Unsupervised Learning
for
Natural Language Processing

(Afternoon Session)

Keh-Yih Su and Jing-Shin Chang
{kysu,shin}@bdc.com.tw

(1999/12/10)

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

Part IV: Advanced Topics: Potential Traps, Sources of Problems, and Why

- Criteria Mismatch: Human Preference in Testing Set vs. Model Fitting in Training Set
  - Mismatch of Measuring Functions
  - Mismatch of Measuring Sources
  - Implied Assumptions During Problem Solving
- Sources Causing Mismatch
  - Model Deficiency
  - Local Traps
  - Insufficient Training Data
  - Statistical Characteristics Variation
- Methods to Reduce Mismatch Effect
  - Reduce Measuring Function Mismatch Effect
  - Reduce Measuring Source Mismatch Effect

Fundamental Problem with Unsupervised Learning -- Criteria Mismatch (I)

- Criteria Mismatch: Human Preference in Testing Set vs. Model Fitting in Training Set
  - System Performance: Error rate in the testing set
    - Error rate measures the fitting for human preference
  - Unsupervised Learning Convergence Direction: Maximum of Likelihood Values in the training set
    - Likelihood value measures the fitting for the adopted model
  - Two measures are not necessarily to be closely correlated, if not under proper setting
  - Sources Resulting Mismatch
    - Adopting different measuring functions
    - Sampling from different sources

Criteria Mismatch (II):

- Unsupervised Learning Wish:
  - System Performance is getting improved iteration by iteration
  - The iteration process will finally converge to the point in the parameter space which possesses the minimum error rate performance (measured in the testing set)

- Implied Assumption for the success of unsupervised learning:

  Increasing Training Set Likelihood Values => Decreasing Testing Set Error Rate

  - Monotonically increasing of likelihood value in the training set (no problem, it is guaranteed)
  - Likelihood Value Increases => Error Rate Decreases (in both training set and testing set)
  - Maximizing Likelihood Value => Minimizing Error Rate (in both training set and testing set)

Criteria Mismatch (III):

- Implied Assumption for the success of unsupervised learning (cont.):
  - Increasing Training Set Likelihood Value => Increasing Testing Set Likelihood Value
  - Maximizing Training Set Likelihood Values => Maximizing Testing Set Likelihood Values
- Implied Conclusion:
  - Increasing Likelihood Value in Training Set => Decreasing the Error Rate in the Testing Set
  - Maximizing Training Set Likelihood Values => Minimizing Testing Set Error Rate

Mismatch between Measuring Functions (I):

- *Model Fitting* (Maximizing Likelihood Value) versus *Preference Finding* (Minimizing Error Rate)

- Many learning methods (designed for the recognition task) pursue “minimizing error rate” indirectly via training the model with other “optimizing criteria”

  - Possible criteria

    - Minimal Sum of Square Error (e.g., Clustering, VQ, etc.)
    - Minimal Inter-Cluster Distance (e.g., Clustering, VQ, etc.)
    - Maximal Likelihood Value (e.g., EM, Viterbi)
    - Maximum Entropy (e.g., IBM Maximum Entropy approach), etc.

  - Implicit Assumption: the model that can optimize the chosen Criterion can also achieve the minimum error rate performance

Mismatch between Measuring Functions (II):

- Traditional statistical pattern recognition obtain the recognition model indirectly through independently searching the most fitted model (governed by a set of parameters) for each individual class
  - Criteria adopted for searching the most fitted model
    - Maximum Likelihood Value (e.g., EM, Viterbi)
    - Maximum Entropy (e.g., IBM Maximum Entropy Approach)
  - Each model is trained only with the data inside its own class, instead of jointly considering all those competing classes (I.e., directly pursuing correct ranking order); however, it is the rank not the value that we really care

$$\hat{c} = \arg\max_{c_i} P\left(c_i \mid f_1^k, \Lambda\right) = \arg\max_{c_i} P\left(f_1^k \mid c_i\right) \times P\left(c_i\right)$$

Mismatch between Measuring Functions (III):

