Time Series Modeling with Hidden Variables and Gradient-Based Algorithms

Piotr Mirowski
Advisor: Prof. Yann LeCun
Courant Institute of Mathematical Sciences, New York University

mirowski@cs.nyu.edu
http://cs.nyu.edu/~mirowski
Overview

Background

Method

Applications
• Not all time series are stationary...
• Beware of the “narrative fallacy”
• “Hidden variables” (memory) could (perhaps) help

[taleb, 2007]
Examples of time series with hidden variables

- Human motion capture
  - Recorded visible markers
  - Many (hidden) physiological variables

- Gene regulation networks
  - Measured mRNA micro-array data (sometimes missing values)
  - Unknown protein expression levels

[Mirowski & LeCun, ECML, 2009; Krouk, Mirowski et al, Genome Biol, 2010]
Time series problems involving hidden variables

Likelihood estimation
- the cat sat on a mat
- my cat sat on the mat
- the cat sat on the rug
- the rat spat on the mat

Inference of hidden representation

Prediction and imputation

Classification and regression

Learning a dynamical system
Reminder: time series models without hidden variables

1\textsuperscript{st} order Markov dynamical model
\[ y_t \approx f(y_{t-1}) \]

\[ P(y_t | y_{t-1}) \]

\[ P(y_t | y_{t-p}) \]

\[ y_t \approx f(y_{t-p}) \]

\[ P(y_t | y_{t-p}) \]

- n-grams on discrete sequences (e.g. language) [Katz, 1987]
- Linear Auto-Regressive models (AR, ARMA) [Kolmogorov]
- Time-Delay Neural Networks (TDNN) [Lang & Hinton, 1988]
- Weighted Kernel Regression, Support Vector Regression (SVR) [Nadaraya, 1964; Cortes & Vapnik, 1995; Muller et al, 1999]
- Gaussian Process Regression [Williams & Rasmussen, 1996]
- Limitation: no long-range dependencies ("short memory")

Maximum Likelihood learning
Adding hidden variables to represent the “memory”

$p^{th}$ order Markov dynamical model

\[
y(t) \approx f(y_{t-p}) \quad P(y_t | y_{t-p})
\]

$p^{th}$ order Markov hidden dynamical model

\[
P(z_t | z_{t-p})
\]
Features of an ideal dynamical model with hidden variables

a. Potentially **high-dimensional continuous** hidden states (variables)

b. Highly **nonlinear** dynamics or observations (outputs/inputs) (e.g. convolutional nets)

c. **Tractable inference** of hidden variables representation

d. Handle long sequences in **linear time**

[Kschischang et al, 2001; Mirowski & LeCun, ECML 2009]
Existing hidden variable models and limitations

- **Hidden Markov Models**
  Discrete hidden variables (“states”)
  [Rabiner, 1989; Baum et al, 1970]

- **Linear Dynamical Systems**
  Continuous hidden variables
  [Kalman, 1960; Nelson, 1976]

- **Nonlinear Dynamical Systems**
  Continuous hidden variables,
  Restricted set of single time-step dynamics

- **Gaussian Process Latent Variable Models**
  Single time-step dynamics, $O(T^2)$ learning
  [Wang et al, 2006; Moon & Pavlovic, 2008]

- **Temporal Restricted Boltzmann Machines**
  Stochastic binary hidden units (sampling)
  [Sutskever & Hinton, 2006; Taylor et al, 2006]

Inference possible; closed-form of Expectation-Maximization
[Dempster et al, 1977]

E-step of EM needs approximations:
sampling or Variational Bayes
Overview

Background

Method

Applications

turkey’s weight

weight

time

Predicted behavior

$Z(t+n)$

$KNO^3/T_0$

15 min

20 min

26 consistent genes

12 min

15 min

53 consistent genes

Observation model

$Y(t)$

$Y(t+1)$

$Y(t+2)$

$Z(t)$

$Z(t+1)$

dynamic model

$f$

Input

Output

$B$

$C$

function: Influence Matrix

$\text{Learned set}$

$\text{Learned set}$

Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec

−0.2
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6

log

GM

JPM

Student Version of MATLAB

Background

Method

Applications
Dynamic Factor Graphs
Definition of Factor Graphs

- Random **variables**

- **Factors**
  - conditional (in)dependencies

- **Dynamic Factor Graphs**: our formalization of a dynamical system

- Joint likelihood:
  \[ P(X, Y, Z) = \prod_{t} P(Z_t|Z_{t-1}) P(Z_t|X_t) P(Y_t|Z_t) P(X_t) \]

