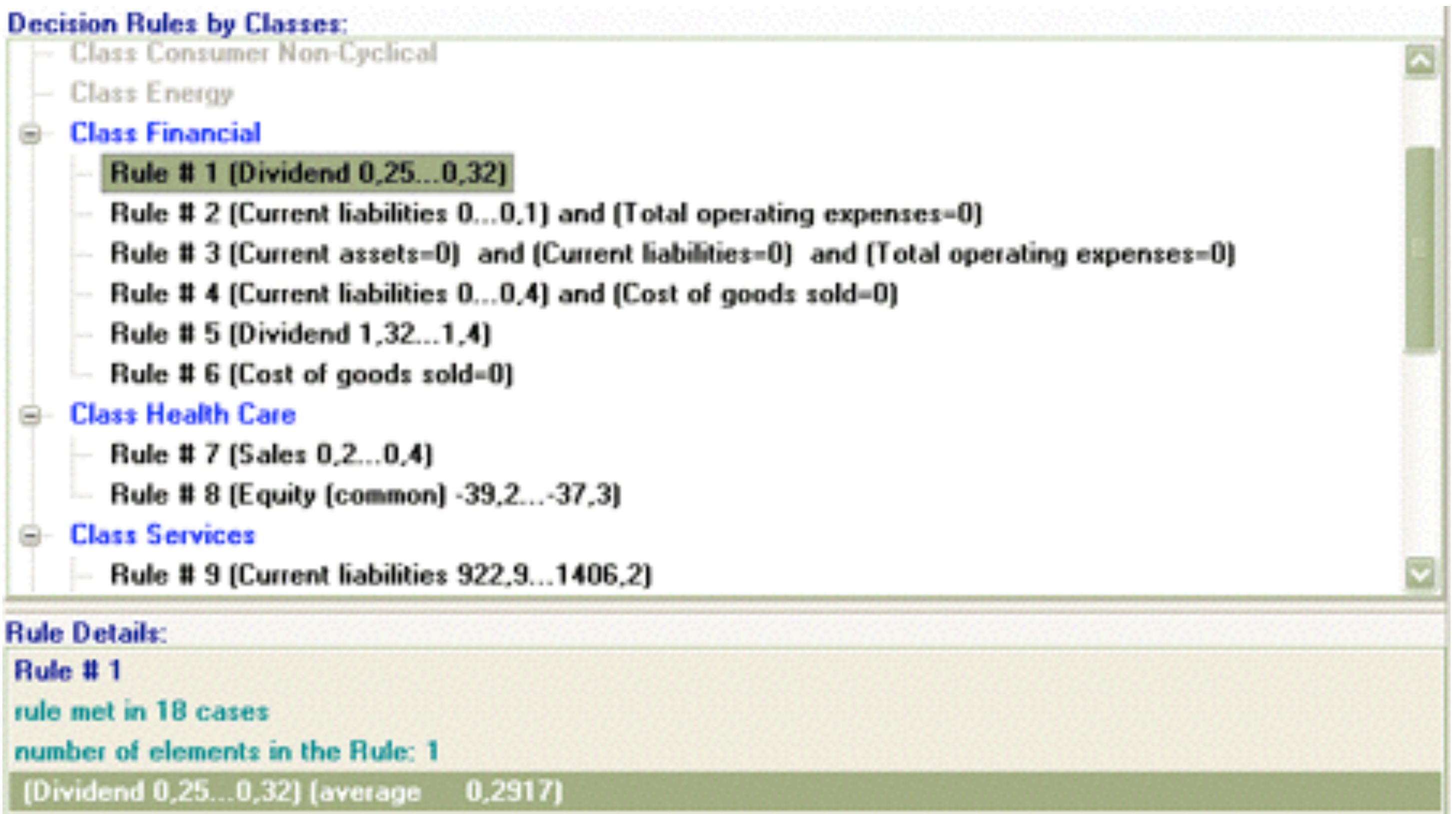
Tree depth	Number of trees
1	10
2	8×10^{2}
3	3×10^{6}
4	2×10^{13}
5	5×10^{25}

