Text Mining
Project/Lab

Behrang Q. Zadeh
behrangatoffice@gmail.com

Text Classification(2)

Pattern Recognition and Classification

- Detecting patterns and structures is a central theme in text mining.
- We usually start with the hypothesis that certain observable patterns in text are correlated to a particular task we address in text mining.
- In the previous session we learned how to do basic classification tasks in order to detect these observable patterns:
    - Naïve Bayes
    - Decision Tree

Goal of this session

- We will dig more into the evaluation of a classification task;
- We will have a closer look at some learning techniques.

Evaluation

- Evaluation is required
  - To decide whether a classifier is reliable and to quantify its quality;
  - To guide us in the process of feature selection;
  - Also, to compare classification techniques.

Evaluation

- Evaluation is required
  - To decide whether a classifier is reliable and to quantify its quality;
  - To guide us in the process of feature selection;
  - Also, to compare classification techniques.
- Depending on the evaluation procedures, we can distinguish different types of evaluation tasks:
  - Intrinsic evaluation: an isolated classifier is evaluated with respect to a pre-defined gold standard result (i.e. a widely accepted dataset);
  - Extrinsic: a classifier is evaluated when it provides a precise functionality for a human user (often more complex, can say why?).

Evaluation

- There exists other classification of evaluation techniques:
  - Black-box vs. glass-box evaluation (similar to software engineering);
  - Automatic vs. manual evaluation (automatic is preferred).
- Evaluation is an important theme in NLP (sometimes a research question itself).

Evaluation

- There exists other classification of evaluation techniques:
  - Black-box vs. glass-box evaluation (similar to software engineering);
  - Automatic vs. manual evaluation (automatic is preferred).
- Evaluation is an important theme in NLP (sometimes a research question itself).
- For an evaluation task, we first need a performance measure.
  - There exists an open list of performance measures, some of them task dependant:
    - Accuracy
    - BLEU
    - Cohen's kappa
    - Usability metrics

Evaluation

- There exists other classification of evaluation techniques:
  - Black-box vs. glass-box evaluation (similar to software engineering);
  - Automatic vs. manual evaluation (automatic is preferred).
- Evaluation is an important theme in NLP (sometimes a research question itself).
- For an evaluation task, we first need a performance measure.
  - There exists an open list of performance measures, some of them task dependant:
    - Accuracy
    - BLEU
    - Cohen's kappa
    - Usability metrics

Our focus is on automatic intrinsic evaluation.

Intrinsic Evaluation

- For an intrinsic evaluation we need a test set (or evaluation set).
- We then calculate a score (performance measure) for a classifier by comparing the labels that it generates for the inputs in the test set with the correct asserted labels for them.

Intrinsic Evaluation

- For an intrinsic evaluation we first need a test set (or **evaluation set**) .
- We then calculate a score (performance measure) for a classifier by comparing the labels that it generates for the inputs in the test set with the correct asserted labels for them.
- Test set is often has the same structure as training set.
- However, it is very important that the test set be distinct from the training corpus to avoid over-fitting:
    - a model can simply memorized its input, without really learning how to generalize to new examples.
    - A misleading score!

Intrinsic Evaluation

- There is often a trade-off between the amount of data available for testing and the amount available for training.
- The number of instances in the test set depends on the classification task and the distribution of labels (sometimes as few as 100 instances).
- The test set must be balanced and diverse.

Intrinsic Evaluation

- The test set must be balanced and diverse.
    - If there are a large number of labels, or some of labels are infrequent, then the test set must be big enough to ensure that the least frequent label are occurring enough (e.g. 50 times).
    - If the test set contains many closely related instances, then the size of the test set should be increased to ensure that this lack of diversity does not skew the evaluation results.
- When large amounts of annotated data are available, it is common to use 10% of the overall data for evaluation.

Intrinsic Evaluation

- An obtained score is more reliable, if test set and train set are less similar:
  - For a PoS tagger that is trained on a new corpus, the obtained score from the evaluation of the tagger on another news corpus is less reliable than the evaluation of the tagger on a corpus of text of genres other than news.
- In short, test set must be a real representative of the actual text data that a classifier is going to deal with.

Intrinsic Evaluation

- Accuracy is one simple metric that can be used to evaluate a classifier.
- Accuracy measures the percentage of inputs in the test set that the classifier correctly labelled.
  - A classifier that predicts the correct labels 60 times in a test set containing 100 names would have an accuracy of 60/100 = 60%.
  - The function **`nltk.classify.accuracy()`**can be used to measure accuracy.

Intrinsic Evaluation

- Accuracy is one simple metric that can be used to evaluate a classifier.
- Accuracy measures the percentage of inputs in the test set that the classifier correctly labelled.
  - A classifier that predicts the correct labels 60 times in a test set containing 100 names would have an accuracy of 60/100 = 60%.
  - The function **`nltk.classify.accuracy()`**can be used to measure accuracy.
- If the test set is not balanced, accuracy is not a very good measure :
  - The frequencies of the individual class labels in the test set are ignored.

Intrinsic Evaluation

- Precision and Recall are other performance measures that are used instead of accuracy in tasks such as information retrieval.

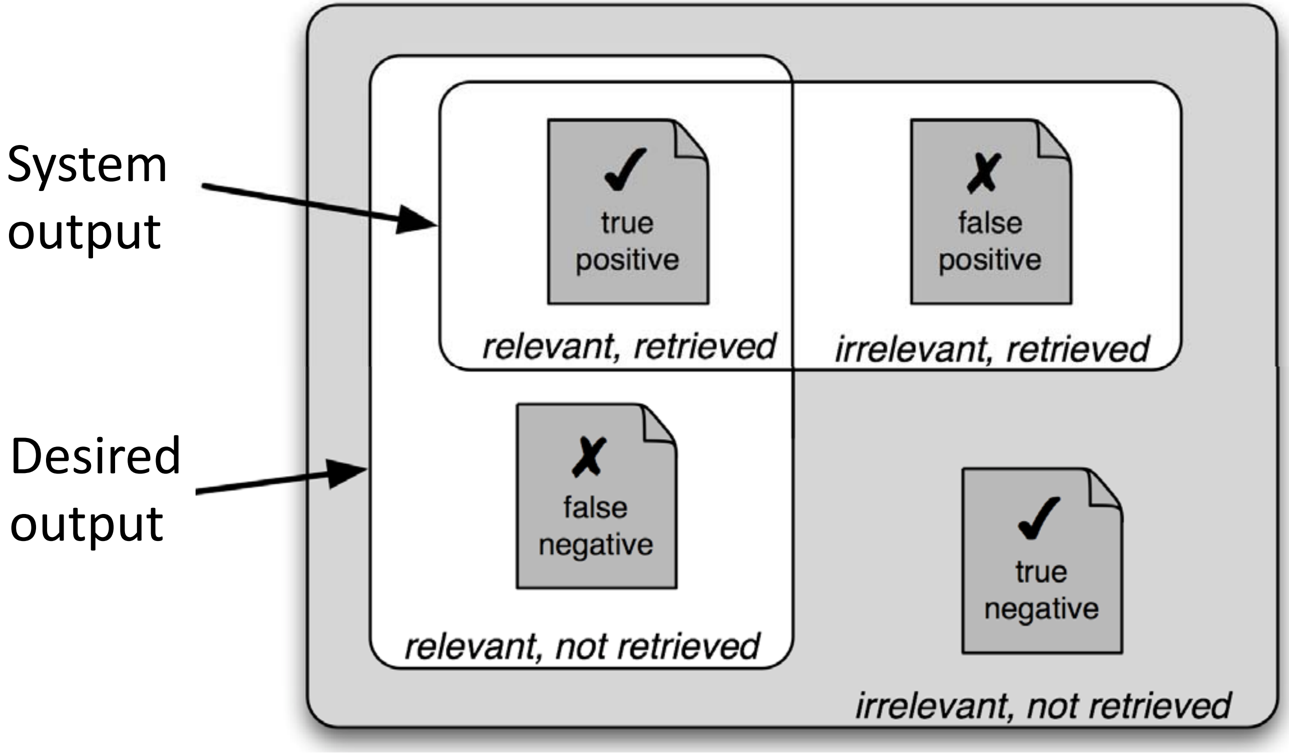
Intrinsic Evaluation

- Precision and Recall are other performance measures that are used instead of accuracy in tasks such as information retrieval.
- Precision indicates how many of the items that are identified were relevant (or expected).
- Recall, however, indicates how many of the all relevant items we could identify.

Intrinsic Evaluation

- Precision and Recall are other performance measures that are used instead of accuracy in tasks such as information retrieval.
- Precision indicates how many of the items that are identified were relevant (or expected).
- Recall, however, indicates how many of the all relevant items we could identify.
- The goal is a high precision and high recall.
- The F-Measure (or F-Score), is thus used to combine these two measures.

Intrinsic Evaluation



Precision = *TP/(TP+FP)*

Recall = *TP/(TP+FN)*

F-Score = (2 × *Precision* × *Recall*) / (*Precision* + *Recall*).

Confusion Matrices

- We can extend the idea of evaluation using precision and recall for a binary class classification to the evaluation of a multi-class classification using confusion matrices.

Confusion Matrices

- We can extend the idea of evaluation using precision and recall for a binary class classification to the evaluation of a multi-class classification using confusion matrices.
- A confusion matrix is a table where each cell [*i*,*j*] indicates how often label *j* was predicted when the correct label was *i*.
- Thus, the diagonal entries (i.e., cells [ii]) indicate labels that were correctly predicted.
- You can use **`nltk.ConfusionMatrix`** to generate a confusion matrix.