- In log-space,
  \[ \log P(X, Y, Z) = c + \sum_{t} \log P(Z_t|Z_{t-1}) + \log P(Z_t|X_t) + \log P(Y_t|Z_t) \]

[Kschischang et al, 2001]
Learning and inference in DFGs

• **Learning**: fitting a model (parameterized by \( W \)) minimizing the negative log-likelihood of observed data \( X, Y \)

\[
nll (X, Y; W) = - \log P(X, Y; W)
\]

• **Inference**: finding latent (\( \sim \)hidden) \( Z \)

• Graphical models try to integrate out \( Z \) (often intractable)

\[
nll (X, Y; W) = - \log \int_z P(X, Y, z; W) \, dz
\]

• We do not integrate out \( Z \)

• **Energy function** for a given configuration of \( X, Y, Z \)

\[
E(X, Y, Z; W) \propto - \log P(X, Y, Z; W) + \text{const}
\]

• **maximum a posteriori**

\[
\arg \max_z P(X, Y; W) = \arg \min \tilde{z} E(X, Y; W)
\]

[Ghahramani, 1997; Dempster et al, 1977; LeCun et al, 2006; Mirowski & LeCun, ECML 2009]
DFGs as a state-space model

**Observation model**

\[ y_t = g(z_t) + \eta_t \]

**Dynamical model** (\(p^{\text{th}}\) order Markov) with complex dependencies on \(Y\) and \(Z\)

\[ z_t = f(z_{t-p}, y_{t-1}) + \epsilon_t \]

[Mirowski & LeCun, ECML 2009]
**DFGs enable highly nonlinear factors: convolutional networks**

- **Observation** or **dynamic** models with higher-order nonlinearity than graphical models:
  - radial basis functions [Ghahramani & Roweis, 1999]
  - single hidden-layer Perceptrons [Ilin et al, 2004]
  - No close-form optimization

- **use gradient-based inference and learning**
  [LeCun et al, 1998a; Waibel et al, 1989; Wan, 1993; Mirowski et al, AAAI 2007]

n-dimensional input with time embedding $p=11$:
- size $n \times p$
- $1 \times 3$ convolution (across time); time-step of 2
- Layer 1: 12 filters; $n \times 5$
- Layer 2: 12 filters; $1 \times 3$
- Layer 3: Full connection; n-dimensional vector
- $n \times 12 \times 3$ convolution (across time, filters and components); time-step of 1

"channels" (dimensions)
- time lags

$y(t-1)$, $y(t)$, $y(t+1)$
$z(t-1)$, $z(t)$, $z(t+1)$

$dynamical model f$

12 filters:
- $1 \times 3$
- $n \times 5$

Layer 1

Layer 2

Layer 3

Full connection: n-dimensional vector

12x3 full connection
Learning and inference:
deterministic gradient-based EM

\[ E_g (y_{t-1}, \bar{y}_{t-1}) \]

\[ g (z_{t-1}; W_g) \]

\[ E_f (z_t, \bar{z}_t) \]

\[ f (z_{t-1}^{p}; W_f) \]

[Mirowski & LeCun, ECML 2009]
Learning and inference: deterministic gradient-based EM

\[
\Delta z_{t-1} = -\eta_z \left( \frac{\partial E_f}{\partial z_{t-1}} + \frac{\partial E_g}{\partial z_{t-1}} \right)
\]

\[
\Delta z_t = -\eta_z \left( \frac{\partial E_f}{\partial z_t} + \frac{\partial E_g}{\partial z_t} \right)
\]