Perceptron

$$f(x) = \begin{cases} 1 & \sum w_j x_j > 0 \\ 0 & \sum w_j x_j \leq 0 \end{cases}$$

Model space:
weights *w*, for each of j attributes

Decision rule



Model space:
all possible rules formed from conjunctions of features

Parametric vs. non-parametric models

- Parametric
    - Particular functional form is assumed (e.g., Binomial)
    - **Number of parameters is fixed in advance**
    - Examples: Naive Bayes, perceptron
- Non-parametric
    - Few assumptions are made about the functional form
    - **Model structure is determined from data**
    - Examples: classification tree, nearest neighbor

Predictive modeling: learning

Learning predictive models

- Choose a **data representation**
- Select a **knowledge representation** (a “model”)
  - Defines a **space** of possible models $M=\{M_1, M_2, \ldots, M_k\}$
- Use **search** to identify “best” model(s)
  - Search the space of models (i.e., with alternative structures and/or parameters)
  - Evaluate possible models with **scoring function** to determine the model which best fits the data

Learning predictive models

- Choose a **data representation**
- Select a **knowledge representation** (a "model")
  - Defines a **space** of possible models $M=\{M_1, M_2, \ldots, M_k\}$
- Use **search** to identify "best" model(s)
  - Search the space of models (i.e., with alternative structures and/or parameters)
  - Evaluate possible models with **scoring function** to determine the model which best fits the data

Scoring functions

- Given a model M and dataset D, we would like to “score” model M with respect to D
  - Goal is to rank the models in terms of their utility (for capturing D) and choose the “best” model
  - Score function can be used to search over ***parameters*** and/or ***model structure***
- Score functions can be different for:
  - Models vs. patterns
  - Predictive vs. descriptive functions
  - Models with varying complexity (i.e., number parameters)

Predictive scoring functions

- Assess the quality of predictions for a set of instances
  - Measures **difference** between the prediction M makes for an instance *i* and the true class label value of *i*

$$S(M) = \underbrace{\sum_{i=1}^{N_{test}}}_{\text{Sum over examples}} \underbrace{d}_{\text{Distance between predicted and true}} \Big[\underbrace{f(x(i); M)}_{\text{Predicted class label for item } i}, \underbrace{y(i)}_{\text{True class label for item } i} \Big]$$

Predictive scoring functions

- Common score functions:
  - Zero-one loss

$$S_{0/1}(M) = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} I\left[f(x(i);M), y(i)\right]$$

$$\text{where } I(a,b) = \begin{cases} 1 & a \neq b \\ 0 & \text{otherwise} \end{cases}$$