Confusion Matrices

		Predicted		
		Cat	Dog	Rabbit
Actual class	Cat	5	3	0
	Dog	2	3	1
	Rabbit	0	2	11

Confusion Matrices

		Predicted		
		Cat	Dog	Rabbit
Actual class	Cat	5	3	0
	Dog	2	3	1
	Rabbit	0	2	11

Good at catching Rabbits!

Confusion Matrices

Not so good with cats and dogs!

		Predicted		
		Cat	Dog	Rabbit
Actual class	Cat	5	3	0
	Dog	2	3	1
	Rabbit	0	2	11

Confusion Matrices

		Predicted		
		Cat	Dog	Rabbit
Actual class	Cat	5	3	0
	Dog	2	3	1
	Rabbit	0	2	11

But very good in distinguishing Rabbits from Cats

Confusion Matrices

		Predicted			
		Cat	Dog	Rabbit	
Actual class	Cat	**5**	3	0	**8**
	Dog	2	**3**	1	**6**
	Rabbit	0	2	**11**	**13**
		7	**8**	**12**	

Confusion Matrices

		Predicted				
		Cat	Dog	Rabbit		
Actual class	Cat	**5**	3	0	**8**	**0.625**
	Dog	2	**3**	1	**6**	**0.5**
	Rabbit	0	2	**11**	**13**	**0.846154**
		7	**8**	**12**		
		0.71	**0.38**	**0.916666667**		

Cross-Validation

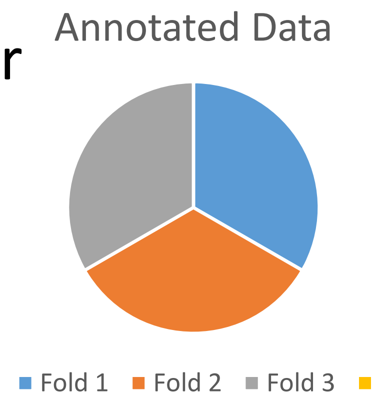
- In order to evaluate a classifier, we must reserve a portion of the annotated data for the test set.
- Test set must be big enough to have an accurate.
- However, if a limited amount of annotated data is available, making the test set larger is not plausible.
- Larger test dataset means making the training set smaller, which impact the classifier performance significantly.
- One solution to solve this problem is to perform **cross-validation.**

Cross-Validation

- Cross-Validation technique consists of multiple evaluations on different test sets and the combination of the obtained scores.
- We first divide the original corpus into *n* subsets called **folds**.
- For each of these folds, we train a model using all of the data *except* the data in that fold, and then test that model on the fold.
- The obtained scores from the individual folds (which might be too small to give accurate evaluation scores on their own) are combined to report an evaluation score.
- The final score is thus based on a large amount of data, and is therefore reliable.

Cross-Validation

- Additionally, cross-validation allows us to examine how widely the performance varies across different training sets:
  - If scores are similar for all *n* training sets, then we can be fairly confident that the score is accurate.
  - If scores vary widely across the *n* training sets, then we should probably be sceptical about the accuracy of the evaluation score.
- Remember, each fold must be balance and diverse (i.e. a fair representative of the real data).



A closer look at learning techniques

- It's possible to treat learning methods as black boxes:
    - Simply train and use a learning technique without understanding how it work.

A closer look at learning techniques

- It's possible to treat learning methods as black boxes:
  - Simply train and use a learning technique without understanding how it work.
- However, an understanding of a learning method can help us in several ways:
  - Selection of appropriate features;
  - Encoding feature values;
  - Setting model parameters;
  - Etc.

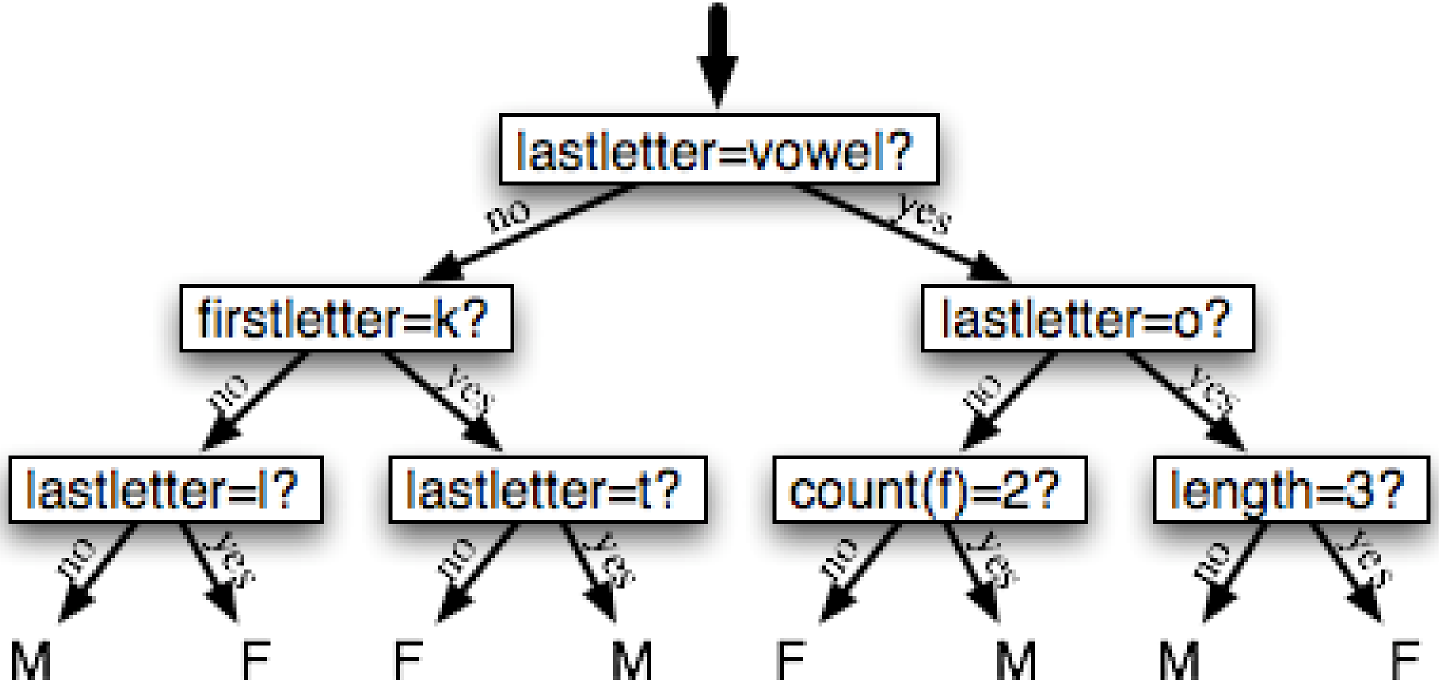
A closer look at learning techniques

- We thus have a slightly closer look at:
  - Decisions trees
  - Naïve Bayes Classifiers
  - Maximum Entropy Classifiers

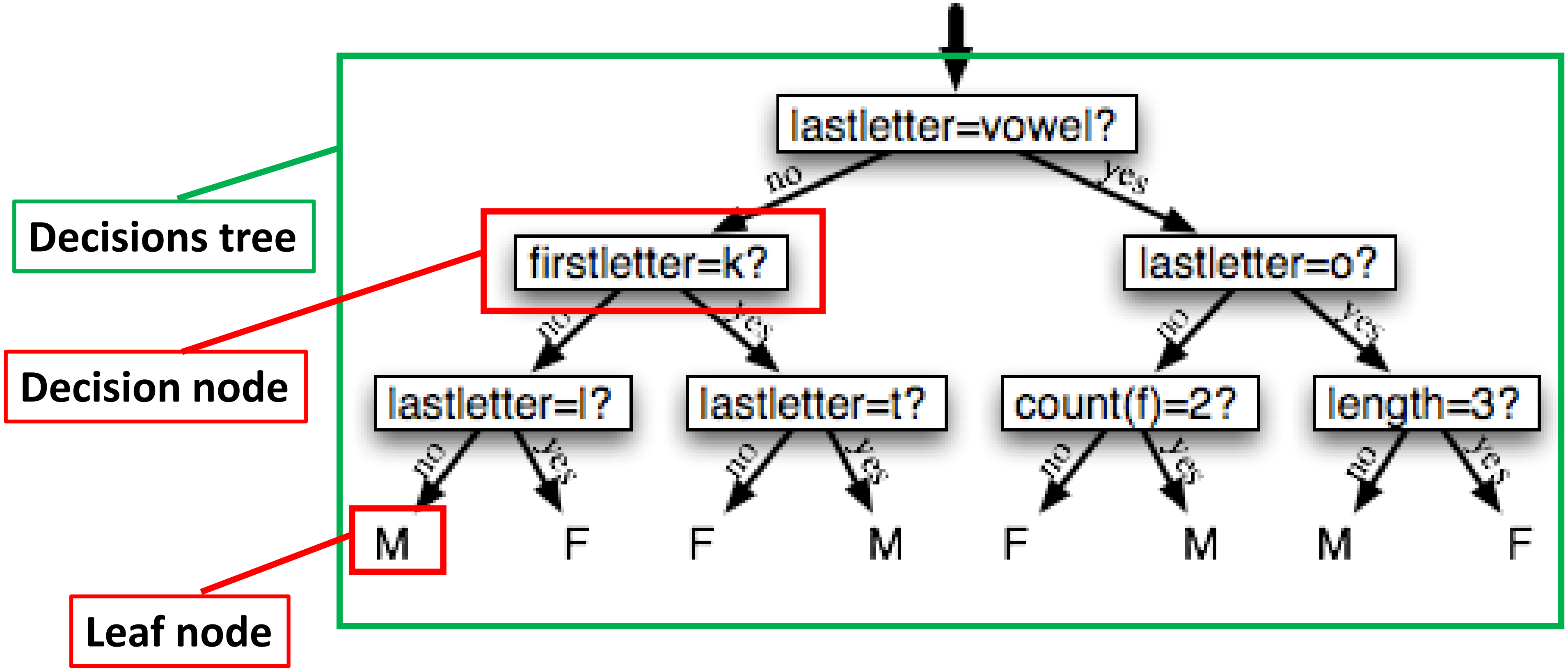
Decisions trees

- A **decision tree** is a flowchart that selects labels for input values.
- This flowchart consists of **decision nodes** (check feature values), and **leaf nodes** (assign labels).
- To choose the label for an input value, we begin at the initial decision node (known as **root node)**.

