Text Mining
Statistical Modeling of Textual Data
Lecture 1

Måns Magnusson, Mattias Villani

Division of Statistics and machine learning
Dept. of Computer and Information Science
Linköping University

Overview 'Probabilistic text modeling'

- **Probabilistic text analysis**
- **Text classification**
- **Text clustering**
- **Topic models**

Overview

- Probabilistic text analysis
- Some probability theory (and important distributions)
- Statistical inference
- Example: Language models
- Example: PoS-tagging

Section 1

Probabilistic text analysis

Probabilistic machine learning

- "Classical" Machine learning: **The toolbox view**
  - Feed data into model and do inference
  - If not working, find an other tool or duck tape...
- Probabilistic Machine Learning: **Model view of your data**
  - Specify your problem as a probabilistic model
  - Do inference conditioned on data: $p(\Theta|\mathbf{w})$
  - If not working, diagnose problems and extend model.
- Another perspective from previous parts of the course

Probabilistic machine learning

- Create (or define) a **model** using **probability theory** and unknown model **parameters**
  - Generative models $p(\mathbf{y}, \mathbf{x})$
  - Discriminative models $p(\mathbf{y}|\mathbf{x})$
  - Can always simulate data from your model.
- Infer the unknown parameters in the model using **generic** inference procedures
  - MCMC, Variational Bayes, Maximum Likelihood, Maximum aposterior
- Inference (learning) and model are two different things!

PROBABILISTIC MODELING OF TEXT

- Assume a probabilistic (or statistical) generative model

$$p(\mathbf{w}_1^n) = p(w_1, w_2, w_3, ..., w_n)$$

  where w_i is a word/token.

- Can use different structures in texts

$$p(\mathbf{w}_1^n|\mathbf{x}) \text{ or } p(\mathbf{w}_1^n, \mathbf{x})$$

  - Sentences, documents etc.

- **Example:** Generative model for a simple unigram model
  - For all words 1 to n
    - $w_n \sim \mathit{Multinomial}(\theta)$

Special issues with probabilistic modeling of text

- Discrete
- High dimensional
- Sparse

Some definitions

"A neutron walks into a bar and asks how much for a drink. The bartender replies 'for you, no charge'."

- Tokens
- Types / word types
- Vocabulary
- Sentence / Document / text segment / context
- Corpus

Section 2

Some probability theory (and important distributions)

Recap: Probability

- **We want to:** Formulate our model in probabilistic terms, i.e. **probability distributions**
- Probability distribution: $p(A)$
  - A function $p(\cdot)$ that gives a probability for an event A
  - **Example:** $p(\text{HEAD}) = 0.5$
  - **Example:** $p(-2 < X < 0) = 0.5$
- Parameters governs probability distributions;
  - **Example:** μ and σ^2 in the Normal distribution
- *Conditional probability:*
  $p(A|B) = \frac{p(A,B)}{p(B)} \iff p(A, B) = p(A|B) \cdot p(B)$
  - **Example:** $p(x > 3|\mu = 3, \sigma^2 = 1)$ where $X \sim N(\mu, \sigma^2)$
  - **Example:** $p(x > 3|z = 1)$ where $X \sim N(z, 1 + z)$ and $z \sim Bernoulli(p)$
- *Chain rule of probability*:
  $p(A_1, ..., A_k) = p(A_1|A_2, ..., A_k) \cdot p(A_2, ..., A_k)$

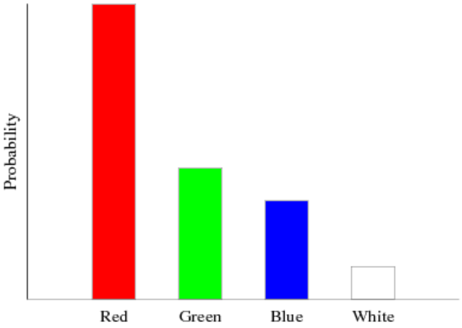
MULTINOMIAL DISTRIBUTION (MN)*

- **Multinomial distribution**: random *discrete* variable $X \in \{1, 2, ..., K\}$ that can assume exactly one of K (unordered) values with values n_k.
  - Probability distribution for categories (words frequencies)
  - $Pr(X = k) = \theta_k$
  - Parameters: $\theta = (\theta_1, ..., \theta_K)$ where $\sum \theta_j = 1$ and all $\theta_j > 0$
- **Probability mass function**:

$$
\begin{aligned}
p(\mathbf{n}|\theta) &= \frac{N!}{n_1! \cdots n_K!} \theta_1^{n_1} \cdots \theta_K^{n_k} \\
&= \frac{\Gamma(\sum_k n_k + 1)}{\prod_k \Gamma(n_k + 1)} \theta_1^{n_1} \cdots \theta_K^{n_k}
\end{aligned}
$$