- Statistical approaches (cont.)
  - Example (rank vs. value):
    - Correct Class is C_1, and is recognized as C_2
    - True probability: [0.55, 0.45]
    - Estimated probability: [0. 49, 0.51]; Small estimation error, but incorrect (in rank)
    - If adjust to [0.7, 0.3]; Large parameter error, but correct (in rank).
  - Why those approaches were adopted?
    - Baysian Classifier guarantees the minimum error rate; therefore, it is natural to infer that the remaining work is just to better estimate the density functions of each individual class
    - There is no existing parametric form for directly estimating the rank (parametric estimation is more efficient and easier)
    - Jointly considering several classes (for estimating rank) is more complicated than only considering one class at a time (density functions are independently estimated)

Mismatch between Measuring Functions (IV):

- Statistical approaches (cont.)
  - Result
    - The parameter set that maximizes the likelihood in the training set is not the one which can really minimize the error rate in the training set.
    - Indirectly adjusting parameters is relatively ineffective (and sometimes awkward)
  - However, it is still used as a good starting point
    - There is still no good statistical model that can directly pursue the correct ranking order so far.
    - Baysian framework is sound and relatively good (comparing to other approaches)
    - It just needs a little twist for fine tuning

Mismatch between Measuring Functions (V):

- For remedying the drawbacks above mentioned, Discriminative Training was proposed to directly pursue “Minimizing error rate”
  - Approximate Each Error by an analytical Loss Function (e.g., arctan or sigmoid)
  - Searching the parameter space for minimizing the corresponding Risk Function
  - Result: Better performance, more effective in adjusting parameters
- Under Discriminative Training, minimizing risk function in the training set does imply minimizing error rate in the training set
  - Note, commonly used Gradient-Descending Search only converges to a local minimum point; global optimum point is not guaranteed
  - Global minimal error rate point is possible to find (e.g., globally searching the parameter space using Genetic Algorithm); however, it is seldom adopted (as it is usually too time consuming)

Mismatch between Measuring Functions (VI):

- In unsupervised learning, however, human preference is not known; therefore, Discriminative Training cannot be applied
  - Errors can no longer be perceived in the training set
  - Therefore, the error rate cannot be used as the searching criterion
- Mismatch between measuring functions is thus unavoidable
  - Result: optimizing the chosen criterion in the training set does not imply we can also minimizing the error rate in the training set

$$\hat{\Lambda}_{MLE}(TR) \neq \hat{\Lambda}_{Err}(TR); \quad \hat{\Lambda}_{MLE}(TS) \neq \hat{\Lambda}_{Err}(TS)$$

Mismatch between Measuring Functions (VII):

- To make unsupervised learning work, a high correlation between those two measures (likelihood in the training set & error rate in the testing set) must be inherited (or implied) from the model
  - The higher the degree of correlation, the better the chance for obtaining good performance
  - However, these two measures will not automatically closely correlate with each other if not under proper setting

Mismatch between Sampling Sources:

- Mismatch of Sampling Sources: *Training Set* vs. *Testing Set*
  - Statistical learning methods implicitly assume that the parameters obtained from the training set are also applicable to the testing set
  - Implicit Assumption:
    - Both the training set and the testing set have identical statistical characteristics
    - Both the training set and the testing set have almost infinitive sampling size
  - Implied Conclusion:
    - Maximizing Training Set Likelihood Value => Maximizing Testing Set Likelihood Value; however, it is not guaranteed
    - Minimizing Training Set Error Rate => Minimizing Testing Set Error Rate ; however, it is not guaranteed

Mismatch between Sampling Sources (cont.):

- Factors for causing mismatch:
  - Statistical characteristics variation (possible sources: sampling from different domains) between training set and testing set
  - Finite sampling size: causing estimation error (Note: the estimation error cannot be perceived in the training set)
- Result:
  - The parameter set that can maximize the *likelihood value* in the *training* set might not be the one that can also do the same in the *testing* set
  - The parameter set that can minimize the *error rate* in the *training* set might not be the one that can also minimize the error rate in the *testing* set

$$\hat{\Lambda}_{MLE}(TR) \neq \hat{\Lambda}_{MLE}(TS); \quad \hat{\Lambda}_{Err}(TR) \neq \hat{\Lambda}_{Err}(TS)$$

Sources for Causing Mismatch:

- Model Deficiency
  - Inappropriate Feature Set
  - Inappropriate Feature Dependency Relationship
  - Causing the mismatch between two measuring functions
- Local Traps
  - Multiple local optimum points inherent in the parameter space
  - Causing the mismatch between two measuring functions
- Insufficient Training Data
  - Large estimation error perceived in the testing set
  - Causing the mismatch between two measuring functions
- Statistical Characteristic Variation
  - Different statistical characteristics between training set and testing set
  - Causing the mismatch between two sampling sources

Inappropriate Feature Set (I)

- The selected Feature Space decides Performance Upper Bound
  - Once the feature space is specified, the best reachable performance is also determined for the given task. The system designers can only try to find a good discriminator to approach the upper bound.
  - Feature selection is probably the most important step
    - Problem Analysis is usually required (versus black-box approach)
- Feature Set Mismatch:
  - Causing the mismatch between two measuring functions
  - Using naive raw features instead of preference-based features:
    - Surface-level features (e.g., words) are used, instead of deeper level features, in the adopted stochastic language model
    - Unable to catch underlying linguistic units based on which human really uses to make preference

Inappropriate Feature Set (II)

- Feature Set Mismatch (cont.):
  - Naive stochastic language model usually fails to catch Long-distance Dependency (frequently adopted by the human preference model)
    - N-gram (either word or POS) was usually adopted
      - with heuristically determined window size (to avoid exponential explosion of the number of parameters)
      - can only handle local dependency
    - At the cost of lower performance by ignoring the features that the long distance dependency requires

Inappropriate Feature Set (III)

- Mismatch of Feature Set (cont.):
  - Examples:
    - Semantic tags assignment: Semantic Markov Chain (which adopts Semantic Tag N-gram) versus Head-Features
      - with heuristically determined window size
    - Parse tree selection (or PP-attachment): Stochastic Context Free Grammar versus Context-Sensitive Layered-Scoring Function [Lin 99]
      - A language can be represented by a context-free grammar does not imply that its constituents can be mixed in a context-free manner (most constituents have selection restriction on its context)
      - Normalization Issue: parse trees with less nodes get a higher score (introducing errors un-related to the linguistics characteristics)
    - IBM Machine Translation Model (I): Free-order word-string versus BDC BehaviorTran linguistic structure
    - OCR: crossing count versus strokes

Inappropriate Feature Dependencies (I)

- Dependency Relationship Mismatch will make the measuring functions to be unmatched
- Inappropriate Markov Assumption is widely assumed
  - Most Markov models only keep a few nearest adjacent neighbors, and drop those constituents that are relatively farther (i.e., only handle local dependency)
  - May not reflect real dependencies among constituents (i.e., the human preference network in which long distance dependency is usually implied)
  - Example: use bi-gram model to predict the next word when the next word really depends on a head word that is ten words away.
    - The prediction power, implied by the dependency, provided by the head word will be attenuated to almost nothing after 10-step state transitions

Inappropriate Feature Dependencies (II)

- Conditional Independence is inappropriately assumed
  - Assuming features are conditional independent (which is frequently used to drop terms) while they are actually highly correlated
    - Example: $P(f_1, f_2|c_i) \cong P(f_1|c_i) \times P(f_2|c_i)$
  - Some features in the adopted feature set are highly correlated, the strong dependency should be utilized in the model
    - Better Reduced Form: $P(f_1, f_2|c_i) \cong P(f_1|f_2) \times P(f_2|c_i)$

Local Maximum Trap (I)

- Multiple local maximums or non-unique global maximum points in the parameter space trap the searching process frequently