Decisions trees



Decisions trees


Decisions tree

Decision node

Leaf node

Decisions trees

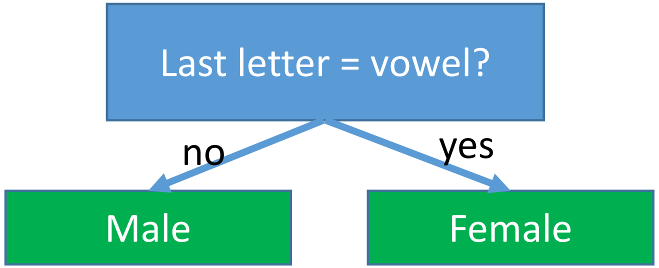
- It is very clear how to use decision trees, but how to build one?
    - What is the root node?
    - What are the decision nodes?
    - How to organize other decision nodes?
    - How many levels these tree must have?

Decisions trees

- It is very clear how to use decision trees, but how to build one?
    - What is the root node?
    - What are the decision nodes?
    - How to organize other decision nodes?
    - How many levels these tree must have?
- We focus on a simplified case of “decision stump”:
    - A decision tree with a single node that decides how to classify inputs based on a single feature;
    - That is, we have one leaf for each possible feature value.

Decisions trees

- It is very clear how to use decision trees, but how to build one?
  - What is the root node?
  - What are the decision nodes?
  - How to organize other decision nodes?
  - How many levels these tree must have?



- We focus on a simplified case of “decision stump”:
  - A decision tree with a single node that decides how to classify inputs based on a single feature;
  - That is, we have one leaf for each possible feature value.

Building a decision stump

- As expected, we must first decide which feature should be used.
- Afterwards, the simplest method is to:
  - Build a decision stump for each possible feature;
  - For each feature, assign a label to each leaf based on the most frequent label for the selected examples in the training set.
  - Choose the one that achieves the highest accuracy as the decision stump!

Building a decision stump

- As expected, we must first decide which feature should be used.
- Afterwards, the simplest method is to:
  - Build a decision stump for each possible feature;
  - For each feature, assign a label to each leaf based on the most frequent label for the selected examples in the training set.
  - Choose the one that achieves the highest accuracy as the decision stump!

Note alternative processes are available!

Building a decision tree

- Now that we can build a decision stump, we can build a larger decision tree:
  - Build the decision stumps;
  - Select the overall best decision stump for the classification task;
  - Replace leaves that do not achieve sufficient accuracy with new decision stumps.
- We can use the notion of Information Gain to choose the most informative feature!

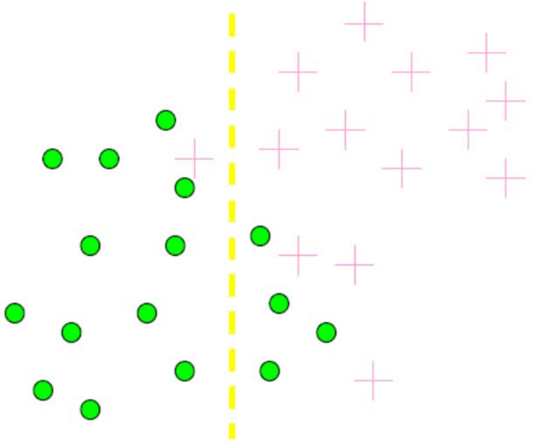
Entropy and Information Gain

- **Information gain** tells us how important a feature is.
- We can use this measure to organize/order features in a decision tree.
- Vice versa, **entropy** measure how disorganized the original set of input values are.

Entropy and Information Gain

Which test is more informative?

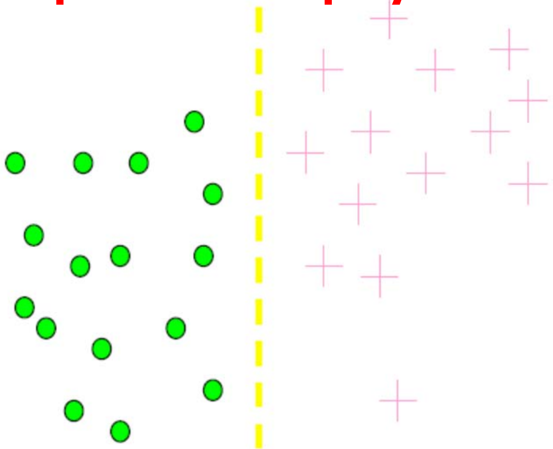
Split over whether saving balance exceeds 100K



Saving over 100K

Saving Less or Equal 100K

Split over whether person is employed



Employed

Unemployed

Acknowledgement: Content and Example for this introduction are taken from Shapiro and Stockman (2001)

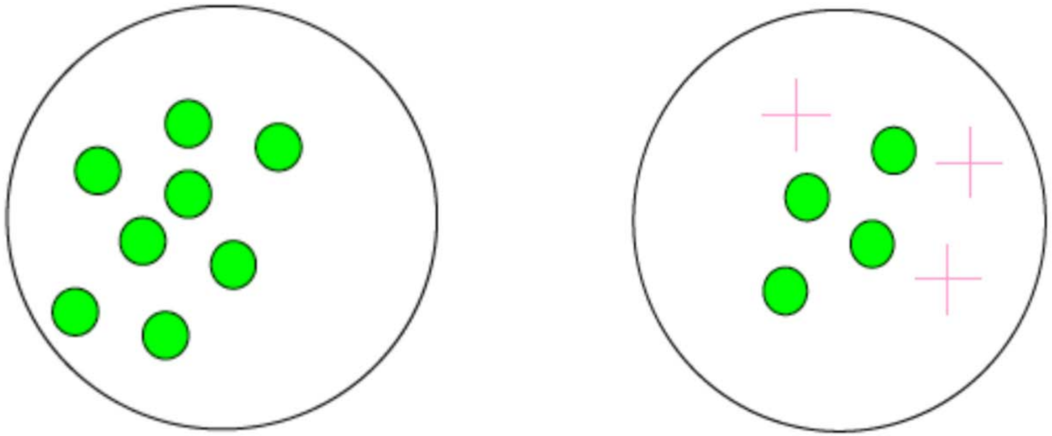
Entropy and Information Gain

- Entropy measures the level of impurity in a group of instances

Entropy and Information Gain

- Entropy measures the level of impurity in a group of instances

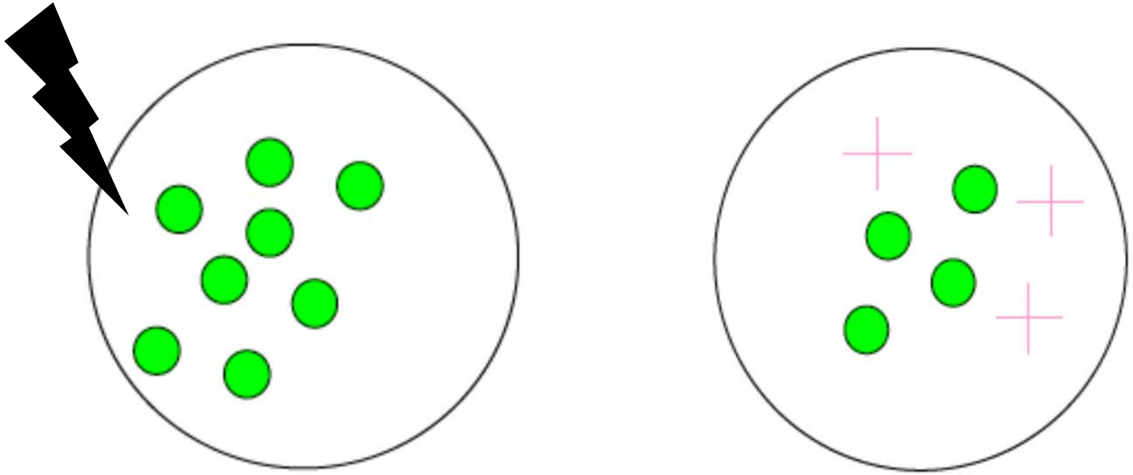
Which set is pure (not mixed)?



Entropy and Information Gain

- Entropy measures the level of impurity in a group of instances

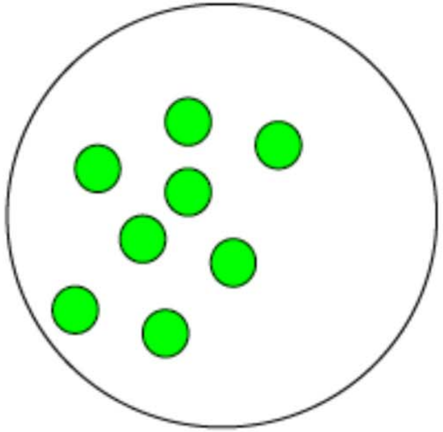
Which set is pure (not mixed)?