[Mirowski & LeCun, ECML 2009]
Learning and inference:
deterministic gradient-based EM

\[
E_g(y_{t-1}, \bar{y}_{t-1})
\]

\[
g(z_{t-1}; W_g)
\]

\[
E_f(z_t, \bar{z}_t)
\]

\[
f(z_{t-p}; W_f)
\]

[Mirowski & LeCun, ECML 2009]
Learning and inference:
deterministic gradient-based EM

\[ \frac{\partial E_g}{\partial \bar{y}_{t-1}} \]

\[ \frac{\partial E_g}{\partial z_{t-1}} \]

\[ \frac{\partial E_f}{\partial \bar{z}_t} \]

\[ \frac{\partial E_f}{\partial z_t} \]

\[ \frac{\partial E_f}{\partial z_{t-1}} \]

\[ \frac{\partial E_f}{\partial z_{t-p}} \]

"backprop"

[Mirowski & LeCun, ECML 2009]
Learning and inference:
deterministic gradient-based EM

\[ E_g(y_t, \bar{y}_t) \]

observation
energy
“fprop”
observation
parameters \( W_g \)

\[ g(z_{t-1}; W_g) \]

\[ g(z_t; W_g) \]

\[ E_g(y_{t-1}, \bar{y}_{t-1}) \]

\[ E_g(y_{t-1}, \bar{y}_{t-1}) \]

[Mirowski & LeCun, ECML 2009]
Learning and inference:
deterministic gradient-based EM

\[ \Delta W_g = -\eta_W \sum_t \frac{\partial E_g(t)}{\partial W_g} \]

\[ \Delta W_g = \sum_t \frac{\partial E_g(t-1)}{\partial W_g} \]

\[ \Delta W_g = \sum_t \frac{\partial E_g(t)}{\partial W_g} \]

- one gradient step
- full gradient descent
- conjugate gradient [LeCun et al, 1998b]
- stochastic gradient descent [Bottou, 2004]
- other: LARS [Tibshirani, 1996], ridge regression, etc...

[Mirowski & LeCun, ECML 2009]
Learning and inference: deterministic gradient-based EM

\[ y_{t-1} \quad y_t \]

\[ z_t \quad \bar{z}_t \]

\[
E_f (z_t, \bar{z}_t) = f (z_{t-1}; W_f)
\]

\[
\text{dynamical energy}
\]

\[
\text{dynamical parameters} \quad W_f
\]

[Mirowski & LeCun, ECML 2009]
Learning and inference: deterministic gradient-based EM

\[ \Delta W_f = -\eta_W \sum_t \frac{\partial E_f(t)}{\partial W_f} \]

- one gradient step
- full gradient descent
- conjugate gradient [LeCun et al., 1998b]
- stochastic gradient descent [Bottou, 2004]
- other: LARS [Tibshirani, 1996], ridge regression, etc...

"backprop"
Learning and inference: deterministic gradient-based EM

\[ E_g(y_{t-1}, \bar{y}_{t-1}) \]

\[ g(z_{t-1}; W_g) \]

\[ E_f(z_t, \bar{z}_t) \]

\[ f(z_{t-1}; W_f) \]

\[ \min_{z} \sum_t E_f(t) + \gamma E_g(t) \]
Learning and inference: deterministic gradient-based EM

\[ E_f(z_t, \bar{z}_t) \]

\[ g(z_{t-1}; W_g) \]

\[ E_g(y_{t-1}, \bar{y}_{t-1}) \]

\[ E_f(z_t, \bar{z}_t) \]

\[ g(z_t; W_g) \]

\[ E_g(y_t, \bar{y}_t) \]

\[ f(z_{t-1}; W_f) \]

\[ \min_{z} \sum_{t} E_f(t) + \gamma E_g(t) \]

\[ \min_{W} \sum_{t} E_f(t) + \gamma E_g(t) \]
Stochastic vs. batch learning and inference

[LeCun et al, 1998b; Mirowski & LeCun, ECML 2009]
Regularization during learning and inference

\[ \mathcal{L}(Y, Z; W) = \sum_t E_g(t) + \gamma \sum_t E_f(t) + R_W(t) + R_Z(t) - \log \Gamma_{Y,Z} \]