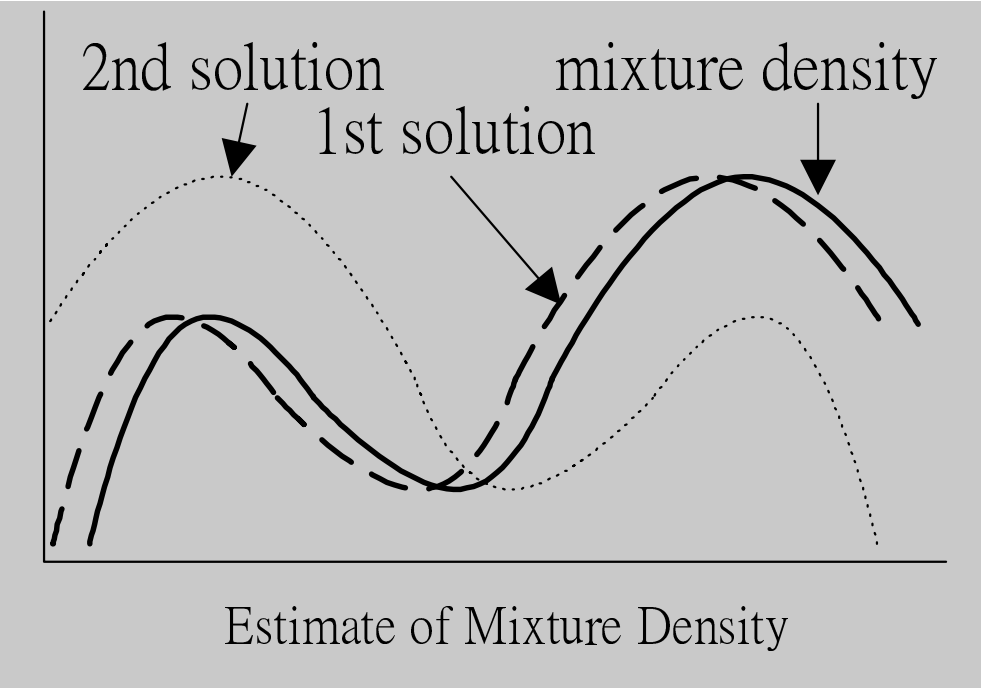
- Poor initial guess might cause the searching process converges to an undesired local maximum not preferred by the human
  - Causing the mismatch between two measuring functions
  - Seed corpus can be used to provide a better starting point

- Example: using a bi-gram model for part of speech tagging; however, each word in the corpus has exactly two tags (e.g., noun and verb)
  - Switching noun and verb of the best (human preferred) tag sequence results in the same (and also the maximum) likelihood value: just an exchange of the labels
  - May be trapped to the completely reversed (& the worst) candidate if not guided by human preference

Local Maximum Trap (II)

- Complicated tasks usually have many local maximum points
  - Task complexity can be measured by the perplexity factor
  - Less chance for the unsupervised learning process converging to the desired local maximum point in complicated tasks
  - Need implicit or explicit hints if unsupervised learning is necessary

- Local Maximum Trap: Example of non-unique global maximum



Insufficient Training Data (I)

- The size of the training samples might not be large enough to support the complexity of the adopted model

- Causing the mismatch between two sampling sources (resulted from the problem of Over Fitting)

  - Likelihood Value always increase through non-trivially refining the features (I.e., increase the dimensionality of the feature vector; thus, it also increases the number of parameters to be estimated)

  - Decreasing Modeling Error in the training set might Increase the Estimation Error in the testing set, as the size of the available training data is fixed.

