[Day-1] Introduction to Statistical Natural Language Processing

(Appendix: Related Techniques)

Keh-Yih Su and Jing-Shin Chang
kysu@bdc.com.tw, shin@nlp.csie.ncnu.edu.tw

(2002/8/17)

Behavior Design Corporation (BDC)
No. 5, 2F, Industrial East Road IV, Science-Based Industrial Park
Hsinchu, Taiwan 30077, R.O.C.

Department of Computer Science and Information Engineering
National Chi-Nan University
Puli, Nantou, Taiwan 545, R.O.C.

Day-1: Introduction to Statistical Natural Language Processing (mainly on Supervised Learning)

- Part I: Introduction (1)
  - Problems and Characteristics of Natural Language Processing
- Part II: Introduction (2)
  - What, When and Why Statistical Approach
- Part III: Basic Concepts and Background
  - Feature Space, Probability, Estimator, Stochastic Process, Data Set Classification, and Performance Measure
- Part IV: Typical Applications
  - Word Segmentation, Tagging, Selecting Parse Tree, Aligning Bilingual Corpus
- Part V: Techniques for Improving Performance
  - Smoothing, Class-Based Model, Adaptive Learning, Tips for Checking
- Part VI: Advanced Topics: SVM, ME
  - Support Vector Machine, Maximum Entropy Models
- **Appendix: Related Techniques**
  - **Parameter Estimation, Fractional Factorial Experiment Design, Decision Tree**

Appendix: Related Techniques

- Parameter Estimation
  - Basic Concepts
  - Maximum Likelihood Estimation
- Fractional Factorial Experiment Design
  - Design Procedure
- Decision Tree
  - CART (Classification and Regression Tree)

Parameter Learning via Estimation: Estimator Characteristics (1)

- Statistic:
  - A *statistic* is any real or vector-valued function of the observation (e.g., $T(\mathbf{X})$).
    - e.g. X: Head/Tail of a fair coin; T: number of heads
    - Performance measure (e.g., error rate) is also a statistic
- Estimators:
  - An estimator is a statistic calculated from sample data that provide either point estimates or interval estimates for some population parameter.

Parameter Learning via Estimation: Estimator Characteristics (2)

- Unbiased:
  - An estimator $\hat{\theta}$ is unbiased if its mean is equal to the population parameter θ being estimated, i.e., $E\left[\hat{\theta}\right] = \theta$
- Efficiency:
  - An estimator $\hat{\theta}$ of θ is said to be more efficient than any other unbiased estimator $\hat{\hat{\theta}}$ if $\mathrm{var}(\hat{\theta}) \le \mathrm{var}(\hat{\hat{\theta}})$
  - An estimator is a minimum variance unbiased estimator if the variance of its sampling distribution is the smallest of all other unbiased estimators.

Parameter Learning via Estimation: Estimator Characteristics (3)

- Consistency: $\lim_{n\to\infty} P\left(\left|\hat{\theta}(n) - \theta\right| \ge \varepsilon\right) = 0$
  - An estimator is said to be a consistent estimator if it converges to the parameter to be estimated in the probability sense. An estimator is a consistent one if the following conditions are met:
    - Asymptotically unbiased
    - Variance converges to 0 when sample size *n* gets large
  - Example:
    - Mean: $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} X_i$, $E\left[\hat{\mu}\right] = \mu$, $Var\left[\hat{\mu}\right] = \frac{\sigma^2}{n}$, $\lim_{n\to\infty} Var\left[\hat{\mu}\right] \to 0$
    - Probability estimator: relative frequencey
    - MLE estimator for mean and variance for Gaussian

Parameter Learning via Estimation: Estimator Characteristics (4)

- Sufficiency [Bickel 77]:
  - A way for doing data compression in estimation process: by only keeping necessary information required for estimation
  - Definition: A statistic $T(X)$ is called *sufficient* for a parameter θ, if and only if, the conditional distribution of X given $T(X) = t$ does not involve θ (Bickel and Doksum, page 63).
    - Once the value of a sufficient statistic T is known, the sample $X = (X_1, \ldots, X_n)$ does not contain any further information about θ.
  - Example:
    - Sum of observations is a sufficient statistic for estimating mean
    - Event count is a sufficient statistic for estimating the associated probability

