Natural Language Processing

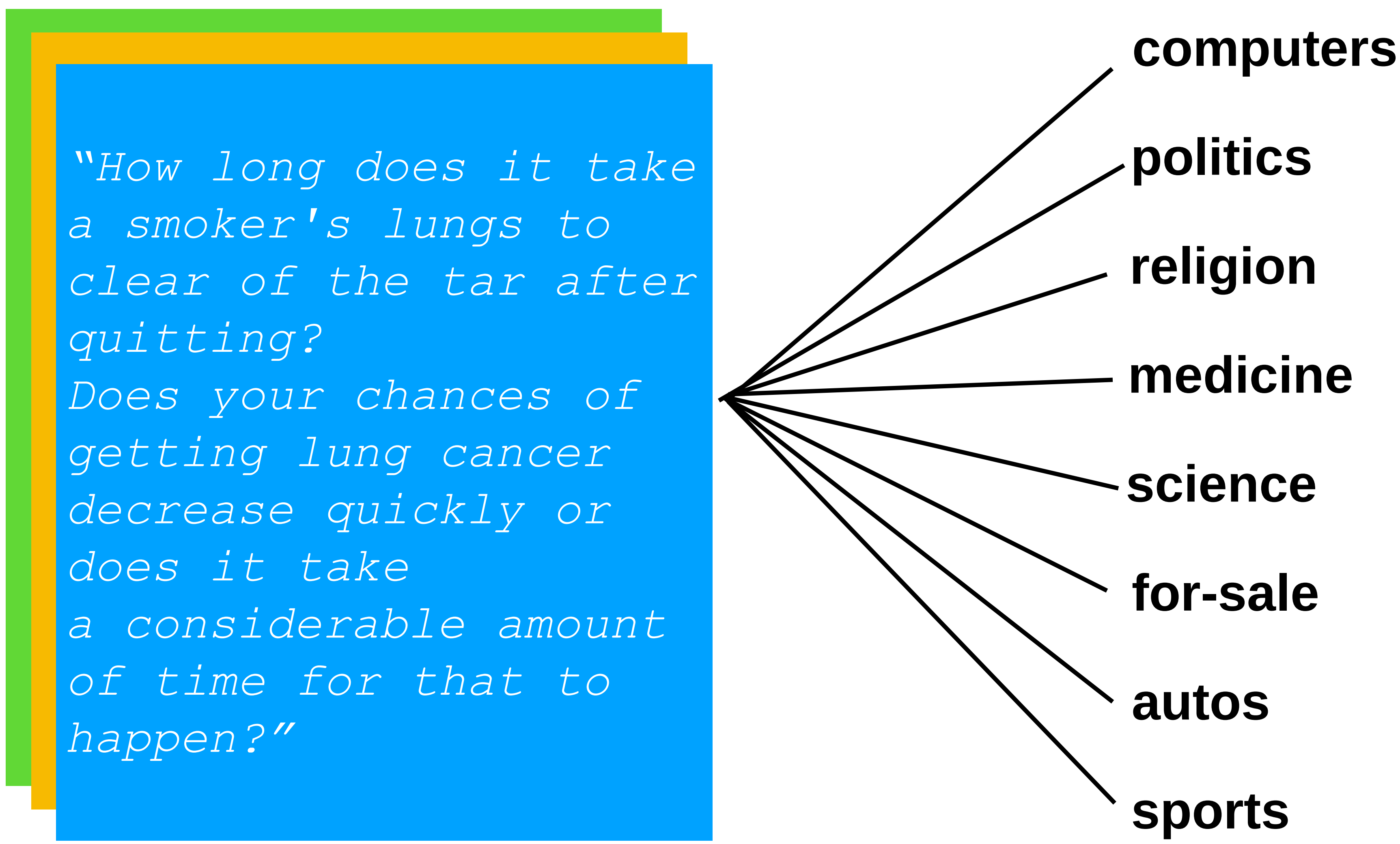
Lectures 2: Language Classification. Probability Review.
Machine Learning Background. Naive Bayes' Classifier.