  - Squared loss

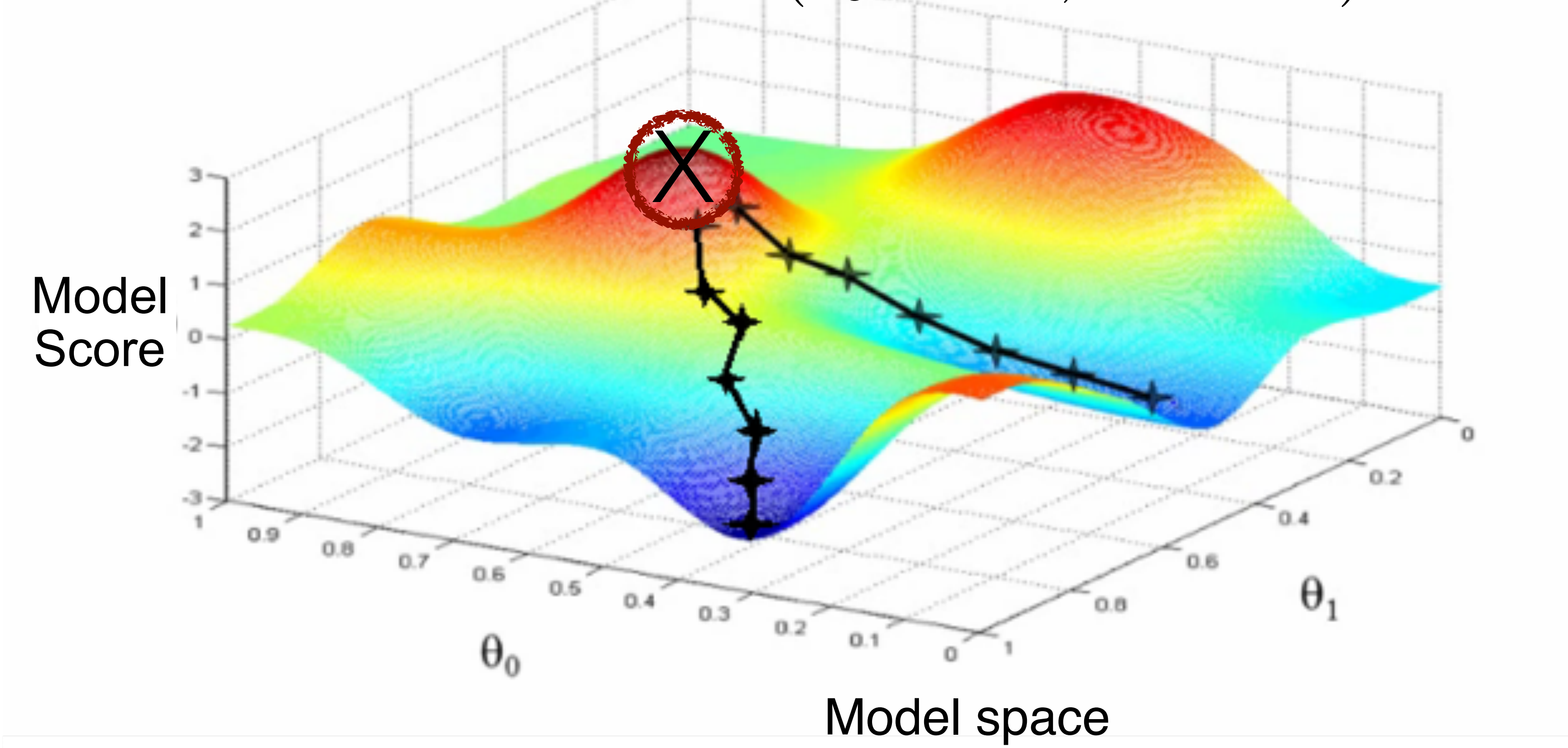
$$S_{sq}(M) = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \left[f(x(i);M) - y(i)\right]^2$$

- Do we minimize or maximize these functions?

Where’s the search?

What space are we searching?

$$\text{Learned model} \approx (\theta_0 = 0.8, \theta_1 = 0.4)$$



Alex Holehuse, Notes from Andrew Ng's Machine Learning Class, http://www.holehouse.org/mlclass/01_02_Introduction_regression_analysis_and_gr.html

Searching over models/patterns

- Consider a **space** of possible models $M=\{M_1, M_2, ..., M_k\}$ with parameters θ
- Search could be over model structures or parameters, e.g.:
    - **Parameters**: In a linear regression model, find the regression coefficients (**β**) that minimize squared loss on the training data
    - **Model structure**: In a decision trees, find the tree structure that maximizes accuracy on the training data

Example model:
Naive Bayes classifiers

Classification as probability estimation

- Instead of learning a function f that assigns labels
- Learn a conditional probability distribution over the output of function f
- $P(\, f(x) \mid x\,) = P(\, f(x) = y \mid x_1, x_2, \ldots, x_p\,)$
- Can use probabilities for the other two tasks
    - Classification
    - Ranking

Knowledge representation and model space

Bayes rule for probabilistic classifier

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

Bayes rule

$$= \frac{P(\mathbf{X}|Y)P(Y)}{[P(\mathbf{X}|Y=+)P(Y=+)] + [P(\mathbf{X}|Y=-)P(Y=-)]}$$

$$\propto P(\mathbf{X}|Y)P(Y)$$

Denominator: normalizing factor to make probabilities sum to 1 (can be computed from numerators)