Parameter Estimation and Performance

- All probabilistic parameters are estimated from a finite set of samples.
  - Some criteria of a good estimator: unbiased, consistent, efficient.
  - Estimation Criteria is not directly linked to Performance Criteria (e.g., error rate, joint precision-recall)
- Some frequently used estimation methods:
  - Maximum Likelihood Estimation (MLE)
  - Bayesian Estimation
  - Least Square Estimation

Maximum Likelihood Estimation

- To choose a set of parameters θ in a way that maximizes the likelihood function $L(\theta)$:

$$L(\theta) = f(x_1, x_2, \cdots, x_n|\theta) = \prod_{i=1}^{n} f(x_i|\theta)$$

  - where $x_1, x_2, \ldots, x_n$ is a set of random samples from the distribution of a random variable $\boldsymbol{X}$ with density f and the associated parameter θ.

- The ML estimation $\hat{\theta} = (\hat{\theta}_1, \hat{\theta}_2, \cdots, \hat{\theta}_k)$ is the set of estimated values that maximizes $L(\theta)$, or values that satisfy the simultaneous equations

$$\frac{\partial L(\theta)}{\partial \theta_i} = 0, \; i = 1, \cdots, k$$

  - Usually, a log function is first taken before taking the derivative operation

Maximum Likelihood Estimation (Cont.)

- Properties:
  - Maximum-likelihood estimates are (1) consistent, and (2) asymptotically efficient.
  - Let $\hat{\theta}_{ML}$ be an MLE of θ , then $g(\hat{\theta}_{ML})$ is an MLE of g(θ) , i.e.,

$$[\hat{g}(\theta)]_{ML} = g(\hat{\theta}_{ML})$$

  if g(.) is a monotonic function (e.g., log-function).

Examples:

- The MLE for the "success" probability *p* of Bernoulli distribution is

$$\hat{p}_{MLE} = \overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

where x_i is the outcome of the *i*-th Bernoulli trial.

- $\hat{p}_{MLE}$ can be interpreted as the relative frequency of success over the *n* trials.
- The MLEs for the mean and variance of the normal density are:

$$\hat{\mu}_{MLE} = \overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\hat{\sigma}^2_{MLE} = \frac{1}{n}\sum_{i=1}^{n} (x_i - \overline{x})^2$$

Bayesian Estimation

- To choose the parameters which maximizes the likelihood function $L(\pi(\theta), \theta)$: $L(\pi(\theta),\theta) = \pi(\theta) f(x_1, x_2, \cdots, x_n|\theta)$
- Where $\pi(\theta)$ is the prior probability density of θ before sampling.
- The Bayesian estimation $\hat{\theta} = (\hat{\theta}_1, \hat{\theta}_2, \cdots, \hat{\theta}_k)$ is the set of estimated values that satisfies the equations

$$\frac{\partial L(\theta)}{\partial \theta_i} = 0, \quad i = 1, \cdots, k.$$

- The Bayesian estimation $\hat{\theta}$ of parameter θ is the expected value of the parameter taken with respect to the posterior distribution of θ given the outcome of the random sample *x*, i.e.,

$$\hat{\theta} = E[\theta | x]$$

Bayesian Decision vs. Maximum-Likelihood Decision

- Bayesian Decision Rule: Find the most probable source model (M) for a given observation (O) by choosing the one with the maximum conditional probability P(M|O): $\hat{M}_B = \arg\max_M P(M|O)$
  - It is the optimal classifier that minimizes the error rate.
- Maximum Likelihood Decision Rule: Find the model (M) which is most likely to generate the observation (O): $\hat{M}_M = \arg\max_M P(O|M)$
  - It is the same as the Bayesian Decision if prior probability, i.e., P(M), is uniformly distributed (since P(M|O) = P(O|M) x P(M) / P(O))

Hypothesis Testing

- Goal:
  - To make a binary decision on a hypothesis based on the given observations.
- Hypotheses:
  - Null Hypothesis (H_0): The hypothesis that we are interested in rejecting or refuting.
  - Alternative Hypothesis (H_1): The contradictory hypothesis of H_0.
- Decision Regions:
  - The observation space is partitioned into acceptance region $R(H_0)$ and rejection region $R(H_1)$; if the observed features fall within the acceptance region, hypothesis H_0 is confirmed, otherwise, H_0 is rejected.
- Types of errors:
  - Type I error: H_0 is true but the observation suggests H_1
  - Type II error: H_0 is false but the observation suggests H_0.