10/29/2020

COMS W4705
Yassine Benajiba

Text Classification

- Given a representation of some document d, identify which class $c \in C$ the document belongs to.



From the 20-Newsgroups data set:
http://www.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes.html

Text Classification

- Applications:
  - Spam detection.
  - Mood / Sentiment detection.
  - Author identification.
  - Identifying political affiliation.
  - Word Sense Disambiguation.
  - …

Text Classification

- This is a machine learning problem.
  - How do we represent each document?
    (feature representation).
  - Can use different ML techniques.
    - **Supervised ML:** Fixed set of classes C.
      Train a classifier from a set of labeled <document,class> pairs.
      - Discriminative vs. Generative models.
    - Unsupervised ML: Unknown set of classes C.
      Topic modeling.

Types of Feedback

- **Supervised learning:** Given a set of input-output pairs, learn a function that maps inputs to outputs.

- **Unsupervised learning:** Learn patterns in the input without any explicit feedback.
  One typical approach: clustering, identify clusters of input examples.

- **Semi-supervised learning:** Start with a few labeled input/output pairs, then use a lot of unlabeled data to improve.

- **Reinforcement learning:** Start with a *policy* determining the agent's actions. Feedback in the form of reward or punishment.

Supervised Learning

- Given: Training data consisting of training examples $(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_n}, y_n)$, where $\mathbf{x_i}$ is an input example (a d-dimensional vector of attribute values) and y_i is the label.

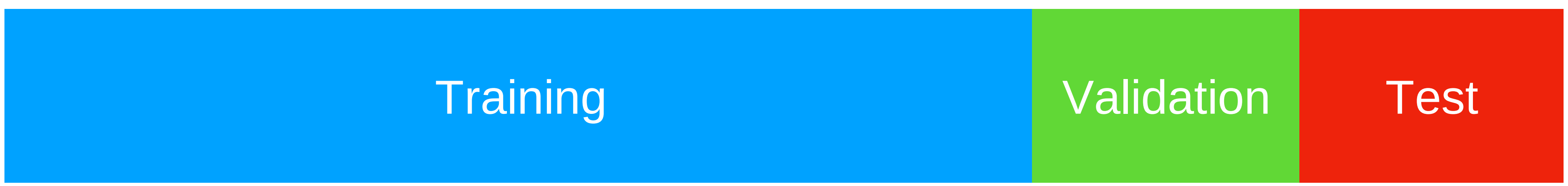
example					label
1	x_{11}	x_{12}	…	x_{1d}	y_1
…	…	…	…	…	…
i	x_{i1}	x_{i2}	…	x_{id}	y_i
…	…	…	…	…	…
n	x_{n1}	x_{n2}	…	x_{nd}	y_n

- Goal: learn a hypothesis function h(x) that approximates the true relationship between x and y. This function should: 1) ideally be consistent with the training data. 2) generalize to unseen examples.

- In NLP y_i typically form a finite, discrete set.

Running Machine Learning Experiments

- When running machine learning experiments we typically split the labeled data in three sections:



- For example: 80% Training, 10% Validation (development), 10% Test or 90/5/5

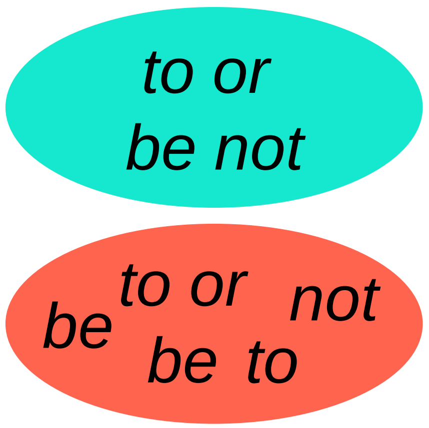
- Validation set is used to tune model parameters (for example smoothing parameters), but cannot be used for training. This can help with overfitting.

- Test set is used to assess the performance of the final model and provide an estimation of the test error.

Note: Never train or tune parameters on the test set!