Naive Bayes classifier

$$\boxed{P(Y|\mathbf{X}) \propto P(\mathbf{X}|Y)P(Y)}$$

Bayes rule

$$\propto \boxed{\prod_{i=1}^{m} P(X_i|Y)} P(Y)$$

Naive assumption

Assumption: Attributes are *conditionally independent* given the class

Naive Bayes classifier

$$
\begin{aligned}
p(y|\mathbf{x}) &= \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})} \\
&= \frac{\prod_i p(x_i|y)\, p(y)}{\sum_j p(\mathbf{x}|y_j)p(y_j)}
\end{aligned}
$$

Model space:
parameters in conditional distributions $p(x_i|y)$
parameters in prior distribution $p(y)$

NBC learning

$$P(BC|A,I,S,CR) = \frac{P(A,I,S,CR|BC)P(BC)}{P(A,I,S,CR)}$$

$$= \frac{P(A|BC)P(I|BC)P(S|BC)P(CR|BC)P(BC)}{P(A,I,S,CR)}$$

$$\propto P(A|BC)P(I|BC)P(S|BC)P(CR|BC)P(BC)$$

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

NBC parameters = CPDs+prior

CPDs: $P(A|BC)$
$P(I|BC)$
$P(S|BC)$
$P(CR|BC)$
Prior: $P(BC)$

Score function

Likelihood

- Let $D = \{x(1), ..., x(n)\}$
- Assume the data *D* are independently sampled from the same distribution:

$$p(X|\theta)$$

- The likelihood function represents the probability of the data as a function of the model parameters:

$$\begin{aligned} L(\theta|D) &= L(\theta|x(1), ..., x(n)) \\ &= p(x(1), ..., x(n)|\theta) \\ &= \prod_{i=1}^{n} p(x(i)|\theta) \end{aligned}$$

If instances are independent, likelihood is product of probs

Likelihood (cont')

- Likelihood is not a probability distribution
    - Gives relative probability of data given a parameter
    - Numerical value of *L* is not relevant, only the ratio of two scores is relevant, e.g.,:

$$\frac{L(\theta_1|D)}{L(\theta_2|D)}$$

- **Likelihood function**: allows us to determine unknown parameters based on known outcomes
- **Probability distribution**: allows us to predict unknown outcomes based on known parameters

NBCs: Likelihood

- NBC likelihood uses the NBC probabilities for each data instance (i.e., probability of the class given the attributes)

