[Day-2] Unsupervised Learning for Natural Language Processing

(Part IV: Potential Traps & Source of Problems)

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

(2002/8/18)

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-2: Unsupervised Learning for Natural Language Processing

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 (1)

- 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 highly correlated, if not under proper setting
  - Sources Resulting Mismatch
    - Adopting different measuring functions
    - Sampling from different sources

Criteria Mismatch (2):

- 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 (3):

- 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 (1):

- *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 (2):

- Mismatch 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 could still be used as a good starting point in supervised learning, and no other better way in unsupervised learning
  - 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 in supervised learning

Mismatch between Measuring Functions (3):

- For remedying the drawbacks above mentioned, Discriminative Training was proposed to directly pursue "Minimizing error rate" in supervised learning
  - Approximate each error by an analytical Loss Function (e.g., arctan or sigmoid)
  - Searching the parameter space for minimizing the corresponding Risk Function
  - Under discriminative training, minimizing risk function in the training set does imply minimizing error rate in the training set
  - Result: Better performance, more effective in adjusting parameters

Mismatch between Measuring Functions (4):

- In unsupervised learning, however, human preference is not known; therefore, discriminative training cannot be applied in many situations
  - 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 minimize 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 (5):

- Example: using a bi-gram model for part of speech tagging; assume 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 likelihood value: just an exchange of the labels; however, it would result 100% and 0% accuracy rates, respectively
- To make unsupervised learning work, a high correlation between those two measures (likelihood in the training set & error rate in the training 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
    - The parameter estimation error is negligible (i.e., the training 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; again, it is not guaranteed

Mismatch between Sampling Sources (cont.):

- Factors that cause 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
  - Which would cause the mismatch between two measuring functions
- Local Traps
  - Multiple local optimum points inherent in the parameter space
  - Which would cause the mismatch between two measuring functions

Sources for Causing Mismatch (Cont.):

- Insufficient Training Data
  - Large estimation error perceived in the testing set
  - Which would cause the mismatch between two sampling sources
- Statistical Characteristic Variation
  - Different statistical characteristics between training set and testing set
  - Which would cause the mismatch between two sampling sources

Inappropriate Feature Set (1)

- 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:
  - 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
  - Causing the mismatch between two measuring functions

Inappropriate Feature Set (2)

- Naive stochastic language model usually fails to catch Long-distance Dependency (frequently adopted by the human preference model) implied in the deep structure
  - Surface word N-gram was usually adopted
    - With heuristically determined window size (to avoid exponential explosion of the number of parameters)
    - Believe "Data is the King": just get more data (better to be annotated)
    - Usually require a huge number of parameters (as no classes are adopted)
  - At the cost of lower performance by ignoring the features that the long distance dependency requires

Inappropriate Feature Set (3)

- Mismatch of Feature Set Examples:
  - Semantic tags assignment: Semantic Markov Chain (which adopts Semantic Tag N-gram) versus Head-Features
    - with heuristically determined window size
  - Aligning bi-lingual sentences: using length-based feature instead of transfer dictionary
  - IBM Machine Translation Model (I): Free-order word-string versus BDC BehaviorTran linguistic structure
  - OCR/OLCR: crossing counts versus Chinese strokes

Inappropriate Feature Dependencies (1)

- 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 attenuate to almost nothing after 10-step state transitions

Inappropriate Feature Dependencies (2)

- Conditional Independence is inappropriately assumed
  - Assuming features are conditional independent (which is frequently used to drop terms) while they are actually highly correlated
    - Example: $P(f_1, f_2, f_3 | c_i) \cong P(f_1 | c_i) \times P(f_2 | c_i) \times P(f_3 | c_i)$
  - Some features in the adopted feature set are highly correlated, the strong dependency should be utilized in the model
    - For example, it might be better reduced to:

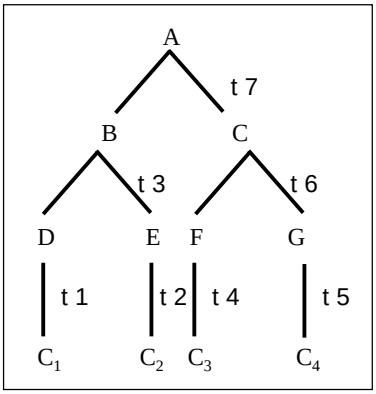
$$P(f_1, f_2, f_3 | c_i) \cong P(f_1 | f_2, c_i) \times P(f_3 | f_2, c_i) \times P(f_2 | c_i)$$

Inappropriate Feature Dependencies (3)

- Example: Parse tree selection with Stochastic Context Free Grammar versus Context-Sensitive Layered-Scoring Function [Su 88, Chiang 96]
  - 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)

Inappropriate Feature Dependencies (4)

- Syntactic Score [Su 88]
- $S_{syn} \equiv P(\text{Syn, Lex} \mid \text{Wrd})$
- Decomposition of Syntax Tree (into phrase levels for score computation in bottom-up GLR parser)



	Reduce	Shift
L8 = {A }	… A …	$ …
L7 = {B, C }	… C …	… … …
L6 = {B, F, G}	… G …	… … …
L5 = {B, F, C_4}	… F …	C_4 …
L4 = {B, C_3, C_4}	… B …	C_3 …
L3 = {D, E, C_3, C_4}	… E …	… … …
L2 = {D, C_2, C_3, C_4}	… D …	C_2 …
L1 = {C_1, C_2, C_3, C_4}	… … …	C_1 …

Inappropriate Feature Dependencies (5)

- Basic Context-Sensitive Formulation

$$P\left(L_j \mid L_1^{j-1}, c_1^n, w_1^n\right) \approx P\left(L_j \mid L_{j-1}\right)$$
$$\equiv P\left(\{l_A, A, r_A\} \mid \{l_A, X_1, X_2, \cdots, X_m, r_A\}\right)$$

  - L_j: the j-th phrase level

$$L_j : \{l_A, A, r_A\} \leftarrow L_{j-1} : \{l_A, X_1, X_2, \cdots, X_m, r_A\}$$

  - Encode context-sensitivity within context-free framework
  - Evaluated after each **reduce** action

- Example:

$$\begin{aligned}
&\mathrm{SCORE}_{\mathrm{syn}}\left(Syn_A\right) \\
&= P\left(L_8, L_7, \cdots, L_2 \mid L_1\right) \\
&= P\left(L_8 \mid L_7, \cdots, L_1\right) \times P\left(L_7 \mid L_6, \cdots, L_1\right) \times \cdots \times P\left(L_2 \mid L_1\right) \\
&\approx P\left(L_8 \mid L_7\right) \times P\left(L_7 \mid L_6\right) \times \cdots \times P\left(L_2 \mid L_1\right) \\
&\approx P\left(\{A\} \mid \{l_7, B, C, r_7\}\right) \times P\left(\{C\} \mid \{l_6, F, G, r_6\}\right) \times \cdots \times P\left(\{D\} \mid \{l_1, c_1, r_1\}\right)
\end{aligned}$$

Inappropriate Feature Dependencies (6)

- Run-Time Formulation (Normalization Form)
  - Compact multiple highly correlated phrase levels when evaluating score (Head-Lexicon in L_1 can be kept in the following example)
  - Evaluated after each **shift** action
  - Avoid "normalization problem" (same input with different number of transition probabilities)
- Example:

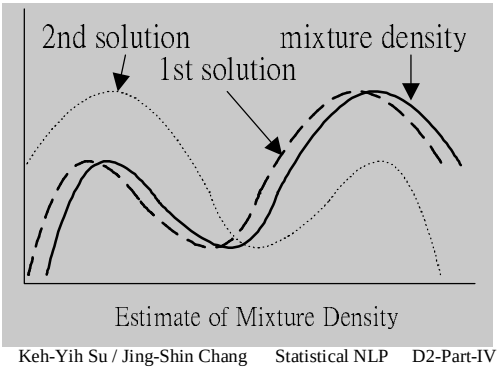
$$\begin{aligned}
&\mathrm{SCORE}_{\mathrm{syn}}\left(Syn_A\right) \\
&= P\left(L_8, L_7, \cdots, L_2 \mid L_1\right) \\
&= P\left(L_8, L_7, L_6 \mid L_1^5\right) \times P\left(L_5 \mid L_1^4\right) \times P\left(L_4, L_3 \mid L_1^2\right) \times P\left(L_2 \mid L_1\right) \\
&\approx P\left(L_8, L_7, L_6 \mid L_5\right) \times P\left(L_5 \mid L_4\right) \times P\left(L_4, L_3 \mid L_2\right) \times P\left(L_2 \mid L_1\right) \\
&\approx P\left(L_8 \mid L_5\right) \times P\left(L_5 \mid L_4\right) \times P\left(L_4 \mid L_2\right) \times P\left(L_2 \mid L_1\right) \quad \text{[transition between shifts]}
\end{aligned}$$

Local Maximum Trap

- 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 generates the same likelihood value
  - May be trapped to the completely reversed (& the worst) candidate if not guided by human preference

Local Maximum Trap (Cont.)

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



Insufficient Training Data (1)

- 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)
  - Cross Entropy always increases 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) in the training set;
    - On the other hand, perplexity always decreases with the same procedure (as prediction capability enhanced)
  - 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 (2)

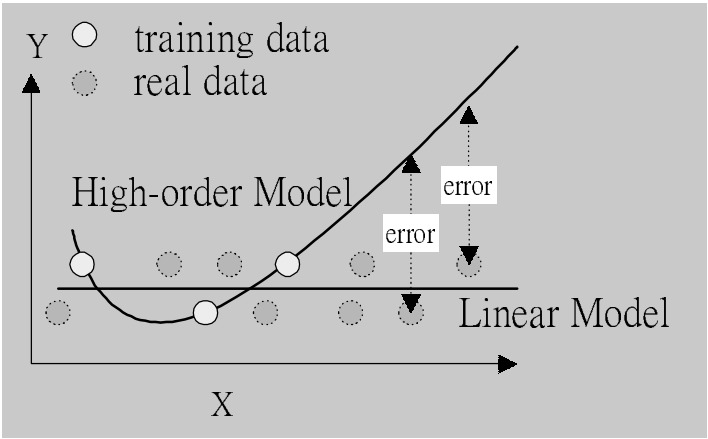
- Example I: increase “N” in an N-gram model
  - Increasing “N” also increases the maximum likelihood value that we can obtain in the training set
  - It also decreases the error rate in the training set under the supervised mode, as the modeling error will be reduced too (through covering wider context)
  - However, the error rate in the testing set will go up eventually if you keep increasing the “N”.

Insufficient Training Data (3)

- 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 (4)

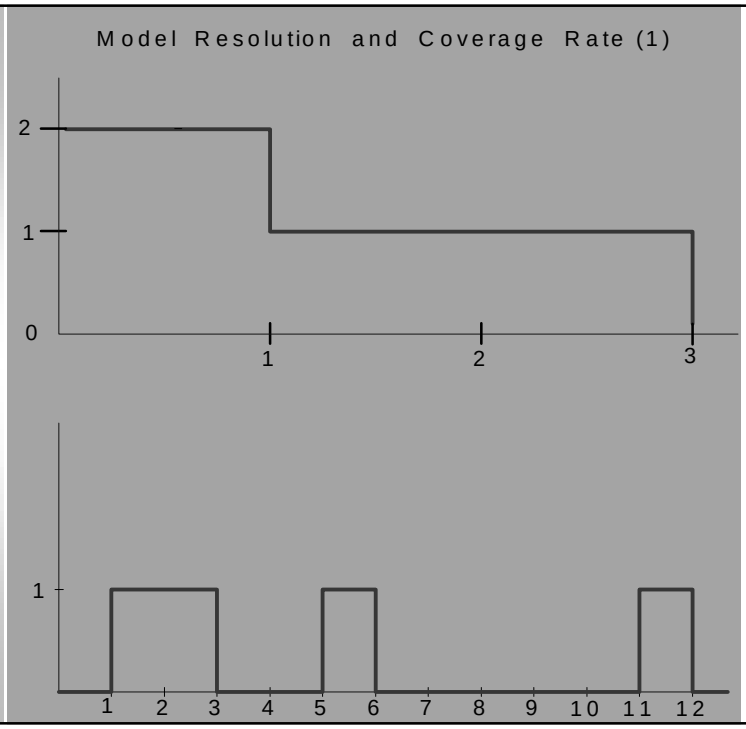
- Over-Tuning Effect
  - Usually, in the first few iterations, the performance in the testing set goes up; however, as the iteration keeps going, the performance in the testing set starts to deteriorate with iteration (although the associated training set likelihood value still keeps increasing)
  - The main reason is that if the model tuned to fit the training set data too much, the data sampling variation between the training set and the testing set will be unveiled
  - This effect is very similar to the over-tuning effect in the adaptive learning of supervised learning
  - A cross-validation set can be adopted to help decide when to stop

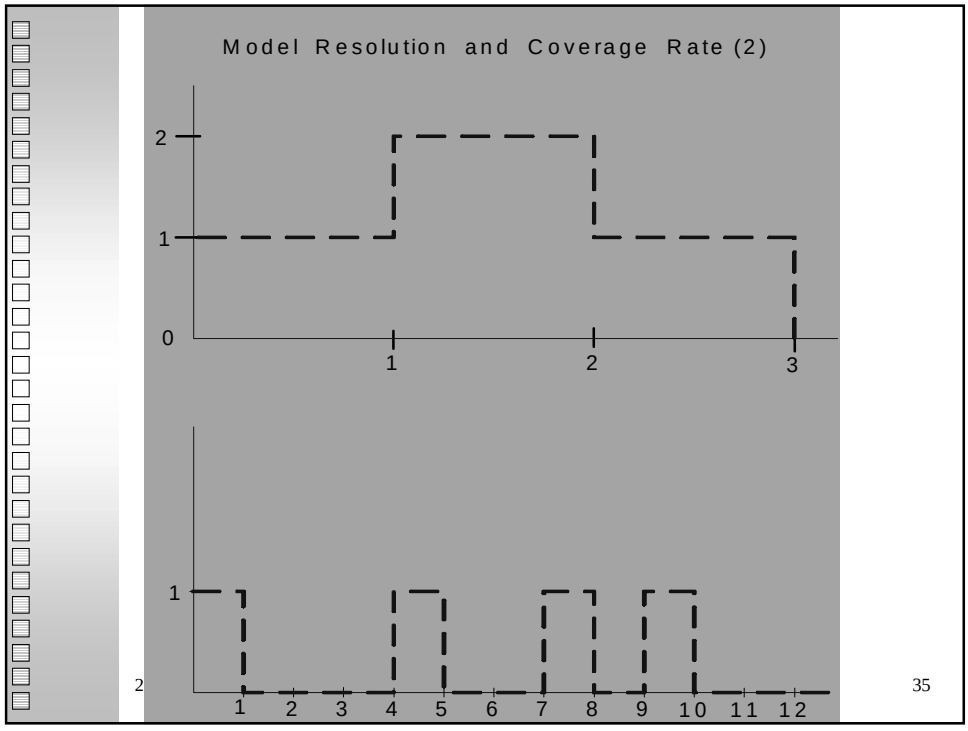
Insufficient Training Data (5)

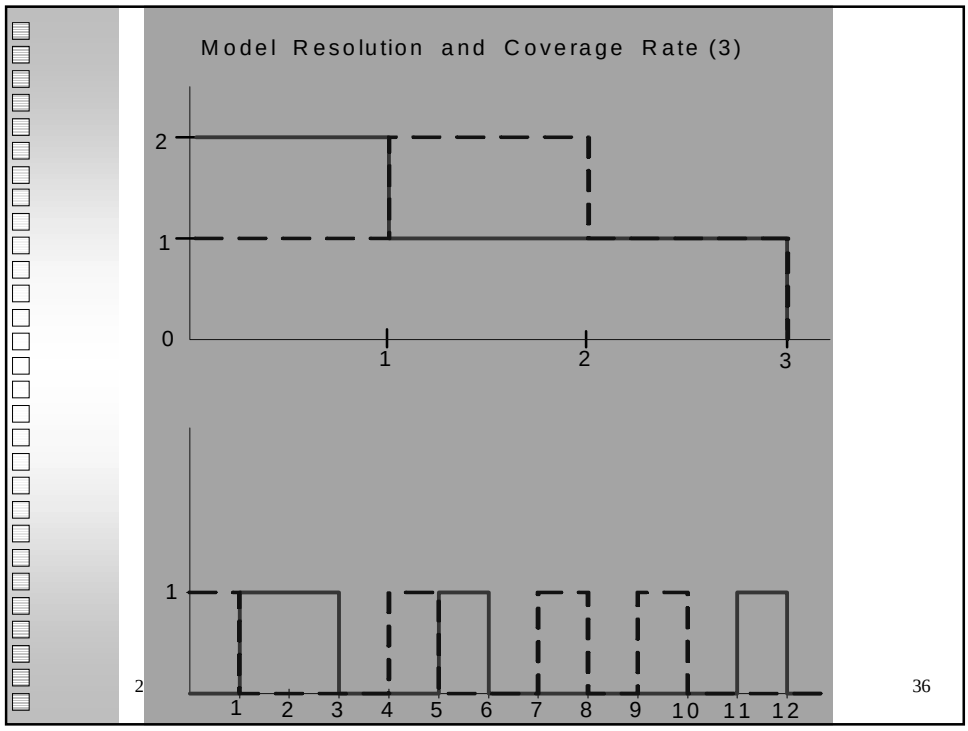
- Model Resolution versus Coverage Rate in the Feature Space
  - Increasing the model resolution (by increasing the model complexity, or by reducing the model covering scope) usually decreases the coverage rate in the testing set
    - Increasing the model resolution increases the discrimination power 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) !

Insufficient Training Data (6)

- Model Resolution versus Coverage Rate (Cont.)
  - 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)
    - Please see following figures




Model Resolution and Coverage Rate (2)
2
1
0
1
2
3
1
1 2 3 4 5 6 7 8 9 10 11 12
2
35


Model Resolution and Coverage Rate (3)
2
1
0
1
2
3
1
1 2 3 4 5 6 7 8 9 10 11 12
2
36

Insufficient Training Data (7)

- 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

General 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

Statistic Characteristics Mismatch

- 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 hold 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 (Cont.)

- 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.
    - The features that peoples really use for understanding utterance are robust (e.g., four-tones in Chinese); otherwise, he cannot be understood
  - 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)