Representing Documents

to be, or not to be



- Set-of-words representation.
- Bag-of-words representation (Multi-set).
- Vector-space model: Each word corresponds to one dimension in vector space. Entries are either:
  - Binary (Word appears / does not appear)
  - Raw or normalized frequency counts.
  - Weighted frequency counts
  - Probabilities.

$$\begin{array}{c} \text{be} \\ \vdots \\ \text{not} \\ \vdots \\ \text{or} \\ \vdots \\ \text{to} \end{array} \begin{bmatrix} \mathbf{2} \\ \vdots \\ \mathbf{1} \\ \vdots \\ \mathbf{1} \\ \vdots \\ \mathbf{2} \end{bmatrix}$$

What is a Word?

- e.g., are "*Cat*", "*cat*" and "*cats*" the same word?
- "*September*" and "*Sept*"?
- "*zero*" and "*oh*"?
- Is "*_*" a word? "*.*"? "***"? '*(*'?
- How many words are there in "*don't*" ? "*Gonna*" ? "*I.B.M.*"?
- In Japanese and Chinese text -- how do we identify a word?
- …

Text Normalization

- Every NLP task needs to do some text normalization.
  - Segmenting / tokenizing words in running text.
  - Normalizing word forms (lemmatization or stemming, possibly replacing named-entities).
  - Sentence splitting.

Linguistic Terminology

- **Sentence**: Unit of written language.
- **Utterance**: Unit of spoken language.
- Word **Form:** the inflected form as it actually appears in the corpus. *“produced”*
- Word **Stem**: The part of the word that never changes between morphological variations. *“produc”*
- **Lemma**: an abstract base form, shared by word forms, having the same stem**,** part of speech, and word sense – stands for the **class** of words with stem. *“produce”*
- **Type**: number of distinct words in a corpus (vocabulary size).
- **Token**: Total number of word occurrences**.**

Tokenization

- Tokenization: The process of segmenting text (a sequence of characters) into a sequence of tokens (words).

“Mr. O'Neill thinks that the boys' stories about Chile's capital aren't amusing.”

$\boxed{\texttt{mr.}}$ $\boxed{\texttt{o'neill}}$ $\boxed{\texttt{thinks}}$ $\boxed{\texttt{that}}$ $\boxed{\texttt{the}}$ $\boxed{\texttt{boys}}$ $\boxed{\texttt{'}}$ $\boxed{\texttt{stories}}$ $\boxed{\texttt{about}}$ $\boxed{\texttt{Chile}}$ $\boxed{\texttt{'s}}$ $\boxed{\texttt{capital}}$ $\boxed{\texttt{are}}$ $\boxed{\texttt{n't}}$ $\boxed{\texttt{amusing}}$ $\boxed{\texttt{.}}$

- Simple (but weak) approach: Separate off punctuation. Then split on whitespaces.

- Typical implementations use regular expressions (finite state automata).

Tokenization Issues

- Dealing with punctuation (some may be part of a word)
  “Ph.D.”, “O’Reilly”, “pick-me-up”

- Which tokens to include (punctuation might be useful for parsing, but not for text classification)?

- Language dependent: Some languages don’t separate words with whitespaces.
  de: *“Lebensversicherungsgesellschaftsangestellter”*

  zh: 日文章鱼怎么说? - Japanese Octopus how say?

  日文章鱼怎么说? - Sun article fish how say?

Chinese example from Sproat (1996)

Lemmatization

- Converting Lemmas into their base form.

"Mr. O'Neill thinks that the boys' stories about Chile's capital aren't amusing."

mr. *o'neill* *think* *that* *the* *boy* *story* *about* *chile* *'s* *capital* *are* *n't* *amusing* *.*

PER *PER* *think* *that* *the* *boy* *story* *about* *LOC* *'s* *capital* *are* *n't* *amusing* *.*

Probabilities in NLP

- Ambiguity is everywhere in NLP. There is often *uncertainty* about the "correct" interpretation. Which is more likely:
  - Speech recognition: *"recognize speech"* vs. *"wreck a nice beach"*
  - Machine translation: *"l'avocat general"*: *"the attorney general"* vs. *"the general avocado"*
  - Text classification: is a document that contains the word *"rice"* more likely to be about politics or about agriculture?
    What if it also includes several occurrences of the word *"stir"*?
- Probabilities make it possible to combine evidence from multiple sources systematically to (using Bayesian statistics)

Bayesian Statistics

- Typically, we observe some evidence (for example, words in a document) and the goal is to infer the “correct” interpretation (for example, the topic of a text).