$$L(\theta|D) = \prod_{i=1}^{n} p(y_i|\mathbf{x}_i;\theta)$$

General likelihood

$$\propto \prod_{i=1}^{n} p(\mathbf{x}_i|y_i;\theta)p(y_i|\theta)$$

Bayes rule

$$\propto \prod_{i=1}^{n} \prod_{j=1}^{p} p(x_{ij}|y_i;\theta)p(y_i|\theta)$$

Naive assumption

Search

Maximum likelihood estimation

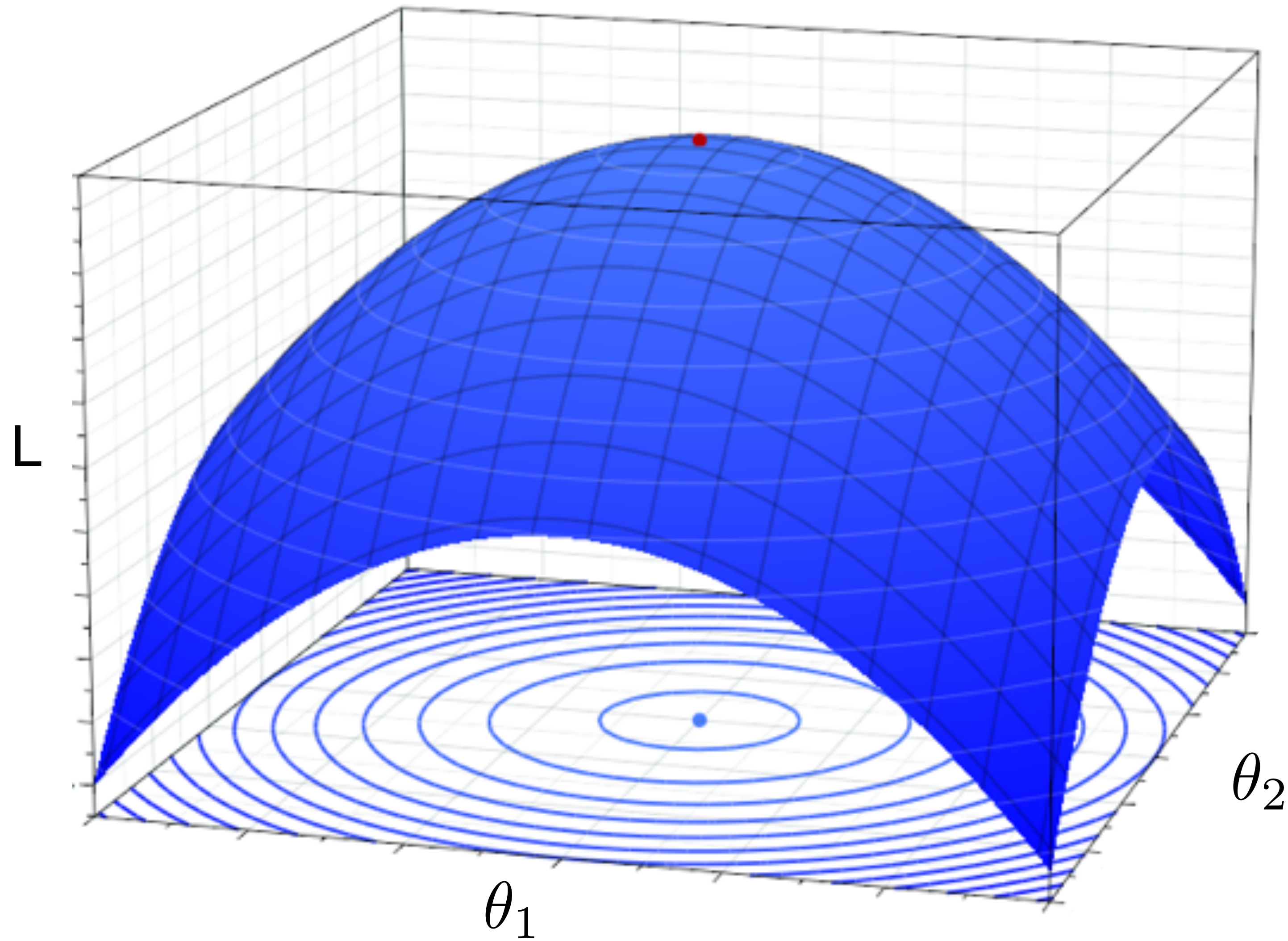
- Most widely used method of parameter estimation
- "Learn" the best parameters by finding the values of θ that maximizes likelihood:

$$\hat{\theta}_{MLE} = \arg\max_{\theta} L(\theta)$$

- Often easier to work with loglikelihood:

$$\begin{aligned} l(\theta|D) &= log\, L(\theta|D) \\ &= log \prod_{i=1}^{n} p(x(i)|\theta) \\ &= \sum_{i=1}^{n} logp(x(i)|\theta) \end{aligned}$$

Likelihood surface



If the likelihood surface is convex we can often determine the parameters that maximize the function analytically

MLE for multinomials

- Let $X \in \{1, \ldots, k\}$ be a discrete random variable with k values, where $P(X{=}j){=}\theta_j$
- Then P(X) is a multinomial distribution:

$$P(X|\theta) = \prod_{j=1}^{k} \theta_j^{I(X=j)}$$

  where I(X=j) is an indicator function

- The likelihood for a data set D=[x_1, ..., x_N] is:

$$P(D|\theta) = \prod_{n=1}^{N} \prod_{j=1}^{k} \theta_j^{I(x_n=j)} = \prod_{j} \theta_j^{N_j}$$

- The maximum likelihood estimates for each parameter are: $\boxed{\hat{\theta}_j = \frac{N_j}{N}}$
  (using Lagrange multipliers)