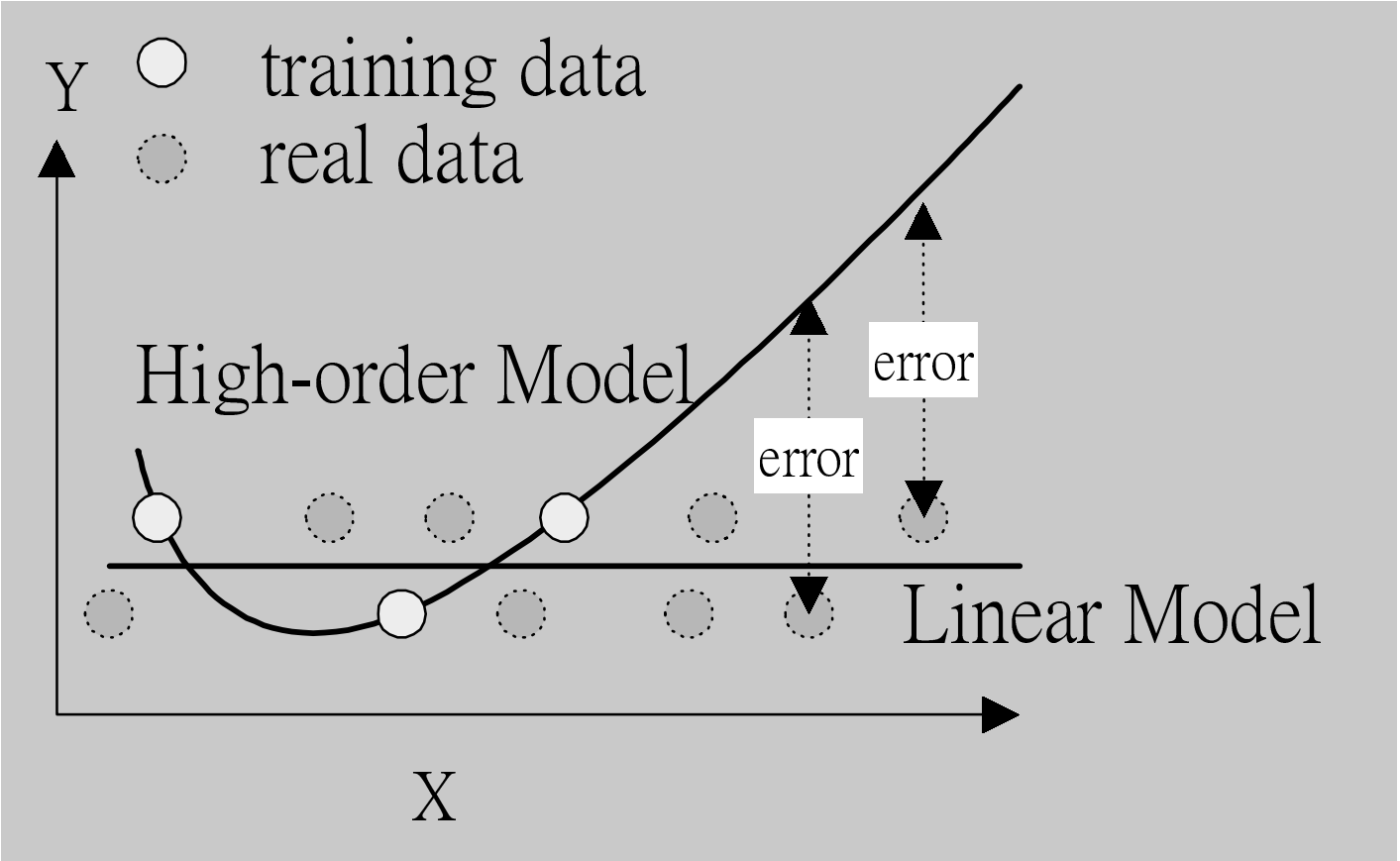
  - The extra errors induced (by the increase of the estimation error) in the testing set might out run those errors to be wiped out (by the decrease of the Modeling Error)

Insufficient Training Data (II)

- Example I: increase “N” in an N-gram model
  - Increases “N” increases the maximum likelihood value we can obtain in the training set
  - It also decreases the error rate in the training set under the supervised mode, as the modeling error will be reduced too (through covering wider context)
  - However, the error rate in the testing set will go up eventually if you keep increasing the “N”.
- Example II: Line Fitting
  - Assume data are really generated from a linear model with noise independently added
  - A high order polynomial function (y = a x^99 + b x^98 +…+ c x +d) is adopted as the model
  - Now trained with 3 data points:
    - Modeling error would be observed in the training set for the linear model
    - Obtain zero modeling error in the training set for any high order model (perfectly fitted by the quadratic curve of the form y = a’ x^2 + b’ x + c’)
    - BUT, the linear model enjoys smaller error in the the testing set

Over Fitting: (Example - Line Fitting)