- **Categorical distribution:** Multinomial with one draw, $\sum^K n_k = 1$
- **Bernoulli distribution:** Multinomial with only two classes with parameter θ and $1 - \theta$
- **Example:**
  - A dice is a $Multinomial(\theta)$ with $J = 6$ and all $\theta_j = 1/6$

Multinomial distribution (MN)



Figur: Source: `https://izbicki.me`

MULTIVARIATE BERNOULLI

- Multivariate random **vector** $X = (X_1, ..., X_K)$ of binary outcomes (i.e. $(0, 1, 1, ..., 0, 0)$).
    - Parameters: $\mathbf{p} = (p_1, ..., p_K)$ for $j = 1, ..., K$ where all $1 \geq p_j \geq 0$
- **Probability mass function**:

$$
\begin{aligned}
p(X) &= \text{assume independence} \\
&= \prod_{i=1}^{K} p(X_k) = \prod^{K} p_k^{x_k}(1-p_k)^{x_k-1}
\end{aligned}
$$

DIRICHLET DISTRIBUTION*

- **Dirichlet distribution**: random **vector** $X = (X_1, ..., X_K)$ satisfying the constraint $X_1 + X_2 + ... + X_K = 1$.
  - Unit simplex (Probability distribution over proportions)
  - Parameters: $\alpha = (\alpha_1, ..., \alpha_K)$ for $j = 1, ..., K$ where all $\alpha_j > 0$
  - Uniform distribution: $\alpha = (1, 1, ..., 1)$
  - Small variance (informative) when the α's are large.
  - "Bathtub shape" when $\alpha_k < 1$ for all k.
- **Probability density function**:

$$p(\mathbf{x}|\boldsymbol{\alpha}) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^{K} x_i^{\alpha_i - 1}$$

where

$$\mathrm{B}(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}$$

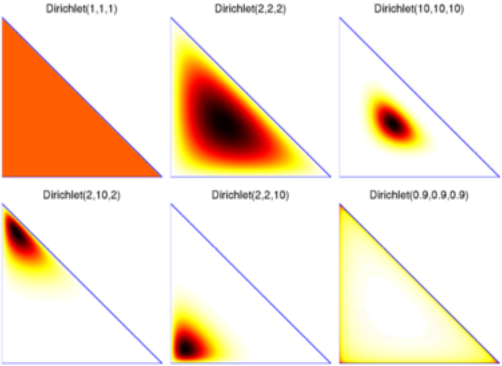
Dirichlet distribution

- **Beta distribution:** A Dirichlet distribution with only two parameters α_1 and α_2
- **Expected value** and **variance** of the *Dirichlet*$(\alpha_1, ..., \alpha_K)$ distribution

$$\mathrm{E}(\theta_b) = \frac{\alpha_b}{\sum_{j=1}^{B} \alpha_j} \qquad \mathrm{V}(\theta_b) = \frac{\mathrm{E}(\theta_b)\left[1 - \mathrm{E}(\theta_b)\right]}{1 + \sum_{j=1}^{B} \alpha_j}$$

- **Example:** A random proportion

DIRICHLET DISTRIBUTION



FIGUR: Source: https://csail.mit.edu

Section 3

Statistical inference

Inference in probabilistic models

- Given data, $\mathbf{w}$, parameters Θ and the model (likelihood) $p(\mathbf{w}|\theta)$
  - learn the parameters
- **Bayesian inference**

$$p(\theta|\mathbf{w}) = \frac{p(\mathbf{w}|\theta) \cdot p(\theta)}{p(\mathbf{w})}$$

- **Maximum likelihood inference**
  - Identify parameters $\hat{\theta}$ that maximize $p(\mathbf{w}|\theta)$
- *Difference*
  - Priors on all parameters: $p(\theta)$
  - Posterior probability / point estimate of θ

Maximum likelihood inference for multinomial data

- **Data**: $y = (n_1, ...n_K)$, where n_k counts the number of observations in the kth category. $\sum_{j=1}^{K} n_j = N$.
- **Example:** A recent survey among consumer smartphones owners in the U.S. showed that among the N =513 respondents:
  - $n_1 = 180$ owned an iPhone
  - $n_2 = 230$ owned an Android phone
  - $n_3 = 62$ owned a Blackberry phone
  - $n_4 = 41$ owned some other mobile phone.

