[Day-1] Introduction to Statistical Natural Language Processing

(Part III: Basic Concepts and Background)

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

Part III: Basic Concepts and Background

- Feature Space, Probability
  - Definition and formulations frequently used in Statistical NLP
- Estimator and Stochastic Process
  - Definition and formulations frequently used in Statistical NLP
- Information Theory
  - Entropy, Mutual Information, Perplexity
  - Typical applications in Statistical NLP
- Data Set Classification
  - Training-Set, Cross-Validation-Set, and Testing-Set
- Performance Measure
  - Error-Rate, Precision, Recall, F-measure, etc.

Feature Space

- Experiment:
  - The process of observing a phenomenon that has variation in its outcomes.
  - Example 1: Observing the outcomes of tossing a fair coin twice.
  - Example 2: Tagging Part-of-Speech of a Token-Sequence.
- Outcome Space (Sample Space, Feature Space):
  - The totality of the possible outcomes of a random experiment.
  - Flip coin example:
    - Flip two coins (or flip a coin twice): the associated sample space is: S={HH, HT, TH, TT}, where H: head; T: tail.
  - Tagging part-of-speech:
    - Every possible combinations of [Token, Tag] pair-sequences

Feature Space (Cont.)

- Feature Space decides the performance upper bound
  - Once outcomes of different classes are mixed in the same feature space, they cannot be separated again without introducing errors (disambiguation would be difficult)
    - Similar to the case that various radio stations must occupy different frequency bands
  - Example 1: Consider a new outcome space of {0, 1, 2} for tossing a coin (each outcome denotes the total number of head occurring), instead of original {HH, HT, TH, TT}
    - Question: what is the chance that a first flip is a “Head”?
  - Example 2: SCFG vs. Lexicon-driven parser
    - Lexicon related information has been lost in SCFG

Probability of an Event

- Event:
  - An event is a subset of the sample space.
  - Flipping coin example:
    - A: at least one head in two tosses.
    - B: tail at the second toss.
    - A = {HT,TH, HH}, ~A = {TT}, B = {HT,TT}
  - Tagging example: “design” is tagged as “n”: {any left context, [design, n], any right context}
- Probability of an Event
  - Intuitive explanation: the probability of an event expresses the long-run frequency for the event to occur in many repeated independent experiments.
  - Coin example: P(A) = ¾, P(~A) = ¼, P(B) = ½.
  - Tagging example: P (n | design)

Probability Space

- Probability Space (three axioms)

1. $P(A) \geq 0$ $[A:$ any event$]$
2. $P(\Omega) = 1$ $[\Omega:$ the event for all outcomes$]$
3. $P(A \cup B) = P(A) + P(B)$ if $A \cap B = \Phi$ [disjoint]

- Not everything between 0 and 1 is a probability
  - For example, |Cos (x)| is not a probability

Joint Event

- Joint Probability of Event A and B: the probability that these two events occur simultaneously
  - Notation: P(A, B) or P(A∩B)
  - Coin example: P(O1 = H, O2 = T); Event (O1) = {HH, HT}, Event (O2) = {HT, TT}, Joint Event = {HT}
  - Tagging example: P(C(i) = adj, C(i+1) = n); Joint Event = { adj, n }
  - Marginal: the probability of an event that can be acquired by summing over all events that can jointly occur with it:

$$P(A) = \sum_{B_i \in S} P(A, B_i)$$

Conditional Probability

- Conditional Probability: event A occurs given that B event has occurred:
  - Notation: P(A|B) = P(A, B) / P(B)
  - Coin example: P(O2 = T | O1 = H)
  - Tagging example: P(C(i+1) = n | C(i) = adj)
  - Conditional Probability is itself a probability
    (it can be regarded as if it is generated from another new outcome space)