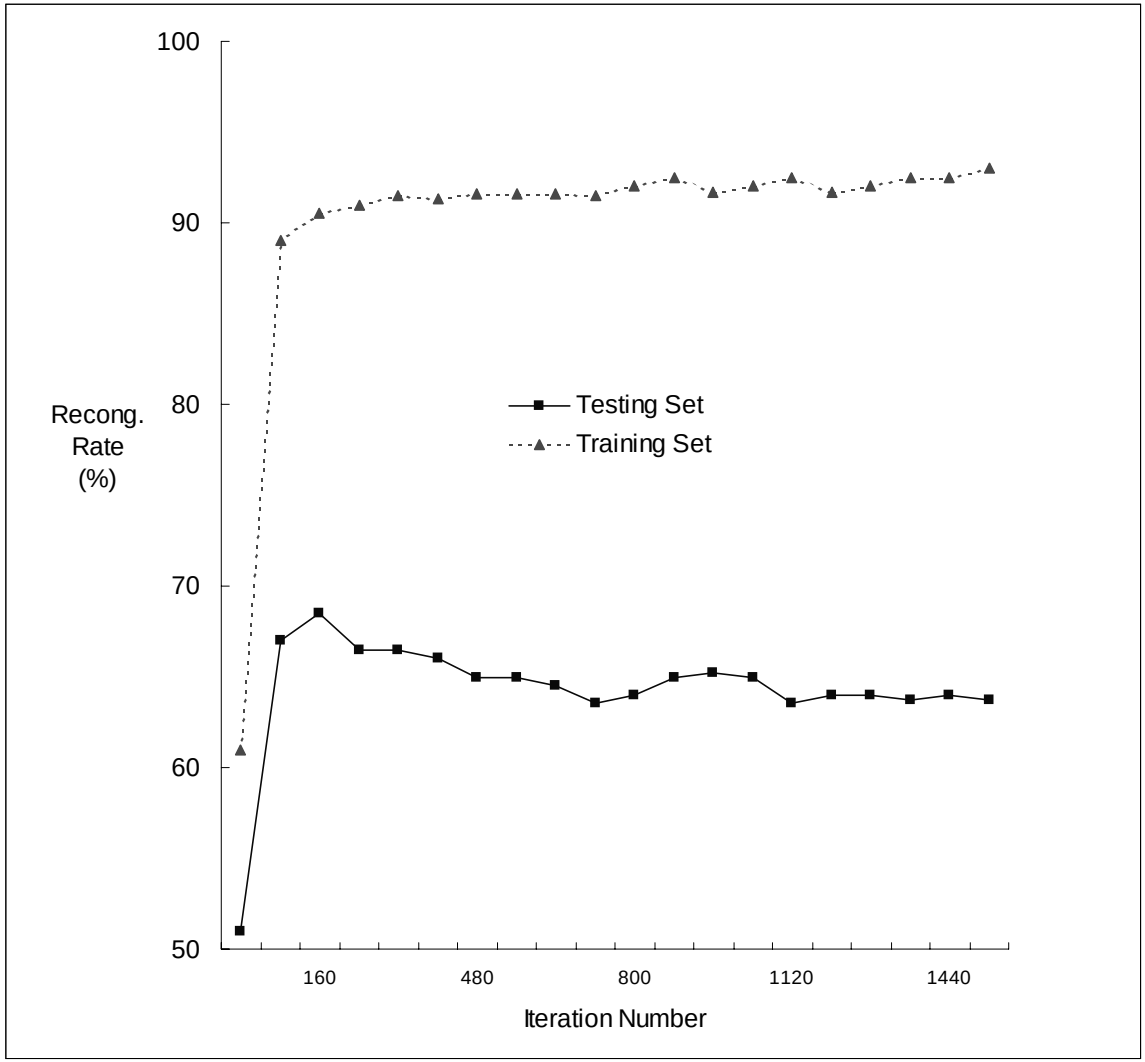
- Training Set and Testing Set Errors in Fitting Lines



Insufficient Training Data (III)

- Over-Tuning Effect
  - Effect: Over-optimistic *training* set performance, after the adaptive learning process has been conducted
  - Why: Having too many adjustable parameters that are affordable, with respective to the size of the given training corpus
  - The mean of the performance measure in the training set monotonically increases with iterations (as the stochastic gradient descending search is adopted), however, the performance measure in the testing set might fall off after a number of iterations.
  - See the curve in the next page.