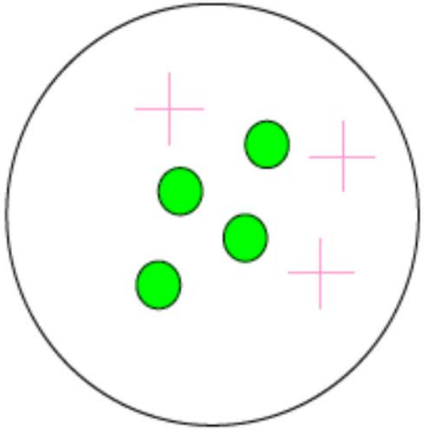
Entropy and Information Gain

- Entropy measures the level of impurity in a group of instances

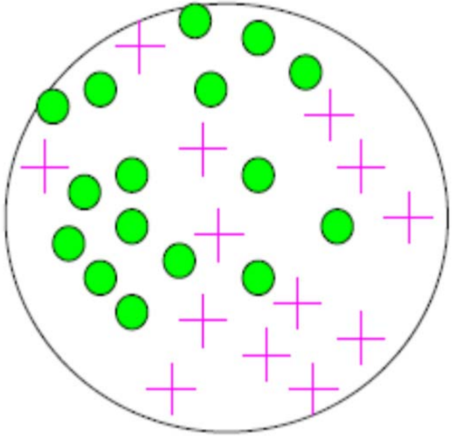
Minimum impurity



Less impure



Very impure



Entropy and Information Gain

- Entropy is a common mathematical tool to measure impurity:

$$\mathrm{E} = \Sigma_i - p_i \log_2 p_i$$

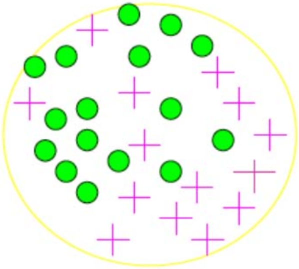
where p_i is the probability for class *i.*

Entropy and Information Gain

- Entropy is a common mathematical tool to measure impurity:

$$\mathrm{E} = \Sigma_i - p_i \log_2 p_i$$

where p_i is the probability for class *i*



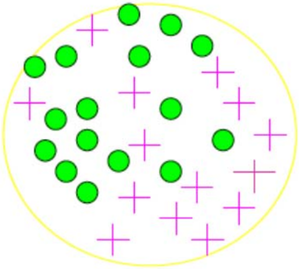
$$\mathrm{E} = -\frac{16}{30} \times \log_2 \frac{16}{30} - \frac{14}{30} \times \log_2 \frac{14}{30} = 0.999$$

Entropy and Information Gain

- Entropy is a common mathematical tool to measure impurity:

$$\mathrm{E} = \Sigma_i - p_i \log_2 p_i$$

where p_i is the probability for class *i*



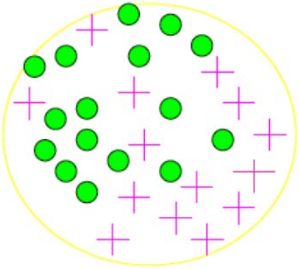
$$\mathrm{E} = \boxed{-\frac{16}{30} \times \log_2 \frac{16}{30}} - \frac{14}{30} \times \log_2 \frac{14}{30} = 0.999$$

Entropy and Information Gain

- Entropy is a common mathematical tool to measure impurity:

$$\mathrm{E} = \Sigma_i - p_i \log_2 p_i$$

where p_i is the probability for class *i*



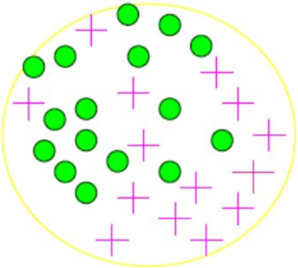
$$\mathrm{E} = -\frac{16}{30} \times \log_2 \frac{16}{30} \boxed{-\frac{14}{30} \times \log_2 \frac{14}{30}} = 0.999$$

Entropy and Information Gain

- Entropy is a common mathematical tool to measure impurity:

$$\mathrm{E} = \Sigma_i - p_i \log_2 p_i$$

where p_i is the probability for class *i*



$$\mathrm{E} = -\frac{16}{30} \times \log_2 \frac{16}{30} - \frac{14}{30} \times \log_2 \frac{14}{30} = 0.999$$

- It comes from information theory.

Developed by Shannon for code-breaking and secure telecommunications.
http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf

Entropy and Information Gain

- Entropy is a common mathematical tool to measure impurity:

$$\mathrm{E} = \Sigma_i - p_i \log_2 p_i$$

where p_i is the probability for class *i.*

- The higher the entropy the more the information content.

Entropy and Information Gain

- Entropy is a common mathematical tool to measure impurity:

$$\mathrm{E} = \Sigma_i - p_i \log_2 p_i$$

where p_i is the probability for class *i.*

- The higher the entropy the more the information content.
    - For a set of two things/labels, when do we have the highest entropy?

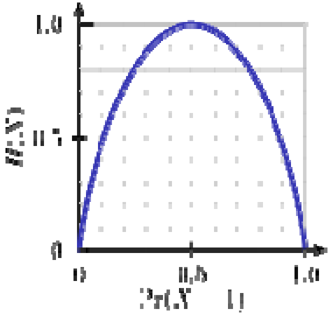
Entropy and Information Gain

- Entropy is a common mathematical tool to measure impurity:

$$\mathrm{E} = \Sigma_i - p_i \log_2 p_i$$

where p_i is the probability for class *i.*

- The higher the entropy the more the information content.
  - For a set of two things/labels, when do we have the highest entropy?



Entropy and Information Gain

- Entropy is a common mathematical tool to measure impurity:

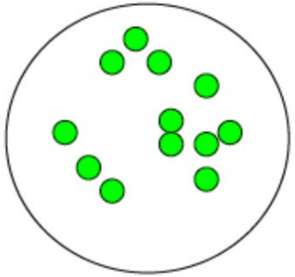
$$\mathrm{E} = \Sigma_i - p_i \log_2 p_i$$

where p_i is the probability for class *i.*

- The higher the entropy the more the information content.
- What does entail for learning from examples?!

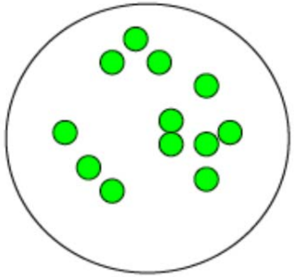
Entropy and Information Gain

- Let’s concentrate on a binary class problem:
  - What is the entropy of a group in which all examples belong to the same class?



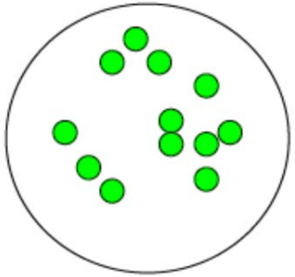
Entropy and Information Gain

- Let’s concentrate on a binary class problem:
  - What is the entropy of a group in which all examples belong to the same class?
    - The answer is E $= -1 \log_2 1 = 0$

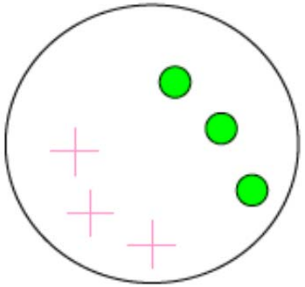


Entropy and Information Gain

- Let's concentrate on a binary class problem:
  - What is the entropy of a group in which all examples belong to the same class?
    - The answer is E $= -1\log_2 1 = 0$



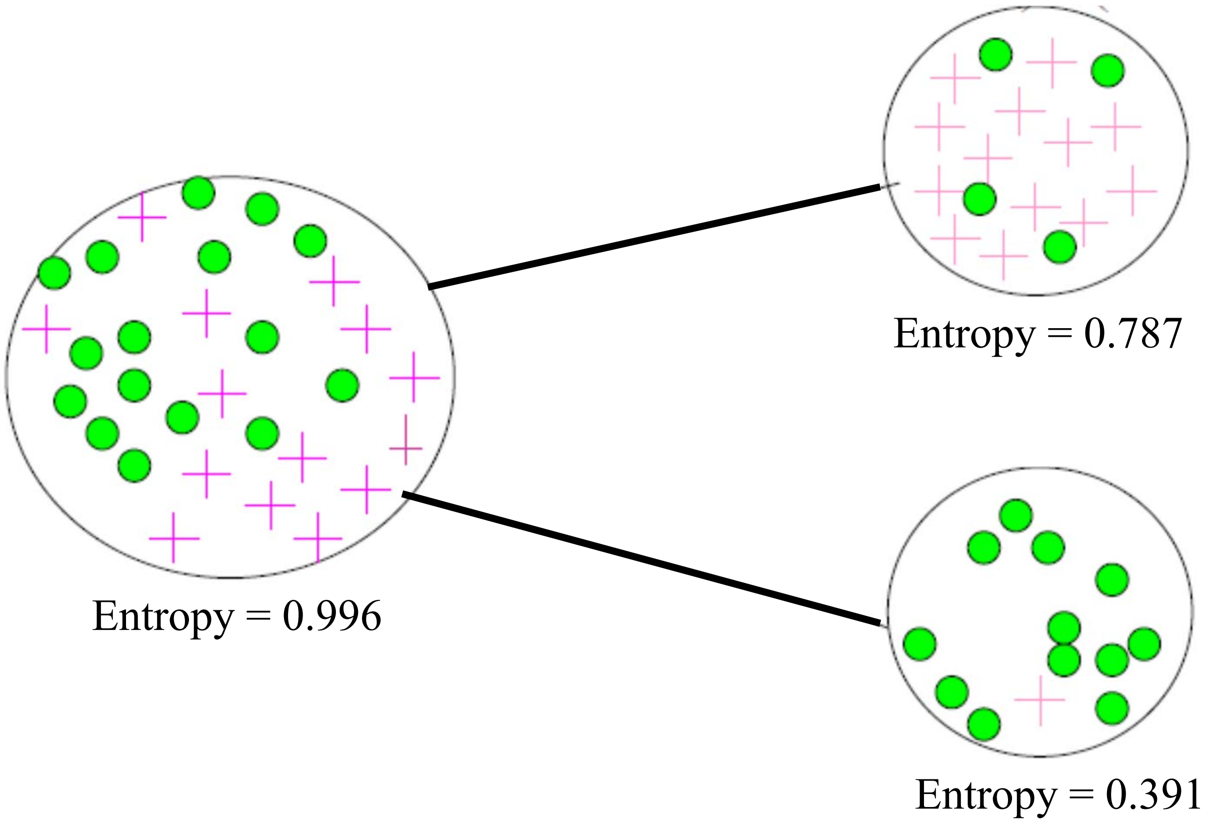
  - What is the entropy of a group in which with 50% in either class?
    - The answer is E $= -0.50\log_2 0.50\ - 0.50\log_2 0.50 = 1$