Multiplication Theorem of Probability

- Theorem:
$$P(A, B) = P(A \mid B) \times P(B) = P(B \mid A) \times P(A)$$
  - Coin example: P(HT) = P(O2 =T | O1 = H) x P(O1 = H)
  - Tagging example: P(adj, n) = P(C(i+1) = n | C(i) = adj) x P(C(i) = adj)
- Generalization:
$$\begin{aligned} & P(A_1, A_2, \cdots, A_k) \\ &= P(A_1 \mid A_2, \cdots, A_k) \times P(A_2 \mid A_3, \cdots, A_k) \\ &\quad \cdots \times P(A_{k-1} \mid A_k) \times P(A_k) \\ &= \prod_{i=1,k-1} P(A_i \mid A_{i+1}^k) \times P(A_k) = P(A_1) \prod_{i=2,k} P(A_i \mid A_1^{i-1}) \end{aligned}$$
  - Coin example: P(HTHH) = P(H|HTH) x P(H|HT) x P(T|H) x P(H)
  - Tagging example: P(det adj n) = P(n | det adj) x P(adj | det) x P(det)

Independent and Conditional Independent (1)

- Independent
  - Definition: P(A, B) = P(A) x P(B) => P(A|B) = P(A) and P(B|A) = P(B)
  - Coin example: P(HT) = P(H) x P(T)
  - SCFG: $P(\text{ParseTree}) = \prod_{R_i \in Tree} P(R_i : N_i \rightarrow \alpha_i)$
- Conditional Independent
  - Definition: P(A, B| C) = P(A| B, C) x P(B| C) = P(A| C) x P(B| C)
    => P(A| B, C) = P(A| C) and P(B|A, C) = P(B | C)
  - A word sense model: P(CW1, CW2 | W-Sense) is frequently simply assumed to be [P(CW1 | W-Sense) x P(CW2 | W-Sense)]
  - Naïve Baiysian: assumes conditional independency among features $P(F_1, \cdots, F_n \mid M_j) = \prod_{i=1,n} P(F_i \mid M_j)$

Independent and Conditional Independent (2)

- Independency does not imply Conditional Independency
  - Example: P(1st-Throw-Dice-Point, 2nd-Throw-Dice-Point) = P(1st-Throw-Dice-Point) x P(2nd-Throw-Dice-Point)
  - However, P(1st-Throw-Dice-Point | 2nd-Throw-Dice-Point, Sum=7) \= P(1st-Throw-Dice-Point | Sum=7)
- Conditional Independency does not imply Independency
  - Example: P(Lung-Cancer, Buy-Cigarette | Smoke) = P(Lung-Cancer | Smoke) x P(Buy-Cigarette | Smoke)
  - However, P(Lung-Cancer, Buy-Cigarette) \= P(Lung-Cancer) x P(Buy-Cigarette)

Independent and Conditional Independent (3)

- Conditional Independent property is frequently used to decouple different nodes in Bayesian inference network [Pearl 88]
  - Example: Diamond-Shape causal-chain, E4 causes E2 and E3, then E2 and E3 cause E1
  - P(E1, E2, E3, E4) = P(E1| E2, E3) x P(E2| E4) x P(E3| E4) x P(E4)
- An error that should be avoided
  - Wrong: P (C_i | W_i, C_{i-1}) = P (C_i | W_i) x P(C_i | C_{i-1})
  - In general, P(A | B, C) \= P(A | B) x P(A | C)

Bayes' Rule:

$$P(A \mid B) = \frac{P(A,B)}{P(B)} = \frac{P(A,B)}{P(A,B)+P(\overline{A},B)} = \frac{P(A)\times P(B \mid A)}{P(A)\times P(B \mid A)+P(\overline{A})\times P(B \mid \overline{A})}$$

- Example: $$P(C_1,\cdots,C_n \mid W_1,\cdots,W_n) = \frac{P(W_1,\cdots,W_n \mid C_1,\cdots,C_n)\times P(C_1,\cdots,C_n)}{P(W_1,\cdots,W_n)}$$
- If independent, P(A|B) = P(A)
- Generalization: $$P(A_k \mid B) = \frac{P(B \mid A_k)\times P(A_k)}{\sum_{A_i} P(B \mid A_i)\times P(A_i)}$$
  - where $A_1, A_2, \ldots, A_n$ are partitions of the sample space; i.e.,

$$A_1 \cup A_2 \cup \cdots \cup A_n = S$$
$$A_i \cap A_j = \phi, \ \forall i \neq j$$

Bayes' Rule (Cont.)