WHMM-based Recognizer: Recognition Rate vs. Iteration Number



E-Set [Su & Lee 94]

Insufficient Training Data (IV)

- Model Resolution versus Coverage Rate in the Feature Space
  - Increasing the model resolution (by increasing the model complexity, or by reducing the model covering scope) usually decreases the coverage rate in the testing set
    - Increasing the model resolution increases the discrimination power in the training set
    - However, if the local description function gets sharper, the scope that it can cover gets smaller
    - No information would be available on those uncovered regions
    - Thus, it would induce low coverage rate on the real data (testing set)
    - Example: Regard each word as a class (IBM first statistical MT) !
  - Example: Histogram and Kernel functions (data-driven approaches)
    - If you divide a histogram into too many divisions, many cells will be empty (and they tell us almost nothing about the real distribution)

Insufficient Training Data (V)

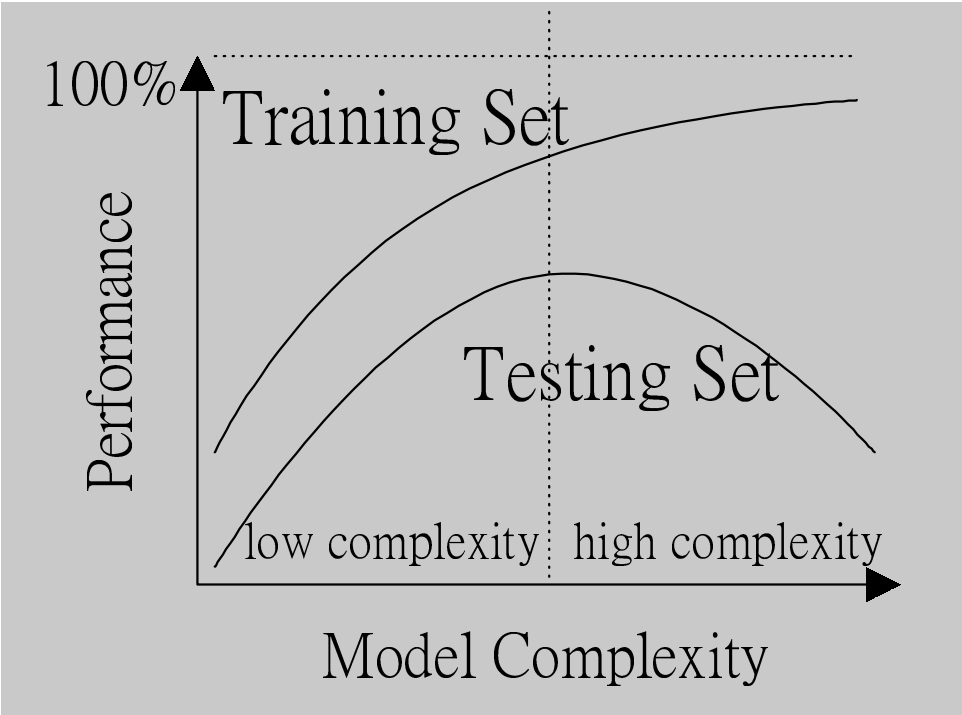
- How much is enough?
  - Usually 5 to 10 times is considered to be enough (i.e., similar performance will be observed in the testing set) for most applications
  - However, the cases that use much less data (typically less than 1 times) to train their NLP models are not rare
  - The suitable size actually depends on the problems and the models adopted
  - Class-based approach and back-off smoothing can greatly relieve the adverse data sparseness phenomenon

Performance Trends

- Increasing Model Complexity (in the same family) always increase the likelihood in the training set
- First rising then falling of the performance curve (in the testing set) are frequently observed, if we keep increasing the model complexity
- Coverage Rate decreases while Model Complexity increases
- Coverage Rate decreases while the Corpus-size of the training set decreases