- **Loss function**
- **Observation energy**
- **Dynamic energy**
- **Latent variable regularization**

**Parameter regularization:**
- \( L_1 \)-norm (sparsity)
- \( L_2 \)-norm (bound)

**Constant log-partition function (ignored)**

**“Lagrange” coefficient:**
- Importance of factor \( f \) during inference

[Vapnik, 1995; Tibshirani, 1996; Mirowski & LeCun, ECML 2009]
Overview

Background

Method

Applications

Figure 3

Z(t+n)

Predicted behavior

>8

<8

KNO

3

T0

Leave-out-two-last

12 min

15 min

20 min

26 consistent genes

Leave-out-last

15 min

20 min

53 consistent genes

Learnt set

Learnt set

Predicted behavior

B

C

f

function: Influence Matrix

Input

Output

15

12 min

15 min

15

20 min

15

20 min

A

Observation

model

Y(t+1)

Y(t+n)

Y(t)

Y(t+2)

Z(t)

Z(t+1)

Z(t+1)

dynamic model

15

20 min

15

20 min

GMGMQ

C

JPM

Student Version of MATLAB

Background

Method

Applications
Application #1: nonlinear time series modeling

[Mirowski & LeCun, ECML 2009]
Time series problems

Likelihood estimation

the cat sat on a mat
my cat sat on the mat
the cat sat on the rug
the rat spat on the mat

Inference of hidden representation

Prediction and imputation

Classification and regression

Learning a dynamical system
Highly nonlinear dynamic factors: convolutional networks (TDNN)

n-dimensional input with time embedding $p=11$:
- **Layer 1**: 12 filters: $n \times 5$
  - $1 \times 3$ convolution (across time); time-step of 2
- **Layer 2**: 12 filters: $1 \times 3$
  - $n \times 12 \times 3$ convolution (across time, filters and components); time-step of 1
- **Layer 3**: Full connection: n-dimensional vector

"channels" (dimensions) + time lags

Observation model $g$

Dynamical model $f$

[LeCun et al, 1998a; Waibel et al, 1989; Wan, 1993; Mirowski & LeCun, ECML 2009]
Results: Inferring the Lorenz chaotic attractor

Data

Lorenz chaotic attractor

Latent state attractor inferred on test data is similar to Lorenz attractor ($|Z| = 3$)

Results

Partial observation:

$$y(t) = y_1(t) + y_2(t) + y_3(t)$$

1-step prediction error of -46.2 dB smaller than in SVR (-41.6 dB)

Lorenz attractor reconstructed on hidden variables

[ Lorenz, 1963; Mattera et al, 1999; Mirowski & LeCun, ECML 2009]
Results: Missing MoCap markers reconstructed

Data

- **Observations Y**: 49-dimensional **Motion Capture markers**

Problem

- **Model data occlusion**
  - Test sequence: 260 frames
  - 2 subsequences of 65 frames with missing data:
    - Left leg
    - Entire **upper body**

Approach

- Infer **hidden variables Z** (E-step) on test sequence (|Z|=147) (without gradient from missing $Y_i(t)$), and **generate Y from Z**

[Taylor et al, 2006; Mirowski & LeCun, ECML 2009]
Results: Missing MoCap markers reconstructed

Original data

Reconstruction of missing upper body

Reconstruction of missing left leg

Lower NMSE than nearest neighbors;
Inferred smooth, realistic motion

<table>
<thead>
<tr>
<th></th>
<th>Nearest neighbors</th>
<th>DFG</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE left leg 1</td>
<td>0.77</td>
<td>0.59</td>
</tr>
<tr>
<td>MSE left leg 2</td>
<td>0.47</td>
<td>0.39</td>
</tr>
<tr>
<td>MSE upper 1</td>
<td>1.24</td>
<td>0.9</td>
</tr>
<tr>
<td>MSE upper 2</td>
<td>0.8</td>
<td>0.48</td>
</tr>
</tbody>
</table>

[Taylor et al, 2006; Mirowski & LeCun, ECML 2009]
Results: CATS time series

Data and problem

CATS time series prediction competition

Noisy chaotic time series (5000 points) with missing data (100 points)