- Used to avoid an infinite number of outcome spaces, e.g.,

$$X \sim N(\mu, \sigma^2) \text{ (Gaussian)} \Rightarrow P(M_i \mid x) = P(x \mid M_i) P(M_i) / P(x)$$

- Used to factorize features, e.g.,

$$P(M_i \mid F_1, \cdots, F_n) = P(F_1, \cdots, F_n \mid M_i) P(M_i) / P(F_1, \cdots, F_n)$$
$$\text{where } P(F_1, \cdots, F_n \mid M_j) \approx \prod_{i=1,n} P(F_i \mid M_j)$$

- Used to decompose original optimization function
  - Example: decompose translation score into transfer score and generation score
    - Note: $P(EW_1, \ldots, EW_n)$ can be ignored when selecting the best candidate

$$P(FW_1, \cdots, FW_m \mid EW_1, \cdots, EW_n)$$
$$= P(EW_1, \cdots, EW_n \mid FW_1, \cdots, FW_m) P(FW_1, \cdots, FW_m) / P(EW_1, \cdots, EW_n)$$

Random Variable

- A random variable X on a sample space S is a function X: S -> R that assigns a real number X(s) to each sample point $s \in S$.
- Example:
  - *X*: number of head in the two tosses,
  - [*X* = 0] = { TT }
  - [*X* = 1] = { TH, HT }
  - [*X* = 1] = { TT }.
- Discrete RV, Continuous RV
  - Discrete/Continuous RV: X takes discrete/continuous values
- Examples:
  - Discrete: C: the lexical category (part-of-speech) of a word
    - [C({beautiful, adj}) = adj-index], [C({computer, noun}) = noun-index]
  - Continuous: Temperature reading

Random Variable (Cont.)

- Why random variable
  - Outcome is a physical attribute generated from an experiment, and is associated with a data type (e.g., number of points, color, etc.)
  - Abstract the outcome into a real number would make it convenient for further processing and discussion without tying to any specific experiment
  - Mathematic operation can be easily taken on a real number (On the other hand, how can we time a "blue color" by 2?)
    - For example, Y = |X|, X and Y are RVs
    - Another example: Normalized target sentence length (in bilingual sentences alignment)

$$\delta(l_1, l_2) = (l_2 - l_1 \cdot c) / \sqrt{l_1 \cdot s^2} \sim N(0,1)$$
$$c, s^2 : \text{mean} \quad \& \text{ variance} \quad \text{of} \quad (l_2 / l_1)$$

Multinomial Distribution

- Example: throw a dice six times, get {3, 6, 2} ⇔ 2, 1, 3 times

$$P(n_3 = 2, n_6 = 1, n_2 = 3) = \frac{6!}{2!1!3!} (p_3)^2 (p_6)^1 (p_2)^3$$

- Example: likelihood of part-of-speech sequence in a text

$$P(n_N, n_{ADJ}, n_V, \cdots) = k(n_N, n_{ADJ}, n_V, \cdots) \cdot p_N^{n_N} p_{ADJ}^{n_{ADJ}} p_V^{n_V} \cdots$$

- Probability distribution:

$$P(n_1, \cdots, n_v) = \frac{n!}{n_1! \cdots n_v!} p_1^{n_1} \cdots p_v^{n_v}, \ \sum_v n_v = n$$