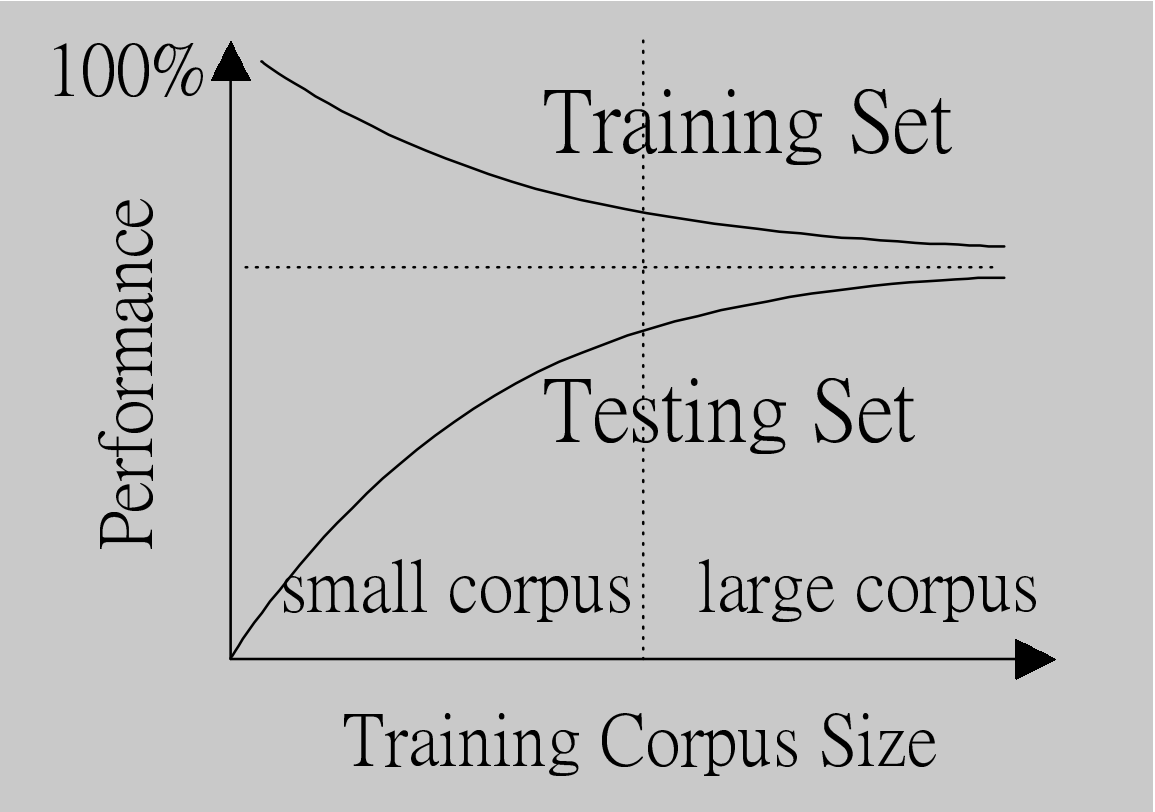
Performance Trends versus Model Complexity

- Problems with High Model Complexity
  - Reducing Modeling Error by Increasing Model Complexity does not increase the testing set performance without limit



Performance Trends versus Training Corpus Size

- Coverage Rate: increases while Corpus-size increases



Statistic Characteristics Mismatch (I)

- Caused by adopting the testing set with different domains or styles (via sampling from different sources/ locations, at different time, etc.)
  - Language usage is usually very dynamic in the real world (very difficult to precisely predicate every possible situation that will occur in the real applications)
  - Pre-assumed conditions rarely can keep long
- Generating the mismatch between two sampling sources
  - Mismatch between Lexicon usage statistics (mainly in domain mismatch)
  - Mismatch between other syntactic (or semantic) patterns statistics (e.g. Style)

Statistic Characteristics Mismatch (II)

- Model Sensitivity (versus Characteristics Variation) is low
  - If the adopted features are invulnerable (e.g., having large inter-class distance, and small intra-class variance)
  - If the adopted estimation method is robust (e.g., adopting smoothing techniques, discarding outliers, etc.)
- Model Sensitivity usually goes up when the model complexity goes up
  - Simple is beautiful (if it can provide the similar training set performance, then it will usually deliver better testing set performance) !
  - Less parameters is better (if both give the similar training set performance)

Methods to Reduce Mismatch Effect

- Reduce Measuring Functions Mismatch Effect
  - Adopting good language model that is closely related to the human preference model
  - Adopting heuristic initial guess (or adopting seed corpus) to avoid local trap
- Reduce Measuring Sources Mismatch Effect
  - Adopting the language models and the estimation methods that are robust (I.e., insensitive) to the statistical characteristics variation (also the sampling variation) between the training set and the testing set
  - Adopting class-based approaches, if necessary, and smoothing techniques to lessen the effect caused by the finite sampling size (remember, the estimation error cannot be perceived in the training set)