Hypothesis Testing: Level of Significance

- Level of Significance:
  - The level of significance, denoted by α, is the maximum probability of making a *Type I* error.
- One-Tailed Test vs. Two-Tailed Test:
  - For a test statistic *T* computed on the sample data:
  - a upper one-tailed test has the decision rule:
    --- Reject H_0 if $T > T_U$; otherwise accept H_0.
  - a lower one-tailed test has the decision rule:
    --- Reject H_0 if $T < T_L$; otherwise accept H_0.
  - a two-tailed test has the decision rule:
    --- Reject H_0 if $T > T_U$ or $T < T_L$
    --- Accept H_0, otherwise.
  - The values of T_U and T_L are critical values that are selected so that the test will have the desired level of significance α.
- p-Value:
  - The *p-value* associated with a test statistic is the smallest level of significance that would have allowed the null hypothesis to be rejected.

Hypothesis Testing: Procedures

- Procedures:
  1. State the null hypothesis, H_0.
  2. State the alternative hypothesis , H_1.
  3. Decide on the level of significance, α.
  4. Choose an appropriate testing procedure and determine the acceptance region.
  5. Compute the test statistic from the sample data.
  6. Make the decision: reject H_0 if the p-value is less than the level of significance α; otherwise accept H_0.
- Example:
  - To test H_0: p =0.6 against H_1: $p \neq 0.6$.
  - For a two-tailed test of the level of significance $\alpha = 0.05$, the critical values of the normal distribution are T_U =1.96 ; T_L = -1.96 .
  - Suppose the computed test statistic T=2.06 which corresponds to the p-value of 0.0394.
  - We will accept H_0 of the level of significance $\alpha = 0.05$.
  - However, we will reject H_0 if the level of significance $\alpha = 0.01$.

Likelihood Ratio Test

- The likelihood ratio λ: $\lambda = \frac{f_0(X)}{f_1(X)}$,

  where
  - H_0: the pdf of the data is $f_0(X)$,
  - H_1: the pdf of the data is $f_1(X)$.
  - Accept H_0 if $\lambda > \lambda_T$ (λ_T is a preset threshold), otherwise accept H_A.

- Example: For automatic compound noun extraction [Su 94]:
  - H_0: the feature vector $\vec{x}$ for the input pattern is generated by a compound model M_C.
  - H_A: the feature vector for the input pattern is generated by a non-compound model M_{nc}.
  - The likelihood ratio λ is $\lambda = \frac{P(M_C \mid \vec{x})}{P(M_{NC} \mid \vec{x})}$

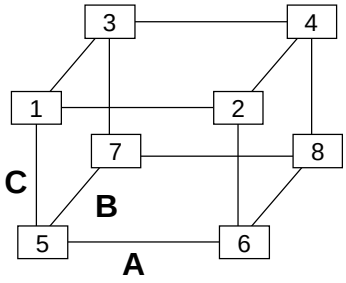
Fractional Factorial Experiment Design (1)

- Test features one at a time is not efficient
  - Three factors (A, B, C) need 4x4 = 16 runs
- Factorial Design [Montgomery 01]
  - Just eight runs to test all three factors
  - Interaction effects
  - Hidden replication
  - Wider inductive basis
- Experiment Design Table

A	B	C
Low	Low	Low
High	Low	Low
Low	High	Low
High	High	Low
Low	Low	High
High	Low	High
Low	High	High
High	High	High

Fractional Factorial Experiment Design (2)

- Effect of each factor
  - A = (2 + 4 + 6 + 8 – 1 – 3 – 5 – 7) / 4
  - B = (3 + 4 + 7 + 8 – 1 – 2 – 5 – 6) / 4
  - C = (1 + 2 + 3 + 4 – 5 – 6 – 7 – 8) / 4
  - AB = (1 + 4 + 5 + 8 – 2 – 3 – 6 - 7) / 4
  - AC = (2 + 4 + 5 + 7 – 1 – 3 – 6 - 8) / 4
  - BC = (3 + 4 + 5 + 6 – 1 – 2 – 7 - 8) / 4
  - ABC = (1 + 4 + 6 + 7 – 2 – 3 – 5 - 8) / 4