Mean and Variance

- Mean: $$\mu_X = \mathrm{E}[\mathrm{X}] = \begin{cases} \sum_{x_i} x_i P(x_i) & \text{(discrete)} \\ \int_R x f(x)\,dx & \text{(continuous)} \end{cases}$$
  - An indication for the central location
  - Example: binominal distribution: ***E[X] = np*** (flip the coin *n* times with *x* H's)

    where $$b(x; n, p) = \binom{n}{x} p^x (1-p)^{1-x} \quad (\mathrm{x} \in \{0\} \cup \mathrm{N}^+)$$
- Variance $$\sigma_X^2 = \mathrm{E}\left[(\mathrm{X}-\mu_\mathrm{X})^2\right] = \begin{cases} \sum_{x_i} (x_i - \mu_\mathrm{X})^2 P(x_i) & \text{(discrete)} \\ \int_R (x_i - \mu_\mathrm{X})^2 f(x)\,dx & \text{(continuous)} \end{cases}$$
  - An indication for the degree of spreading
  - Example: binominal distribution: ***Var[X] = np(1-p)***

Covariance and Correlation Coefficient

- Covariance:
  - The covariance C_{XY} of two random variables X and Y is defined as follows:

$$C_{xy} = E[(X - \mu_X)(Y - \mu_Y)]$$

- Correlation Coefficient:
  - The correlation coefficient ρ_{XY} of two random variables X and Y is defined as follows:

$$\rho_{xy} = \frac{C_{xy}}{\sigma_x \sigma_y}, \quad -1 \le \rho \le 1, \quad \begin{cases} \rho > 0 & \text{positive correlation} \\ \rho < 0 & \text{negative correlation} \end{cases}$$

  - Un-correlated: $\rho = 0$
  - E[XY] = E[X] x E[Y] => $\rho = 0$.

More on Correlation Coefficient

- Correlation does not imply *Causality*
  - Causality: IF X-Event, then Y-Event. P(Y|X) = 1 > P(Y) > P(~Y|X) = 0
  - Example: "Lung-Cancer" and "Buy-Cigarette" are correlated, because they come from the same source of "Smoke"; however, no causality remained between these two events
  - Causality can let us have conditional independent form: P(Lung-Cancer, Buy-Cigarette | Smoke) = P(Lung-Cancer | Smoke) x P(Buy-Cigaratte | Smoke)
- independency imply un-correlatedness
  - P(A | B) = P(A)
- Un-correlatedness does not imply independent
  - It is possible that a non-linear dependent case has $\rho = 0$
  - Example: X ~ zero mean Gaussian, $Y = X^2$
  - However, if X and Y are bivariate Gaussian, then this statement holds

Parameter Estimator

- Statistic:
  - A *statistic* is any real or vector-valued function of the observation (*T*(**X**)).
    - e.g. X: Head/Tail of a fair coin; $T(X_1, \ldots, X_n)$: number of heads
- Estimator:
  - An estimator is a statistic calculated from sample data that provide either point estimates or interval estimates for some parameters. Usually, the term "*estimate*" is used to denote its associated value
  - Coin example: p(H) = k/n
  - Tagging example: P(n | det) = [# of det-n] / [# of det]
- An estimator is a function of a RV, and can be regarded as a RV itself
  - Estimator variance (i.e., estimation accuracy) depends on the size of the sampling data: proportional to 1/n

Parameter Estimation

- All probabilistic parameters are estimated from a finite set of samples.
  - Some criteria of a good estimator: unbias, consistent, efficient (see appendix for the definition).
- Some frequently used estimation methods:
  - Maximum Likelihood Estimation (MLE)
  - Least Square Estimation
  - Bayesian Estimation

Maximum Likelihood Estimation

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

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

  - where $x_1, x_2, \ldots, x_n$ is a set of random samples from the distribution of a random variable ***X*** with density *f* and 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(θ)*, or values that satisfies the simultaneous equations

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

MLE Examples:

- The MLE for the part-of-speech probabilities:

$$L(n_N, n_{ADJ}, n_V, \cdots) = p_N^{n_N} p_{ADJ}^{n_{ADJ}} p_V^{n_V},$$

$$LogLikelihood(n_N, n_{ADJ}, n_V, \cdots) = [n_N \times Log(p_N)] + [n_{ADJ} \times Log(p_{ADJ})] + \cdots$$

$$\text{Optimize} \left\{ LogLikelihood(n_N, n_{ADJ}, n_V, \cdots) + \lambda \cdot \sum_C P_C \right\}$$

$$\frac{\partial LL}{\partial p_C} = 0, \forall p_C \Rightarrow \frac{n_C}{\hat{p}_C} - \lambda = 0 \Rightarrow \hat{p}_C = \frac{n_C}{\lambda} \propto n_C$$

$$\Rightarrow \hat{p}_C = \frac{n_C}{\sum_C n_C} = \frac{n_C}{n} \quad \text{(MLE estimate)}$$

- $\hat{p}_{MLE}$ can be interpreted as the relative frequency of occurrence over the *n* words (parts of speech).

MLE Examples (Cont.):

- Normal (Gaussian) distribution with parameters: mean μ and standard deviation σ

$$N(x; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- The MLEs for the mean and variance of the normal density are:

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

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

Stochastic Process

- A stochastic process { X(t) ,t ∈ T }
  - is a collection of random variables; i.e, for each t ∈ T, X(t) is a random variable (a family of RVs).
  - Where T is the index set of the process (e.g., time-index or word-position)
  - { X(t)} is a discrete-time process, if T is a countable set; e.g., $\{X_n, n=0,1\}$
- Interpretations:
  - A stochastic process X(t) = X(t, ζ) is a single time function (a sample of the given process) if ζ is fixed.
  - X(t, ζ) becomes a random variable equal to the state of the given process at time t, if t is fixed.
  - If t and ζ are fixed, then X(t, ζ) is a constant.
- Tagging example
  - {C(1) = noun, C(2) = verb, …, C(n) = pron}: C(t): part of speech generated sequentially at time t (or position t)

Markov Chain

- A discrete-time discrete-state stochastic process $\{X_n, n=0,1,\ldots\}$, having the property that given the present state, the past states have no influence on the future, is called a discrete-time Markov chain.
  - For example, we only need today's data to predict tomorrow's weather
- The Markov property:
  - $$P(X_n = x_n \mid X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, \cdots, X_0 = x_o) = P(X_n = x_n \mid X_{n-1} = x_{n-1}) \quad \text{[1st order]}$$
  - $P(X_n = x_n \mid X_{n-1} = x_{n-1})$ are called the transition probabilities of the chain.

Markov Chain (Cont.)

- Example: The formula for tagging part-of-speech is approximated as:

$$\arg\max_{t_1^n} \prod_{i=1}^{n} P(w_i \mid t_i) \cdot P(t_i \mid t_{i-1})$$

  - where t_i corresponds to the part-of-speech attached to the i-th word w_i.
  - The probability $P(t_i \mid t_{i-1})$ in the above formula is the transition probability of the assumed Markov model.

- Note, X(i+1) and X(i-1) are still correlated
  - Conditional independence does not imply independent

Self-information & Entropy

- Self-information $I(x_k)$:
  - One desired property: information of two independent events should be the sum of the associated information of each individual event. It is the Log function that can satisfy those properties
  - $I(x_k) = -\log P(x_k)$
  - $I(x_k)$ is the amount of information (or uncertainty) associated with the known occurrence of output x_k.
- Entropy H(x) : $H(X) = -\sum_{x_i} \left[P(x_i) \cdot \log_2 P(x_i) \right]$
  - H(x) is the average information (or uncertainty) of the source X.
  - Word segmentation example (Left Context Entropy):

$$H_L(C; w) = -\sum_{c_i:\text{Left-Of}(w)} \left[P(c_i) \cdot \log_2 P(c_i) \right]$$

Conditional Entropy

- Conditional Entropy:

$$H(X \mid y_j) = -\sum_X \left[P(x_i \mid y_j) \cdot \log_2 P(x_i \mid y_j)\right]$$
$$H(X \mid Y) = \sum_Y P(y_j) H(X \mid y_j) = \sum_X \sum_Y P(x_i, y_j) \cdot \log_2 P(x_i \mid y_j)$$

- Chain Rule:

$$\begin{aligned} H(X,Y) &= -\sum_X \sum_Y \left[P(x_i, y_j) \cdot \log_2 P(x_i, y_j)\right] \\ &= -\sum_X \sum_Y \left[P(x_i, y_j) \cdot \left(\log_2 P(x_i \mid y_j) + \log_2 P(y_j)\right)\right] \\ &= H(X \mid Y) + H(Y) \end{aligned}$$
$$H(X_n, X_{n-1}, \cdots, X_1) = H(X_n \mid X_1^{n-1}) + \cdots + H(X_2 \mid X_1) + H(X_1)$$

Mutual Information

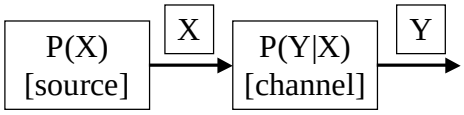
- (Specific) Mutual Information I(X; Y):

$$I(x; y) = \log_2 \frac{P(x, y)}{P(x) \cdot P(y)}$$

- I(X; Y) is the mutual information between X and Y.
- Example: Use $I(w_x; w_y)$ as a measure for the preference of "strong economy" and "powerful economy" [K. Church 89]:
- $I(w_x; w_y) >> 0$, w_x and w_y are highly associated.
- $I(w_x; w_y) \approx 0$, w_x and w_y are independent.
- $I(w_x; w_y) << 0$, w_x and w_y are in complementary distribution.

Entropy and Mutual Information

- Relation between (Average) Mutual Information and Conditional Entropy



$$I(X;Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X)$$
$$= \sum_X \sum_Y \left[P(x_i, y_j) \cdot \log \frac{P(x_i, y_j)}{P(x_i) P(y_j)} \right]$$

- I(X;Y): change of uncertainty about source symbol X with/without observing Y (through a noisy channel or not); that is, *extra* information from Y
- I(X;Y) = 0 ⇔ H(X|Y) = H(X) (independent, no extra information by knowing Y)

Perplexity

- The perplexity is a measure of the constraint imposed by the grammar, or the level of uncertainty given the grammar.
- Let P(w | s) be the probability that w will be the next word when the current state is *s*.
  - The entropy $H_s(w)$, associated with state s is

$$H_s(w) = -\sum_w [P(w \mid s) \cdot \log_2 P(w \mid s)]$$

  - The entropy H(w) of the task is the average value of $H_s(w)$, i.e.

$$H(w) = \sum_s \pi(s) H_s(w)$$

  - where $\pi(s)$ is the probability of being in state *s* during the production of a sentence.

Perplexity (Cont.)

- The perplexity *S(w)* of the task [Bahl 83]: $S(w) = 2^{H(w)}$
- In Practice, define *logprob* (*LP*) as [Jelinek 97]:

$$LP \cong \lim_{n\to\infty} -\frac{1}{n}\sum_{i=1}^{n} \log P\left(w_i \middle| w_1, \ldots, w_{i-1}\right)$$

  - In above equation, $P\left(w_i \middle| w_1, \ldots, w_{i-1}\right)$ will be replaced by your adopted language model (e.g., bi-gram or tri-gram)
- And Perplexity is defined as: $PP \cong 2^{LP}$

Bayesian Classifier

- "max" and "argmax" operators
  - "max": the maximum value among given members
  - "argmax": the associated member index of the member that possesses the maximum value
- Maximum Likelihood Classifier : find the model that has the maximal probability to generate the observed features

$$\hat{M} = \arg\max_{M_i} P\left(f_1^d \mid M_i\right)$$

Where *f* is the feature vector with dimensionality of d.

Bayesian Classifier (Cont.)

- Bayesian classifier: find the best model based on given features

$$\hat{M} = \arg\max_{M_i} P(M_i \mid f_1^d) = \arg\max_{M_i} P(f_1^d \mid M_i) P(M_i) / P(f_1^d)$$
$$= \arg\max_{M_i} P(f_1^d \mid M_i) P(M_i)$$

  - If the model is correct, it can achieve the minimum error rate; however, it is not the only classifier that can achieve the minimum error rate
  - If priori probability is uniformly distributed, it becomes Maximum Likelihood Classifier

- Tagging example:

$$\hat{c}_1^n = \arg\max_{c_1^n} P(c_1^n \mid w_1^n) = \arg\max_{c_1^n} P(w_1^n \mid c_1^n) P(c_1^n)$$

Elements of Parameter Learning

- Parameter Learning: Statistical Language Model + Training Corpus + Parameters Estimators
- Training Corpus: known instances used for learning
  - The information source used to learn the desired knowledge
  - The amount of implied Information is related to the Corpus Size and the Degree of Annotation (if under the supervised learning mode)
- Parameter Estimation Error
  - The estimated parameter is a statistic measure based on a set of finite samples
    - Has estimation errors as other statistics do
    - Values obtained from different sets of data (e.g., training set and testing set) are usually different
  - Variance of parameters (i.e., estimation accuracy) depends on the size of the evaluation corpus: proportional to 1/n

Data Set Classification (1)

- Training Set
  - The data set used to obtain model parameters
  - Performance measured in the training set reflects the model capability to fit available training instances
- Testing Set
  - A data set which is independently sampled other than the training set
  - It is mainly used to measure the true system performance in the real world, which also reflects the model capability to fit other instances in the real world
  - Testing set is frequently implicitly tuned without awareness
  - Development Testing Sets and Real Testing Set (testing set can still be implicitly tuned)
- Cross-Validation Set
  - Another set of data which is independently sampled other than both the training set and the testing set
  - It is mainly used to help making design decision (e.g., model complexity adopted, etc.)

Data Set Classification (2)

- Why testing set
  - The performance measured in the training set is generally over-optimistic (which could be resulted from over-fitting, or over-tuning).
    - 100% accuracy is possible if the number of parameter is greater than that is needed
    - Over-fitting usually occurs when the number of training data is not enough to support the model complexity adopted
    - Over-tuning happens during the adaptive learning process while we have too many adjustable parameters that we can afford
  - We need another independent data set to reflect the true performance when the customer deploys the system in the real world

Data Set Classification (3)

- To keep the testing set away from contamination, it is not allowed to see (or involve) the details in any design/training phase
  - You cannot use it to decide the suitable model complexity, dimensionality of the feature space, or when to stop during the adaptive learning process
  - Design decision should be made on the cross-validation set
- Adopt a new testing set after every certain period
  - When you compare various approaches and try to select one with the best testing set performance, the performance has already implicitly tuned on the testing set

Data Set Classification (4)

- Why Cross-Validation Set
  - Bad training set performance normally implies methodological flaw, so we can immediately know that we must re-do the design
  - In contrary, good training set performance may result from:
    - A really good model (which will also get good testing set performance)
    - An over-fitted model (which will give bad testing set performance)
    - A set of over-tuned parameters resulted from the adaptive learning process (which gives bad testing set performance too)
  - As the testing set is not allowed to be used to help making design decision, we cannot disambiguate the above situations
  - We need another independent set of data to provide a simulated testing test performance to help us making design decision

Performance Evaluation

- Performance is estimated from a set of finite samples (also a statistic measure)
  - Has estimation errors as other statistics do
  - Variance of performance measure (i.e., estimation accuracy) depends on the size of the evaluation corpus: proportional to 1/n
  - Values obtained from different sets of data (e.g., training set, cross-validation set, and testing set) are usually different

Performance Criteria

- Contingency Table

1st class:"+" 2nd class: "-"		Real Label	
		+	-
Predicted Label	+	N11	N21
	-	N12	N22

- Error rate (E), precision (P), recall (R), and F-measure
  - E = number_of_incorrect_identification / total_number_of_instances = (N12+N21) / (N11+N12+N21+N22)
  - P = number_of_correct_identification / number_of_candidates_in_list = N11 / (N11+N21)
  - R = number_of_correct_identification / number_of_correct_instances = N11 / (N11+N12)
  - F-measure (beta = 1): 2PxR/(P+R)

Performance Evaluation Methods (1)

- Re-substitution Estimate:
  - Use the same set of samples to design and test a model (training set performance, or close set performance)
- Holdout Estimate:
  - Use two mutually exclusive sets of samples to design and test a model
  - Testing set performance, or open set performance
  - Less data is left in the training set; thus it would result in a worse system
  - The true testing set performance would be deteriorated, although its value can be more accurately estimated

Performance Evaluation Methods (2)

- Leave-one-out Estimate:
  - Use one sample for testing and the other samples for design; test the model in rotation for each single sample, then report the performance by averaging the obtained result
  - Retain the largest amount of training set data (thus have the best model) while provide the most accurate testing set performance measure
  - Very time-consuming, as it demands to repeat the design process N times

Performance Evaluation Methods (3)

- Rotation Estimate (v-fold cross validation):
  - Use one subset of the samples for testing and the other subsets for design; test the model in rotation for each subset
  - Example: 10-fold rotation
    - 1. Divide all data (D) into 10 subsets of equal size: $D = D_1 \cup D_2 \cup \ldots \cup D_{10}$
    - 2. For Iteration I =1 to 10:
      - use D_i as testing set, $D-D_i$ as training set, and evaluate testing set error rate E_i
    - 3. Report the estimated error rate as: $(E_1+E_2+\ldots+E_{10})/10$
  - A compromise between achieving better true testing set performance and obtaining more accurate measure on that

Performance Upperbound

- Even human does not completely agree with each other
  - Different groups of people would make different Golden Data Sets
  - The agreement between humans will be the upperbound for the performance of our systems
- Human Agreement Check: Kappa Statistics [Carletta, 96]
  - $K = [P(A) - P(E)] / [1 – P(E)]; \ 0 <= K <= 1$
    - P(A): the proportion of times that the coders agree
    - P(E): the proportion of times that we expect them to agree by chance
  - $K > 0.8$, good reliability
  - $0.67 < K < 0.8$, allowing tentative conclusions to be drawn

Traps to Avoid

- Compare two event probability in two different outcome spaces
  - Example: compare P(f1 | M1) with P(f2| M2).
  - Usually happen during model simplification after Bayesian formulation has been adopted
- Abuse of the approximation operator (the following two equations are in wrong expression)

Wrong: $P\left(c_1^n \mid w_1^n\right) \approx \prod_i P\left(w_i \mid c_i\right) P\left(c_i \mid c_{i-1}\right)$

Wrong: $\hat{c}_1^n = \arg\max_{c_1^n} P\left(c_1^n \mid w_1^n\right) \approx \arg\max_{c_1^n} \prod_i P\left(w_i \mid c_i\right) P\left(c_i \mid c_{i-1}\right)$

Traps to Avoid (Cont.)

- Blind Attempt
  - Try a popular method for every problem without knowing what is the characteristics of the problem and why this approach should be adopted
    - "If you have a hammer, then everything looks like a nail"
    - Since many different approaches have been tried to the same testing set (or have tried the same popular approach to different problems), the testing set performance, in some sense, has already been tuned (i.e., biased)
  - Regard every linguistic symbol as just a symbol of "X1", "X2", etc.
    - Without knowing the linguistic meaning of each symbol, it has only little chance that you will come out with a good design

Tips to Keep in Mind

- Forms (the relationship between candidate and various features) are more important than the way to estimate the associated parameter values (once any reasonable approach has been adopted)
  - What kinds of features should be adopted is the most important issue
  - Don't mix up the outcomes (i.e., generate overlapping classes in feature space by ignoring the important cue), and then try to clean it up later
  - Dependency between different features should be carefully expressed in your model.
- A good language model (feature space plus dependency relationship) with a simple parameter estimation method is better than a bad language model with a more complicated parameter estimation approach (such as ME, SVM, etc.)