Results

Our predictions beat the CATS benchmark

<table>
<thead>
<tr>
<th></th>
<th>Kalman Smoothers (CATS winner)</th>
<th>DFG</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE all 5 segments</td>
<td>4.08</td>
<td>3.90</td>
</tr>
<tr>
<td>MSE first 4 segments</td>
<td>3.46</td>
<td>2.88</td>
</tr>
</tbody>
</table>

[LeBandasse et al, 2004; Mirowski & LeCun, ECML 2009]
Application #2: discovery of gene regulation networks

[Krouk, Mirowski et al, Genome Biology 2010]
Time series problems

Likelihood estimation

the cat sat on a mat
my cat sat on the mat
the cat sat on the rug
the rat spat on the mat

Prediction and imputation

Inference of hidden representation

Classification and regression

Learning a dynamical system

the cat sat on a
my cat sat on the
the cat sat on the rug
the rat spat on the mat

the cat sat on a mat
my cat sat on the mat
the cat sat on the rug
the rat spat on the mat
Problem 4: Learning Genetic Regulatory Networks (GRN)

Micro-array data

Arabidopsis in reaction to NO$_3^-$

4 time series (replicates)
28 time-points

>22k genes, clustering restricts to subset of 76 genes

Problem: learning a GRN from mRNA

GRN assumption:

\[ y_t \approx f(y_{t-1}) \]

[4]

[Krouk, Mirowski et al, Genome Biology 2010]
Problem 4: Learning Genetic Regulatory Networks (GRN)

1) DFG models **data uncertainty**
2) Simple, regularized linear dynamics
3) Bootstrapping (random initializations) to compute **statistically significant links**
Results: Learning Genetic Regulatory Networks (GRN)

Out-of-sample tests: **predict the trend** (up/down) of mRNA

<table>
<thead>
<tr>
<th></th>
<th>Fit on train data (SNR)</th>
<th>Leave-out-last accuracy on test data</th>
</tr>
</thead>
<tbody>
<tr>
<td>naive trend forecast</td>
<td>-</td>
<td>52%</td>
</tr>
<tr>
<td><strong>DFG + grad</strong></td>
<td>32.4dB</td>
<td>69%</td>
</tr>
<tr>
<td><strong>DFG + LARS</strong></td>
<td>32.4dB</td>
<td>74%</td>
</tr>
<tr>
<td><strong>LARS</strong></td>
<td>32.1dB</td>
<td>71%</td>
</tr>
</tbody>
</table>

**DFG can successfully predict gene behavior in unseen conditions**

**Biological confirmation (QPCR):** hubs in GRN are key TFs

[Zou & Hastie, 2005; Krouk, Mirowski et al, Genome Biology 2010]
Problem 5: Inferring hidden protein transcription factors

rate of mRNA concentration
rate of protein (transcription factor) concentration

basal rate
decay term
protein-mRNA transcription

\[
\frac{dy_i}{dt} = b_i - d_i y_i(t) + \sum_{j=1}^{M} s_{i,j} \sigma(z_j(t)) + \epsilon_{i,t}
\]

protein-mRNA translation

\[
\frac{dz_j}{dt} = -\delta_j z_j(t) + w_j y_j(t) + \eta_{j,t}
\]
mRNA-protein translation

[Jaeger & Monk, 2010; Gao et al, 2008]
Results\(^5\): Inferring multiple hidden protein TFs

Human cancer TGF-\(\beta\) dataset:
mRNA for 70 genes
8 time points

4 hidden proteins
biological ODE kinetics,
known experimental profile (iTRAQ)

Inferred protein profiles for 3 proteins matched experimental iTRAQ data

[Human cancer TGF-\(\beta\) dataset: mRNA for 70 genes
8 time points]

4 hidden proteins
biological ODE kinetics,
known experimental profile (iTRAQ)

Inferred protein profiles for 3 proteins matched experimental iTRAQ data

[Keshamouni et al, 2009]
Application #3: topic modeling of time-stamped documents