MAXIMUM LIKELIHOOD INFERENCE FOR MULTINOMIAL DATA

- Let $\theta_1 = Pr(\text{owns iPhone})$, $\theta_2 = Pr(\text{owns Android})$ etc
- **Likelihood**

$$p(n_1, n_2, ..., n_K | \theta_1, \theta_2, ..., \theta_K) = \frac{N!}{n_1! \cdots n_K!} \prod_{j=1}^{K} \theta_j^{n_j}$$

- **Maximum likelihood** (ML) estimator

$$\hat{\theta}_k = \frac{n_k}{N}$$

- **ML problematic when data is sparse.** $n_k = 0 \Rightarrow \hat{\theta}_k = 0$.

BAYESIAN INFERENCE FOR MULTINOMIAL DATA

$$p(\theta|\mathbf{w}) = \frac{p(\mathbf{w}|\theta) \cdot p(\theta)}{p(\mathbf{w})}$$

- **Prior:** $p(\theta) \sim \text{Dirichlet}(\alpha_1, ..., \alpha_K)$ with density

$$p(\theta_1, \theta_2, ..., \theta_K) \propto \prod_{j=1}^{K} \theta_j^{\alpha_j - 1}$$

  is *conjugate* to the multinomial

- **Posterior** distribution (Likelihood $\times$Prior)

$$\theta|n_1, ..., n_K \sim \text{Dirichlet}(n_1 + \alpha_1, ..., n_K + \alpha_K)$$

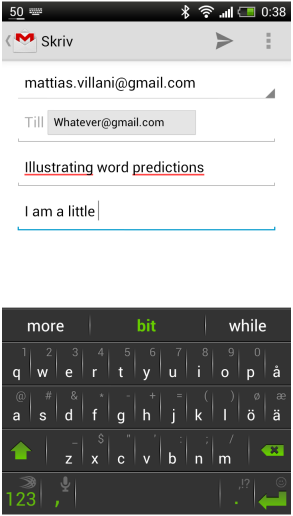
- **Posterior expected value**

$$E(\theta_k|n_1, ..., n_K) = \frac{n_k + \alpha_k}{N + \sum_{j=1}^{K} \alpha_j}$$

Section 4

EXAMPLE: LANGUAGE MODELS

EXAMPLE: LANGUAGE MODELS



Probabilistic language models

- Let w_i denote the ith word in a text segment. Let $\mathbf{w}_1^k = w_1 w_2 \cdots w_k$ denote a text with k tokens.
- The probability of a text (using chain rule of probability)

$$p(\mathbf{w}_1^n) = p(w_1) \cdot p(w_2|w_1) \cdot p(w_3|\mathbf{w}_1^2) \cdots p(w_n|\mathbf{w}_1^{n-1})$$

- Probability distribution over the next token in a sentence:

$$p(w_k|\mathbf{w}_1^{k-1})$$

- Example:

$$p(\text{mall}|\text{I like to go to the}) = 0.2$$

$$p(\text{school}|\text{I like to go to the}) = 0.001$$

(Add beginning of sentence token/tag <s>)

UNIGRAM MODELS

- **Unigram language models** ignores the previous words **and** the order of the words:

$$p(w_n|w_1, ..., w_{n-1}) = p(w_n)$$

- **Bag-of-word assumption**
- Simulating a text from a bag-of-words model gives rubbish:

  "much asks into neutron asks. "

- Generative model: $p(\mathbf{w}|\theta)$
  - $\theta \sim \text{Dir}(\alpha)$ (prior)
  - For all 1 to n
    - $w_n \sim Multinomial(\theta)$ (likelihood)

Unigram models

- $p(w_n)$ can be estimated using **maximum likelihood (ML)** estimation as:

$$\hat{\theta}_v = \frac{C(v_n)}{N}$$

where $C(v_n)$ is the number of tokens of word type v_n (no prior)

- **Problem with MLE:** words not in training corpus are deemed impossible!

$$C(v_n) = 0 \Rightarrow \hat{\theta} = 0$$

UNIGRAM MODELS

- $p(w_n)$ can be estimated using **Bayesian** inference as:

$$E(\theta_v) = \frac{C(v_n) + \alpha}{N + V\alpha}$$

with Dirichlet prior.
- **Add-one (Laplace) smoothing** obtained with uniform prior $\alpha_1 = ... = \alpha_B = 1$

$$E(\theta_v|\mathbf{w}) = \frac{C(v_n) + 1}{N + V}$$

- *Most smoothing techniques are just different priors!*

LANGUAGE MODELS - N-GRAMS

- The **bigram** model

$$p(w_n|w_1, ..., w_{n-1}) = p(w_n|w_{n-1})$$

- **Trigram model**: $p(w_n|w_{n-1}, w_{n-2})$ and so on.
- **n-grams** looks for pairs of consecutive words $w_1 w_2 ... w_n$.
- **Heaps law**: $V \approx \sqrt{N}$.
- n-grams can have a **huge outcome space** $B = V^n$.

