Introduction to Statistical NLP

Corina Forăscu
corinfor@info.uaic.ro

Chris Manning and Hinrich Schütze, *Foundations of Statistical Natural Language Processing*

MIT Press. Cambridge, 1999

Why Stats in NLP

- **Processing**: use probabilistic & statistical models or algorithms to process natural language input or output
- **Learning**: use inferential statistics to learn from examples (corpus data), or even estimate the parameters of probabilistic models that can be used in processing
- **Evaluation**: use statistics to assess the performance of language processing systems

Statistics in NLP

Probability theory: mathematical theory of uncertainty (***random experiments***)

$$P(x) = P(\{u \in \Omega \mid u \mapsto x\})$$

Descriptive statistics: Methods for summarizing (large) datasets

$$f_n(x) = \frac{C(x)}{n}$$

Inferential statistics: Methods for drawing inferences from (large) datasets

$$P(x) = f_n(x) \pm i$$

Probability theory (1)

- Mathematical theory of uncertainty, based on *random experiments* (throw a dice, predict the weather)
- Models of stochastic (non-deterministic) systems

For an experiment *E* , we denote by *X* its result from Ω – finite set where *X* takes values, *sample space*

- *Event* – any subset of Ω
- X – *random variable*
- *Probability function P:* Ω → [0, 1], *P(X=x)*

  P(A) ≥ 0 (for any event A).

  P(Ω) = 1

  If A and B are disjoint events, then P(A ∪ B) = P(A) + P(B).

Probability theory (2)

$P(\Omega - A) = 1 - P(A)$

$P(\emptyset) = 0$

$P(A \cup B) = P(A) + P(B) - P(A \cap B)$

$P(A \cap B) \leq P(A \cup B) \leq P(A) + P(B)$

If $A \subseteq B$, then $P(B - A) = P(B) - P(A)$

If $(A_1, ..., A_n)$ is a partitioning of Ω, then

$P(B) = P(A_1 \cap B) + ... + P(A_n \cap B)$ (*law of total probability*).

Conditional probability

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

$P(A \cap B) = P(A)\,P(B|A) = P(B)\,P(A|B)$

(*chain/multiplication rule*)

$P(A|B) = P(A)\,P(B|A) / P(B)$ (*Bayes theorem*)

A, B – ***independent*** if the following three (equivalent) conditions hold:

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

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

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

Conditional probabilities in practice

p(grammar | sentences).
p(parse | sentence).
p(ml-tag |word)
p(sem-tag|word)
p(tag1tag2…tagn| word1word2…wordn)
p(syn-rel | word1, tag1, word2, tag2)

Probability theory: stochastic variables

A ***stochastic variable*** is a function X from a sample space Ω to a *value space* Ω_X.

X is a *numeric variable* if Ω_X is a subset of the set of real numbers

X is a *discrete variable if* Ω_X is finite or countable infinite

$f_X(x) = P(X = x)$ - *frequency function* (or probability function) f_X giving the probability of each possible value x of X

If X is discrete: $f_X(x) = P(\{u \in \Omega \mid X(u) = x\}) = \sum_{u\,:\,X(u) = x} P(u)$

If X is numeric, F_X is the *distribution function* $F_X(x) = P(X \leq x)$

If X is discrete numerical: $F_X(x) = \sum_{y \leq x} f_X(y)$

Parameters of the stochastic variables

For X a discrete numerical variable:

E(X) = $\sum_{x \in \Omega X} x \cdot f_X(x)$ - *expectation* is the (weighted) average value

Var(X) = $\sum_{x \in \Omega X} (x - E(X))^2 \cdot f_X(x)$ – *variance* is the expected

(squared) deviation from the expectation

$$Var(X) = E((X - E(X))^2) = E(X^2) - E^2(X)$$

H(X) = $- \sum_{x \in \Omega X} f_X(x) \cdot \log_2 f_X(x)$ – *entropy* is the expected amount of information (measured in bits) when learning the value of X

I(w_1:w_2) = $P(w_1,w_2)/P(w_1)*P(w_2)$ – *mutual information* measures how “interesting” is a given sequence (w_1, w_2)

Eveniments

$f_{(X, Y)}(x, y) = P(X = x, Y = y) =$
$= P(\{ (u, v) \in \Omega_X \times \Omega_Y \mid X(u) = x, Y(v) = y \})$ –
- joint probability of X and Y

