Applied Machine Learning: Algorithms, Practice and Theory

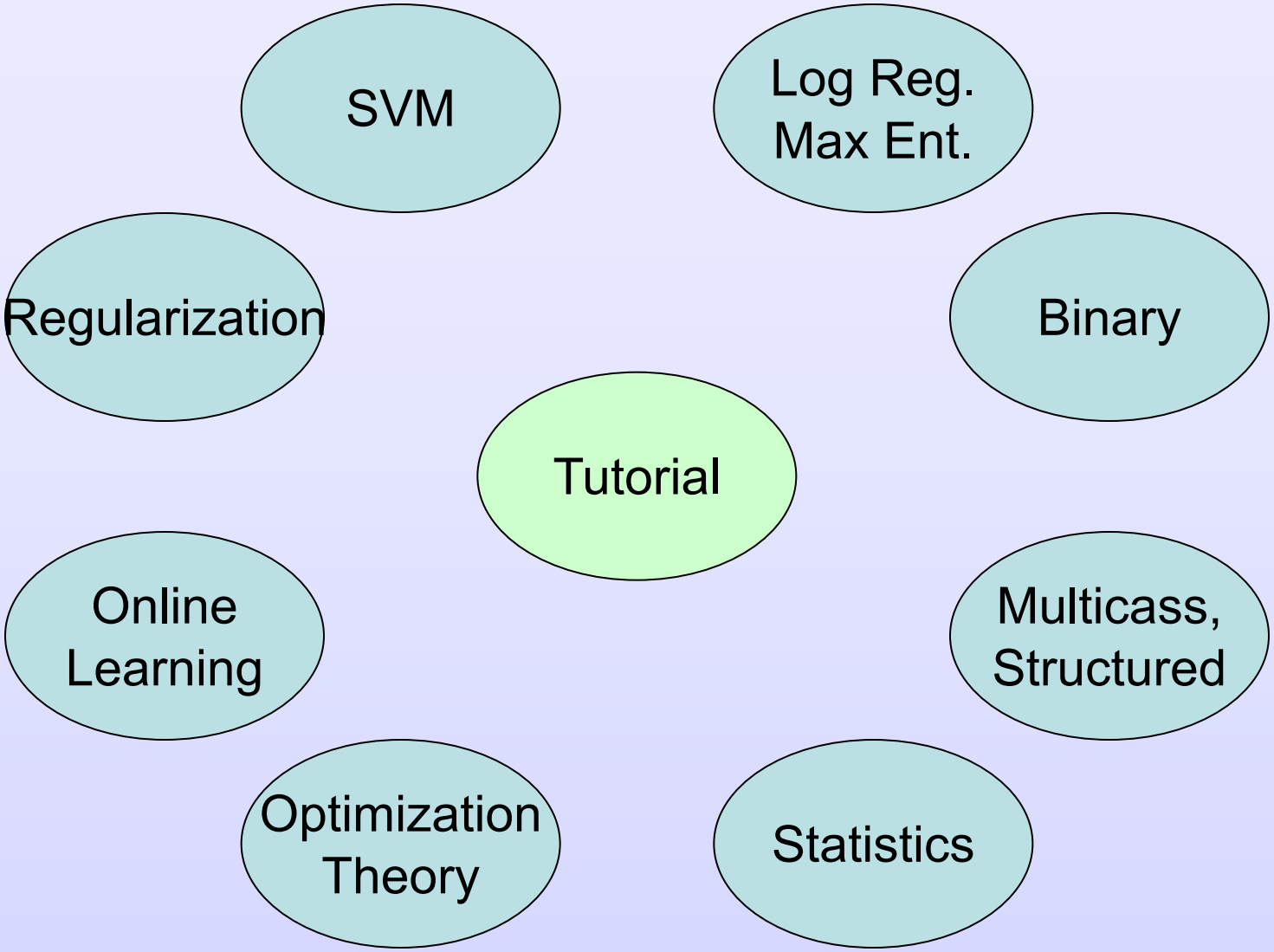
Predicting & Naïve Bayes

Koby Crammer
Technion – Israel Institute of Technology

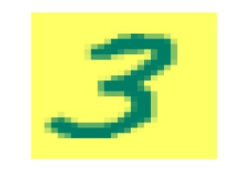
Thanks

- Ofer Dekel
- Josheph Keshet
- Shai Shalev-Schwatz
- Yoram Singer
- Mark Dredze
- Ryan McDonald
- Fernando Pereira
- Partha Pratim Talukdar
- Alex Kulesza

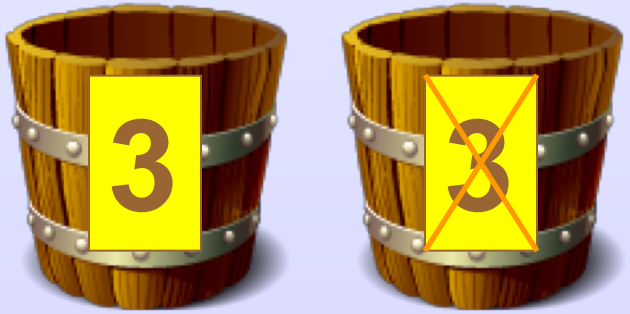
Tutorial Context



Binary Classification



If it's not one class, It most be the other class



Outline

- Making Predictions
- Algorithms:
  - Naïve Bayes
  - Logistic Regression
  - Support Vector Machines
- Online Learning:
  - Perceptron
  - Passive-Aggressive

Outline

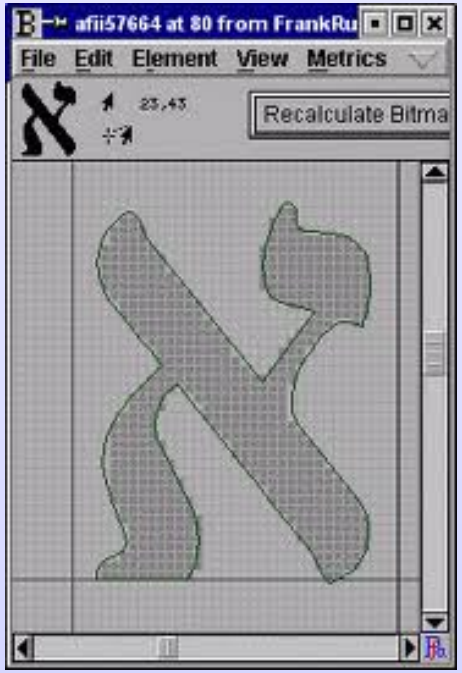
- Building Predictors
  - Data
  - Evaluation
  - Prediction
- Models
  - Assumptions
  - Naïve Bayes
  - Learning
- Non-Parametric models

Daddy, teach me to read
Expert Approach

- Of course my daughter …
- The first letter of the Hebrew alphabet is called Allef and it looks like א
- You arite Allef with three strokes:
  - Diagonal line that goes from top-left to bottom-right.
  - Short vertical line from top-right that goes down until the first line
  - Short vertical line from bottom-left that goes up until the first line
- Daddy, can you repeat ;-(

Daddy, teach me to read
Data-driven Approach

- Of course my daughter …



Do you get it?

- Of course …
- Lets see:
  - א yes 
  - ב no 
  - א no 
  - ג no 

- Predictors are evaluated using data
- This data is not the same used for training
- Yet, it has the same statistical properties

Outline

- Building Predictors
  - Data
  - Evaluation
  - Prediction
- Models
  - Assumptions
  - Naïve Bayes
  - Learning
- Non-Parametric models

Setting

- Making Predictions
- Input-output relation

- Communication with “Problem” via “Data”
  - Training set used to build (or find) prediction functions
  - Test set used to evaluate predictions
- Assume both sets come from the same “source”

- Evaluation using loss-function

Formal Setting – Data

- Instances
  - Images, Sentences
- Labels
  - Parse tree, Names
- Statistical Assumption
  - Used for training and evaluation
- I.I.D. Sample

$$\mathbf{x} \in \mathcal{X}$$

$$y \in \mathcal{Y} = \{-1 \; ; \; 1\}$$

$$(X, Y) \sim P$$

$$S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$$

$$(\boldsymbol{x}_i, y_i) \sim P$$

Formal Setting - Predictions

- Discrete Predictions: $f : \mathcal{X} \rightarrow \{-1; 1\}$
  - Hard to optimize
- Continuous predictions : $f : \mathcal{X} \rightarrow \mathbb{R}$
  - Label $\text{sign}(f(\mathbf{x}))$
  - Confidence $|f(\mathbf{x})|$
- Notation: $f(\mathbf{x}) = \hat{y}$

Outline

- Building Predictors
  - Data
  - Evaluation
  - Prediction
- Models
  - Assumptions
  - Naïve Bayes
  - Learning
- Non-Parametric models