- However, still too many runs required when k is large
  - Total 2 to the K runs. If k = 8, it needs 256 runs
  - Fractional Factorial design is recommended

Fractional Factorial Experiment Design (3)

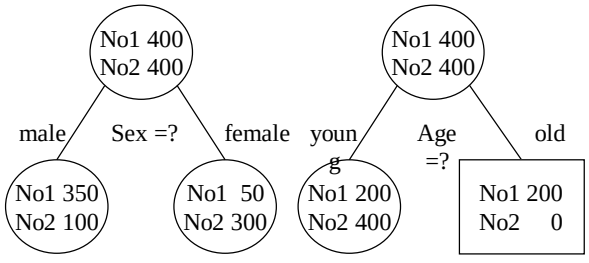
- Fractional Factorial Experiment Design [Montgomery 01]
  - Only run a fraction of complete factorial experiment design
  - Mostly used in screening experiments
  - Will generate Aliases (with various resolution)
    - Resolution III: no main effects are aliased with any other main effect
    - Resolution IV: no main effects are aliased with any other main effect or with any two-factor interaction
    - Resolution V: no main effects or two-factor interactions are aliased with any other main effect or with any two-factor interaction
  - Check the result, and identify those significant factors. Design new experiments to resolve the aliases involved
  - Select the desired fraction (2 to k-p), and look up the table for the appropriate generator (see the following example)

Fractional Factorial Experiment Design (4)

- Fractional Factorial Experiment Design Example
  - k = 5 (i.e., A, B. C, D, E), p = 2; ¼ fraction of 5 factors in 8 runs with Resolution III ([Montgomery 2001]: Appendix XII, Table (c), page 663)
  - Design Generators: D = AB, E = AC
  - Aliases:
    - A = BD = CE
    - B = AD = CDE
    - C = AE = BDE
    - D = AB = BCE
    - E = AC = BCD
    - BC = DE = ACD = ABE
    - CD = BE = ABC = ADE

Classification and Regression Tree (CART) [Breiman 1984]

- Binary Tree Construction
  - Select features to split the node (see the following figure)
  - Decide when to stop splitting the node
  - Assign label to each node
  - Use Cross-Validation to select the best tree structure
- Classification Process
  - Just follow the binary decision tree from the root to terminal nodes
- Example of binary decision trees in CART method:



Splitting

- How to choose features to split the node
  - Minimize impurity criterion after splitting
  - Impurity criteria
- Estimated classification error after splitting
- Estimated entropy after splitting

$$\Delta I(s,t) = I(t) - P_L I(t_L) - P_R I(t_R)$$
$$I(t) = -\sum_{i=1}^{N} P(j \mid t) \cdot \log P(j \mid t)$$

- Others, e.g., minimum or maximum value.

Termination

- When to stop
  - Grow the tree until only one class of data points in a node, or
  - Grow the tree until the number of data points in a node is less than a preset value, or
  - Grow the tree until the levels of splitting is larger than a preset value, or
  - Grow the tree until $\max_{s \in S} \Delta I(s,t) < \beta$
- Cross Validation
  - Construct sequence of subtrees, T_0 ,..., T_n, ranging from full tree to just the root node
  - Estimate "honest" error rate for each subtree by using Cross-Validation
  - Choose tree size with minimum "honest" error rate
- Terminal Node Assignment
  - Majority Vote: Choose most frequent class to label the node, choose mean value for regression.

CART: Advantages & Applications

- Advantages of Tree-based Modeling [Breiman 84]
  - It is very flexible, and can be applied to any data structure through the appropriate selection of the set of features (the set of questions).
  - The final classification has a simple form which can be compactly stored and that efficiently classifies new data.
  - The tree procedure output gives easily understood and interpreted information regarding the predictive structure of the data.
- Examples: End of sentence detection
  - Features: number of words, case before and after ".", etc.
  - Performance: 99.84% (cross-validated)
- Another Decision Tree: C4.5 [Quilan 1993], C5.0 (enhanced commercial product)
  - Publicly available: http://www.cse.unsw.edu.au/~quinlan/