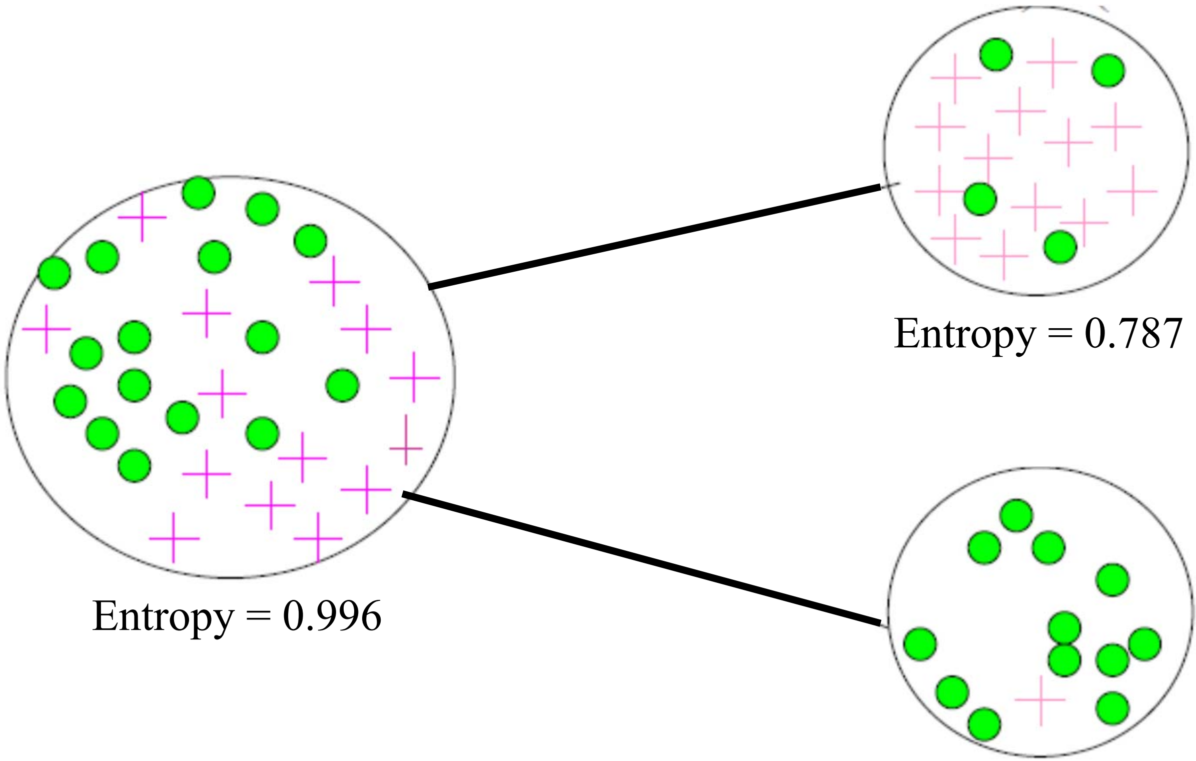
Entropy and Information Gain

- We would like to find which feature in a given set of features is most discriminative between classes to be learned.
- Information gain tell can tell us how important a given feature is!
- We use that to decide the order of features/decision nodes in a decision tree!

Entropy and Information Gain



Entropy and Information Gain



Entropy = 0.996

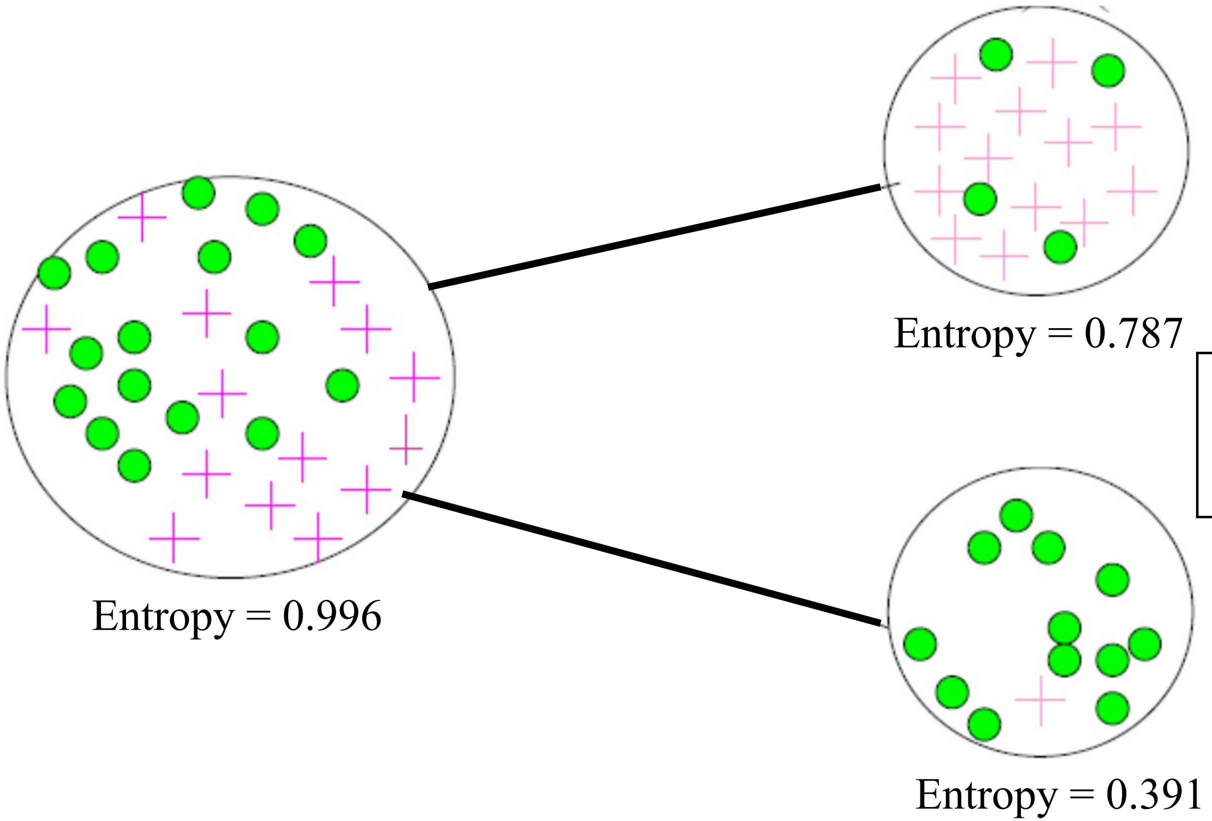
Entropy = 0.787

Entropy = 0.391

Weighted Average Entropy for Children =

$$\frac{17}{30} \times 0.787 + \frac{13}{30} \times 0.391 = 0.615$$

Entropy and Information Gain



Entropy = 0.996

Entropy = 0.787

Entropy = 0.391

Weighted Average Entropy for Children =

$$\frac{17}{30} \times 0.787 + \frac{13}{30} \times 0.391 = 0.615$$

Information Gain =
0.996 – 0.615 = 0.38

Entropy-Based Decision Tree Construction

- To construct a decision tree we must answer two questions (reminder):
  - What features and what values must be used?
- ID3 (Iterative Dichotomiser 3) by Quinlan exploits information gain to construct a decision tree!
- Use entropy to calculate information gain and then answer the above questions.

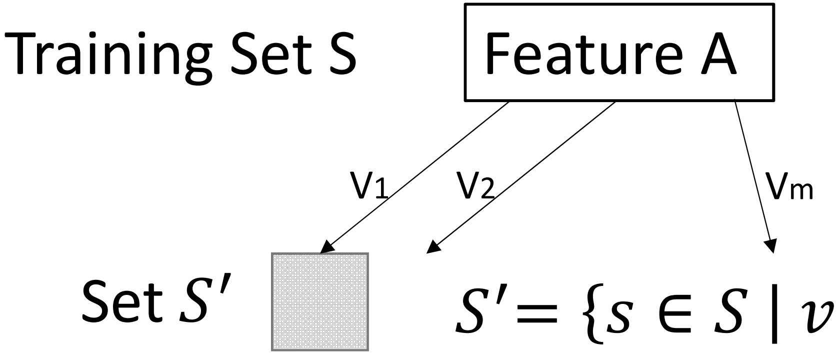
Entropy-Based Decision Tree Construction

Training Set S

Feature A

Find feature A that has the highest information gain over the training dataset and make it the root of the tree

Entropy-Based Decision Tree Construction



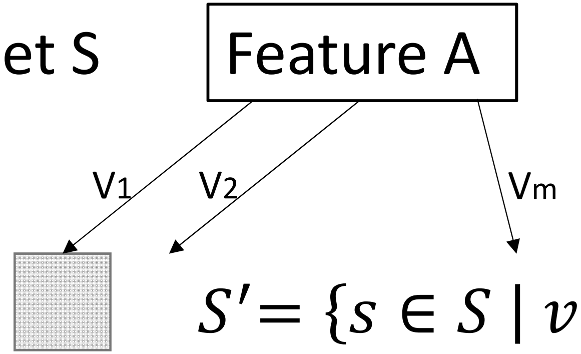
$S' = \{s \in S \mid value(A) = v_1\}$

Find feature A that has the highest information gain over the training dataset and make it the root of the tree

Construct child nodes for each value of A, i.e. find a subset of training samples of which A has certain value.