- Probabilities express the degree of belief we have in the possible interpretations.

  - **Prior probabilities:** Probability of an interpretation prior to seeing any evidence.

  - **Conditional (Posterior) probability:** Probability of an interpretation after taking evidence into account.

Probability Basics

- Begin with a **sample space** Ω
  - Each $\omega \in \Omega$ is a possible basic outcome / "possible world" (e.g. the 6 possible rolls of a die).
- A **probability distribution** assigns a probability to each basic outcome.

$$P(\omega) \leq 1.0 \text{ for every } \omega \in \Omega$$

$$\sum_{\omega \in \Omega} P(\omega) = 1.0$$

- E.g: six-sided die

$$P(1) + P(2) + P(3) + P(4) + P(5) + P(6) = 1.0$$

Events

- An *event A* is any subset of Ω .

$$P(A) = \sum_{\omega \in A} P(\omega)$$

- Example:

$$P(\text{die roll} < 4) = P(1) + P(2) + P(3) = 1/6 + 1/6 + 1/6 = 1/2$$

Random Variables

- A random variable is a function from basic outcomes to some range, e.g. real numbers or booleans.

$$Odd(1) = true$$

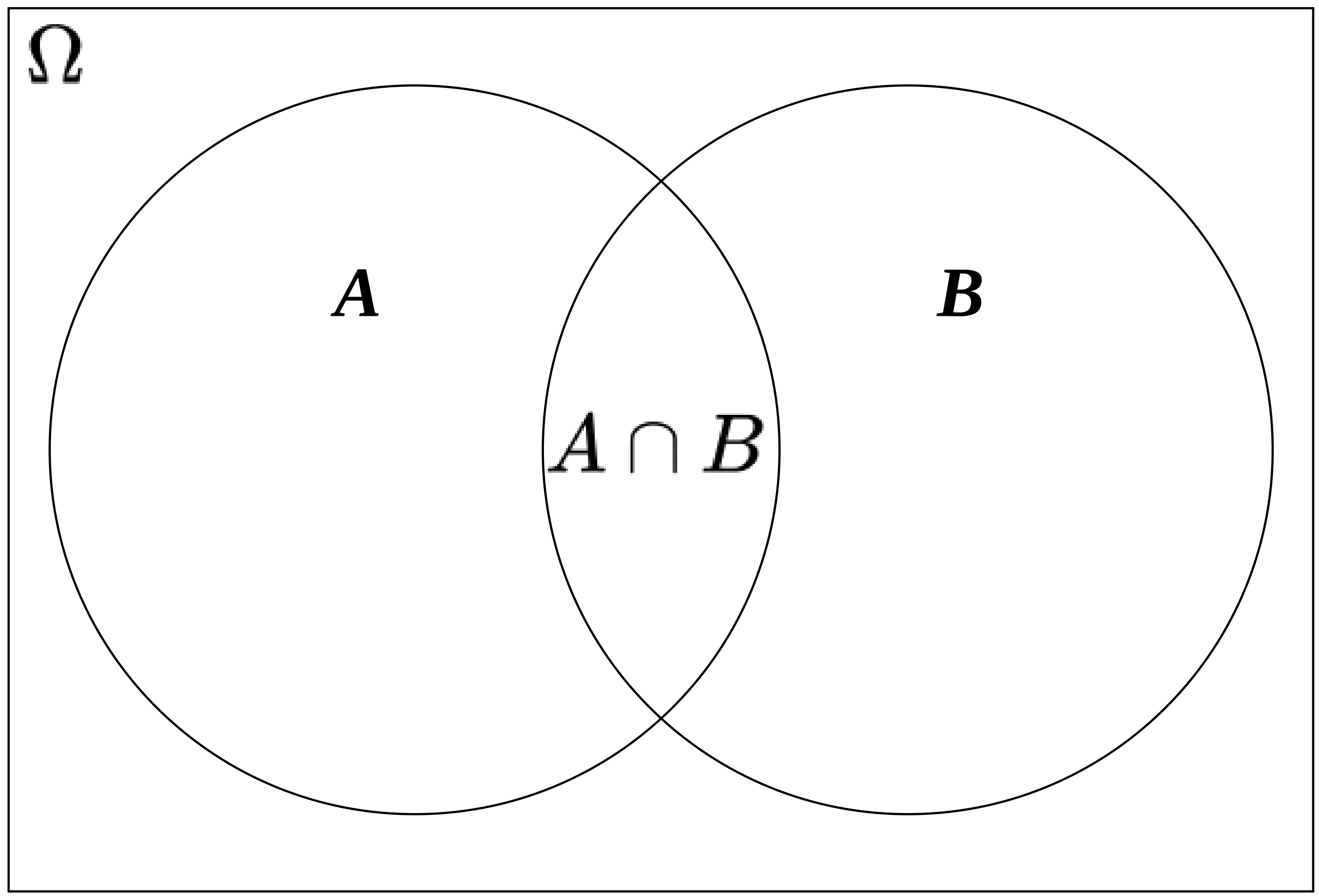
- A distribution P induces a probability distribution for any random variable.