$f_{X|Y}(x \mid y) = P(X = x \mid Y = y) = P(X = x, Y = y) / P(Y = y)$ –
- conditional probability of X given Y

$H(X, Y) = - \sum_{x \in \Omega X} \sum_{y \in \Omega Y} f_{(X, Y)}(x, y) \cdot \log_2 f_{(X, Y)}(x, y)$ – joint entropy

$H(X|Y) = - \sum_{x \in \Omega X} \sum_{y \in \Omega Y} f_{(X, Y)}(x, y) \cdot \log_2 f_{X|Y}(x \mid y)$ – conditional entropy

$H(Y|X) = H(X) + H(Y|X) = H(Y) + H(X|Y)$

$I(X; Y) = H(X) - H(X|Y)$ – **mutual information** – amount of information X contain about Y

X and Y are *independent* if and only if
$P(X = x, Y = y) = P(X = x) P(Y = y)$, for all x and y

Independent eveniments

For X and Y independent:

$P(X = x|Y = y) = P(X = x)$ for all x, y

$P(Y = y|X = x) = P(Y = y)$ for all x, y

$H[X|Y] = H[X]$

$H[Y|X] = H[Y]$

Example

Consider the experiment of randomly choosing a pair of two adjacent words from a text. Let X_1 be the stochastic variable which maps the first word to its word class (POS), and let X_2 be the stochastic variable which maps the second word to its word class. Suppose we know the following probabilities:

$P(X_2 = \text{noun}) = 0.2$
$P(X_2 = \text{adjective}) = 0.05$
$P(X_1 = \text{article} \mid X_2 = \text{noun}) = 0.3$
$P(X_1 = \text{article} \mid X_2 = \text{adj}) = 0.6$
$P(X_1 = \text{article} \mid X_2 \text{ is neither noun nor adjective}) = 0$

Example

$P(X_1\text{=article}) =$
$P(X_2\text{=noun})\ P(X_1\text{=article}|X_2\text{=noun}) + P(X_2\text{=adjective})\ P(X_1\text{=article}|X_2\text{=adjective}) =$
$0.06 + 0.03 = 0.09$

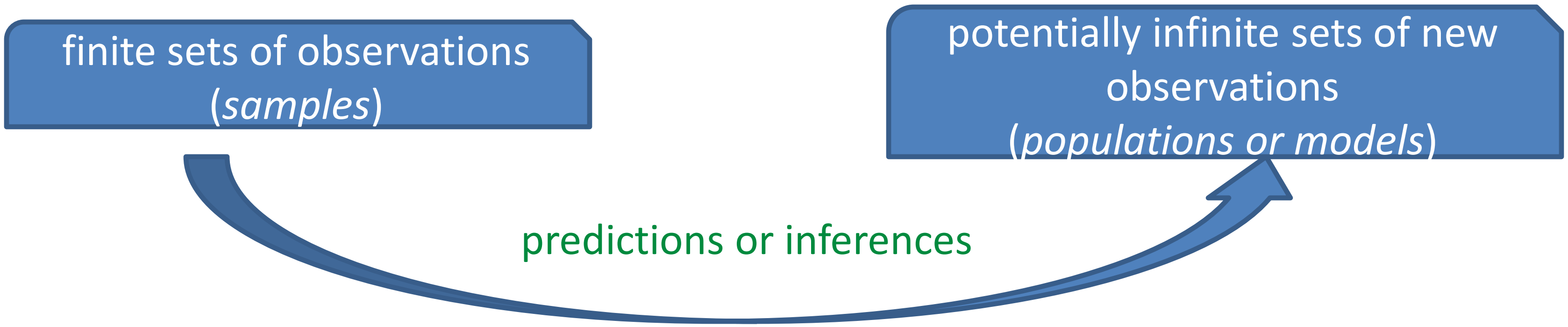
$P(X_2\text{=noun}|X_1\text{=article}) =$
$P(X_2\text{=noun})\ P(X_1\text{=article}|X_2\text{=noun}) / P(X_1\text{=article}) =$
$0.06 / 0.09 = 0.67$

$P(X_2\text{=adjective}|X_1\text{=article}) =$
$P(X_2\text{=adjective})\ P(X_1\text{=article}|X_2\text{=adjective}) / P(X_1\text{=article}) =$
$0.03 / 0.09 = 0.33$

$P(X_2\text{=noun or adjective}|X_1\text{=article}) =$
$1 - 0 = 1$

X_1 and X_2 are not independent:
$P(X_1\text{=article},X_2\text{=noun}) = P(X_2\text{=noun})\ P(X_1\text{=article}|X_2\text{=noun}) = 0.06$,
while $P(X_1\text{=article})\ P(X_2\text{=noun}) = 0.018$.

