CS57300
PURDUE UNIVERSITY

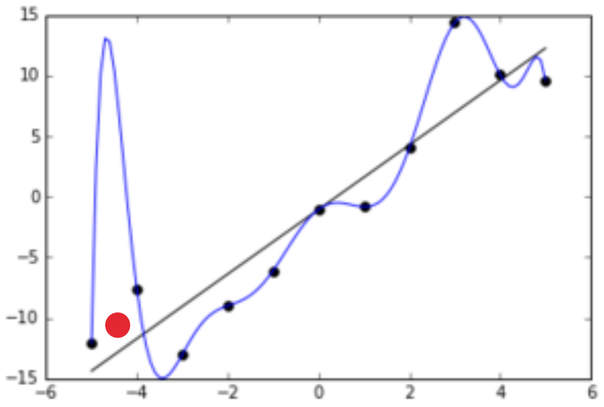
JANUARY 31, 2019

DATA MINING

PREDICTION AND EVALUATION

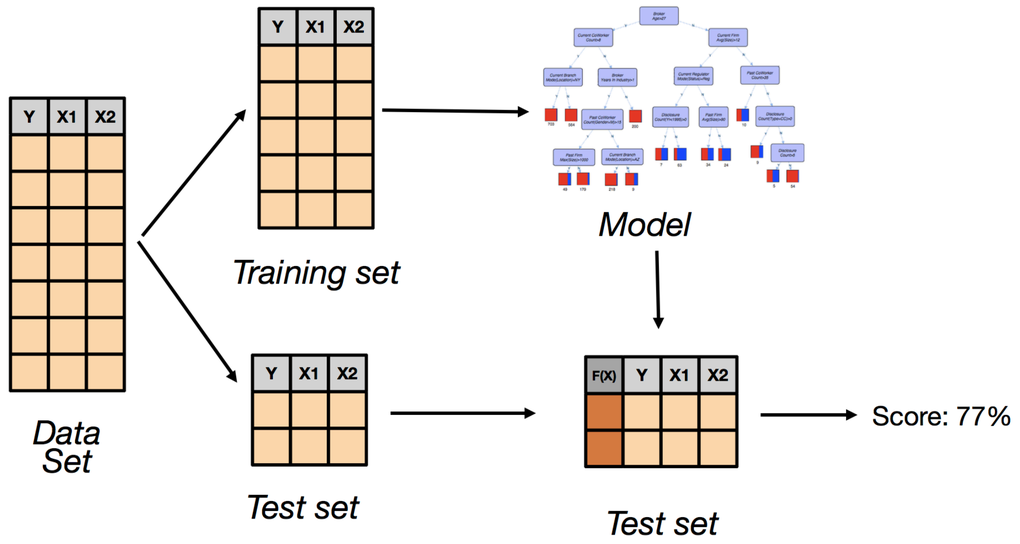
PREDICTING UNSEEN DATA

- Use the learned model to predict class label *y* for unseen data ***x***
- Evaluating the performance of the learned model (i.e., its capability to generalize) by computing the scoring functions on the unseen data
- How to evaluate the performance of a model given a limited amount of data?
  - Can we train the model on the entire dataset and use the scoring function value on it as an estimate for the model's performance on the unseen data?



SPLIT INTO TRAINING DATA AND TESTING DATA

- Split the dataset into disjoint two sets: training set and testing set
- Learn the model using the training set and evaluate the model's performance on testing set



OVERFITTING

- When the performance of your model on the training data is much better than its performance on the testing data, you are likely overfitting…
- Consider a distribution D of data representing a population and a sample D_S drawn from D, which is used as training data
- Given a model space M, a score function S, and a learning algorithm that returns a model $m \in M$, the algorithm **overfits** the training data D_S if:
  $\exists m' \in M$ such that $S(m, D_S) > S(m', D_S)$ but $S(m, D) < S(m', D)$
  - In other words, there is another model (m') that is better on the entire distribution and if we had learned from the full data we would have selected it instead