[Mirowski et al, NIPS Deep Learning Workshop 2010]
Time series problems

Likelihood estimation

the cat sat on a mat
my cat sat on the mat
the cat sat on the rug
the rat spat on the mat

Inference of hidden representation

Classification and regression

Prediction and imputation

Learning a dynamical system
Idea: Dynamics on latent topics in time-stamped documents

- Intuition: drift of latent news topics + “scoops”

- Examples of datasets investigated, exhibiting a temporal structure:
  - Financial/business news articles
    1 year of Bloomberg news (2008) for 30 stocks + stock volatility
    Reuters 21578 + classification
  - Political speeches
    US State of the Union addresses (1790-2010)
  - Scientific publications
    NIPS 1987-2003

- Field recently explored by unsupervised techniques:
  Dynamic Topic Models, Dynamical Hierarchical Dirichlet Processes

[Taleb 2007; Blei et al, 2006; Wang et al, 2008; Pruteanu-Malinci et al, 2009]
Idea: Dynamic Factor Graphs for time-stamped documents

- DFG loss expresses trade-off between:
  - reconstruction of word histograms
  - classification
  - simple topic dynamics between consecutive time units
Idea: Stacked DFGs for time-stamped documents

Stacked Dynamic Factor Graphs with non-linear dimensionality reduction

[Ranzato et al, 2008; Hinton et al, 2006]
In our maximum a posteriori approach we replace the full distribution over variational inference or Gibbs sampling with the standard approximation made by LDA which is that the topic assignment small number of topics, documents and vocabulary size, although approximate techniques.

Estimating the likelihood of a document given a topic model is intractable even for the marginal distribution of documents to be a bag of independent words. Each document is associated with a latent representation that maximizes the likelihood. We rewrite a delta distribution with a mode at  by a delta distribution with a mode at 

\[
\bar{\theta}_{i,n} \approx 0
\]

In the field of topic models, the perplexity measures the difficulty of predicting documents after training model

\[
P \equiv p \left( \{w_i\}_{i=1}^T | \Omega \right) \approx \exp \left( - \frac{\sum_{i=1}^T \log p(w_i|\Omega)}{\sum_{i=1}^T N_i} \right)
\]

**Results: Unsupervised DAE as dynamic topic models**

Out-of-sample topic-model perplexity

\[
P \equiv p \left( \{w_i\}_{i=1}^T | \Omega \right) \approx \exp \left( - \frac{\sum_{i=1}^T \log p(w_i|\Omega)}{\sum_{i=1}^T N_i} \right)
\]

Results on State of the Union addresses (1790-2010), yearly dynamics

2000 words or entities

Our DAE's architecture is 100-30-10-2

Train on 17k paragraphs (1790-1989)

Test on 1965 paragraphs (1990-2010)

**Lower perplexity than LDA**

**L1 preserves the dynamics**

<table>
<thead>
<tr>
<th>K</th>
<th>LDA</th>
<th>DAE no dyna</th>
<th>DAE L1</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>739</td>
<td>197</td>
<td>218</td>
</tr>
<tr>
<td>30</td>
<td>951</td>
<td>481</td>
<td>514</td>
</tr>
<tr>
<td>10</td>
<td>1154</td>
<td>1008</td>
<td>859</td>
</tr>
<tr>
<td>2</td>
<td>1428</td>
<td>1553</td>
<td>1206</td>
</tr>
</tbody>
</table>

Results\textsuperscript{7}: Topic trajectories of time-stamped documents

State of the Union addresses yearly average (over paragraphs) of 2-dimensional codes

Auto-encoder

2000-100-30-10-2

L\textsubscript{1} dynamics

Hierarchical trajectory over time of latent topics

Auto-encoder

2000-100-30-10-2

no dynamics

3-topic LDA

[Van der Maaten & Hinton, 2008; Blei et al, 2003; Mirowski et al, NIPS Deep Learning 2010]
Results: Text categorization and information retrieval