$$P(X = x_i) = \sum_{\{\omega : X(\omega) = x_i\}} P(\omega)$$

- E.g $P(Odd = true) = P(1) + P(3) + P(5) = 1/2$

Joint and Conditional Probability

Joint probability: $P(A \cap B)$ also written as $P(A, B)$



Conditional probability: $P(A|B) = \frac{P(A, B)}{P(B)}$

Rules for Conditional Probability

- Product rule: $P(A, B) = P(B) \cdot P(A|B) = P(A) \cdot P(B|A)$

- Chain rule (generalization of product rule):

$$P(A_n, \ldots, A_1) = P(A_n|A_{n-1}, \ldots, A_1) \cdot P(A_{n-1}, \ldots, A_1)$$

- **Bayes' Rule:**

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Independence

- Two events are independent if $P(A) = P(A|B)$

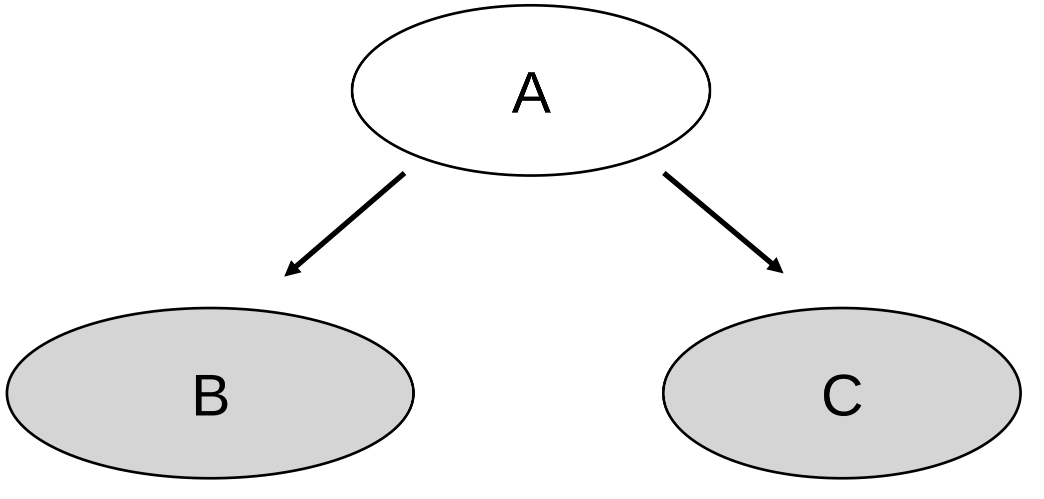
  or equivalently $P(A,B) = P(A) \cdot P(B)$ (if $P(B) > 0$)

- Two events are **conditionally independent** if:

$$P(B,C|A) = P(B|A)P(C|A)$$

  or equivalently

$$P(B|A,C) = P(B|A) \text{ and } P(C|A,B) = P(C|A)$$



Probabilities and Supervised Learning

- Given: Training data consisting of training examples
  data = $(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_n}, y_n)$,
  Goal: Learn a mapping h from x to y.