In this case, MLE can be determined analytically by counting

Learning CPDs from examples

	X_1		
	Low	Medium	High
Y: Yes	10	13	17
Y: No	2	13	0

$$P[\,X_1 = \text{Low} \mid Y = \text{Yes}] = \frac{10}{(10+13+17)}$$

$$P[\,Y = \text{No}] = \frac{(2+13)}{(2+13+10+13+17)}$$

NBC learning

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

- Estimate prior P(BC) and conditional probability distributions P(A | BC), P(I | BC), P(S | BC), P(CR | BC) independently with maximum likelihood estimation

P(BC)

BC	θ
yes	$9/14$
no	$5/14$

P(A I BC)

BC	A	θ
yes	$<= 30$	$2/9$
	$31..40$	$4/9$
	> 40	$3/9$
no	$<= 30$	$3/5$
	$31..40$	$0/5$
	> 40	$2/5$

P(I I BC)

BC	I	θ
yes	high	$2/9$
	med	$4/9$
	low	$3/9$
no	high	$2/5$
	med	$2/5$
	low	$1/5$

P(S I BC)

BC	S	θ
yes	yes	$6/9$
	no	$3/9$
no	yes	$1/5$
	no	$4/5$

P(CR I BC)

BC	CR	θ
yes	exc	$3/9$
	fair	$6/9$
no	exc	$4/5$
	fair	$1/5$

NBC prediction

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no
31..40	**high**	**no**	**excellent**	**?**

- What is the probability that a new person will buy a computer?

$$P(BC = yes|A = 31..40, I = high, S = no, CR = exc)$$
$$\propto P(A = 31..40|BC = yes)P(I = high|BC = yes)$$
$$P(S = no|BC = yes)P(CR = exc|BC = yes)P(BC = yes)$$

P(BC)

BC	θ
yes	$9/14$
no	$5/14$

P(A I BC)

BC	A	θ
	$<= 30$	$2/9$
yes	$31..40$	$4/9$
	> 40	$3/9$
	$<= 30$	$3/5$
no	$31..40$	$0/5$
	> 40	$2/5$

P(I I BC)

BC	I	θ
	high	$2/9$
yes	med	$4/9$
	low	$3/9$
	high	$2/5$
no	med	$2/5$
	low	$1/5$

P(S I BC)

BC	S	θ
yes	yes	$6/9$
	no	$3/9$
no	yes	$1/5$
	no	$4/5$

P(CR I BC)

BC	CR	θ
yes	exc	$3/9$
	fair	$6/9$
no	exc	$4/5$
	fair	$1/5$

Zero counts are a problem

- If an attribute value does not occur in training example, we assign **zero** probability to that value
- How does that affect the conditional probability P[f(x) | x] ?
- It equals 0!!!
- Why is this a problem?
- Adjust for zero counts by “smoothing” probability estimates

Smoothing: Laplace correction

X_1

		Low	Medium	High
Y	Yes	10	13	17
	No	2	13	0

Laplace correction
Numerator: ***add 1***
Denominator: ***add k***,
where k=number of possible values of X

$$P[X_1 = \text{High} \mid Y = \text{No}] = \frac{0 + 1}{(2 + 13 + 0) + 3}$$

Adds uniform prior

Is assuming independence a problem?

- What is the effect on probability estimates?
  - Over-counting evidence, leads to overly confident probability estimate
- What is the effect on classification?
  - Less clear…
  - For a given input x, suppose f(x) = True
  - Naïve Bayes will correctly classify if P[f(x) = True | x] > 0.5
    ...thus it may not matter if probabilities are overestimated

Naive Bayes classifier

- Simplifying (naive) assumption:
  attributes are conditionally independent given the class
- Strengths:
  - Easy to implement
  - Often performs well even when assumption is violated
  - Can be learned incrementally
- Weaknesses:
  - Class conditional assumption produces skewed probability estimates
  - Dependencies among variables cannot be modeled

NBC learning

- Model space
  - Parametric model with specific form
    (i.e., based on Bayes rule and assumption of conditional independence),
  - Models vary based on parameter estimates in CPDs
- Search algorithm
  - MLE optimization of parameters
    (convex optimization results in exact solution)
- Scoring function
  - Likelihood of data given NBC model form