Dynamic Bayesian Networks

Peter Nabende

Alfa Informatica,
Center for Language and Cognition Groningen,
University of Groningen
email: p.nabende@rug.nl
 Outline

• Graphical Models
• Bayesian Networks
  – Description
  – Main Tasks involved
• Dynamic Bayesian Networks
  – Description
  – Main Tasks Involved
• Related work on Dynamic Bayesian Networks for similarity estimation
  – Dynamic Bayesian Network Framework to model context and memory in edit distance learning
  – Evaluation of several phonetic similarity algorithms on the task of cognate identification
Graphical Models

- Way of representing probability distributions in a graph
- Consists of variables (nodes) $X = \{x_1, x_2, \ldots, x_k\}$ with a set $E$ of dependencies (edges) between the variables and set $P$ of probability distribution functions for each variable
- Used to illustrate and work with conditional independencies among variables in a given problem
- Two variables are independent when they have no direct impact on each other’s value. In Fig. 1, $x_1$ is conditionally independent of $x_3$ given $x_2$ if: $P(x_1 \mid x_2, x_3) = P(x_1 \mid x_2)$

Fig. 1: illustration of conditional independence. $x_1$ is conditionally independent of $x_3$ given $x_2$
Graphical Models

• Two major categories exist: Directed & Undirected (Fig.2)

Fig. 2: Classification showing examples of graphical models
Bayesian Networks

• Specific type of graphical model which is a *Directed Acyclic Graph* (DAG), i.e. there are no cycles
• All inter-node connections have a direction, indicated by an arrowhead
• Directionality in the edge indicates parent or cause ($x_p$) to child or effect ($x_c$) relationship

Fig. 3: Directed Graphical Model show causal relationship between parent and child variables
Bayesian Networks

• Compact representation of a joint probability distribution over a set of variables

\[ P(x_1, x_2, x_3) = P(x_1 \mid x_2) \cdot P(x_2) \cdot P(x_3 \mid x_2) \]

Fig. 4: Bayesian Network

• In general, given nodes \( X=\{x_1, \ldots, x_n\} \) the joint probability function for any Bayesian Network (BN) is:

\[ P(X) = \prod_{i=1}^{n} P(x_i \mid \text{parents}(x_i)) \]
Bayesian Networks

- Two major types of Bayesian Networks:
  - Singly connected => only one path for any two variables (Fig. 5a)
  - Multiply connected (Fig. 5b). Can be fully connected

Fig. 5: a) Singly connected DAG (b) Multiply connected DAG [Adapted from Ghahramani, 2007]
Bayesian Networks

• **Inference in BNs**
  – Task of computing the probability of each value of a node in a BN when other variables’ (nodes) values are known
  
  – Two types of inference support:
    • Predictive support for node $x_i$, based on evidence nodes connected to $x_i$ through its parent nodes (top-down reasoning)
    • Diagnostic support (abduction) for node $x_i$, based on evidence nodes connected to $x_i$ through its children nodes (bottom-up reasoning)
Bayesian Networks

• Learning
  – The role of learning is to adjust the parameters of a Bayesian network so that the probability distributions defined by the network sufficiently describes the statistical behavior of the observed data

<table>
<thead>
<tr>
<th>Structure</th>
<th>Observability</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Known</td>
<td>Full</td>
<td>Sample statistics</td>
</tr>
<tr>
<td>Known</td>
<td>Partial</td>
<td>EM or gradient ascent</td>
</tr>
<tr>
<td>Unknown</td>
<td>Full</td>
<td>Search through model space</td>
</tr>
<tr>
<td>Unknown</td>
<td>Partial</td>
<td>Structural EM</td>
</tr>
</tbody>
</table>

Table 1: Learning methods depending on what is already known about the problem (Source: Murphy and Mian, 1999)
Dynamic Bayesian Networks

- Are Bayesian Networks in which variables have a relation to time.
- For a Dynamic Bayesian Network, the following needs to be defined: a) prior network, b) Transition network, c) Observation network, d) End network

Fig. 6 Networks needed to completely specify a DBN for the simple example of the classic HMM
Dynamic Bayesian Networks

Fig. 7: A HMM, a simple type of DBN

Fig. 8: An HMM with two observation sequences (pair HMM)
Dynamic Bayesian Networks

- Classic HMMs have only one hidden variable and one observation variable
- pair HMM has one hidden variable and two observation variables
- DBNs generally have any number of hidden variables and any number of observation variables
A Dynamic Bayesian Framework to Model Context and Memory in Edit Distance Learning: An Application to Pronunciation Classification

Karim Filali and Jeff Bilmes
Departments of Computer Science & Engineering and Electrical Engineering
University of Washington
Seattle, WA 98195, USA
Introduction

- Edit distance – a common measure of similarity between two strings
- Data-driven methods for the automatic learning of edit costs (Ristad and Yianilos, 1998 – string edit distance; Neuhaus and Bunke, 2004 – graph edit distance)
- Problem of learning string edit distance revisited in a Graphical Models framework
Models of Edit Distance

- $d(s_1^m, t_1^n) =$ string ED between $s_1^m$ and $t_1^n =$ minimum weighted sum of deletions, insertions, and substitutions required to transform $s_1^m$ into $t_1^n$

- A $O(m \cdot n)$ DP algorithm can be used to compute the ED between two strings based on the following recursion:

$$d(s_i^i, t_1^j) = \min \begin{cases} d(s_i^i, t_1^j) + \gamma(< s_i, \varepsilon >), \\ d(s_1^{i-1}, t_1^j) + \gamma(< \varepsilon, t_j >), \\ d(s_1^{i-1}, t_1^{j-1}) + \gamma(< s_i, t_j >), \end{cases}$$

with $d(\varepsilon, \varepsilon) = 0$ and $\gamma: \{< s, t >| (s, t) \neq (\varepsilon, \varepsilon)\} \rightarrow \mathbb{R}_+$, a cost function

- case of Levenshtein distance
Stochastic Models of Edit Distance

- Model joint probability $P(S_1^m = s_1^m, T_1^n = t_1^n \mid \theta)$

- $P(s_1^m, t_1^n \mid \theta) = \sum_z P(Z_1^l = z_1^l, s_1^m, t_1^n \mid \theta)$

- For no dependency between edit operations - memoryless model

$P(Z_1^l, s_1^m, t_1^n \mid \theta)$ is factored as $\prod_i P(Z_i, s_1^m, t_1^n \mid \theta)$

- For context independence

$P(Z_i = z_i, s_1^m, t_1^n \mid \theta) \propto \begin{cases} f_{ins}(b_i) & \text{for } z_i^{(s)} = \epsilon; z_i^{(t)} = t_i \\ f_{del}(s_i) & \text{for } z_i^{(s)} = s_i; z_i^{(t)} = \epsilon \\ f_{sub}(s_i, b_i) & \text{for } (z_i^{(s)}, z_i^{(t)}) = (s_i, t_i) \\ 0 & \text{otherwise} \end{cases}$

- where $a_i$ and $b_i$ are source and target string positions at edit operation $i$
DBNs for learning edit distance

- DBN for memory-less transducer model.

![Diagram of DBN framework with context and memory in edit distance learning.](image)
DBNs for learning edit distance

- Context-dependent model.
DBNs for learning edit distance

- Memory model
Pronunciation Classification

• The task is recognizing words from *surface pronunciations*

• Training data: *lexicon with canonical pronunciations*

• DBN model: $P(w,s,t)$
  – $w$ is the word to predict
  – $s$ is the canonical pronunciation
  – $t$ is the surface pronunciation

• DBN models the transformation from $s$ to $t$

• Models are trained using EM (GMTK toolkit)
Pronunciation Classification

• Data
  – Switchboard data (Godfrey et al., 1992), hand annotated in the context of the Speech Transcription Project in Greenberg et al., 1996)
  – Corpus has much variety in word pronunciations, which can significantly deviate from the prototypical pronunciations found in a lexicon
  – Noise introduced during the annotation of speech segments
  – Reference pronunciation dictionary is a lexicon of 2002 Switchboard speech recognition evaluation
  – Out of 40000 entries, 5000 entries are used
  – For testing STP data is divided into 9495 training words (corresponding to 9545 pronunciations) and 912 (901) test words
DBN Models evaluated

- MCI
- Memory
- Context
- Position
  - Determine whether part of the gain witnessed is due to the implicit dependence on the source-target string position
- Length
  - Models the length of the edit sequence
- Direct
  - Patterned on classical HMM. We have consume a target symbol in each frame
Table 1: DBN based model Results Summary

<table>
<thead>
<tr>
<th>Model</th>
<th>$Z_i$ Dependencies</th>
<th>% Err rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lev</td>
<td>none</td>
<td>35.97</td>
</tr>
<tr>
<td>Baseline</td>
<td>none</td>
<td>35.55</td>
</tr>
<tr>
<td>Memory</td>
<td>$Z_{i-1}$</td>
<td>30.05</td>
</tr>
<tr>
<td></td>
<td>$\text{editOperationType}(Z_{i-1})$</td>
<td>36.16</td>
</tr>
<tr>
<td></td>
<td>$\text{stochastic binary } H_{i-1}$</td>
<td>33.87</td>
</tr>
<tr>
<td></td>
<td>$Z_{i-1}^{(s)}$</td>
<td>29.62</td>
</tr>
<tr>
<td></td>
<td>$Z_{i-1}^{(t)}$</td>
<td>27.65</td>
</tr>
<tr>
<td>Context</td>
<td>$s_i$</td>
<td>21.70</td>
</tr>
<tr>
<td></td>
<td>$t_i$</td>
<td>32.06</td>
</tr>
<tr>
<td></td>
<td>$s_i, s_{i-1}$</td>
<td>20.26</td>
</tr>
<tr>
<td></td>
<td>$t_i, t_{i-1}$</td>
<td>28.21</td>
</tr>
<tr>
<td></td>
<td>$s_i, s_{i-1}, s_{a_i+1}$</td>
<td>29.32</td>
</tr>
<tr>
<td></td>
<td>$s_i, s_{a_i+1}$ ($s_{a_i+1}$ in last frame)</td>
<td>23.14</td>
</tr>
<tr>
<td></td>
<td>$s_i, s_{a_i-1}$ ($s_{a_i+1}$ in first frame)</td>
<td>23.15</td>
</tr>
<tr>
<td>Position</td>
<td>$a_i$</td>
<td>33.80</td>
</tr>
<tr>
<td></td>
<td>$b_i$</td>
<td>31.06</td>
</tr>
<tr>
<td></td>
<td>$a_i, b_i$</td>
<td>34.17</td>
</tr>
<tr>
<td>Mixed</td>
<td>$b_i, s_i$</td>
<td>22.22</td>
</tr>
<tr>
<td></td>
<td>$Z_{i-1}^{(t)}, s_i$</td>
<td>24.26</td>
</tr>
<tr>
<td>Length</td>
<td>none</td>
<td>33.56</td>
</tr>
<tr>
<td></td>
<td>$s_i$</td>
<td>20.03</td>
</tr>
<tr>
<td>Direct</td>
<td>none</td>
<td>23.70</td>
</tr>
</tbody>
</table>
Evaluation of Several Phonetic Similarity Algorithms on the Task of Cognate Identification

Grzegorz Kondrak and Tarek Sherif
Departments of Computing Science
University of Alberta
Edmonton, Alberta, Canada T6G 2E8
Introduction

• Edit distance for measuring phonetic similarity
• Task is cognate identification using machine learning techniques
  – Stochastic Transducer
  – Levenshtein with learned weights
  – CORDI
  – Pair-HMM
  – Dynamic Bayesian Nets
• Compared with manually optimized techniques
  – ALINE
  – A linguistically-motivated metric mainly aimed at comparison with ALINE
Manually Constructed Schemes

• ALINE
  – Originally designed to identify and align cognates in vocabularies of related languages
  – Principle component: a function that calculates the similarity of two phonemes expressed in terms of about a dozen multi-valued phonetic features
    • Example: Phonetic feature manner with values: stop = 1.0, affricative = 0.9, fricative = 0.8, approximant = 0.6, high vowel = 0.4, mid-vowel = 0.2, low-vowel = 0.0
  – Overall similarity score computed by a dynamic programming algorithm (Wagner and Fisher, 1974)
Manually constructed schemes

• A linguistically motivated metric (Mielke, 2005)
  – Aimed at showing that the bias towards phonetically natural classes of phonological classes can be modeled without phonological features
  – Phonetic distance metric based on acoustic and articulatory measures.
  – Method comprises 63 phonetic segments each segment represented by a 7 dimensional vector
  – Phonetic distance between any two phonetic segments is Euclidean distance
Manually constructed schemes

• Comparison of Mielke’s method to ALINE
  – Has limited size of phonetic segment set compared to ALINE
  – Has no obvious reference point
  – Some individual distances in Mielke’s method seem to be unintuitive probably due to lack of perceptual features
  – ALINE is more in agreement with human perceptual judgments than Mielke’s method
Learning algorithms

- **Stochastic Transducer** (Ristad and Yianilos, 1998)
  - Uses EM to learn probabilities of the possible edit operations which are used in either Viterbi or stochastic scoring

- **Levenshtein with Learned Weights** (Initially aimed at improving on the RY model (Mann and Yarowsky, 2001)

- **CORDI**
  - Detects recurrent sound correspondences in bilingual wordlists that can be used for computing similarity score between two words

- **Pair HMM** – trained using the Baum Welch algorithm
Learning algorithms

- DBNs (Filali and Bilmes, 2005)
  - Memoriless context-independent model
  - Memory model
  - Context-dependent model (only context of two letters in the source word)
  - Length model
  - All DBN models implemented using the GMTK toolkit
Experiments

• **Set up**
  
  – Input is a pair of words that have the same meaning in distinct languages
  
  – For each pair, the system produces a score representing the likelihood that the words are cognate
  
  – Candidate pairs are ordered by their scores, and evaluate the ranking using 11 point interpolated average precision (Manning and Schutze, 2001)
  
  – Scores are normalized by the length of the longer word in the pair
Experiments

- **Training Data**
  - Comes from the Comparative Indoeuropean Data Corpus (Dyen et al., 1992)
  - Contains wordlists of 200 basic meanings representing 95 speech varieties from Indo-European family of languages
  - Each word is represented in an orthographic form without diacritics using the 26 letters of the Roman alphabet
  - Approximately 180,000 cognate pairs were extracted from the corpus
Experiments

• Development Data
  – The development set was composed of three language pairs: Italian-Croatian, Spanish-Romanian, and Polish-Russian
  – Represent different levels of relatedness: 25.3%, 58.5%, and 73.5% of the word pairs are cognates respectively

• Test Data
  – The test set consisted of five 200-word lists representing English, German, French, Latin, and Albanian, compiled by Kessler (2001)
  – For the supervised experiments, the test data was converted to have the same orthographic representation as the training data
Experiments

Table 1: 11-point average cognate identification precision for various methods

<table>
<thead>
<tr>
<th>Languages</th>
<th>Proportion of cognates</th>
<th>EDIT</th>
<th>MIEL</th>
<th>ALINE</th>
<th>Method R&amp;Y</th>
<th>LLW</th>
<th>PHMM</th>
<th>DBN</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>German</td>
<td>0.590</td>
<td>0.906</td>
<td>0.909</td>
<td>0.912</td>
<td>0.894</td>
<td>0.918</td>
<td>0.930</td>
</tr>
<tr>
<td>French</td>
<td>Latin</td>
<td>0.560</td>
<td>0.828</td>
<td>0.819</td>
<td>0.862</td>
<td>0.889</td>
<td>0.922</td>
<td>0.934</td>
</tr>
<tr>
<td>English</td>
<td>Latin</td>
<td>0.290</td>
<td>0.619</td>
<td>0.664</td>
<td>0.732</td>
<td>0.728</td>
<td>0.725</td>
<td>0.803</td>
</tr>
<tr>
<td>German</td>
<td>Latin</td>
<td>0.290</td>
<td>0.558</td>
<td>0.623</td>
<td>0.705</td>
<td>0.642</td>
<td>0.645</td>
<td>0.730</td>
</tr>
<tr>
<td>English</td>
<td>French</td>
<td>0.275</td>
<td>0.624</td>
<td>0.623</td>
<td>0.623</td>
<td>0.684</td>
<td>0.720</td>
<td>0.812</td>
</tr>
<tr>
<td>French</td>
<td>German</td>
<td>0.245</td>
<td>0.501</td>
<td>0.510</td>
<td>0.534</td>
<td>0.475</td>
<td>0.569</td>
<td>0.734</td>
</tr>
<tr>
<td>Albanian</td>
<td>Latin</td>
<td>0.195</td>
<td>0.597</td>
<td>0.617</td>
<td>0.630</td>
<td>0.568</td>
<td>0.602</td>
<td>0.680</td>
</tr>
<tr>
<td>Albanian</td>
<td>French</td>
<td>0.165</td>
<td>0.643</td>
<td>0.575</td>
<td>0.610</td>
<td>0.446</td>
<td>0.545</td>
<td>0.653</td>
</tr>
<tr>
<td>Albanian</td>
<td>German</td>
<td>0.125</td>
<td>0.298</td>
<td>0.340</td>
<td>0.369</td>
<td>0.376</td>
<td>0.345</td>
<td>0.379</td>
</tr>
<tr>
<td>Albanian</td>
<td>English</td>
<td>0.100</td>
<td>0.184</td>
<td>0.287</td>
<td>0.302</td>
<td>0.312</td>
<td>0.378</td>
<td>0.382</td>
</tr>
<tr>
<td>AVERAGE</td>
<td></td>
<td>0.2835</td>
<td>0.576</td>
<td>0.597</td>
<td>0.628</td>
<td>0.601</td>
<td>0.637</td>
<td>0.704</td>
</tr>
</tbody>
</table>
Experiments

Table 2: Average Cognate Identification for various DBN models

<table>
<thead>
<tr>
<th>Model</th>
<th>Raw Score</th>
<th>Normalized</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCI</td>
<td>0.515</td>
<td>0.601</td>
</tr>
<tr>
<td>MEM</td>
<td>0.563</td>
<td>0.595</td>
</tr>
<tr>
<td>LEN</td>
<td>0.516</td>
<td>0.587</td>
</tr>
<tr>
<td>CON-FOR</td>
<td>0.582</td>
<td>0.599</td>
</tr>
<tr>
<td>CON-REV</td>
<td>0.624</td>
<td>0.619</td>
</tr>
<tr>
<td>CON-AVE</td>
<td>0.629</td>
<td>0.709</td>
</tr>
</tbody>
</table>
Experiments

Table 3: Phonetic test results

<table>
<thead>
<tr>
<th>Model</th>
<th>Raw Score</th>
<th>Normalized</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCI</td>
<td>0.462</td>
<td>0.430</td>
</tr>
<tr>
<td>MEM</td>
<td>0.351</td>
<td>0.308</td>
</tr>
<tr>
<td>LEN</td>
<td>0.464</td>
<td>0.395</td>
</tr>
<tr>
<td>CON-AVE</td>
<td>0.433</td>
<td>0.414</td>
</tr>
<tr>
<td>CORDI</td>
<td>—</td>
<td>0.629</td>
</tr>
</tbody>
</table>
Discussion

• Results suggest that learning approaches are more effective than manually designed schemes
• ALINE outperforms Mielke’s method
• Among the DBN models, the average context model performs best
• Average context DBN performs just as well as the pair-HMM approach, but substantially better than the RY approach and its modification LLW
• In the unsupervised context, all DBNs fail to perform meaningfully regardless of whether they are normalized or not. CORDI achieves a respectable performance
• Comparison to results from Filali and Bilmes (2005)
  – Memory and length models perform worse overall here
  – Context-dependent model performs well on both tasks
Conclusion

• Propose to work on DBNs for generation tasks, such as machine transliteration
REFERENCES


Models of Edit Distance

- Let \( s_1^m = s_1s_2...s_m \) be a source string over a source alphabet \( A \), 
  \( m = \) length of the string, 
  \( s_i^j = s_i...s_j \) is the substring and \( s_i^j = 0 \), when \( i > j \).

- Likewise \( t_1^n = \) target string over target alphabet \( B \)
  \( n = \) length of \( t_1^n \)

- A source string can be transformed into a target string through a 
a sequence of edit operations 
\(< s, t > ( (s, t) \neq (\varepsilon, \varepsilon) ) \) denotes edit operations in which \( s \) is 
replaced by \( t \). \(< s, t > \) is an insertion if \( s = \varepsilon \) and \( t \neq \varepsilon \),
\(< s, t > \) is a deletion if \( t = \varepsilon \) and \( s \neq \varepsilon \), \(< s, t > \) is a substitution
if \( s \neq \varepsilon \) and \( t \neq \varepsilon \), and in all other cases \(< s, t > \) is an identity.
DBNs for learning edit distance

- Direct model
DBNs for learning edit distance

- Length unrolling model
• Some of the DBNs could be important in learning various properties in languages for example: *inflexion* in highly inflective languages

• How different is *cognate identification* from *machine transliteration*
  – Some models such as CORDI could be better for *Cognate Identification* than the DBNs

• What is the importance of length

• For Russians – *infectives for the female names*?