- We would like to learn this mapping using $P(y|x)$.

- Two approaches:

  - Discriminative algorithms learn $P(y|x)$ directly.

  - Generative algorithms use Bayes rule

$$P(y|x) = \frac{P(x|y) \cdot P(y)}{P(x)}$$

Discriminative Algorithms

- Model conditional distribution of the label given the data $P(y|x)$
- Learns decision boundaries that separate instances of the different classes.
- To predict a new example, check on which side of the decision boundary it falls.
- Examples:
  support vector machine (SVM), decision trees, random forests, neural networks, log-linear models.

Generative Algorithms

- Assume the observed data is being “generated” by a “hidden” class label.

- Build a **different model** for each class.

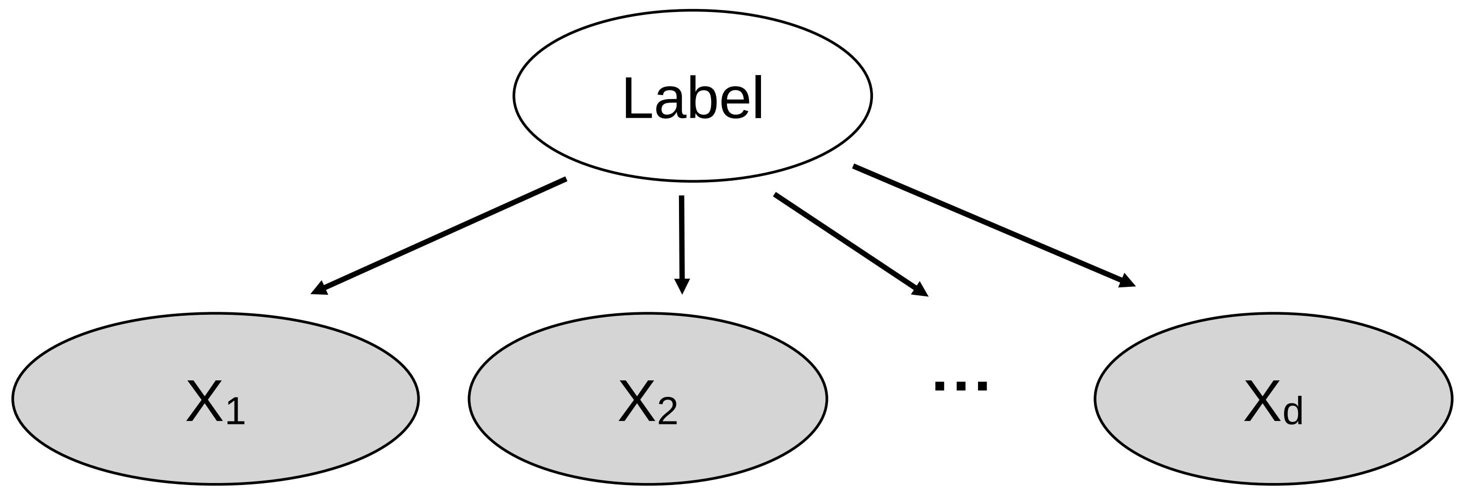
- To predict a new example, check it under each of the models and see which one matches best.