Entropy-Based Decision Tree Construction

Training Set S



Find feature A that has the highest information gain over the training dataset and make it the root of the tree

Set S'

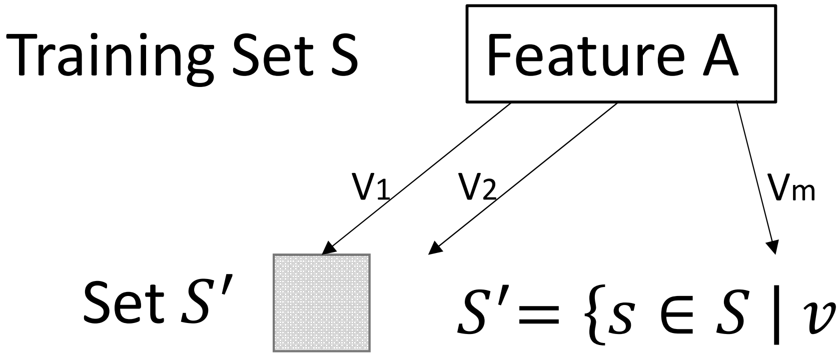
$$S' = \{s \in S \mid value(A) = v_1\}$$

Construct child nodes for each value of A, i.e. find a subset of training samples of which A has certain value.

Repeat recursively until certain conditions are met!

Entropy-Based Decision Tree Construction

Find feature A that has the highest information gain over the training dataset and make it the root of the tree



$$S' = \{s \in S \mid value(A) = v_1\}$$

Construct child nodes for each value of A, i.e. find a subset of training samples of which A has certain value.

Until all the elements of the S' are the same, or there is no more feature to be used!

Entropy-Based Decision Tree Construction

```
>>> import math def entropy(labels):
      freqdist = nltk.FreqDist(labels)
      probs = [freqdist.freq(l) for l in freqdist]
      return -sum(p * math.log(p,2) for p in probs)
```

Entropy-Based Decision Tree Construction

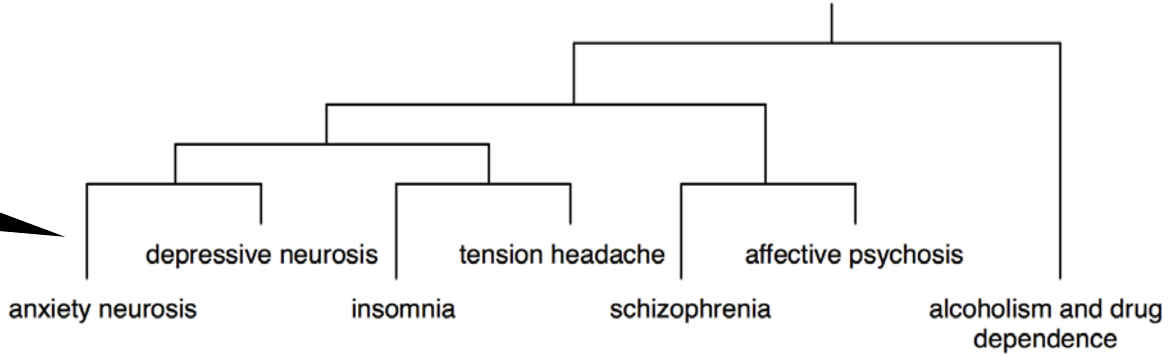
```
>>> import math def entropy(labels):
      freqdist = nltk.FreqDist(labels)
      probs = [freqdist.freq(l) for l in freqdist]
      return -sum(p * math.log(p,2) for p in probs)


>>> print(entropy(['male', 'male', 'male', 'male']))
0.0
>>> print(entropy(['male', 'female', 'male', 'male']))
0.811
```

Decision Trees: a summary

- Some of Advantages:
  - Decision trees are easy to understand and interpret.
  - Decision trees are very well suited for hierarchical classification task. distinctions can be made.

Phylogeny trees



Decision Trees: a summary

- Some of Disadvantages:
    - Decision trees are prone to overfitting, specifically at lower decision nodes (can you explain why?); although some solutions are available:
        - Stop dividing nodes once the amount of training data becomes too small.
        - Using **pruning, i.e.** reduces the size of decision trees by removing nodes that do not improve performance on a dev-test.

Decision Trees: a summary

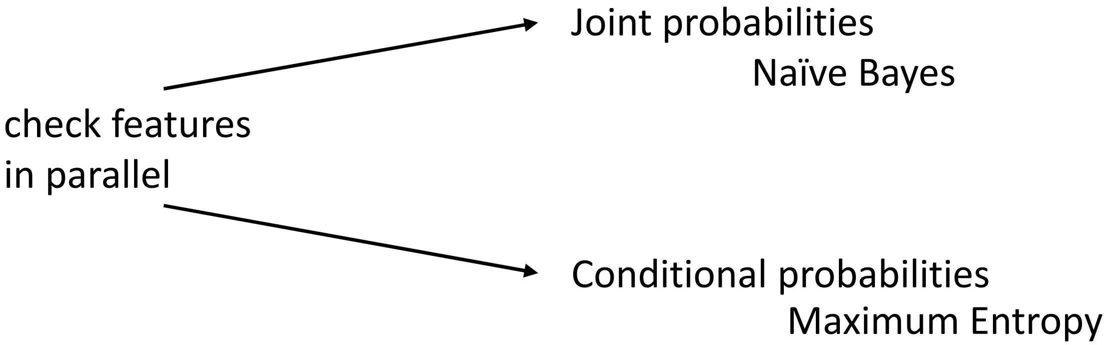
- Some of Disadvantages:
  - Decision trees are prone to overfitting, specifically at lower decision nodes (can you explain why?); although some solutions are available:
    - Stop dividing nodes once the amount of training data becomes too small.
    - Using **pruning, i.e.** reduces the size of decision trees by removing nodes that do not improve performance on a dev-test.
  - They force features to be checked in a particular order, which often results in huge decision trees:
    - E.g. in document classification some words are strong indicative of labels independently. These words are pushed to the bottom of the tree and repeated in different branch (an exponential growth!).

Alternatives to Decision Trees

- As discussed, decision trees check features in a specific order, one at a time and that is not desirable!
- We can however check features in parallel!
- To do so, we can use joint probabilities as well as conditional probabilities!

Alternatives to Decision Trees

- As discussed, decision trees check features in a specific order, one at a time and that is not desirable!
- We can however check features in parallel!
- To do so, we can use joint probabilities as well as conditional probabilities!



check features in parallel

Joint probabilities

Naïve Bayes

Conditional probabilities

Maximum Entropy

Joint Probability & Conditional Probability

- To review these concepts, we start with the famous “Sex, Math and English” example

	Math	**English**	**Total**
Female	1	17	18
Male	37	20	57
Total	38	37	75

Acknowledgement: Slides for this section are based on materials from Mark Liberman and Stephen Isard (see http://www.ling.upenn.edu/courses/cogs501)

Joint Probability & Conditional Probability

- To review these concepts, we start with the f nd English" example:

probability of picking a female math professor!

	Math	English	Total
Female	1	17	18
Male	37	20	57
Total	38	37	75

Number of enrolled students by their gender

$P(\text{female, Math}) = .013$

$P(\text{female, English}) = .227$

$P(\text{male, Math}) = .493$

$P(\text{male, English}) = .267$

"joint probabilities", which are symmetric!

	Math	English	Total
Female	.013	.227	.240
Male	.493	.267	.760
Total	.506	.494	1.00

"joint distribution" of sex and department: probabilities of picking a female or male for each subject in the whole set

Joint Probability & Conditional Probability

- Let’s start with a little game, we want to guess the sex:

	Math	English	Total
Female	1	17	18
Male	37	20	57
Total	38	37	75

P(male) = 57/75 = .760

P(male, math) = 37/75 = .493

P(male | math) = 37/38 = .974

P(math | male) = 37/57 = .649

Joint Probability & Conditional Probability

- Let’s start with a little game, we want to guess the sex:

	Math	English	Total
Female	1	17	18
Male	37	20	57
Total	38	37	75

P(male) = 57/75 = .760

The probability of bumping into a male professor

Joint Probability & Conditional Probability

- Let’s start with a little game, we want to guess the sex:

	Math	English	Total
Female	1	17	18
Male	37	20	57
Total	38	37	75

P(male, math) = 37 / 75 = .493

The probability of bumping into a male math professor

Joint Probability & Conditional Probability

- Let’s start with a little game, we want to guess the sex:

	Math	English	Total
Female	1	17	18
Male	37	20	57
Total	38	37	75

P(male | math) = 37/38 = .974

The probability of bumping into a male professor in the math department!

Joint Probability & Conditional Probability

- Let’s start with a little game, we want to g

The probability of bumping into a math professor in the male employees of the department! Formally, read it as “the probability of *male* given *math”*

	Math	English	Total
Female	1	17	18
Male	37	20	57
Total	38	37	75

P(math | male) = 37/57 = .649

Joint Probability & Conditional Probability

- Let’s start with a little game, we want to guess the sex:

	Math	English	Total
Female	1	17	18
Male	37	20	57
Total	38	37	75

P(male) = 57/75 = .760

P(male, math) = 37/75 = .493

P(male | math) = 37/38 = .974

P(math | male) = 37/57 = .649

These probabilities are different because they represent different assumptions!

Joint Probability & Conditional Probability

- Please think about these numbers, write down the table and calculate the numbers again!

Joint Probability & Conditional Probability

- Would it be possible to calculate conditional probabilities from the joint distributions?!