EXAMPLE LEARNING PROBLEM

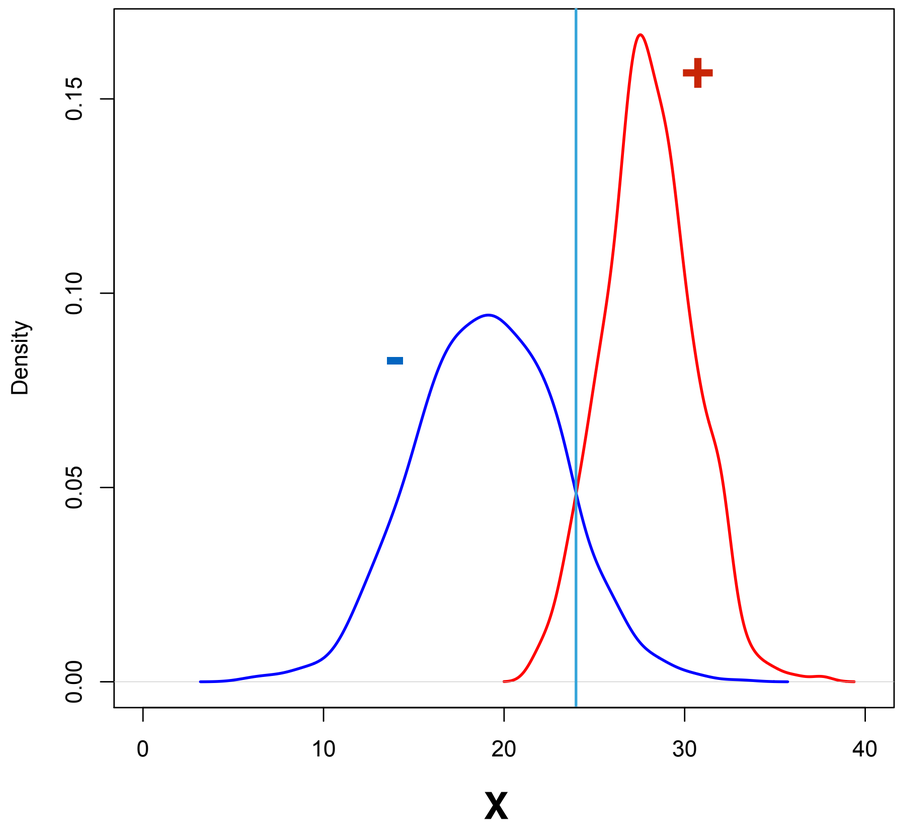
Task: Devise a rule to classify items based on the attribute ***x***

Knowledge representation:
If-then rules

Example rule:
If x > 24 then +
Else -

What is the model space?

All possible thresholds



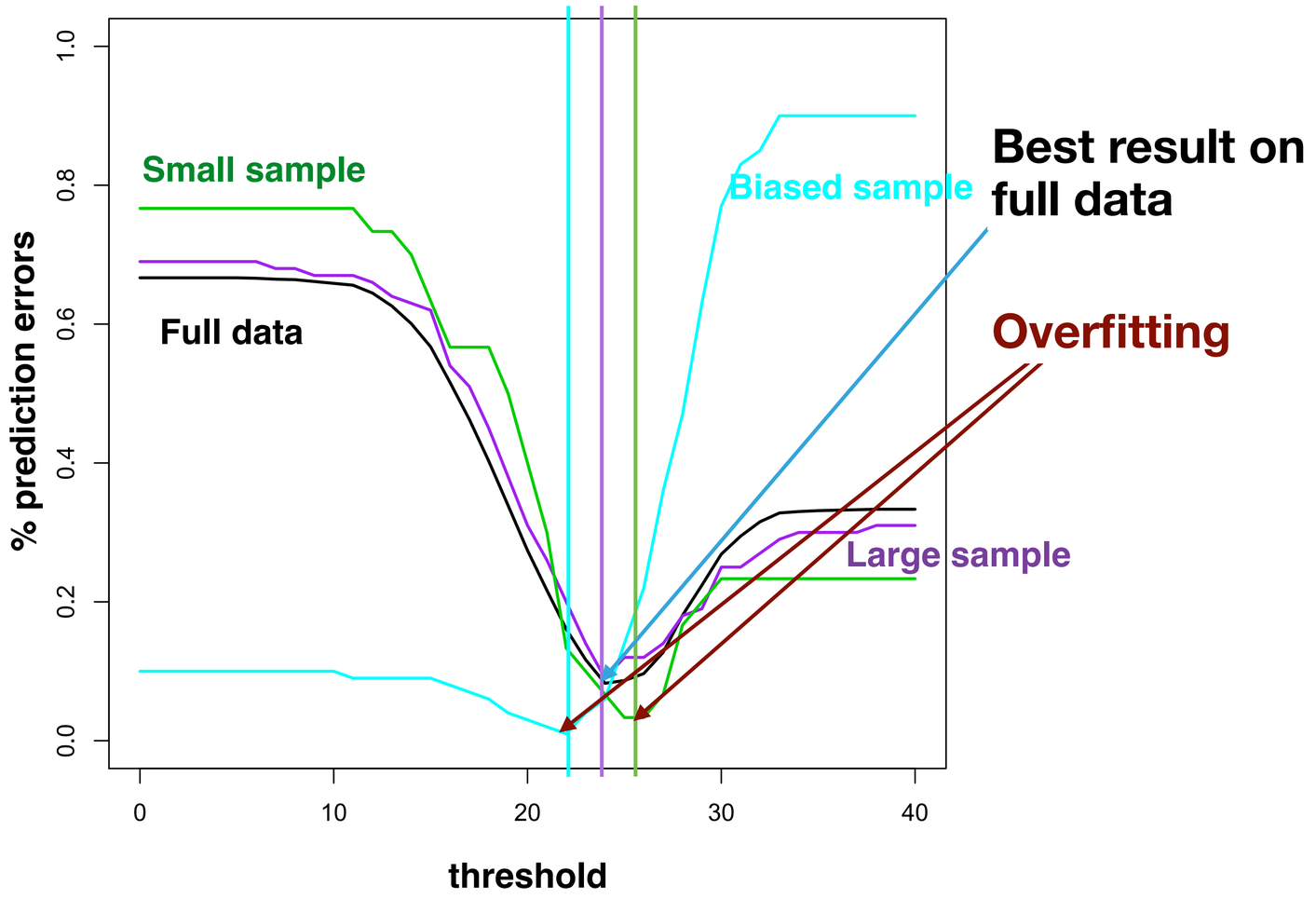
What score function?