- Estimate $P(x|y)$ and $P(y)$. Then use bases rule

$$P(y|x) = \frac{P(x|y) \cdot P(y)}{P(x)}$$

- Examples:
  Naive Bayes, Hidden Markov Models, Gaussian Mixture Models, PC

Naive Bayes



Label

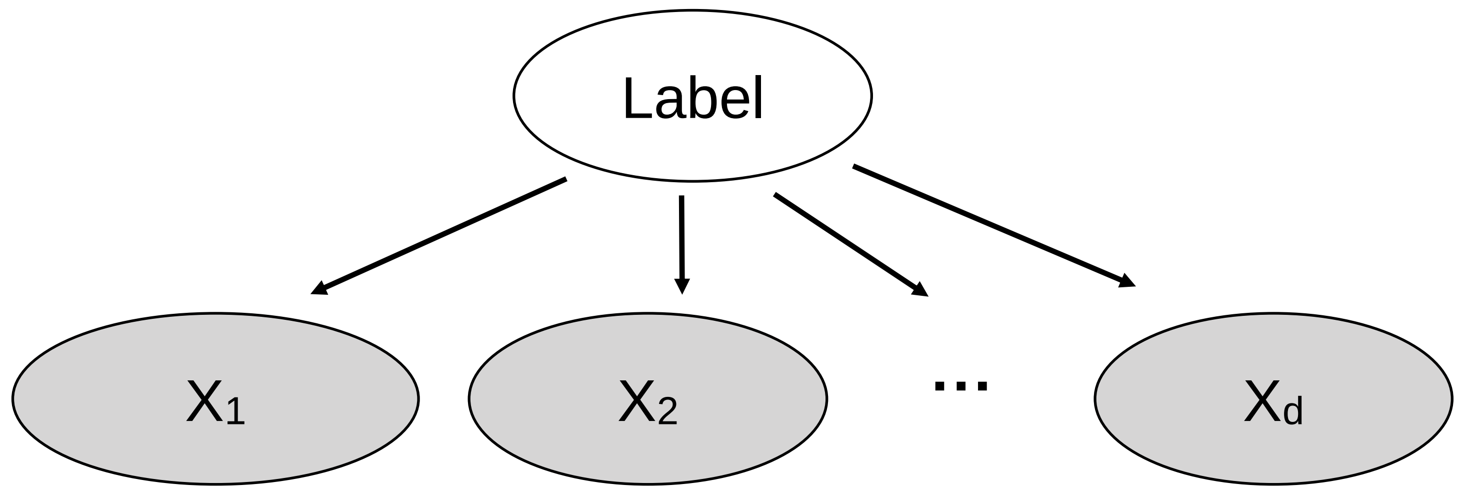
Attributes

$$\mathbf{P}(Label, X_1, \ldots X_d) = \mathbf{P}(Label) \prod_i P(X_i | Label)$$

$$\mathbf{P}(Label | X_1, \ldots X_d) = \frac{\mathbf{P}(Label) \prod_i P(X_i | Label)}{\prod_i P(X_i)}$$

$$= \alpha [\mathbf{P}(Label) \prod_i P(X_i | Label)]$$

Naive Bayes Classifier



$$\mathbf{P}(Label|X_1,\ldots X_d) = \alpha[\mathbf{P}(Label)\prod_i P(X_i|Label)]$$

$$y* = \arg\max_y P(y)\prod_i P(x_i|y)$$

Note that the normalizer α does no longer matter for the argmax because α is independent of the class label.

Training the Naive Bayes' Classifier

- Goal: Use the training data to estimate **P**(Label) and **P**(X_i|Label) from training data.

- Estimate the prior and posterior probabilities using **Maximum Likelihood Estimates (MLE):**

$$P(y) = \frac{Count(y)}{\sum_{y' \in Y} Count(y')}$$

$$P(x_i|y) = \frac{Count(x_i, y)}{\sum_{x'} Count(x', y)} = \frac{Count(x_i, y)}{Count(y)}$$

- I.e. we just count how often each token in the document appears together with each class label.

Why the Independence Assumption Matters

- Without the independence assumption we would have to estimate $\mathbf{P}(X_1, \ldots X_d | Label)$
- There would be many combinations of $x_1, \ldots, x_d$ that are never seen (sparse data).
- The independence assumption allows us to estimate each $\mathbf{P}(X_1 | label)$ independently.

Is this a safe assumption for documents?
Are the words really independent of each other?

Training the Naive Bayes' Classifier

- Ways to improve this model?
- Some issues to consider...
  - What if there are words that do not appear in the training set? What if it appears only once?
  - What if the plural of a word never appears in the training set?
  - How are extremely common words (e.g., "the", "a") handled?

Acknowledgments

- Some slides and examples from:
  - Kathy McKeown, Dragomir Radev