	Math	English	Total
Female	.013	.227	.240
Male	.493	.267	.760
Total	.506	.494	1.00

Joint Probability & Conditional Probability

- Would it be possible to calculate conditional probabilities from the joint distributions?! Let’s say **P(male | math)**

	Math	**English**	**Total**
Female	.013	.227	.240
Male	.493	.267	.760
Total	.506	.494	1.00

Joint Probability & Conditional Probability

- Would it be possible to calculate conditional probabilities from the joint distributions?! Let’s say **P(male | math)**

	Math	English	Total
Female	.013	.227	.240
Male	.493	.267	.760
Total	.506	.494	1.00

Would you say P(male | math) = 493/506 = .974?

Joint Probability & Conditional Probability

- Would it be possible to calculate conditional probabilities from the joint distributions?! Let's say **P(male | math)**

	Math	English	Total
Female	.013	.227	.240
Male	.493	.267	.760
Total	.506	.494	1.00

Would you say P(male | math) = 493/506 = .974?

P(male | math) = P(male, math) / P(math)

Joint Probability & Conditional Probability

- Would it be possible to calculate conditional probabilities from the joint distributions?! Let's say **P(male | math)**

	Math	English	Total
Female	.013	.227	.240
Male	.493	.267	.760
Total	.506	.494	1.00

Would you say P(male | math) = 493/506 = .974?

P(male | math) = P(male, math) / P(math)

P(A | B) = P(A, B) / P(B)

Bayes’ Theorem

- So far we know that:

[1] P(A | B) = P(A, B) / P(B)

[2] P(B | A) = P(B, A) / P(A)

[3] P(A, B) = P(B, A)

Bayes’ Theorem

- So far we know that:

[1] $P(A \mid B) = P(A, B) / P(B)$

[2] $P(B \mid A) = P(B, A) / P(A)$

[3] $P(A, B) = P(B, A)$

Using simple math [1] and [2] can be written as:

$P(A \mid B)\, P(B) = P(A, B)$

$P(B \mid A)\, P(A) = P(B, A)$

Bayes’ Theorem

- So far we know that:

$$[1]\ P(A \mid B) = P(A, B) / P(B)$$

$$[2]\ P(B \mid A) = P(B, A) / P(A)$$

$$[3]\ P(A, B) = P(B, A)$$

Using simple math [1] and [2] can be written as:

$$P(A \mid B)\, P(B) = P(A, B)$$

$$P(B \mid A)\, P(A) = P(B, A)$$

Using [3] we arrive to:

$P(A \mid B)\, P(B) = P(B \mid A)\, P(A)$ and thus $P(A \mid B) = P(B \mid A)\, P(A) / P(B)$

Bayes' Theorem

Why is it a big deal?!

- Think about the relationship between evidence and theory, or feature and class label!
- Suppose we have a set of class labels C_1, C_2, ..., and we've observed some features.
- We'd like to pick the class label that is more likely to be true given our observations.

Why is it a big deal?!

- Think about the relationship between evidence and theory, or feature and class label!
- Suppose we have a set of class labels C_1, C_2, ..., and we've observed some features.
- We'd like to pick the class label that is more likely to be true given our observations.
- This can be formulated by conditional probability **P(C | E),** i.e. the probability of *class label* given *observed features*.

Why is it a big deal?!

- The classification problem can be solved by calculating **P(C | E)** for each class label and picking the maximum P(C | E).
- Now, we can use the Bayes theorem to estimate the P(C | E)

Why is it a big deal?!

- The classification problem can be solved by calculating **P(C | E)** for each class label and picking the maximum P(C | E).
- Now, we can use the Bayes theorem to estimate the P(C | E)
- Let’s solve the problem:

$$P(C \mid E) = P(E \mid C)\, P(C) \,/\, P(E)$$

Can’t we use training data to calculate these probabilities?

Why is it a big deal?!

- The classification problem can be solved by calculating **P(C | E)** for each class label and picking the maximum P(C | E).
- Now, we can use the Bayes theorem to estimate the P(C | E)
- Let’s solve the problem:

$$P(C \mid E) = P(E \mid C)\, P(C) \,/\, P(E)$$

Can’t we use training data to calculate these probabilities?

	Math	**English**	**Total**
Female	1	17	18
Male	37	20	57
Total	38	37	75

Why is it a big deal?!

- The classification problem can be solved by calculating **P(C | E)** for each class label and picking the maximum P(C | E).
- Now, we can use the Bayes theorem to estimate the P(C | E)
- Let's solve the problem:

$$P(C \mid E) = P(E \mid T)\, P(T) / P(E)$$

$$\mathbf{ARGMAX_i\ \ P(E \mid C_i)\, P(C_i)}$$

The best class label accordingly!

Why is it a big deal?!

- The classification problem can be solved by calculating **P(C | E)** for each class label and picking the maximum P(C | E).
- Now, we can use the Bayes theorem to estimate the P(C | E)
- Let’s solve the problem:

$$P(C \mid E) = P(E \mid T)\, P(T) \,/\, P(E)$$

$$\mathbf{ARGMAX_i \quad P(E \mid C_i)\, P(C_i)}$$

The best cla
accordin

Bayes rule is a mathematical formulation of Hermann von Helmholtz statement: what we perceive is our "best guess" given both sensory data and our prior experience.

Naive Bayes Classifiers

- What if we have a set of features instead of one?
    - i.e. if we want to decide class labels by observation made over a feature set?
    - Or, how to calculate $\mathbf{P(C_i \mid E_1 \ldots E_n)}$?!

Naive Bayes Classifiers

- What if we have a set of features instead of one?
  - i.e. if we want to decide class labels by observation made over a feature set?
  - Or, how to calculate $\mathbf{P(C_i \mid E_1 \ldots E_n)}$?!
- Let’s use the Bayes theorem and the **chain rule**:

$$
\begin{aligned}
P(C_i \mid E_1 \ldots E_n) &= P(C_i)\ P(E_1 \ldots E_n \mid C_i) \\
&= P(C_i)\ P(E_1 \mid C_i)\ P(E_2 \ldots E_n \mid C_{i,}\ E_1) \\
&= P(C_i)\ P(E_1 \mid C_i)\ P(E_2 \mid C_{i,}\ E_1)\ P(E_3 \ldots E_n \mid C_{i,}\ E_1\ ,\ E_2) \\
&= P(C_i)\ P(E_1 \mid C_i)\ P(E_2 \mid C_{i,}\ E_1) \ldots P(E_n \mid C_{i,}\ E_{1,}\ E_{2,\ \ldots,}\ E_{n-1})
\end{aligned}
$$

Naive Bayes Classifiers

- What if we have a set of features instead of one?
  - i.e. if we want to decide class labels by observation made over a feature set?
  - Or, how to calculate $\mathbf{P(C_i \mid E_1 \ldots E_n)}$?!
- Let's use the Bayes theorem and the **chain rule**:

Here naivety can plays a role!!!

$$
\begin{aligned}
P(C_i \mid E_1 \ldots E_n) &= P(C_i)\, P(E_1 \ldots E_n \mid C_i) \\
&= P(C_i)\, P(E_1 \mid C_i)\, P(E_2 \ldots E_n \mid C_{i,} E_1) \\
&= P(C_i)\, P(E_1 \mid C_i)\, P(E_2 \mid C_{i,} E_1)\, P(E_3 \ldots E_n \mid C_i, E_1 , E_2) \\
&= P(C_i)\, P(E_1 \mid C_i)\, P(E_2 \mid C_{i,} E_1) \ldots P(E_n \mid C_{i,} E_{1,} E_{2, \ldots,} E_{n-1})
\end{aligned}
$$

Naive Bayes Classifiers

- What if we have a set of features instead of one?
  - i.e. if we want to decide class labels by observation made over a feature set?
  - Or, how to calculate **$P(C_i \mid E_1 \ldots E$**
- Let’s use the Bayes theorem and th

Let’s assume that each feature **E_i** is conditionally independent of every other **E_j** for **$i \neq j$**

$$\begin{aligned}
\mathbf{P(C_i \mid E_1 \ldots E_n)} &= \mathbf{P(C_i)\ P(E_1 \ldots E_n \mid C_i)} \\
&= \mathbf{P(C_i)\ P(E_1 \mid C_i)\ P(E_2 \ldots E_n \mid C_{i,}\ E_1)} \\
&= \mathbf{P(C_i)\ P(E_1 \mid C_i)\ P(E_2 \mid C_{i,}\ E_1)\ P(E_3 \ldots E_n \mid C_{i,}\ E_1\ ,\ E_2)} \\
&= \mathbf{P(C_i)\ P(E_1 \mid C_i)\ P(E_2 \mid C_{i,}\ E_1) \ldots P(E_n \mid C_{i,}\ E_{1,}\ E_{2,}\ \ldots,\ E_{n-1})}
\end{aligned}$$

Naive Bayes Classifiers

- What if we have a set of features instead of one?
  - i.e. if we want to decide class labels by observation made over a feature set?
  - Or, how to calculate $\mathbf{P(C_i \mid E_1 \ldots E}$
- Let's use the Bayes theorem and th

Let's assume that each feature $\mathbf{E_i}$ is conditionally independent of every other $\mathbf{E_j}$ for **i ≠ j**

$$
\begin{aligned}
\mathbf{P(C_i \mid E_1 \ldots E_n)} &= \mathbf{P(C_i)\ P(E_1 \ldots E_n \mid C_i)} \\
&= \mathbf{P(C_i)\ P(E_1 \mid C_i)\ P(E_2 \ldots E_n \mid C_{i,}\ E_1)} \\
&= \mathbf{P(C_i)\ P(E_1 \mid C_i)\ P(E_2 \mid C_{i,}\ E_1)\ P(E_3 \ldots E_n \mid C_{i,}\ E_1\ ,\ E_2)} \\
&= \mathbf{P(C_i)\ P(E_1 \mid C_i)\ P(E_2 \mid C_{i,}\ E_1) \ldots P(E_n \mid C_{i,}\ E_{1,}\ E_{2,}\ \ldots,\ E_{n-1})}
\end{aligned}
$$

$\mathbf{= P(E_2 \mid C_i)}$

$\mathbf{= P(E_n \mid C_i)}$

Naive Bayes Classifiers

- What if we have a set of features instead of one?
  - i.e. if we want to decide class labels by observation made over a feature set?
  - Or, how to calculate $\mathbf{P(C_i \mid E_1 \ldots E}$
- Let's use the Bayes theorem and th

Let's assume that each feature $\mathbf{E_i}$ is conditionally independent of every other $\mathbf{E_j}$ for $\mathbf{i \neq j}$

$$
\begin{aligned}
P(C_i \mid E_1 \ldots E_n) &= P(C_i)\, P(E_1 \ldots E_n \mid C_i) \\
&= P(C_i)\, P(E_1 \mid C_i)\, P(E_2 \ldots E_n \mid C_{i,} E_1) \\
&= P(C_i)\, P(E_1 \mid C_i)\, P(E_2 \mid C_{i,} E_1)\, P(E_3 \ldots E_n \mid C_{i,} E_1 , E_2) \\
&= P(C_i)\, P(E_1 \mid C_i)\, P(E_2 \mid C_{i,} E_1) \ldots P(E_n \mid C_{i,} E_{1,} E_{2,} \ldots, E_{n-1}) \\
&\approx P(C_i) \prod_{i=1}^{n} P(E_i \mid C)
\end{aligned}
$$

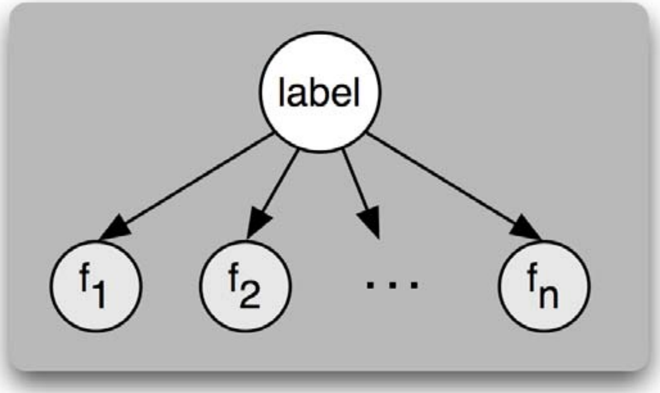
under our naïve assumption (independence assumption)

Naive Bayes Classifiers

- Now that we have a clear picture of what is going on (hopefully), the classification task can be easily formalized by:

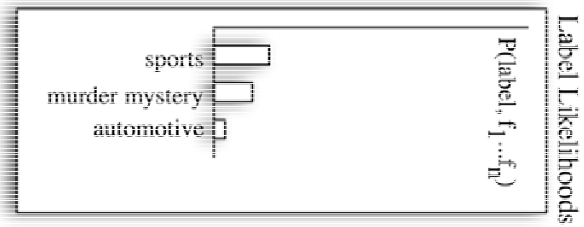
$$Classify(E_1, \ldots, E_n) = \ arg\max_{c} P(C = c) \prod_{i=1}^{n} P(E_i = e_i | C = c)$$

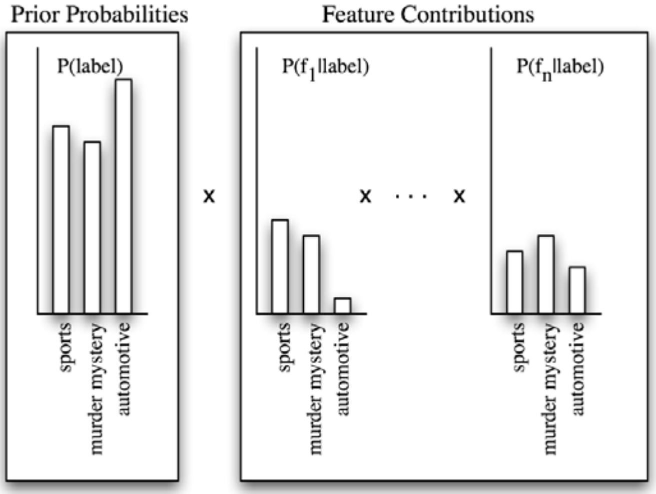
Naive Bayes Classifiers



- Now that we have a clear picture of what is goin classification task can be easily formalized by:

$$Classify(E_1,\dots,E_n) = \underset{c}{argmax}\, P(C=c)\prod^{n} P(E_i=e_i|C=c)$$





Problems with Naive Bayes Classifiers (1)

- What if in a training set a certain feature does not appear with a certain class label, i.e. **P(E|C) = 0**?!
  - The class label likelihood in this situation is 0 and thus regardless of other features, an input will never be assigned this label!
  - Therefore, does a zero count shows an impossible event?!

Problems with Naive Bayes Classifiers (1)

- What if in a training set a certain feature does not appear with a certain class label, i.e. **P(E|C) = 0**?!
  - The class label likelihood in this situation is 0 and thus regardless of other features, an input will never be assigned this label!
  - Therefore, does a zero count shows an impossible event?!
  - For several reasons, the answer can be “NO”!
- Use smoothing to address the limitation stated above:
  - Estimate the value of P(E|C) using techniques other than simple counting:
    - Additive counting (generally a horrible choice!!!), Good-Tring smoothing, …
  - **`nltk.probability`** implements a number of smoothing methods.
  - Also, see Bill MacCartney’s tutorial slides on smoothing (goo.gl/9LCfHE)

Problems with Naive Bayes Classifiers (2)

- What if we want to decide about labels based on multiple features?
  - Put it simply, we question the naïve independence assumption!
  - In other words, using the Naïve Bayes Classifiers we cannot incorporate complex features when making decisions about class labels.

Problems with Naive Bayes Classifiers (2)

- What if we want to decide about labels based on multiple features?
  - Put it simply, we question the naïve independence assumption!
  - In other words, using the Naïve Bayes Classifiers we cannot incorporate complex features when making decisions about class labels.
- For example:
  - Is it reasonable to assume independence between Bavaria and Weißbier?
  - Or, is it reasonable to assume that the features *ends-with(a)* and *ends-with(vowel)* are independent from each other?.

Problems with Naive Bayes Classifiers (2)

- What if we want to decide about labels based on multiple features?
    - Put it simply, we question the naïve independence assumption!
    - In other words, using the Naïve Bayes Classifiers we cannot incorporate complex features when making decisions about class labels.
- For example:
    - Is it reasonable to assume independence between Bavaria and Weißbier?
    - Or, is it reasonable to assume that the features *ends-with(a)* and *ends-with(vowel)* are independent from each other?.
- Bird et. al. describe this as the double counting-counting problem!

Double-Counting Problem

- During training, the contribution of dependant features are computed separately, i.e. the calculation of $P(feature, label)$;
- But when using the classifier to choose labels for new inputs, feature contributions are combined, i.e. we use $\prod P(feature, label)$.

Double-Counting Problem

- During training, the contribution of dependant features are computed separately, i.e. the calculation of $P(feature, label)$;
- But when using the classifier to choose labels for new inputs, feature contributions are combined, i.e. we use $\prod P(feature, label)$.

- Solution:
  - Do not assume the conditional independence between features!
  - Put it simply, this statement means “do not use Naïve Bayes Classifiers”!

Double-Counting Problem

- During training, the contribution of dependant features are computed separately, i.e. the calculation of $P(feature, label)$;
- But when using the classifier to choose labels for new inputs, feature contributions are combined, i.e. we use $\prod P(feature, label)$.

- Solution:
  - Do not assume the conditional independence between features!
  - Put it simply, this statement means “do not use Naïve Bayes Classifiers”!
- Use Maximum Entropy Classifiers (a logistic regression technique).

Maximum Entropy Models

- What is the discarding of the assumption of independence in Naïve Bayes Approach means?

$$\begin{aligned} P(C_i \mid E_1 \ldots E_n) &= P(C_i)\, P(E_1 \ldots E_n \mid C_i) \\ &= P(C_i)\, P(E_1 \mid C_i)\, P(E_2 \ldots E_n \mid C_{i,}\, E_1) \\ &= P(C_i)\, P(E_1 \mid C_i)\, P(E_2 \mid C_{i,}\, E_1)\, P(E_3 \ldots E_n \mid C_{i,}\, E_1 ,\, E_2) \\ &= P(C_i)\, P(E_1 \mid C_i)\, P(E_2 \mid C_{i,}\, E_1) \ldots P(E_n \mid C_{i,}\, E_{1,}\, E_{2,}\, \ldots,\, E_{n-1}) \end{aligned}$$

Maximum Entropy Models

- What is the discarding of the assumption of independence in Naïve Bayes Approach means?

$$\begin{aligned}
P(C_i \mid E_1 \ldots E_n) &= P(C_i)\, P(E_1 \ldots E_n \mid C_i) \\
&= P(C_i)\, P(E_1 \mid C_i)\, P(E_2 \ldots E_n \mid C_{i,} E_1) \\
&= P(C_i)\, P(E_1 \mid C_i)\, P(E_2 \mid C_{i,} E_1)\, P(E_3 \ldots E_n \mid C_{i,} E_1 , E_2) \\
&= P(C_i)\, P(E_1 \mid C_i)\, P(E_2 \mid C_{i,} E_1) \ldots P(E_n \mid C_{i,} E_{1,} E_{2,} \ldots, E_{n-1})
\end{aligned}$$

- Lets continue with Mark Johnson’s slides (goo.gl/k4TYnd)

Next Session: syntax and parsing!