Formal Setting - Evaluation

- Loss
  - Cost, Error, Risk $\ell(f(\boldsymbol{x}), y) \in \mathbb{R}_+$
- Expected Loss

$$\mathrm{E}_{(X,Y)\sim P}\left[\ell(f(X), Y)\right]$$

- Averaged Loss

$$\mathrm{E}_{(X,Y)\sim S}\left[\ell(f(X), Y)\right]$$

$$= \frac{1}{m}\sum_{i=1}^{m} \ell(f(\boldsymbol{x}_i), y_i)$$

Loss Functions

- Natural Loss:
  - Zero-One loss:

$$\ell\left(f(\boldsymbol{x}), y\right) = \begin{cases} 0 & y = \operatorname{sign}(f(\boldsymbol{x})) \\ 1 & y \neq \operatorname{sign}(f(\boldsymbol{x})) \end{cases}$$

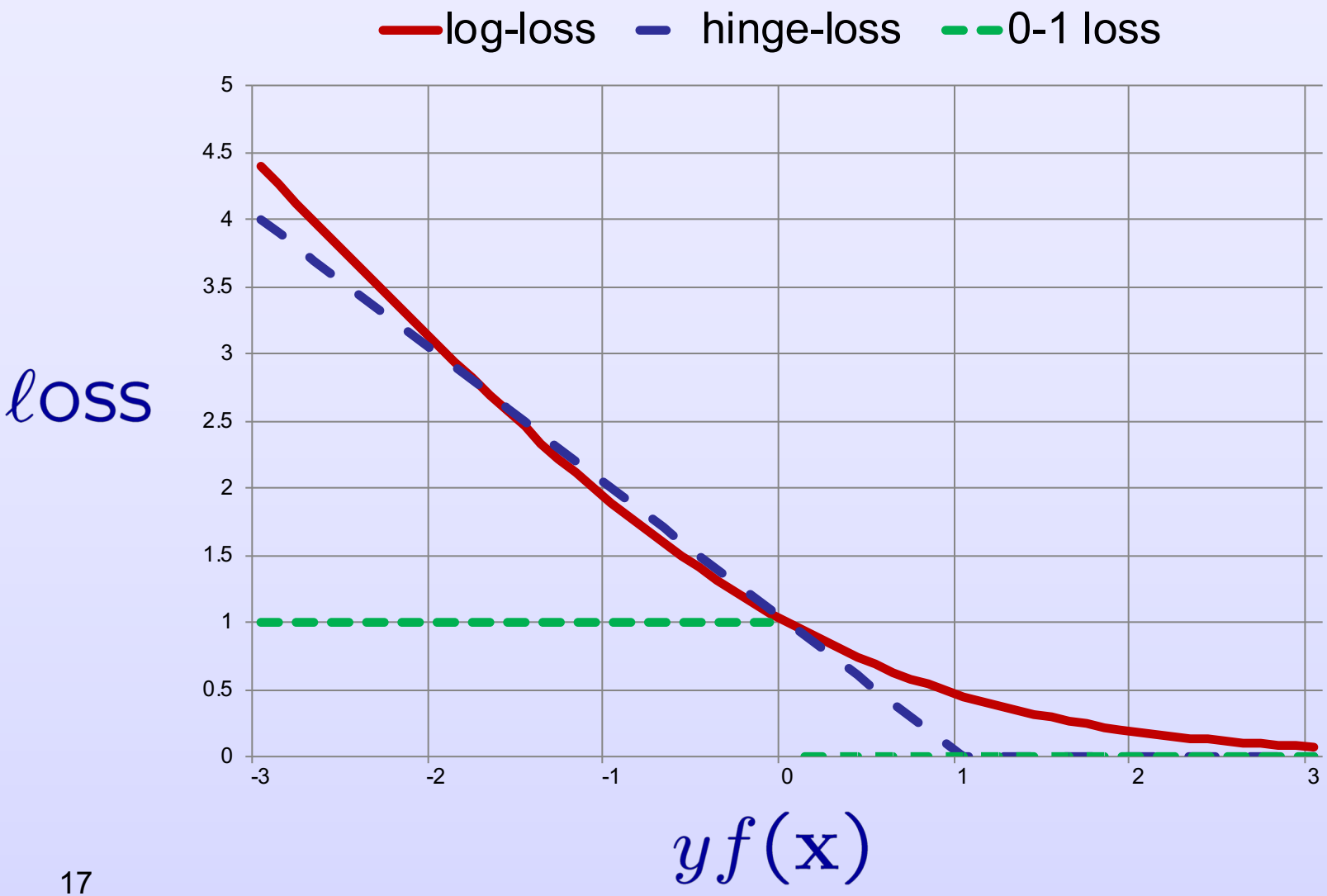
- Real-valued-predictions loss:
  - Hinge loss:

$$\ell\left(f(\boldsymbol{x}), y\right) = \max\left(0, 1 - yf(\boldsymbol{x})\right)$$

  - Log loss (Max Entropy, Boosting)

$$\ell\left(f(\boldsymbol{x}), y\right) = \log\left(1 + \exp\left(-yf(\boldsymbol{x})\right)\right)$$

Loss Functions



Bayes-Optimal Predictor

- Our goal is to minimize:

$$\mathrm{E}_{(X,Y)\sim P}\left[\ell(f(X),Y)\right]$$
$$=\mathrm{E}_{X\sim P}\left[\mathrm{E}_{Y\sim P(Y|X)}\left[\ell(f(X),Y)\right]\right]$$

- The per-input expectation is minimized if :

$$f(X)=\arg\min_{z\in\mathcal{Y}}\mathrm{E}_{Y\sim P(Y|X)}\left[\ell(z,Y)\right]$$

Outline

- Building Predictors
  - Data
  - Evaluation
  - Prediction
- Models
  - Assumptions
  - Naïve Bayes
  - Learning
- Non-Parametric models

Bayes-optimal Predictor

$$f(X) = \arg\min_{z \in \mathcal{Y}} \mathrm{E}_{Y \sim P(Y|\boldsymbol{x})} \left[\ell(z, Y) \right]$$

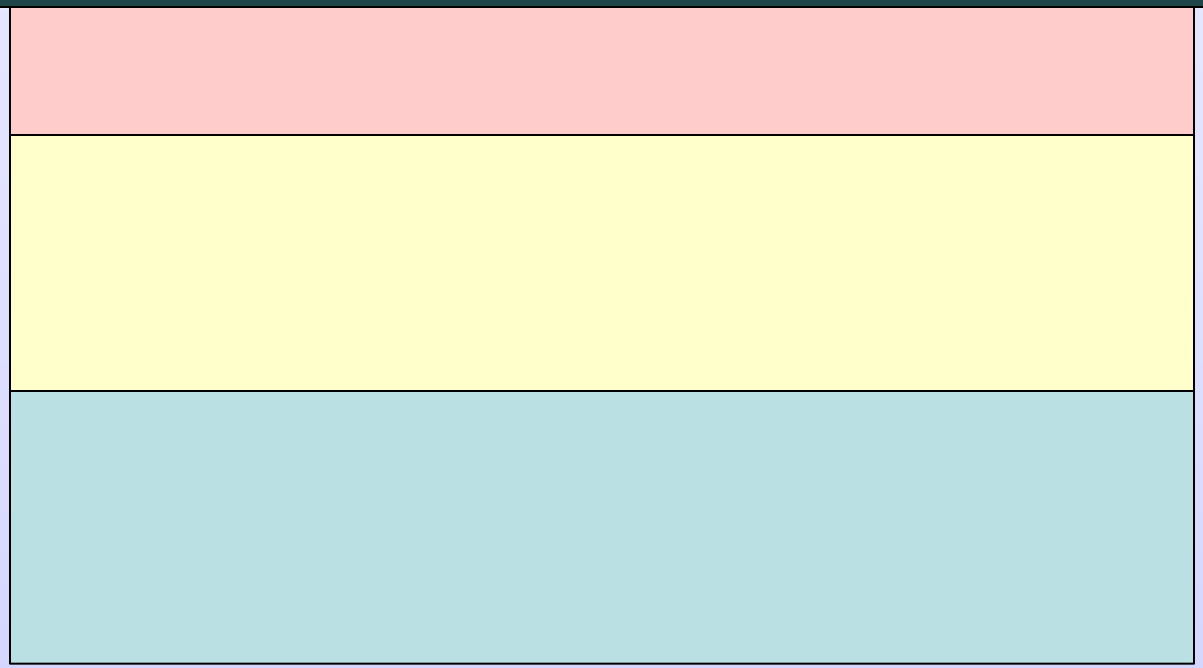
Maximum a-posteriori

$$f(X) = \arg\max_{z \in \mathcal{Y}} P(z|\boldsymbol{x})$$

Bayes-optimal Predictor

- Zero-one loss:

Maximizing the conditional (a-posteriori)
is equivalent to
maximizing the joint distribution



Problem

- Bayes-optimal predictor:

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} P(y|x) = \arg\max_{y \in \mathcal{Y}} P(x, y)$$

- But: $P(x, y)$ is unknown, only $S \sim P^m$ is

- Solution:
  - Estimate a proxy $\hat{P}_S(X, Y)$
  - Use estimated model to make predictions

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} \hat{P}_S(y|x)$$

Three Issues

- Modeling:
  - What estimates to use?

$$\hat{P}_S(X, Y)$$

- Inference:
  - How to use the estimates?
  - How make decisions / predictions?