Statistical Inference



- *random sample of X*: a vector $(X_1, \ldots, X_n)$ of independent variables X_i with the same distribution as X
- *statistical material*: a vector $(x_1, \ldots, x_n)$ of values such that $X_i = x_i$ in a particular experiment

Types of statistical inference

- **Estimation**: Use samples and sample variables to predict population variables
    - **Point estimation**: Use sample variable $f(X_1, ..., X_n)$ to estimate parameter ϕ. **(MLE)**
    - **Interval estimation**: Use sample variables $f_1(X_1, ..., X_n)$ and $f_2(X_1, ..., X_n)$ to construct an interval such that $P(f_1(X_1, ..., X_n) < \phi < f_2(X_1, ..., X_n)) = p$, where p is the confidence level adopted
- **Hypothesis testing**: Use samples and sample variables to test hypotheses about populations and population variables.

Maximum Likelihood Estimation (MLE)

Chooses the estimate that maximizes the probability of the statistical material:

Given a statistical material $(x_1, ..., x_n)$ and a set of parameters θ, the **likelihood function** L is:

$L(x_1, ..., x_n, \theta) = \prod_i P_\theta(x_i)$

where $P_\theta(x_i)$ is the probability that the variable X_i assumes the value x_i given a set of values for the parameters in θ

Maximum likelihood estimation means choosing θ so that the likelihood function is maximized:

$\max_\theta L(x_1, ..., x_n, \theta)$

MLE is a good solution to the estimation problem if the statistical material is large enough.

Hypothesis testing

1. Choose a test statistic t whose distribution is known when the null hypothesis is true.
2. Use t to calculate the probability p of observing the data given that the null hypothesis is true.
3. If $p < \alpha$, reject the null hypothesis, where α is the significance level adopted.

What next

1. Statistical estimators & combined estimators
2. Find collocations in text (mutual information)
3. n-gram models over sparse data
4. …
5. WSD
6. POS tagging
7. Lexical acquisition
8. Probabilistic CFG
9. Statistical alignment
10. MT
11. …

NLP (stat) tools

1. RACAI's linguistic web services (text processing, factored translation – includes Romanian): http://www.racai.ro/webservices/
2. RACAI's Wordnet browser (Romanian, English): http://www.racai.ro/wnbrowser/
3. CMU-Cambridge Statistical Language Modeling toolkit for construction & testing of statistical language models http://svr-www.eng.cam.ac.uk/~prc14/toolkit.html
4. openNLP MAXENT package for training and using maximum entropy models http://maxent.sourceforge.net/
5. SVMTool generator of sequential taggers based on Support Vector Machines (SVM) http://www.lsi.upc.es/~nlp/SVMTool/
6. LIBSVM -- A Library for Support Vector Machines: http://www.csie.ntu.edu.tw/~cjlin/libsvm/
7. Weka collection of machine learning algorithms for data mining tasks (data pre-processing, classification, regression, clustering, association rules, and visualization) http://www.cs.waikato.ac.nz/ml/weka/
8. CLUTO - Family of Data Clustering Software Tools http://glaros.dtc.umn.edu/gkhome/views/cluto/
9. GIZA++: Training of statistical translation models: http://www.fjoch.com/GIZA++.html
10. Ted Pedersen's NLP software: http://www.d.umn.edu/~tpederse/code.html

Recommended readings

Chris Manning and Hinrich Schütze, *Foundations of Statistical Natural Language Processing*

1. Krenn, B. & Samuelsson, C. (1997) *The Linguist's Guide to Statistics*.
2. http://nlp.stanford.edu/links/statnlp.html
3. http://nlp.stanford.edu/fsnlp/
4. http://www-a2k.is.tokushima-u.ac.jp/member/kita/NLP/nlp_tools.html