LANGUAGE MODELS - N-GRAMS

- **ML** estimate:

$$\hat{p}(w_n|w_{n-1}) =$$

$$\hat{\theta}_{v(n)|v(n-1)} = \frac{\text{Number of times word type } v_n \text{ follows directly after } v_{n-1}}{\text{Number of times } v_{n-1} \text{appears in the text}}$$

where $v(n)$ is the word type at position n.

- Alternative formulation

$$\hat{p}(w_n|w_{n-1}) = \frac{C(v_{n-1},v_n)}{C(v_{n-1})}$$

- **Problem with MLE:** n-grams

$$C(v_{n-1},v_n) = 0 \Rightarrow \hat{\theta}_{v(n)|v(n-1)} = 0$$

- Lots of n-grams are unseen in training corpus. **Sparsity** problems!

THE SPARSITY PROBLEM - N-GRAMS

- **Bayesian** estimation (smoothing for bigrams, Dirichlet prior again.)

$$E(\theta_v|\cdot) = \frac{C(v_{n-1},v_n) + \alpha}{C(v_{n-1}) + \alpha V}$$

- **Again:** *Most smoothing techniques are just different priors!*

Section 5

Example: PoS-tagging

A PROBABILISTIC MODEL FOR POS TAGGING

- **Part-of-Speech** (**PoS**) or **word classes** - verb, noun, adjective, preposition etc:
- **PoS tagging**: determine the sequence of POS tags

$$t_1^n = t_1 t_2 \cdots t_n$$

for the words in the sentence

$$w_1^n = w_1 w_2 \cdots w_n$$

- **Note**: each word gets a PoS tag

$$\begin{array}{cccc} w_1 & w_2 & \cdots & w_n \\ t_1 & t_2 & \cdots & t_n \end{array}$$

- We add tags to our probabilistic model.

$$p(\mathbf{w}) = p(\mathbf{w}, \mathbf{t})$$

A PROBABILISTIC MODEL FOR POS TAGGING, CONT.

- Two simplifying model assumptions makes the problem manageable.
- **Assumption 1**: **each word depends only on its tag**:

$$p(\mathbf{w}|\mathbf{t}) = \prod_{i=1}^{n} p(w_i|t_i)$$

- **Assumption 2**: **Bigram assumption** for the **tags** :

$$p(\mathbf{t}) = \prod_{i=1}^{n} p(t_i|t_{i-1})$$

- **Hidden Markov model (HMM)**
- Reduces the dimensionality so n-gram HMM is feasible.

A PROBABILISTIC MODEL FOR POS TAGGING, CONT.*

- Generative model $p(\mathbf{w}|\Theta)$
  - For all 1 to T (prior)
    - $\phi_t \sim \text{Dir}(\alpha)$ (transition probabilities)
    - $\theta_t \sim \text{Dir}(\beta)$ (emission probabilities)
  - For all 1 to n (likelihood)
    - $t_n \sim \mathit{Categorical}(\phi_{t_{n-1}})$
    - $w_n \sim \mathit{Categorical}(\theta_{t_n})$

PART-OF-SPEECH TAGGING, PARAMETER INFERENCE

- Assume we know both **w** and **t** on a training set.
- The PoS parameters can be estimated using ML

$$\hat{p}(t_i|t_{i-1}) = \hat{\phi}_{t_i,t_{i-1}} = \prod_{i=1}^{n} p(t_i|t_{i-1}) = \frac{C(t_i, t_{i-1})}{C(t_{i-1})}$$

as a bigram model from a tagged corpus, or using a bayesian approach

$$E(\phi_{t_i,t_{i-1}}) = \frac{C(t_i, t_{i-1}) + \alpha}{C(t_{i-1}) + T\alpha}$$

- The word distribution (emission) $p(w_i|t_i)$ can be estimated by (MLE)

$$\hat{p}(w_i|t_i) = \hat{\theta}_{w_i,t_i} = \frac{C(t_i, w_i)}{C(t_i)}$$

or using a Bayesian approach

$$\hat{p}(w_i|t_i) = E(\theta_{w_i,t_i}) = \frac{C(t_i, w_i) + \alpha}{C(t_i) + \alpha V}$$