2000w-TFIDF logistic regression $F_1=0.83$
2000w-TFIDF SVM $F_1=0.84$

<table>
<thead>
<tr>
<th>K</th>
<th>ICA</th>
<th>DAE no dynamics</th>
<th>DAE $L_1$ dynamics</th>
<th>R&amp;S</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.81</td>
<td>0.86</td>
<td>0.85</td>
<td>0.85</td>
</tr>
<tr>
<td>30</td>
<td>0.70</td>
<td>0.86</td>
<td>0.85</td>
<td>0.85</td>
</tr>
<tr>
<td>10</td>
<td>0.40</td>
<td>0.86</td>
<td>0.85</td>
<td>0.84</td>
</tr>
<tr>
<td>2</td>
<td>0.09</td>
<td>0.54</td>
<td>0.51</td>
<td>0.19</td>
</tr>
</tbody>
</table>

2000w-TFIDF information retrieval AUPR=0.51

<table>
<thead>
<tr>
<th>K</th>
<th>LDA</th>
<th>DAE no dynamics</th>
<th>DAE $L_1$ dynamics</th>
<th>R&amp;S</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.42</td>
<td>0.90</td>
<td>0.89</td>
<td>0.87</td>
</tr>
<tr>
<td>30</td>
<td>0.49</td>
<td>0.62</td>
<td>0.63</td>
<td>0.93</td>
</tr>
<tr>
<td>10</td>
<td>0.54</td>
<td>0.81</td>
<td>0.78</td>
<td>0.89</td>
</tr>
<tr>
<td>2</td>
<td>0.25</td>
<td>0.71</td>
<td>0.75</td>
<td>0.70</td>
</tr>
</tbody>
</table>

We outperform TF-IDF + (logreg/SVM), all unsupervised embeddings + (logreg/SVM) and match R&S 2008

Application #4: statistical language modeling

the cat sat on a mat

my cat sat on the mat

the cat sat on the rug

the rat spat on the mat

[Mirowski et al, SLT 2010]
Time series problems

Likelihood estimation

the cat sat on a mat
my cat sat on the mat
the cat sat on the rug
the rat spat on the mat

Inference of hidden representation

Classification and regression

Prediction and imputation

Learning a dynamical system
Problem: \textit{n-grams and statistical language models}

\texttt{the cat sat on the mat} \quad P (w_t | w_{t-5}^{t-1}) = 0.15 \\
\texttt{w_{t-5} \ w_{t-4} \ w_{t-3} \ w_{t-2} \ w_{t-1} \ w_{t}}

- \textbf{\textit{n-grams}} define \textbf{conditional probabilities} \\
\[ P (w_t | w_{t-1}, w_{t-2}, w_{t-3}, w_{t-4}, w_{t-5}) \]

- syntactic likelihood of seeing a sub-sequence in a language \\
\texttt{the cat sat on the \textbf{hat}} \quad P (w_t | w_{t-5}^{t-1}) = 0.05 \\
\texttt{the cat sat on the \textbf{sat}} \quad P (w_t | w_{t-5}^{t-1}) = 0
Problem: Limitations of $n$-grams

- there are $V^n$ $n$-grams on $V$ words (exponential in $n$)

- *discrete* model

- most real datasets have *incomplete* coverage:
  the dataset could miss the following $n$-grams:

  \[
  \begin{align*}
  \text{the cat sat on a mat} & \quad P(w_t|w_{t-5}^{t-1}) = \? \\
  \text{my cat sat on the mat} & \quad P(w_t|w_{t-5}^{t-1}) = \? \\
  \end{align*}
  \]

- what about *semantic similarity?* not used...

  \[
  \begin{align*}
  \text{the cat sat on the rug} & \quad P(w_t|w_{t-5}^{t-1}) = \? \\
  \text{the rat spat on the mat} & \quad P(w_t|w_{t-5}^{t-1}) = \? \\
  \end{align*}
  \]

[Katz, 1987; Goodman & Chen, 1996]
Goal: To “learn a language” is to maximize a likelihood on a corpus

\[ P \left( w_1^T \right) = P \left( w_1^{n-1} \right) \prod_{t=n}^{T} P \left( w_t | w_{t-n+1}^{t-1} \right) \]