Prediction error rate

SCORE FUNCTION OVER MODEL SPACE

Search procedure?

Try all thresholds, select one with lowest score



DEAL WITH OVERFITTING

- It's easier for more complex models to overfit
- Approaches to avoid overfitting
  - Regularization
  - Model selection through cross-validation
  - Penalty term in scoring function

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

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

BAYES RULE

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

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

DENOMINATOR: NORMALIZING FACTOR TO MAKE PROBABILITIES SUM TO 1 (CAN BE COMPUTED FROM NUMERATORS)

NAIVE BAYES CLASSIFIER

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

BAYES RULE

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

NAIVE ASSUMPTION

Assumption: Attributes are *conditionally independent* given the class

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

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

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

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

NBC parameters = CPDs+prior

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

SCORING FUNCTION

LIKELIHOOD

- Let $D = \{x(1), ..., x(n)\}$, where $x(i) = \langle \mathbf{x}_i, c_i \rangle$
- 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

NBC: LIKELIHOOD

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

$$L(\theta|D) = \prod_{i=1}^{n} p(x(i)\,|\,\theta)$$ **General likelihood**

$$= \prod_{i=1}^{n} p(\mathbf{x}_i|c_i,\theta)P(c_i|\theta)$$ **Product rule**

$$= \prod_{i=1}^{n}\prod_{j=1}^{m} p(x_{ij}|c_i,\theta)P(c_i|\theta)$$ **Naive assumption**

SEARCH

MAXIMUM LIKELIHOOD ESTIMATION

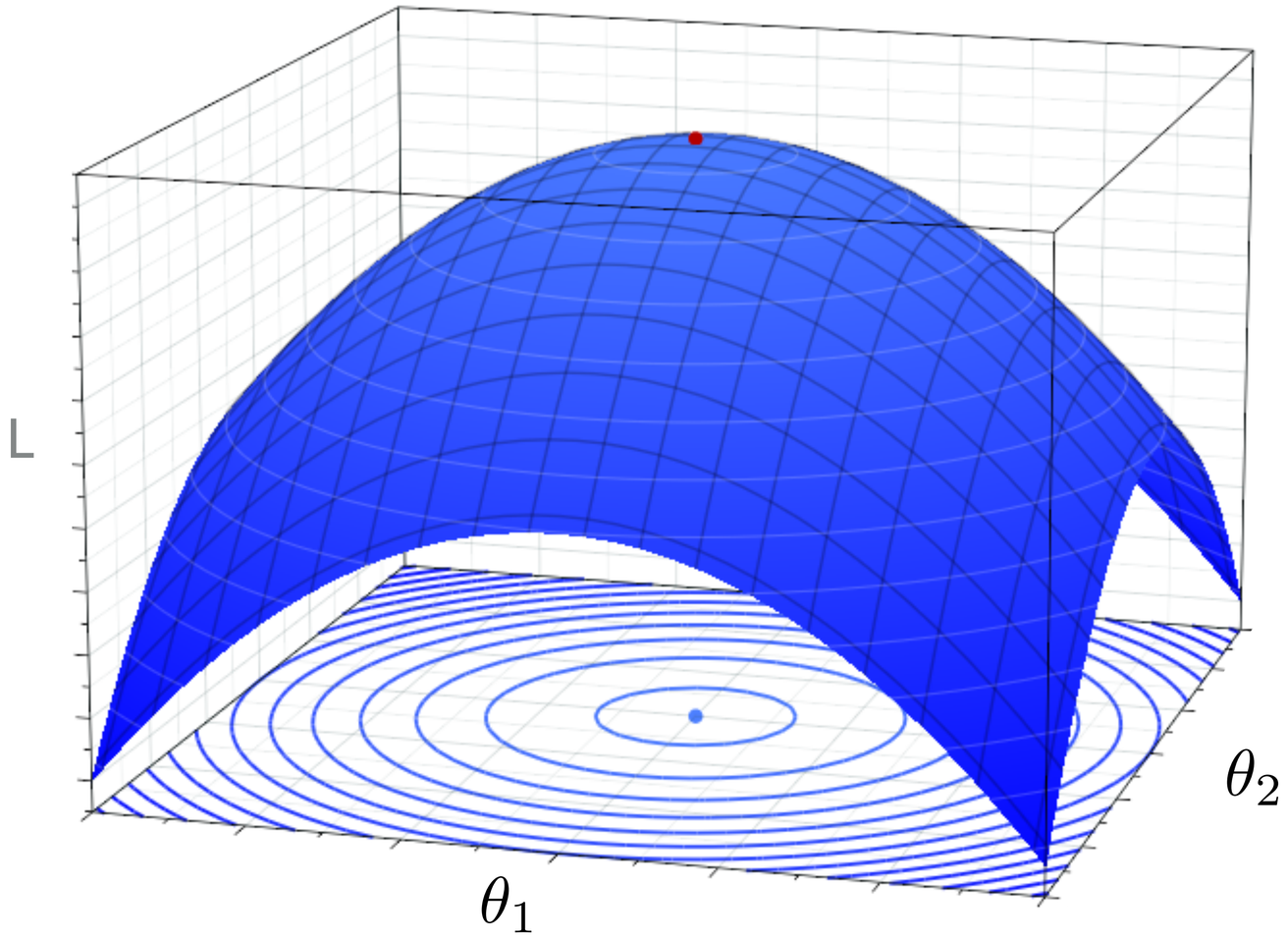
- "Learn" the best parameters by finding the values of θ that maximizes likelihood:

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

- 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 NBC

- Likelihood: $L(\theta \mid D) = \prod_{i=1}^{n} \prod_{j=1}^{m} P(x_{ij} \mid c_i) P(c_i)$
- Parameters:
  - Priors: Denote $p_l = P(c_i = l), l = 1,...,L$ (i.e., there are *L* possible labels)
  - Conditional probability distributions: Denote $q_l^{jk} = P(x_{ij} = k \mid c_i = l), k = 1,...,K(j)$ (i.e., the *j*-th attribute has *K*(*j*) possible value)

MLE FOR NBC

- Rewrite likelihood: $L(\theta \mid D) = (\prod_{l=1}^{L} p_l^{N_l})(\prod_{l=1}^{L}\prod_{j=1}^{m}\prod_{k=1}^{K(j)} (q_l^{jk})^{N_l^{jk}})$
  - $N_l = \sum_{i=1}^{n} I(c_i = l)$, i.e., the number of data points in class *l*
  - $N_l^{jk} = \sum_{i=1}^{n} I(c_i = l, x_{ij} = k)$, i.e. the number of data points in class *l*, and its *j*-th attribute is *k*
- Convex maximization
  - $p_l = N_l / n$, i.e., the fraction of data in the training set where its label is *l*
  - $q_l^{jk} = N_l^{jk} / N_l$, i.e. the fraction of data whose *j*-th attribute is *k* among data whose label is *l*

LEARNING CPDS FROM EXAMPLES

Y	X_1 = Low	X_1 = Med	X_1 = High
Yes	10	13	17
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 | 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 | 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 | BC)

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

P(CR | 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 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 | 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 | 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 | BC)

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

P(CR | BC)

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

IS THERE ANY PROBLEM?

		X_1		
		Low	**Med**	**High**
Y	**Yes**	10	13	17
	No	2	13	0

ZERO COUNTS ARE A PROBLEM

- If an attribute value does not occur in training data, 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

Y	Low	Med	High
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

WHAT ABOUT CONTINUOUS VARIABLES

- Discretize continuous variables through binning
  - Split the range of the continuous variable to several bins, assign a categorical value to each bin, and map continuous values fall into that bin to the assigned categorical value

- Model the probability distribution for continuous variables explicitly
  - For example, assume a Gaussian distribution and introduce additional parameters: $P(x_{ij} = x \mid c_i = l) \sim N(\mu_j^l, \sigma_j^l)$

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

ASSIGNMENT 2 IS OUT!

- Implement Naive Bayes Classifier to predict the outcome of speed dating events!
- Please implement your own version!
- Due date: Wednesday Feb 13, 11:59pm