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} \hat{P}_S(y|x)$$

- Learning:
  - How to construct estimates from (finite) of data

Outline

- Building Predictors
  - Data
  - Evaluation
  - Prediction
- Models
  - Assumptions
  - Naïve Bayes
  - Learning
- Non-Parametric models

Modeling

- How to factor the joint distribution?

$$\hat{P}_S(X,Y) = \hat{P}_S(Y)\hat{P}_S(X|Y)$$
$$= \hat{P}_S(X)\hat{P}_S(Y|X)$$

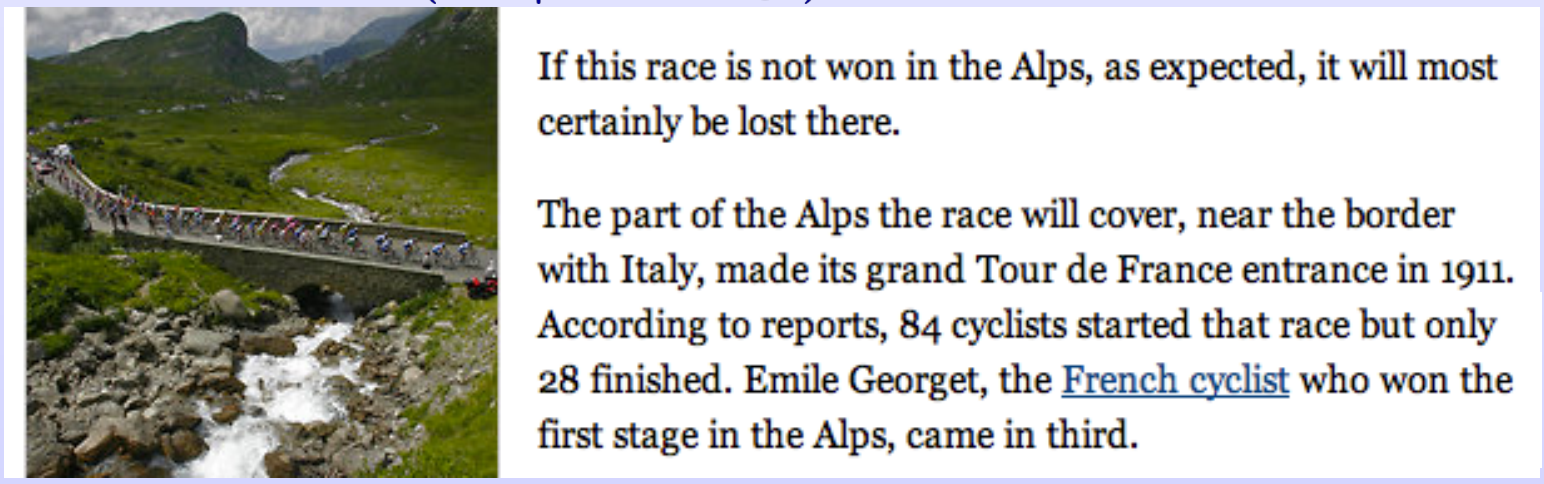
- Type of estimate:
  - Parametric: e.g. Gaussian, Bernoulli
  - Non-parametric: grows with $m = |S|$

Writing a Document

Generative Approach

- Topic(s) [and content] $\hat{P}_S(Y)$

- Then write it $\hat{P}_S(X|Y=y)$



If this race is not won in the Alps, as expected, it will most certainly be lost there.

The part of the Alps the race will cover, near the border with Italy, made its grand Tour de France entrance in 1911. According to reports, 84 cyclists started that race but only 28 finished. Emile Georget, the French cyclist who won the first stage in the Alps, came in third.

Making a Decision

Discriminative Approach

- Factorizing the joint distribution:

$$\hat{P}_S(X,Y) = \hat{P}_S(X)\hat{P}_S(Y|X)$$

Distribution of all documents

Distribution of labels for a specific document

- But, the Bayes-optimal classifier depends only on the conditional term $\hat{y} = \arg\max_{y \in \mathcal{Y}} P(y|x)$
- No need to model the distribution over data

Generative vs Discriminative

- Can be used to generate new examples
- If the model fits the real world, works better
- Often simpler training (counting)

- Used only for predictions
- Works well even nature of data distribution is not known
- Model only one distribution (conditional)

Specific Generative Model: Labels

- Labels are picked from a finite set

$$\hat{P}_S(Y)$$

- Modeled by a multinomial distribution
- Simple estimate: Fraction of documents per class

Specific Generative Model: Documents

- We pick a topic y
- Basic unit: words / tokens
- Notation: The t*th* word of the document $\phi_t(\boldsymbol{x})$

$\phi_1(\boldsymbol{x}) = \text{If}$ $\phi_3(\boldsymbol{x}) = \text{race}$ $\phi_9(\boldsymbol{x}) = \text{Alps}$

If this race is not won in the Alps, as expected, it will most certainly be lost there.

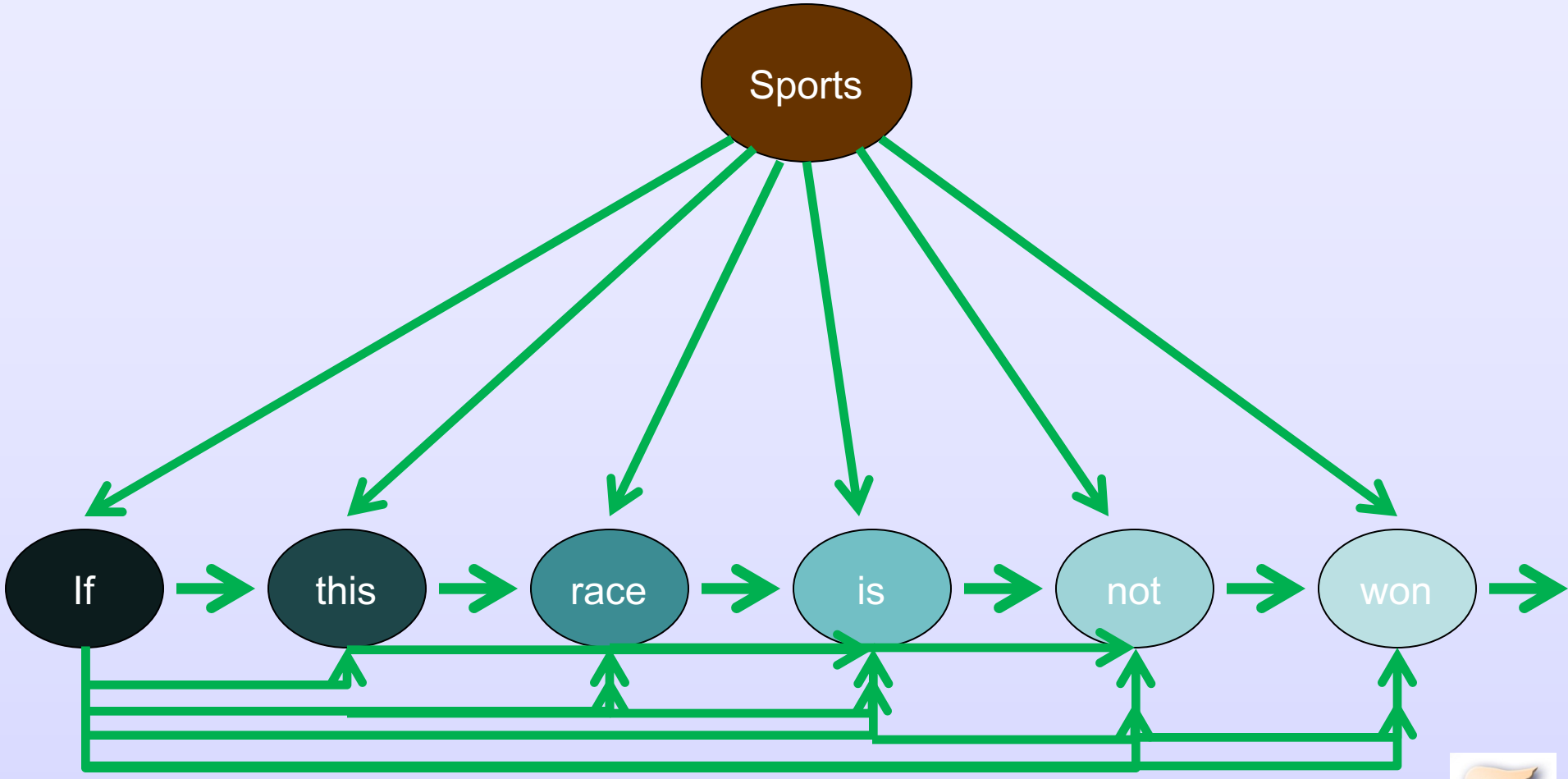
The part of the Alps the race will cover, near the border with Italy, made its grand Tour de France entrance in 1911. According to reports, 84 cyclists started that race but only 28 finished. Emile Georget, the French cyclist who won the first stage in the Alps, came in third.

Specific Generative Model: Documents

- Identify a document with an ordered list of words

$$P(\boldsymbol{x}|y) = P\left(\phi_1(\boldsymbol{x}) \ldots \phi_T(\boldsymbol{x}) \middle| y\right)$$

Specific Generative Model: Documents



Outline

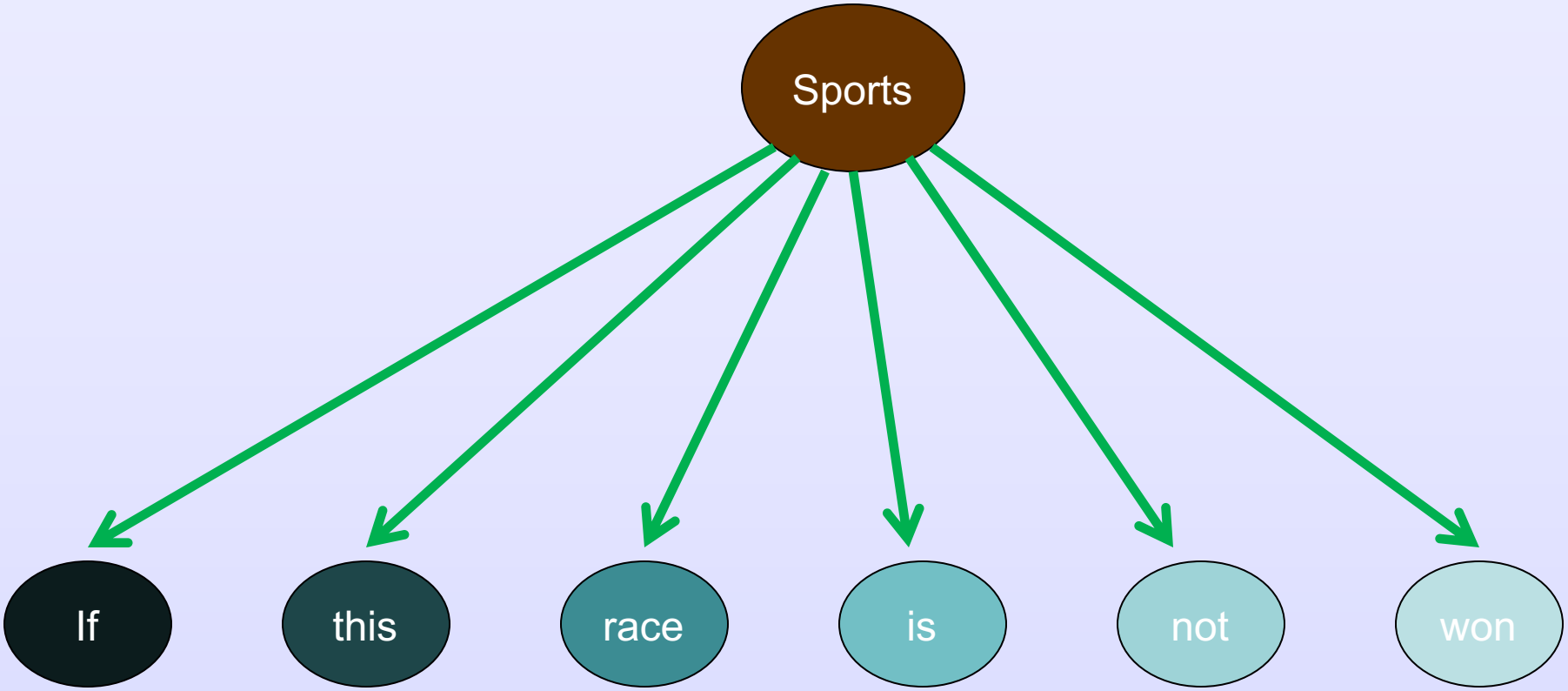
- Building Predictors
  - Data
  - Evaluation
  - Prediction
- Models
  - Assumptions
  - Naïve Bayes
  - Learning
- Non-Parametric models

Specific Generative Model: Documents

- First approximation: Independence given topic (our modeling choice)

$$P(\boldsymbol{x}|y) = P\left(\phi_1|y\right) \times P\left(\phi_2|\phi_1, y\right) \times \cdots \times P\left(\phi_T|\phi_1, \ldots, \phi_{T-1}, y\right)$$

Specific Generative Model: Documents

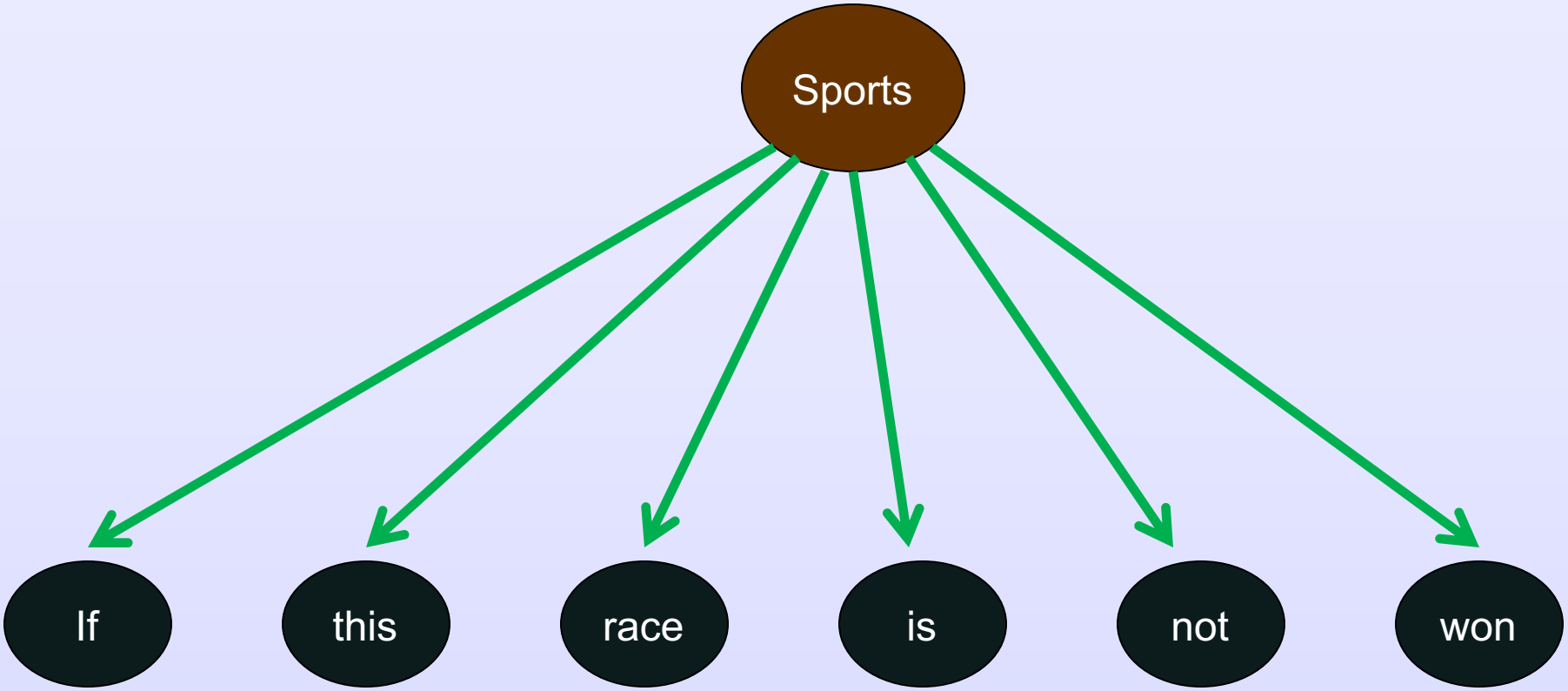


Specific Generative Model: Documents

- Second approximation: Identical distribution (our modeling choice)
- Ignoring location of words

$$P(\boldsymbol{x}|y) = \prod_{t=1}^{T} P_t\left(\phi_t|y\right)$$

Specific Generative Model: Documents



Invariant under permutation of words

Specific Generative Model: Documents

- Per location representation

$$P(\boldsymbol{x}|y) = \prod_{t=1}^{T} P\left(\phi_t|y\right)$$

- Notation: $\boldsymbol{x}_q$
  - number of times a word appears

- Bag-of-word representation

$$P(\boldsymbol{x}|y) = \prod_{q=1}^{Q} P\left(\text{word}_q|y\right)^{\boldsymbol{x}_q}$$

Occam's Razor

Occam's Razor

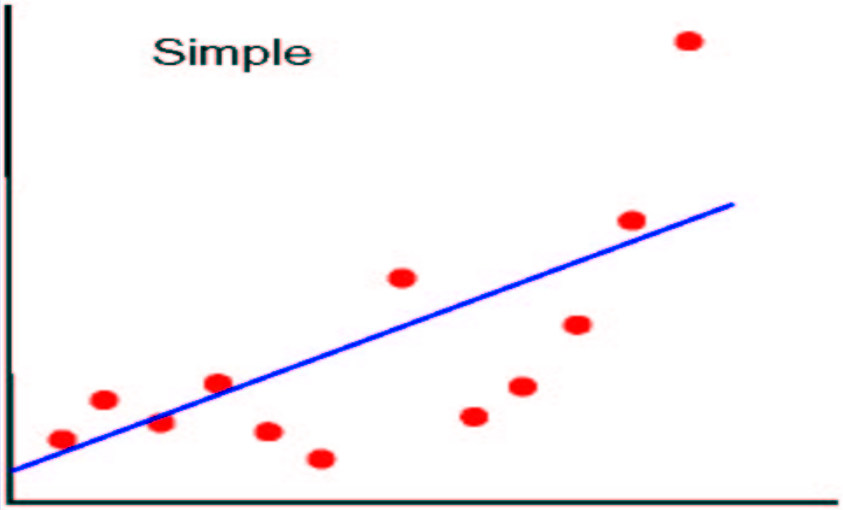
through the ages...

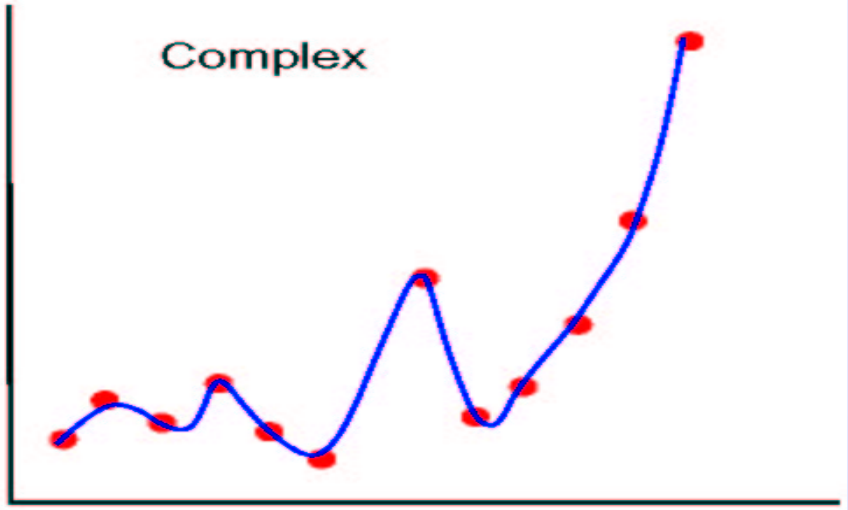
Pluralitas non est ponenda sine necessitate.
(Plurality should not be posited without necessity.)
- William of Ockham

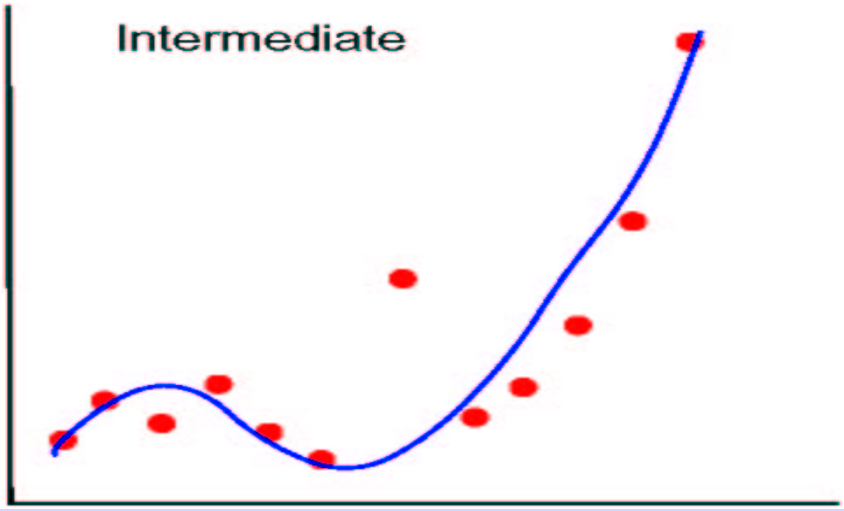
Everything should be made as simple as possible, but not simpler.
- Albert Einstein

Keep
It
Simple,
Stupid !

Occam's Razor







Outline

- Building Predictors
  - Data
  - Evaluation
  - Prediction
- Models
  - Assumptions
  - Naïve Bayes
  - Learning
- Non-Parametric models

Naïve Bayes: Model

$$P(\boldsymbol{x}, y) = P(y)P(\boldsymbol{x}|y)$$

$$= P(y) \prod_{q=1}^{Q} P\Big(\text{word}_q|y\Big)^{\boldsymbol{x}_q}$$

- Parameters:
  - Distribution over classes

$$\big(P(1) \dots P(K)\big) \in \Delta^{K-1}$$

  - K distributions over dictionary words, one per class

$$\big(P(\text{word}_1|y)) \dots P(\text{word}_Q|y))\big) \in \Delta^{Q-1}$$

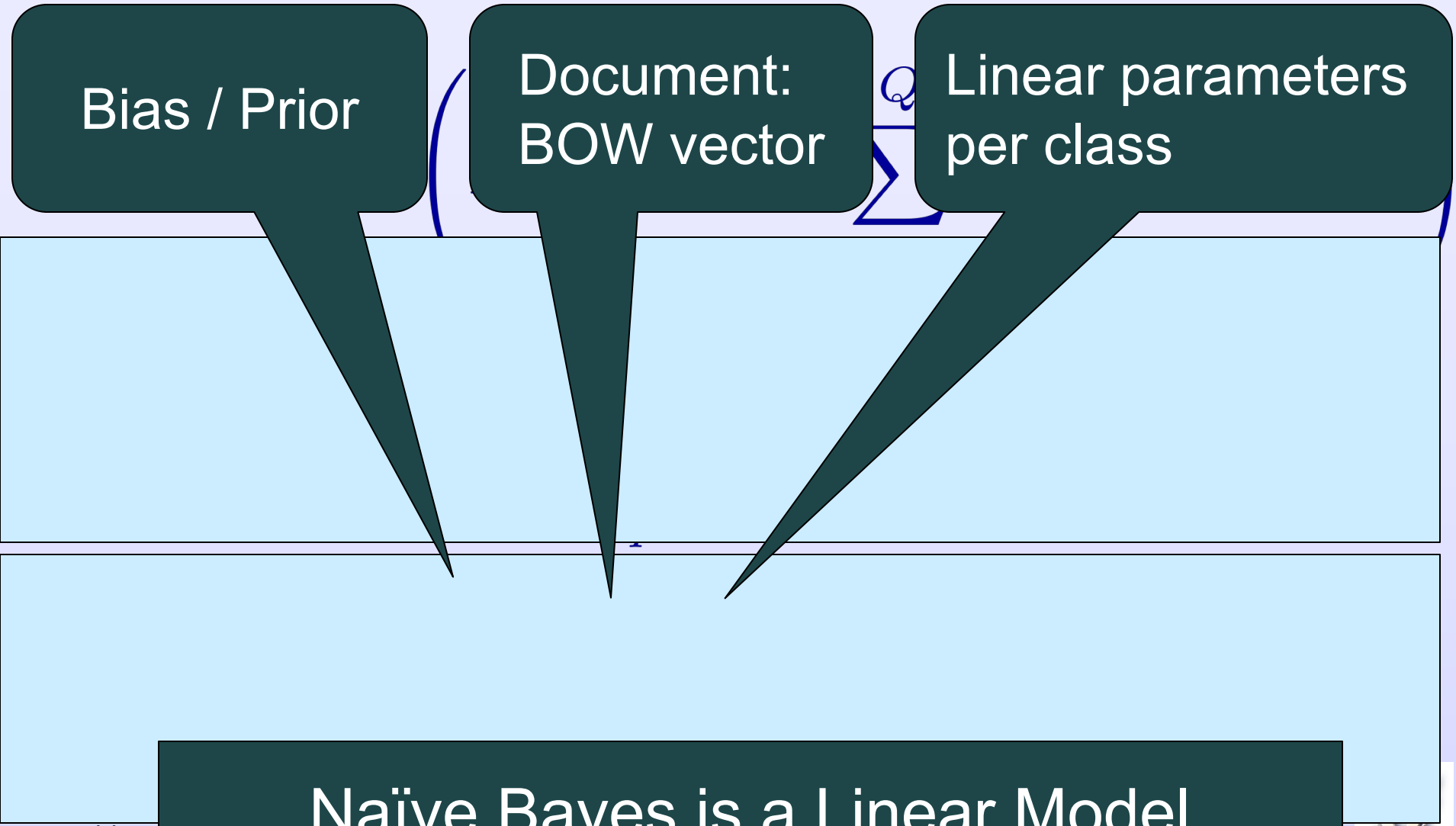
$$y = 1 \dots K$$

Naïve Bayes: Inference

$$\hat{y} = \arg\max_{z} P(\boldsymbol{x}, z)$$

$P(\text{word} \mid y)$

Naïve Bayes: Inference



Naïve Bayes is a Linear Model

Outline

- Building Predictors
  - Data
  - Evaluation
  - Prediction
- Models
  - Assumptions
  - Naïve Bayes
  - Learning
- Non-Parametric models

Naïve Bayes: Learning

- Input: labeled set $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$

$$\boldsymbol{x}_i \in \mathbb{R}_{+}^{Q} \ , \ y_i \in \{1 \dots K)\}$$

- Output:
  - Distribution over classes

$$\left(\hat{P}_S(1) \dots \hat{P}_S(K)\right) \in \Delta^{K-1}$$

  - K distributions over dictionary words, one per class

$$\left(\hat{P}_S(1|y)) \dots \hat{P}_S(Q|y))\right) \in \Delta^{Q-1} \quad y = 1 \dots K$$

Maximum Likelihood Estimation

- Find parameters that maximize probability of input sample:

$$\arg \max_{\{P(z)\},\{P(q|z)\}} \prod_{i=1}^{m} \left(P(y_i) \prod_{q=1}^{Q} P(q|y_i)^{\boldsymbol{x}_{i,q}} \right)$$

- Compare with inference:

$$\hat{y} = \arg \max_{z} \left(P(z) \prod_{q=1}^{Q} P(q|z)^{\boldsymbol{x}_q} \right)$$

Example

Data Set

- { (football ; SPORTS) }

Model I

- P(SPORTS)=0.5
- P(POLITICS)=0.5

- P(football | SPORTS) = 0.2
- P(basketball | SPROTS) = 0.8
- P(election | POLITICS) = 1.0

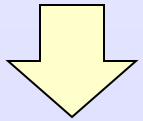
Model II

- P(SPORTS) = 1.0
- P(football | SPORTS) = 1.0

Maximum Likelihood Estimation

- Take the logarithm of the objective

$$\prod_{i=1}^{m}\left(P(y_i)\prod_{q=1}^{Q}P(q|y_i)^{\boldsymbol{x}_{i,q}}\right)$$



$$\sum_{i=1}^{m}\log P(y_i)+\sum_{i=1}^{m}\sum_{q=1}^{Q}\boldsymbol{x}_{i,q}\log\left(P(q|y_i)\right)$$

Maximum Likelihood Estimation

- Rewrite the sums

Enumerate over all labels

Enumerate over all examples per label

Enumerate over all words in dictionary

$$\sum_{y=1}^{K} \sum_{i:\, y_i = y} \log P(y) + \sum_{y=1}^{K} \sum_{i:\, y_i = y} \sum_{q=1}^{Q} x_{i,q} \log \left(P(q|y) \right)$$

Maximum Likelihood Estimation

- Add constraints:

$$\arg \max_{\{P(z)\},\{P(q|z\}} \sum_{y=1}^{K} \sum_{i\,:\,y_i=y} \log P(y)$$

$$+ \sum_{y=1}^{K} \sum_{i\,:\,y_i=y} \sum_{q=1}^{Q} \boldsymbol{x}_{i,q} \log\left(P(q|y)\right)$$

$$\text{s.t.} \sum_{z} P(z) = 1$$

$$\sum_{q} P(q|z) = 1 \quad \text{for } q = 1 \ldots K$$

Maximum Likelihood Estimation

- Solving with Lagrange Method:

$$\hat{P}_S(y) = \frac{m_y}{m}$$

$$\hat{P}_S(q|y) = \frac{\sum_{i\,:\,y_i=y} \boldsymbol{x}_{i,q}}{\sum_{i\,:\,y_i=y} \sum_{q=1}^{Q} \boldsymbol{x}_{i,q}}$$

Number of documents with label y

Total number of documents

Total no. times word q in all documents with label y

Total no. of words in all documents with label y

Need for Smoothing

Data Set

- { (football ; SPORTS) }

Model

- P(SPORTS) = 1.0
- P(football | SPORTS) = 1.0

Test example

- P(SPORTS | “basketball game”) = 0.0 !!

Smoothing

- What about rare words?
  - Words that not in the training set have zero estimate!
- Solution:
  - Smoothing
  - Add “fake” count of “1” per dictionary-word
  - Equivalent: use uniform prior over dictionary-words
  - Many smoothing methods

$$\hat{P}_S(q|y) = \frac{1 + \sum_{i:\, y_i = y} x_{i,q}}{m + \sum_{i:\, y_i = y} \sum_{q=1}^{Q} x_{i,q}}$$

Computational Properties

- Size:
  - No. of labels, plus $\left(\hat{P}_S(1) \dots \hat{P}_S(K)\right) \in \Delta^{K-1}$
  - No. of dictionary-words times number of labels

$$\left(\hat{P}_S(1|y)) \dots \hat{P}_S(Q|y))\right) \in \Delta^{Q-1} \quad y = 1 \dots K$$

- Time:

$$\hat{P}_S(q|y) = \frac{1 + \sum_{i\,:\,y_i = y} \boldsymbol{x}_{i,q}}{m + \sum_{i\,:\,y_i = y} \sum_{q=1}^{Q} \boldsymbol{x}_{i,q}}$$

  - No. of documents
  - Total no. of words times no. of documents
  - Can be parallelized!

Outline

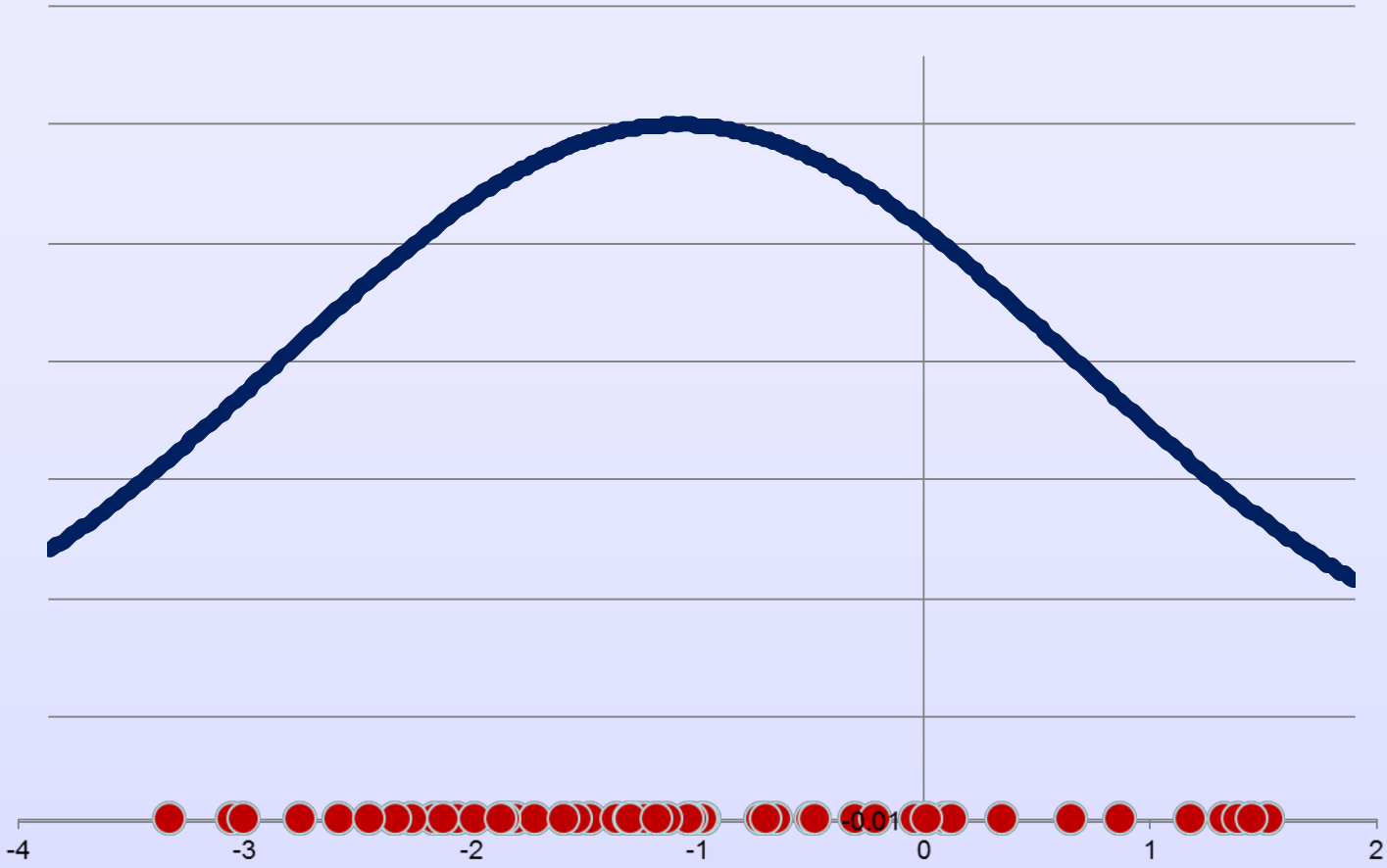
- Building Predictors
  - Data
  - Evaluation
  - Prediction
- Models
  - Assumptions
  - Naïve Bayes
  - Learning
- Non-Parametric models

Parametric vs. Non Parametric

- Parametric:
  - Implicitly modeled distribution over document with a parametric (multinomial) distribution
  - Size of model is fixed in the training-set size!
  - Works well if model is similar to true distribution

- Non-Parametric:
  - Models that grow in training-set size
  - Flexible enough to model (almost) all distributions

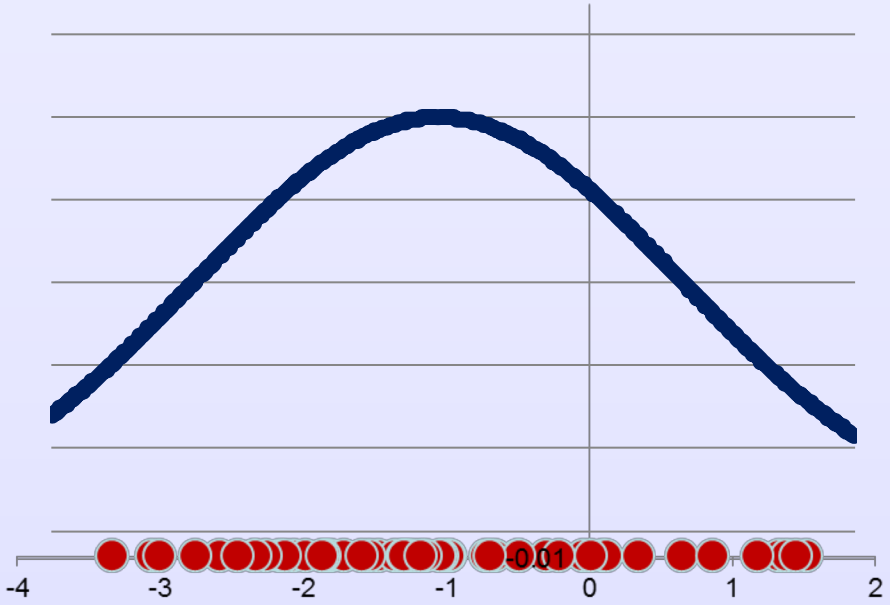
1-D Regression



1-D Regression

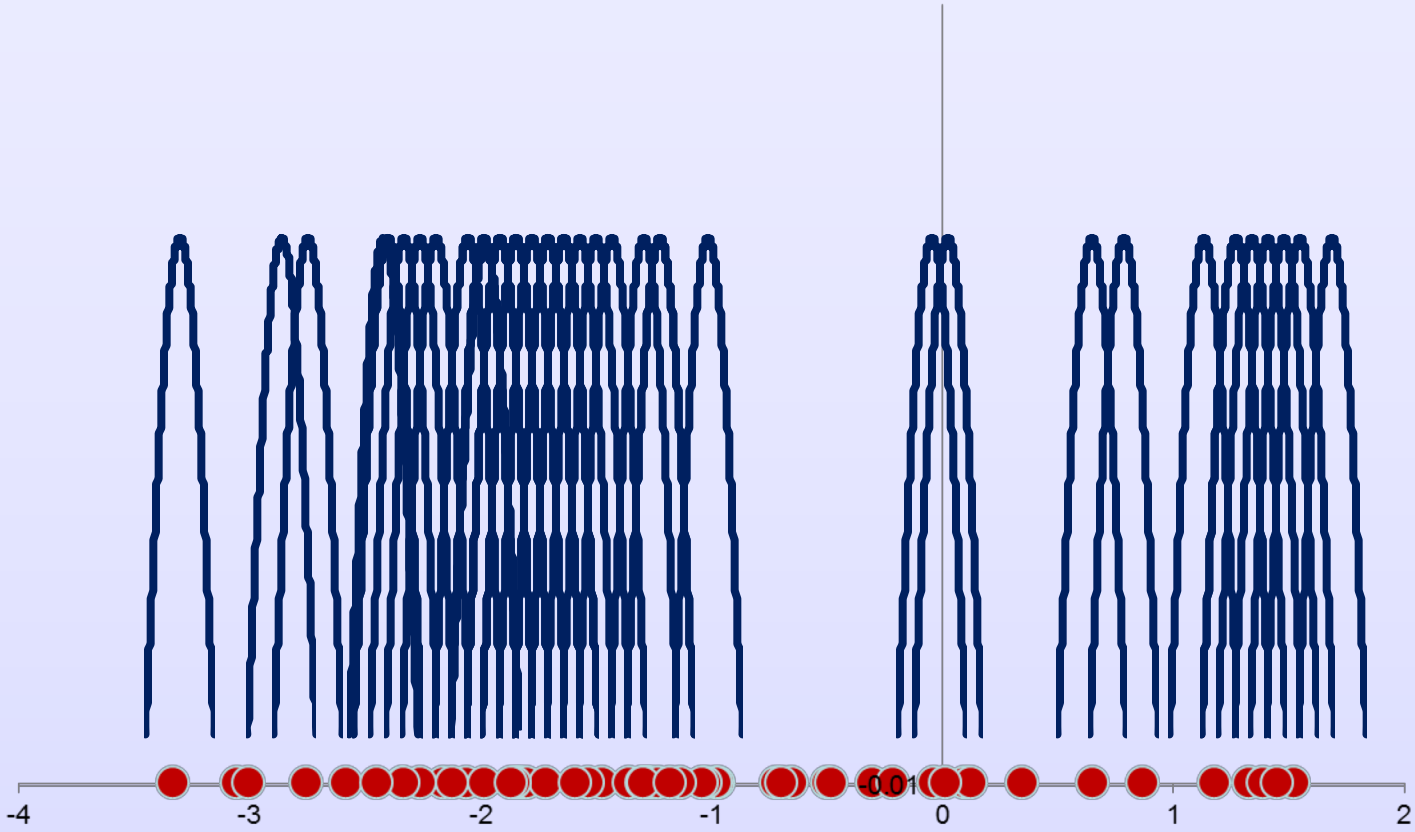
$$\hat{\mu} = \frac{\sum_i x_i}{m}$$

$$\hat{\sigma}^2 = \frac{\sum_i (x_i - \hat{\mu})^2}{m}$$



$$X \sim \mathcal{N}(\hat{\mu}, \hat{\sigma}^2) = \frac{1}{\sqrt{2\pi\hat{\sigma}^2}} \exp\left(-\frac{(X - \hat{\mu})^2}{2\hat{\sigma}^2}\right)$$

1-D Regression



$$X \sim \sum_{i=1}^{m} \mathcal{N}(x_i, s)$$

Summary: Naïve Bayes

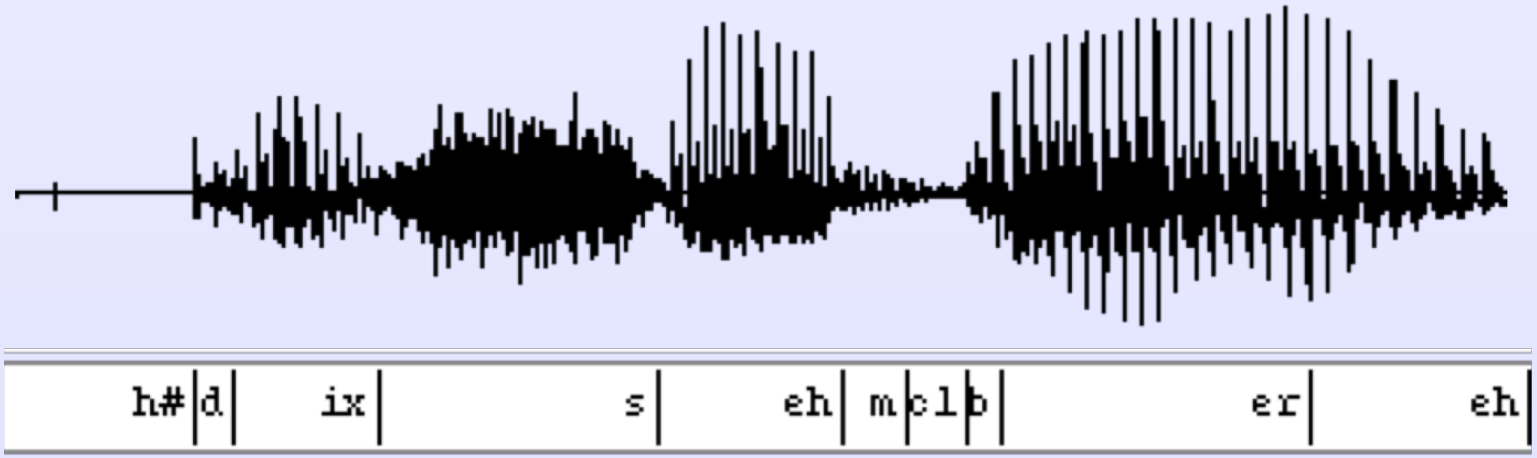
- Models:
  - Multinomial distribution over labels
  - Multinomial distributions over words/features given label
- Modeling choices:
  - Generative
  - Parametric
  - Word appearance are I.I.D
  - Small & Simple
- Inference:
  - Optimal Bayes using estimated model
- Learning:
  - Count and Smooth

Multi Class Single Label

Elimination of a single class is not enough

Hierarchical Classification

Phonetic transcription of DECEMBER

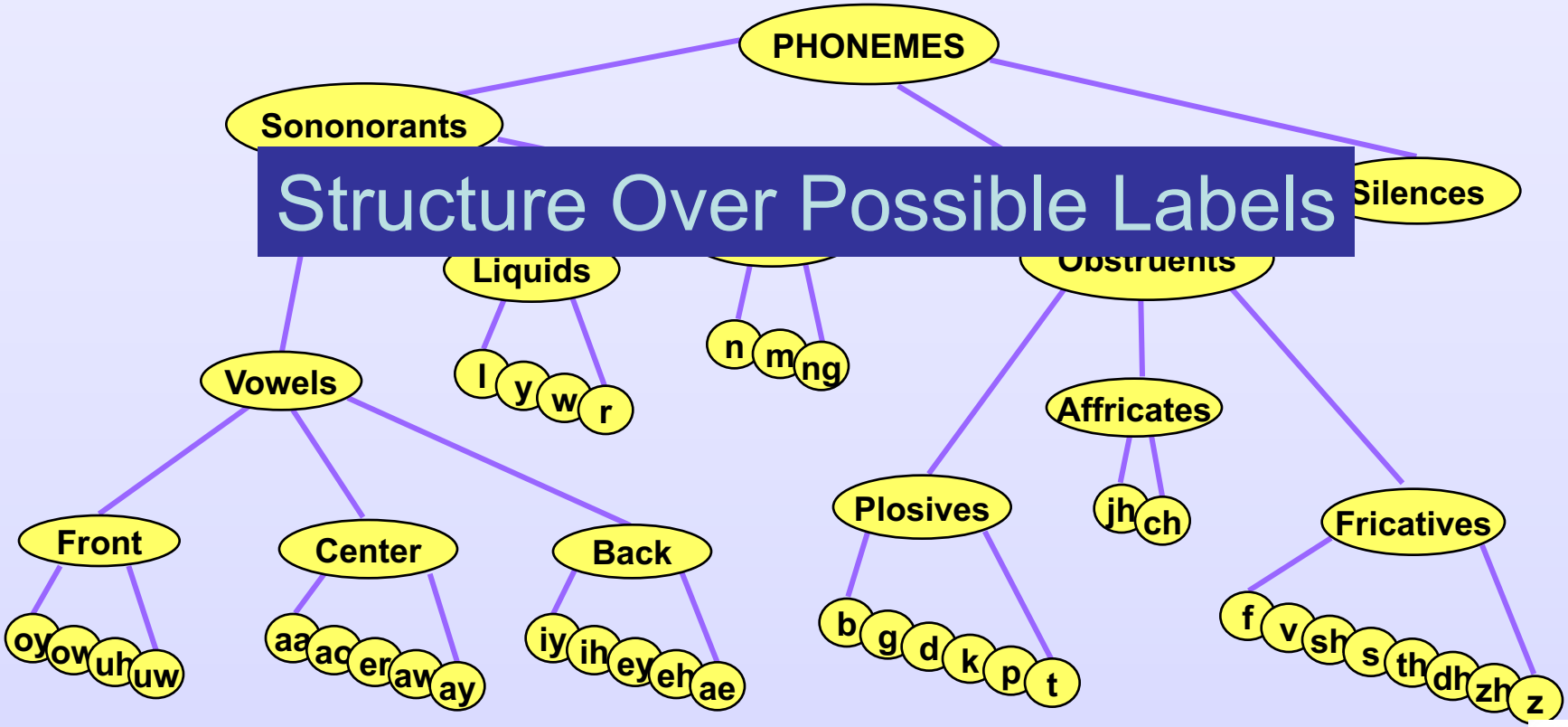


Gross error `d ix CH eh m bcl b er`

Small errors `d AE s eh m bcl b er`

`d ix s eh NASAL bcl b er`

Phonetic Hierarchy



Structure Over Possible Labels

Multi-Class Multi-Label

Document

The higher minimum wage signed into law… will be welcome relief for millions of workers … . The 90-cent-an-hour increase … .

Relevant topics

- **REGULATION / POLICY**
- **CORPORATE / INDUSTRIAL**
- **GOVERNMENT / SOCIAL**
- **MARKETS**
- **LABOUR**
- **ECONOMICS**

Full topic ranking

- **ECONOMICS**
- **CORPORATE / INDUSTRIAL**
- **REGULATION / POLICY**
- **MARKETS**
- **LABOUR**
- **GOVERNMENT / SOCIAL**
- **LEGAL/JUDICIAL**
- **REGULATION/POLICY**
- **SHARE LISTINGS**
- **PERFORMANCE**
- **ACCOUNTS/EARNINGS**
- **COMMENT / FORECASTS**
- **SHARE CAPITAL**
- **BONDS / DEBT ISSUES**
- **LOANS / CREDITS**
- **STRATEGY / PLANS**
- **INSOLVENCY / LIQUIDITY**

Multi-Class Multi-Label

Document

The higher minimum wage signed into law… will be welcome relief for millions of workers … . The 90-cent-an-hour increase … .

Relevant topics

- REGULATION / POLICY
- CORPORATE / INDUSTRIAL
- GOVERNMENT / SOCIAL
- MARKETS
- LABOUR
- ECONOMICS

Full topic ranking

- ECONOMICS
- INSOLVENCY / LIQUIDITY
- CORPORATE / INDUSTRIAL
- REGULATION / POLICY
- COMMENT / FORECASTS
- LABOUR
- REDITS
- ICIAL
- REGULATION/POLICY
- SHARE LISTINGS
- GOVERNMENT / SOCIAL
- PERFORMANCE
- ACCOUNTS/EARNINGS
- MARKETS
- SHARE CAPITAL
- BONDS / DEBT ISSUES
- STRATEGY / PLANS

Non-trivial Evaluation Measures

Recall Precision Any Error? No. Errors?

Noun Phrase Chunking

Estimated volume was a light 2.4 million ounces .

Simultaneous Labeling

Named Entity Extraction

Bill Clinton and Microsoft founder Bill

Interactive Decisions

Gates met today for 20 minutes .

[McDonald06]

Sentence Compression

- The Reverse Engineer Tool is available now and is priced on a site-licensing basis , ranging from $8,000 for a single user to $90,000 for a multi

Complex Input – Output Relation

- Essentially , design recovery tools read existing code and translate it into the language in which CASE is conversant -- definitions and structured diagrams .

Dependency Parsing

John hit the ball with the bat

Non-trivial Output