Instead of \( n \)-grams, we will model

a distributed representation for the probabilities
**Goal: Energy-based models of word probabilities**

\[
E(t, v) = -\mathbf{z}_t^T \mathbf{z}_v - b_v
\]

Normalized probabilities:

\[
P \left( w_t = w_v \mid \mathbf{w}_{t-n+1} \right) = \frac{e^{-E(t, v)}}{\sum_{v' = 1}^{\left| \mathbf{W} \right|} e^{-E(t, v')}}
\]

- \(\mathbf{z}_t\): representation of target word \(w_t\)
- \(\mathbf{z}_v\): representation of any word \(v\) in the vocabulary
Log-BiLinear Nonlinear model

Our model

[Mirowski et al, SLT 2010]

sentence or document topic simplex from LDA
[Blei et al, 2003]
Idea: Add syntactic features and long-range dependencies

- add Part-Of-Speech features as inputs: 
  *improvement*
- add Supertag features 
  [Joshi, 1997; Bangalore, 1999]:
  *large improvement*
- topic mixtures based on LDA simplex 
  [Blei et al, 2003]:
  *large improvement* on large corpora
Results\textsuperscript{9}: Supertags dramatically improve LM perplexity

Wall Street Journal corpus
10k vocabulary (no proper nouns, no numbers)

<table>
<thead>
<tr>
<th>POS / Supertags</th>
<th>Perplexity</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>86.5</td>
<td>Kneser-Ney 5-grams</td>
</tr>
<tr>
<td>-</td>
<td>84.9</td>
<td>LBLN [Minh et al, 2009]</td>
</tr>
<tr>
<td>POS $</td>
<td>Z_x</td>
<td>= 5$</td>
</tr>
<tr>
<td>POS $</td>
<td>Z_x</td>
<td>= 50$</td>
</tr>
<tr>
<td>Supertags $</td>
<td>Z_x</td>
<td>= 50$</td>
</tr>
</tbody>
</table>

\textsuperscript{9}Katz, 1987; Goodman & Chen, 1996; Bangalore, 1997; Mirowski et al, SLT 2010
# Results

**Lowest perplexity on AP News benchmark**

AP News corpus
17k vocabulary (no proper nouns, no numbers)

<table>
<thead>
<tr>
<th>#topics</th>
<th>POS</th>
<th>Perplexity</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>123.3</td>
<td>Kneser-Ney 5-grams</td>
</tr>
<tr>
<td>0</td>
<td>-</td>
<td>104.4</td>
<td>our LBLN [Minh et al, 2009]</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>Zx</td>
<td>=40</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>98.5</td>
<td>Our work</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Zx</td>
<td>=40</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>99.0</td>
<td>their LBLN [Mnih et al, 2009]</td>
</tr>
</tbody>
</table>

Results:
Lowest perplexity on AP News benchmark

AP News corpus
17k vocabulary (no proper nouns, no numbers)

Train on 14M words
Validate on 1M words
Test on 1M words

# Results:

**Best word accuracy on speech recognition**

HUB-4 TV broadcast transcripts corpus

25k vocabulary (with proper nouns & numbers)

<table>
<thead>
<tr>
<th>#topics</th>
<th>POS</th>
<th>Word accuracy</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>63.7%</strong></td>
<td>AT&amp;T Watson [Goffin et al, 2005]</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>63.5%</strong></td>
<td>KN 5-grams on 100-best list</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>66.6%</strong></td>
<td>Oracle: best of 100-best list</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>57.8%</strong></td>
<td>Oracle: worst of 100-best list</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td><strong>64.1%</strong></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>(</td>
<td>Z_x</td>
<td>= 34</td>
</tr>
<tr>
<td>0</td>
<td>(</td>
<td>Z_x</td>
<td>= 3</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td><strong>64.2%</strong></td>
<td>This work (re-ranking the 100-best list)</td>
</tr>
<tr>
<td>5</td>
<td>(</td>
<td>Z_x</td>
<td>= 3</td>
</tr>
<tr>
<td>5</td>
<td>(</td>
<td>Z_x</td>
<td>= 3</td>
</tr>
</tbody>
</table>
Acknowledgements
Thank you

• Further references: