Spectral Methods for Natural Language Processing

Karl Stratos

Thesis Defense

Committee

David Blei, Michael Collins, Daniel Hsu, Slav Petrov, and Owen Rambow

Latent-Variable Models in NLP

Models with latent/hidden variables are widely used for unsupervised and semi-supervised NLP tasks.

Some examples:

1. Word clustering (Brown et al., 1992)
2. Syntactic parsing (Matsuzaki et al., 2005; Petrov et al., 2006)
3. Label induction (Haghighi and Klein 2006; Berg-Kirkpatrick et al., 2010)
4. Machine translation (Brown et al., 1993)

Computational Challenge

latent variables $\longrightarrow$ (generally) intractable computation

- Learning HMMs: intractable (Terwijn, 2002)
- Learning topic models: NP-hard (Arora et al., 2012)
- Many other hardness results

Common approach: **EM**, **gradient-based** search (SGD, L-BFGS)

- No global optimality guaranteed!
- Heuristics in this sense

Why Not Heuristics?

Heuristics are often sufficient for empirical purposes.

- EM, SGD, L-BFGS: remarkably successful training methods
- Do have weak guarantees (convergence to a local optimum)
- Ways to deal with local optima issues (careful initialization, random restarts, ...)

"So why not just use heuristics?"

Why Not Heuristics?

Heuristics are often sufficient for empirical purposes.

- EM, SGD, L-BFGS: remarkably successful training methods
- Do have weak guarantees (convergence to a local optimum)
- Ways to deal with local optima issues (careful initialization, random restarts, ...)

"So why not just use heuristics?"

At least two downsides:

1. Impedes the development of new theoretical frameworks

   No new understanding of problems for better solutions
2. Limited guidance of rigorous theory

   Black art tricks, unreliable and difficult to reproduce

This Thesis

Derives algorithms for latent-variable models in NLP with **provable guarantees**.

Main weapon

SPECTRAL METHODS

(i.e., methods that use **singular value decomposition (SVD)** or other similar factorization)

This Thesis

Derives algorithms for latent-variable models in NLP with **provable guarantees**.

<u>Main weapon</u>

SPECTRAL METHODS

(i.e., methods that use **singular value decomposition (SVD)** or other similar factorization)

Stands on the shoulders of many giants:

- Guaranteed learning of GMMs (Dasgupta, 1999)
- Dimensionality reduction with CCA (Kakade and Foster, 2007)
- Guaranteed learning of HMMs (Hsu et al., 2008)
- Guaranteed learning of topic models (Arora et al., 2012)

Main Contributions

Novel spectral algorithms for two NLP tasks

TASK 1. **Learning lexical representations**

(UAI 2014) First provably correct algorithm for clustering words under the language model of Brown et al. ("Brown clustering")

(ACL 2015) New model-based interpretation of smoothed CCA for deriving word embeddings

Main Contributions

Novel spectral algorithms for two NLP tasks

TASK 1. **Learning lexical representations**

(UAI 2014) First provably correct algorithm for clustering words under the language model of Brown et al. ("Brown clustering")

(ACL 2015) New model-based interpretation of smoothed CCA for deriving word embeddings

TASK 2. **Estimating latent-variable models for NLP**

(TACL 2016) Consistent estimator of a model for unsupervised part-of-speech (POS) tagging

(CoNLL 2013) Consistent estimator of a model for supervised phoneme recognition

Overview

Motivation

Brown clustering algorithm (Brown et al., 1992)

- An agglomerative word clustering method
- Popular for semi-supervised NLP (Miller et al., 2004; Koo et al., 2008)

This method assumes an underlying clustering of words, but is not guaranteed to recover the correct clustering.

This work:

- Derives a spectral algorithm with a guarantee of recovering the underlying clustering.
  - Also empirically much faster (up to ~ 10 times)

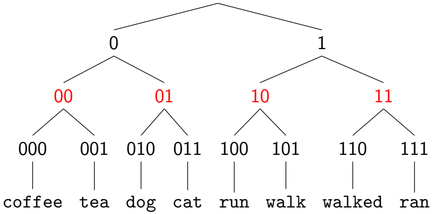
Original Clustering Scheme of Brown et al. (1992)

BrownAlg
Input: sequence of words $x_1 \dots x_N$ in vocabulary $\mathcal{V}$, number of clusters m

1. Initialize each $w \in \mathcal{V}$ to be its own cluster.
2. For $|\mathcal{V}| - 1$ times, merge a pair of clusters that yields the smallest decrease in

$$p\left(x_1 \dots x_N \middle| \textbf{Brown model}\right)$$

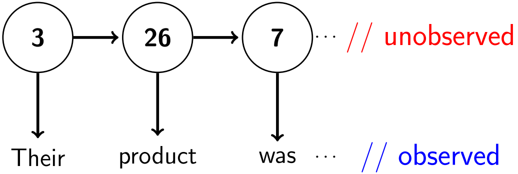
   when merged.
3. Return a pruning of the resulting tree with m leaf clusters.



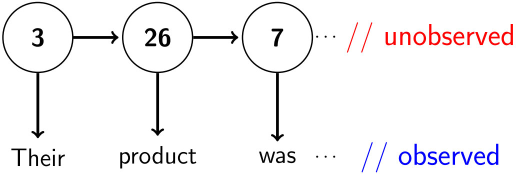
$\boxed{m = 4}$

00	`coffee tea`
01	`dog cat`
10	`walk run`
11	`walked ran`

Brown Model = Restricted HMM



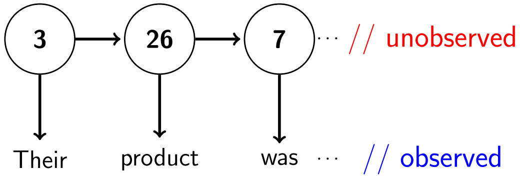
Brown Model = Restricted HMM



- Hidden states: m word classes $\{1 \dots m\}$
- Observed states: n word types $\{1 \dots n\}$
- **Restriction.** Word x belongs to exactly one class $C(x)$.

$$p(x_1 \dots x_N) = \pi_{C(x_1)} \times \prod_{i=2}^{N} T_{C(x_i),C(x_{i-1})} \times \prod_{i=1}^{N} O_{x_i,C(x_i)}$$

Brown Model = Restricted HMM



- Hidden states: m word classes $\{1 \dots m\}$
- Observed states: n word types $\{1 \dots n\}$
- **Restriction.** Word x belongs to exactly one class $C(x)$.

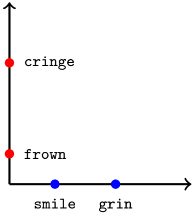
$$p(x_1 \dots x_N) = \pi_{C(x_1)} \times \prod_{i=2}^{N} T_{C(x_i),C(x_{i-1})} \times \prod_{i=1}^{N} O_{x_i,C(x_i)}$$

The model assumes a true class $C(x)$ for each word x. **BrownAlg** is a <u>greedy heuristic</u> with no guarantee of recovering $C(x)$.

Derivation of a Spectral Algorithm

Key observation. Given the emission parameters $O_{x,c}$, we can trivially recover the true clustering (by the model restriction).

$$O = \begin{array}{c c} & \begin{array}{cc} 1 & 2 \end{array} \\ \begin{array}{c} \texttt{smile} \\ \texttt{grin} \\ \texttt{frown} \\ \texttt{cringe} \end{array} & \begin{bmatrix} 0.3 & 0 \\ 0.7 & 0 \\ 0 & 0.2 \\ 0 & 0.8 \end{bmatrix} \end{array}$$



Algorithm: put words x, x' in the same cluster iff

$$\frac{O_x}{||O_x||} = \frac{O_{x'}}{||O_{x'}||}$$

SVD Recovers the Emission Parameters

Theorem. Let $U\Sigma V^\top$ be a rank-m SVD of Ω defined by

$$\Omega_{x,x'} := \frac{p(x,x')}{\sqrt{p(x) \times p(x')}}$$

Then for some orthogonal $Q \in \mathbb{R}^{m \times m}$,

$$U = \sqrt{O}Q^\top$$

Corollary: words x, x' are in the same cluster iff

$$\frac{U_x}{||U_x||} = \frac{U_{x'}}{||U_{x'}||}$$

Clustering with Empirical Estimates

$\widehat{\Omega}$:= empirical estimate of Ω from N samples $x_1 \dots x_N$

$$\widehat{\Omega}_{x,x'} := \frac{\mathsf{count}(x,x')}{\sqrt{\mathsf{count}(x) \times \mathsf{count}(x')}}.$$

$\widehat{U}\widehat{\Sigma}\widehat{V}^\top$:= rank-m SVD of $\widehat{\Omega}$

Clustering with Empirical Estimates

$\widehat{\Omega}$:= empirical estimate of Ω from N samples $x_1 \ldots x_N$

$$\widehat{\Omega}_{x,x'} := \frac{\text{count}(x, x')}{\sqrt{\text{count}(x) \times \text{count}(x')}}.$$

$\widehat{U}\widehat{\Sigma}\widehat{V}^\top$:= rank-m SVD of $\widehat{\Omega}$

The Guarantee. If N is large enough (polynomial in the condition number of Ω), $C(x)$ is given by some m-pruning of an agglomerative clustering of

$$\hat{f}(x) := \widehat{U}_x / \left|\left|\widehat{U}_x\right|\right|$$

Clustering with Empirical Estimates

$\widehat{\Omega}$:= empirical estimate of Ω from N samples $x_1 \ldots x_N$

$$\widehat{\Omega}_{x,x'} := \frac{\text{count}(x, x')}{\sqrt{\text{count}(x) \times \text{count}(x')}}.$$

$\widehat{U}\widehat{\Sigma}\widehat{V}^\top$:= rank-m SVD of $\widehat{\Omega}$

The Guarantee. If N is large enough (polynomial in the condition number of Ω), $C(x)$ is given by some m-pruning of an agglomerative clustering of

$$\hat{f}(x) := \widehat{U}_x / \left|\left|\widehat{U}_x\right|\right|$$

Proof sketch. Large N ensures small $\left|\left|\Omega - \widehat{\Omega}\right|\right|$, which ensures the *strict separation property* for the distance between $\hat{f}(x)$:

$$C(x) = C(x') \neq C(x'') \Longrightarrow \left|\left|\hat{f}(x) - \hat{f}(x')\right|\right| < \left|\left|\hat{f}(x) - \hat{f}(x'')\right|\right|$$

The claim follows from Balcan et al. (2008).

Summary of the Algorithm

- Compute an empirical estimate $\widehat{\Omega}$ from unlabeled text.

$$\widehat{\Omega}_{x,x'} := \frac{\mathsf{count}(x, x')}{\sqrt{\mathsf{count}(x) \times \mathsf{count}(x')}}$$

Summary of the Algorithm

- Compute an empirical estimate $\widehat{\Omega}$ from unlabeled text.

$$\widehat{\Omega}_{x,x'} := \frac{\mathsf{count}(x, x')}{\sqrt{\mathsf{count}(x) \times \mathsf{count}(x')}}$$

- **Compute a rank-m SVD**:

$$\widehat{\Omega} \approx \widehat{U}\widehat{\Sigma}\widehat{V}^\top$$

Summary of the Algorithm

- Compute an empirical estimate $\widehat{\Omega}$ from unlabeled text.

$$\widehat{\Omega}_{x,x'} := \frac{\mathsf{count}(x,x')}{\sqrt{\mathsf{count}(x) \times \mathsf{count}(x')}}$$

- **Compute a rank-m SVD**:

$$\widehat{\Omega} \approx \widehat{U}\widehat{\Sigma}\widehat{V}^{\top}$$

- **Agglomeratively cluster** the normalized rows $\widehat{U}_x / \left|\left|\widehat{U}_x\right|\right|$.

Summary of the Algorithm

- Compute an empirical estimate $\widehat{\Omega}$ from unlabeled text.

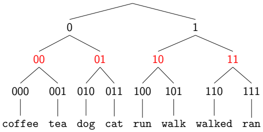
$$\widehat{\Omega}_{x,x'} := \frac{\mathsf{count}(x, x')}{\sqrt{\mathsf{count}(x) \times \mathsf{count}(x')}}$$

- **Compute a rank-m SVD**:

$$\widehat{\Omega} \approx \widehat{U}\widehat{\Sigma}\widehat{V}^\top$$

- **Agglomeratively cluster** the normalized rows $\widehat{U}_x / \left|\left|\widehat{U}_x\right|\right|$.

- Return a pruning of the hierarchy into m leaf clusters.



00	`coffee tea`
01	`dog cat`
10	`walk run`
11	`walked ran`

Experiments: Comparison with Brown et al.

Corpus. RCV1 new articles (205 million words)

- Induced 1000 clusters with both algorithms
- Use them as features in a perceptron-style model for named-entity recognition (NER)

$$\ldots\ \overbrace{\texttt{John Smith}}^{\text{PER}}\ \texttt{works at}\ \overbrace{\texttt{New York Times}}^{\text{ORG}}\ \ldots$$

- NER dataset: CoNLL 2003 shared task

Features	time to induce clusters	dev F1	test F1
—	—	90.03	84.39
Brown	22 hours	92.68	88.76
Spectral	2 hours	92.31	87.76

Overview

Motivation: WORD2VEC as Matrix Decomposition

- WORD2VEC (Mikolov et al., 2013) trains word/context embeddings by maximizing some objective:

$$(v_w, v_c) = \arg\max_{u,v} J(u, v)$$

- Recently cast as a *low-rank decomposition* of *transformed co-occurrence counts* (Levy and Goldberg, 2014):

$$v_w^\top v_c = f(\mathsf{count}(w, c))$$

- **Q. Are there other count transformations whose low-rank decompositions yield effective word embeddings?**

This Work

1. Count transformation under **canonical correlation analysis (CCA)** (Hotelling, 1936)
   - Model-based interpretation
2. Unifies various spectral methods in the literature
3. Empirically competitive with WORD2VEC and GLOVE

Optimization Problem Underlying CCA

Input:

1. $(X, Y) \in \mathbb{R}^d \times \mathbb{R}^{d'}$ // two "views" of an object
2. $m \leq \min(d, d')$ // number of projection vectors

Output: $(a_1, b_1) \ldots (a_m, b_m) \in \mathbb{R}^d \times \mathbb{R}^{d'}$ such that

- (a_1, b_1) is the solution of

$$\arg\max_{a,b} \; \mathrm{Cor}\left(a^\top X,\; b^\top Y\right) \qquad (1)$$

- For $i = 2 \ldots m : (a_i, b_i)$ is the solution of (1) subject to:

$$\mathrm{Cor}\left(a^\top X,\; a_j^\top X\right) = 0 \qquad \forall j < i$$
$$\mathrm{Cor}\left(b^\top Y,\; b_j^\top Y\right) = 0 \qquad \forall j < i$$

Exact Solution via Singular Value Decomposition (SVD)

Theorem. (Hotelling, 1936) Define **correlation matrix** $\Omega \in \mathbb{R}^{d \times d'}$:

$$\Omega := \left(\mathbf{E}[XX^\top] - \mathbf{E}[X]\mathbf{E}[X]^\top\right)^{-1/2} \left(\mathbf{E}[XY^\top] - \mathbf{E}[X]\mathbf{E}[Y]^\top\right) \left(\mathbf{E}[YY^\top] - \mathbf{E}[Y]\mathbf{E}[Y]^\top\right)^{-1/2}$$

Let (u_i, v_i) be the left/right singular vectors of Ω corresponding to the i-th largest singular value. Then

$$a_i = \left(\mathbf{E}[XX^\top] - \mathbf{E}[X]\mathbf{E}[X]^\top\right)^{-1/2} u_i$$

$$b_i = \left(\mathbf{E}[YY^\top] - \mathbf{E}[Y]\mathbf{E}[Y]^\top\right)^{-1/2} v_i$$

Two Views of a Word

Extract samples of $(X, Y) := (\textbf{word}, \textbf{context})$ from a corpus:

$$\texttt{... Whatever } \textbf{our} \ \textbf{souls} \ \textbf{are} \texttt{ made of ...}$$
$$\downarrow$$
$$(\textbf{souls}, \textbf{our}) \quad (\textbf{souls}, \textbf{are})$$

Perform SVD on

$$\hat{\Omega} = \left(\hat{\mathbf{E}}[XX^\top] - \hat{\mathbf{E}}[X]\hat{\mathbf{E}}[X]^\top\right)^{-1/2}$$
$$\left(\hat{\mathbf{E}}[XY^\top] - \hat{\mathbf{E}}[X]\hat{\mathbf{E}}[Y]^\top\right)$$
$$\left(\hat{\mathbf{E}}[YY^\top] - \hat{\mathbf{E}}[Y]\hat{\mathbf{E}}[Y]^\top\right)^{-1/2}$$

Simplified Correlation Matrix

When the number of samples is large,

$$\hat{\Omega} \approx \hat{\mathbf{E}}\left[XX^\top\right]^{-1/2}\ \hat{\mathbf{E}}\left[XY^\top\right]\ \hat{\mathbf{E}}\left[YY^\top\right]^{-1/2}$$

I.e., decompose the following transformed counts!

$$\hat{\Omega}_{w,c} = \frac{\mathsf{count}(w,c)}{\sqrt{\mathsf{count}(w) \times \mathsf{count}(c)}}$$

Previous Work Using CCA for Word Embeddings

- Dhillon et al. (2011, 2012) propose various modifications of CCA, but take the square root of counts,

$$\hat{\Omega}_{w,c} = \frac{\mathsf{count}(w,c)^{1/2}}{\sqrt{\mathsf{count}(w)^{1/2} \times \mathsf{count}(c)^{1/2}}}$$

- The square root was taken for empirical reasons.

- We now provide a **model-based interpretation** that naturally admits this extra transformation.

SVD Still Recovers the Emission Parameters

Theorem. Let $U\Sigma V^\top$ be a rank-m SVD of $\Omega^{\langle a \rangle}$ defined by

$$\Omega_{w,c}^{\langle a \rangle} := \frac{p(w,c)^a}{\sqrt{p(w)^a \times p(c)^a}}$$

(where $a \neq 0$). Then for an orthogonal Q and a positive vector s,

$$U = O^{\langle a/2 \rangle} \mathsf{diag}(s) Q^\top$$

Corollary: normalized rows of U still **cluster-revealing**

- Assuming words generated by the Brown model

Choosing the Value of a

One answer: $a = 1/2$

Why?

- Word counts drawn from a multinomial distribution
- Equivalent to: drawn from independent Poisson distributions (conditioned on the length of the corpus)
- Square-root is a **variance-stabilizing** transformation for Poisson random variables (Bartlett, 1936):

$$X \sim \text{Poisson}(\lambda)$$
$$\text{Var}(X^{1/2}) \approx \mathbf{1/4}$$

Experiments

Corpus: pre-processed English Wikipedia (1.4 billion words)

Comparison with

- GLOVE (Pennington et al., 2014)
- WORD2VEC: CBOW, SGNS (Mikolov et al., 2013)
- Default hyperparameter configurations

Evaluation Tasks

1. **AVG-SIM**: word similarity scores averaged across 3 datasets

w1	w2	human	$\cos(\theta)$
`king`	`queen`	8.58	?
`drink`	`eat`	6.87	?
`professor`	`cucumber`	0.31	?

Evaluation Tasks

1. **AVG-SIM**: word similarity scores averaged across 3 datasets

w1	w2	human	$\cos(\theta)$
`king`	`queen`	8.58	?
`drink`	`eat`	6.87	?
`professor`	`cucumber`	0.31	?

2. **SYN**: accuracy in 8000 syntactic analogies
MIXED: accuracy in 19544 syntactic/semantic analogies
(two datasets provided by Mikolov et al. 2013)

	w1	w2		w3	w4
(syntactic)	`take`	`took`	$\sim$	`sit`	?
("semantic")	`London`	`England`	$\sim$	`Kampala`	?

Effect of Power Transformation in CCA

Different values of a in

$$\hat{\Omega}^{\langle a \rangle}_{w,c} = \frac{\text{count}(w,c)^a}{\sqrt{\text{count}(w)^a \times \text{count}(c)^a}}$$

1000 dimensions

a	AVG-SIM	SYN	MIXED
1	0.572	39.68	57.64
2/3	0.650	60.52	74.00
1/2	**0.690**	**65.14**	**77.70**

Word Similarity and Analogy

- LOG: log transform, no scaling
- PPMI: no transform, PPMI scaling
- CCA: square-root transform, CCA scaling

500 dimensions

Method		AVG-SIM	SYN	MIXED
Spectral	LOG	0.652	59.52	67.27
	PPMI	0.628	43.81	58.38
	CCA	**0.655**	68.38	74.17
Others	GLOVE	0.576	68.30	78.08
	CBOW	0.597	75.79	73.60
	SGNS	0.642	**81.08**	**78.73**

Semi-Supervised Learning

Real-valued extra features for NER (CoNLL 2003 dataset)

30 dimensions

Features	Dev	Test
—	90.04	84.40
BROWN	92.49	88.75
LOG	92.27	88.87
PPMI	92.25	89.27
CCA	**92.88**	**89.28**
GLOVE	91.49	87.16
CBOW	92.44	88.34
SGNS	92.63	88.78

(BROWN: 1000 Brown clusters)

Overview

Motivation

- Goal: induce POS tags

  `John/N has/V a/D light/J bag/N`

- Straightforward approach: learn an HMM with EM
  - Terrible performance (Merialdo, 1994)
  - Model misspecification
  - Suboptimal learning

- **This work**:
  - Introduces a variant of HMM suited for POS tagging.
    - "Anchor" HMM
  - Derives an exact estimation method.
    - Based on NMF (Arora et al., 2012)

Anchor HMM

Relaxation of the Brown et al. disjointedness assumption

Disjointedness: Each word belongs to exactly one state.

$$\Downarrow$$

"Anchor": Each state has at least 1 word that belongs to that state *only*.

h_1	`the`
h_2	`new`
h_3	`on`
h_4	`is`

Bonus: hidden states are lexicalized by anchor words

Learning an Anchor HMM

Define "context" Y and matrix Ω with rows:

$$\Omega_x := \mathbf{E}[Y|X = x]$$

Conditions:

1. Y is independent of X, given the state H of X.
2. Ω has rank m (number of states).

Learning an Anchor HMM

Define "context" Y and matrix Ω with rows:

$$\Omega_x := \mathbf{E}[Y | X = x]$$

Conditions:

1. Y is independent of X, given the state H of X.
2. Ω has rank m (number of states).

One choice of Y: indicator vector of neighboring words

```
the    dog    saw    the    cat
```

Can reduce the dimension as long as $\text{rank}(\Omega) = m$

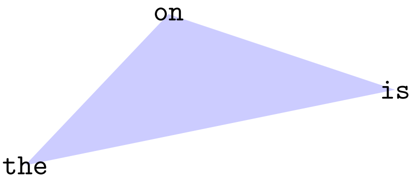
- Random projection, SVD, CCA

Learning an Anchor HMM (Cont.)

Under the conditions, Ω factorizes:

$$\Omega_x = \sum_{h} p(h|x) \times \mathbf{E}[Y|h]$$

where $\Omega_x = \mathbf{E}[Y|h_x]$ if x is an anchor!

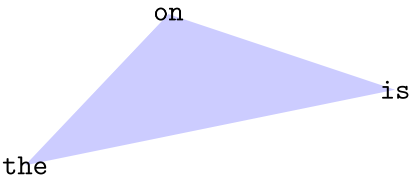


Learning an Anchor HMM (Cont.)

Under the conditions, Ω factorizes:

$$\Omega_x = \sum_h p(h|x) \times \mathbf{E}[Y|h]$$

where $\Omega_x = \mathbf{E}[Y|h_x]$ if x is an anchor!



Algorithm:

1. Find anchor rows (Arora et al., 2012).
2. Estimate convex coefficients $p(h|x)$.
3. Use Bayes' rule to recover emission parameters $o(x|h)$.
4. Given $o(x|h)$, recover $t(h'|h)$ and $\pi(h)$.

Experiments

Dataset. Universal treebank (McDonald et al., 2013)

12 POS tags for 10 languages

Baselines.

- EM: HMM trained with EM
- BROWN: Brown clusters (Brown et al., 1993)
- LOG-LIN: Log-linear model (Berg-Kirkpatrick et al., 2010)

	de	en	es	fr	id	it	ja	ko	pt-br	sv
EM	46	60	61	60	50	52	60	52	60	42
BROWN	60	63	67	66	59	**66**	60	48	**67**	**62**
ANCHOR	63	**71**	**74**	**72**	**67**	60	69	**62**	66	61
LOG-LIN	**68**	62	67	62	61	53	**78**	61	63	57

Discovered Anchor Words (for 12 Tags)

German	English	Spanish	French	Italian	Korean
empfehlen	loss	y	avait	radar	완전
wie	1	hizo	commune	però	중에
;	on	-	Le	sulle	경우
Sein	one	especie	de	-	줄
Berlin	closed	Además	président	Stati	같아요
und	are	el	qui	Lo	많은
,	take	países	(	legge	,
-	,	la	à	al	볼
der	vice	España	États	far-	자신의
im	to	en	Unis	di	받고
des	York	de	Cette	la	맛있는
Region	Japan	municipio	quelques	art.	위한

`loss` ≈ noun   `1` ≈ number   `on` ≈ preposition   . . .

Overview

Refinement HMM for Supervised Phoneme Recognition

Introduces a latent variable for each state.

$$\begin{array}{ccccccccc} \text{ao}^{1} & \rightarrow & \text{ao}^{2} & \rightarrow & \text{ao}^{4} & \rightarrow & \text{ao}^{1} & \rightarrow & \text{ow}^{3} \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\ 15 & & 9 & & 7 & & 900 & & 835 \end{array}$$

$$p(15\ 9\ 7\ 900\ 835,\ \text{ao ao ao ao ow},\ 1\ 2\ 4\ 1\ 3)$$

We derive a **spectral algorithm** for consistently estimating the model parameters without observing the latent states.

- Algorithm: dimensionality reduction with SVD, followed by the method of moments
- Extension of Hsu et al. (2008)

Overview

Summary of Contributions

Novel spectral algorithms for two NLP tasks:

1. **Learning lexical representations.**

   Brown clusters (UAI 2014), word embeddings (ACL 2015)

2. **Estimating latent-variable models.**

   Unsupervised (TACL 2016)/supervised (CoNLL 2013) tagging

Summary of Contributions

Novel spectral algorithms for two NLP tasks:

1. **Learning lexical representations.**
   Brown clusters (UAI 2014), word embeddings (ACL 2015)
2. **Estimating latent-variable models.**
   Unsupervised (TACL 2016)/supervised (CoNLL 2013) tagging

Radically different from previous algorithms

- Central computation: decomposition (SVD and NMF)
- **Guarantees** about the consistency of estimates

Summary of Contributions

Novel spectral algorithms for two NLP tasks:

1. **Learning lexical representations.**

   Brown clusters (UAI 2014), word embeddings (ACL 2015)

2. **Estimating latent-variable models.**

   Unsupervised (TACL 2016)/supervised (CoNLL 2013) tagging

Radically different from previous algorithms

- Central computation: decomposition (SVD and NMF)
- **Guarantees** about the consistency of estimates

Conclusion Spectral methods are viable and effective for NLP

- New understanding of problems
- Scalable and often competitive with the state-of-the-art

Limitations of (Current) Spectral Learning Framework

- "Rigid": specific forms of objective/model
  - Squared-error minimization, trace maximization
  - Relatively simple models (e.g., HMMs, topic models)
- Limited applicability compared to EM, backprop
- Ongoing progress
  - Moments + likelihood (Chaganty and Liang, 2014)
  - More general non-convex objectives (Janzamin et al., 2015)

Future Directions

- Flexible spectral framework

  Ex. Manifold optimization

- Online/randomized spectral methods

  Ex. SVD (Halko et al., 2011), CCA (Ma et al., 2015), matrix sketching (Edo, 2013)

- Incorporate more nonlinearity

  Ex. Deep CCA (Andrew et al., 2013)

- Other NLP applications

  Ex. More word clustering, deciperment, generalized CCA for multi-lingual tasks

Future Directions

- Flexible spectral framework

  Ex. Manifold optimization

- Online/randomized spectral methods

  Ex. SVD (Halko et al., 2011), CCA (Ma et al., 2015), matrix sketching (Edo, 2013)

- Incorporate more nonlinearity

  Ex. Deep CCA (Andrew et al., 2013)

- Other NLP applications

  Ex. More word clustering, deciperment, generalized CCA for multi-lingual tasks

THANK YΩU! QUESTIΩNS?

EXTRA SLIDES

Proof of Spectral Learning of O

What is $\mathbf{E}[\widehat{\Omega}]$?

$$\mathbf{E}[\widehat{\Omega}] = \mathsf{diag}(O\pi)^{-1/2} O \mathsf{diag}(\pi)(OT)^{\top} \mathsf{diag}(OT\pi)^{-1/2}$$

Proof of Spectral Learning of O

What is $\mathbf{E}[\widehat{\Omega}]$?

$$\begin{aligned}\mathbf{E}[\widehat{\Omega}] &= \mathrm{diag}(O\pi)^{-1/2} O \mathrm{diag}(\pi)(OT)^\top \mathrm{diag}(OT\pi)^{-1/2} \\ &= \underbrace{\color{red}{\mathrm{diag}(O\pi)^{-1/2} O \mathrm{diag}(\pi)^{1/2}}}_{\color{red}{A}} \underbrace{\color{blue}{\ldots\ldots\ldots}}_{\substack{\color{blue}{\Theta^\top} \\ \text{(some rank-}m\text{ matrix)}}}\end{aligned}$$

Proof of Spectral Learning of O

What is $\mathbf{E}[\widehat{\Omega}]$?

$$\begin{aligned}\mathbf{E}[\widehat{\Omega}] &= \mathrm{diag}(O\pi)^{-1/2} O \mathrm{diag}(\pi)(OT)^\top \mathrm{diag}(OT\pi)^{-1/2} \\ &= \underbrace{\mathrm{diag}(O\pi)^{-1/2} O \mathrm{diag}(\pi)^{1/2}}_{A} \underbrace{\cdots\cdots\cdots}_{\Theta^\top}\end{aligned}$$

(some rank-m matrix)

What is A?

Proof of Spectral Learning of O

What is $\mathbf{E}[\widehat{\Omega}]$?

$$\begin{aligned}\mathbf{E}[\widehat{\Omega}] &= \text{diag}(O\pi)^{-1/2} O \text{diag}(\pi) (OT)^{\top} \text{diag}(OT\pi)^{-1/2} \\ &= \underbrace{\color{red}{\text{diag}(O\pi)^{-1/2} O \text{diag}(\pi)^{1/2}}}_{\color{red}{A}} \underbrace{\color{blue}{\ldots\ldots\ldots}}_{\color{blue}{\Theta^{\top}}}\end{aligned}$$

(some rank-m matrix)

What is $\color{red}{A}$?

$$\color{red}{A}_{x,h} = \frac{O_{x,h}\sqrt{\pi_h}}{\sqrt{\sum_h O_{x,h}\pi_h}}$$

Proof of Spectral Learning of O

What is $\mathbf{E}[\widehat{\Omega}]$?

$$\begin{aligned}\mathbf{E}[\widehat{\Omega}] &= \mathrm{diag}(O\pi)^{-1/2} O \mathrm{diag}(\pi)(OT)^\top \mathrm{diag}(OT\pi)^{-1/2} \\ &= \underbrace{\mathrm{diag}(O\pi)^{-1/2} O \mathrm{diag}(\pi)^{1/2}}_{A} \underbrace{\cdots\cdots\cdots}_{\Theta^\top}\end{aligned}$$

(some rank-m matrix)

What is A?

$$A_{x,h} = \frac{O_{x,h}\sqrt{\pi_h}}{\sqrt{\sum_h O_{x,h}\pi_h}} = \frac{O_{x,h}\sqrt{\pi_h}}{\sqrt{O_{x,C(x)}\pi_{C(x)}}}$$

Proof of Spectral Learning of O

What is $\mathbf{E}[\widehat{\Omega}]$?

$$
\begin{aligned}
\mathbf{E}[\widehat{\Omega}] &= \mathrm{diag}(O\pi)^{-1/2} O \mathrm{diag}(\pi)(OT)^\top \mathrm{diag}(OT\pi)^{-1/2} \\
&= \underbrace{\mathrm{diag}(O\pi)^{-1/2} O \mathrm{diag}(\pi)^{1/2}}_{A} \underbrace{\cdots\cdots\cdots}_{\Theta^\top}
\end{aligned}
$$

(some rank-m matrix)

What is A?

$$
A_{x,h} = \frac{O_{x,h}\sqrt{\pi_h}}{\sqrt{\sum_h O_{x,h}\pi_h}} = \frac{O_{x,h}\sqrt{\pi_h}}{\sqrt{O_{x,C(x)}\pi_{C(x)}}} = \sqrt{O_{x,C(x)}}
$$

Proof of Spectral Learning of O

What is $\mathbf{E}[\widehat{\Omega}]$?

$$\mathbf{E}[\widehat{\Omega}] = \mathrm{diag}(O\pi)^{-1/2} O \mathrm{diag}(\pi)(OT)^\top \mathrm{diag}(OT\pi)^{-1/2}$$
$$= \underbrace{\mathrm{diag}(O\pi)^{-1/2} O \mathrm{diag}(\pi)^{1/2}}_{A} \underbrace{\cdots\cdots\cdots}_{\Theta^\top}$$

(some rank-m matrix)

What is A?

$$A_{x,h} = \frac{O_{x,h}\sqrt{\pi_h}}{\sqrt{\sum_h O_{x,h}\pi_h}} = \frac{O_{x,h}\sqrt{\pi_h}}{\sqrt{O_{x,C(x)}\pi_{C(x)}}} = \sqrt{O_{x,C(x)}}$$

1. A has the **same sparsity pattern** as O.

Proof of Spectral Learning of O

What is $\mathbf{E}[\widehat{\Omega}]$?

$$\begin{aligned}\mathbf{E}[\widehat{\Omega}] &= \text{diag}(O\pi)^{-1/2} O \text{diag}(\pi)(OT)^\top \text{diag}(OT\pi)^{-1/2} \\ &= \underbrace{\text{diag}(O\pi)^{-1/2} O \text{diag}(\pi)^{1/2}}_{A} \underbrace{\cdots\cdots\cdots}_{\Theta^\top}\end{aligned}$$

(some rank-m matrix)

What is A?

$$A_{x,h} = \frac{O_{x,h}\sqrt{\pi_h}}{\sqrt{\sum_h O_{x,h}\pi_h}} = \frac{O_{x,h}\sqrt{\pi_h}}{\sqrt{O_{x,C(x)}\pi_{C(x)}}} = \sqrt{O_{x,C(x)}}$$

1. A has the **same sparsity pattern** as O.
2. A has **orthogonal columns**: $A^\top A = I_{m \times m}$.

Proof of Spectral Learning of O (Cont.)

If $U \in \mathbb{R}^{n \times m}$ is the top m left singular vectors of $\mathbf{E}[\widehat{\Omega}]$,

$$UU^\top = \mathbf{E}[\widehat{\Omega}](\mathbf{E}[\widehat{\Omega}]^\top \mathbf{E}[\widehat{\Omega}])^+ \mathbf{E}[\widehat{\Omega}]^\top$$

Proof of Spectral Learning of O (Cont.)

If $U \in \mathbb{R}^{n\times m}$ is the top m left singular vectors of $\mathbf{E}[\widehat{\Omega}]$,

$$\begin{aligned} UU^\top &= \mathbf{E}[\widehat{\Omega}](\mathbf{E}[\widehat{\Omega}]^\top \mathbf{E}[\widehat{\Omega}])^+ \mathbf{E}[\widehat{\Omega}]^\top \\ &= A\Theta^\top(\Theta A^\top A \Theta^\top)^+ \Theta A^\top \end{aligned}$$

Proof of Spectral Learning of O (Cont.)

If $U \in \mathbb{R}^{n \times m}$ is the top m left singular vectors of $\mathbf{E}[\widehat{\Omega}]$,

$$\begin{aligned} UU^\top &= \mathbf{E}[\widehat{\Omega}](\mathbf{E}[\widehat{\Omega}]^\top \mathbf{E}[\widehat{\Omega}])^+ \mathbf{E}[\widehat{\Omega}]^\top \\ &= A\Theta^\top(\Theta A^\top A\Theta^\top)^+\Theta A^\top \\ &= A\Theta^\top(\Theta\Theta^\top)^+\Theta A^\top \end{aligned}$$

Proof of Spectral Learning of O (Cont.)

If $U \in \mathbb{R}^{n \times m}$ is the top m left singular vectors of $\mathbf{E}[\widehat{\Omega}]$,

$$\begin{aligned} UU^\top &= \mathbf{E}[\widehat{\Omega}](\mathbf{E}[\widehat{\Omega}]^\top \mathbf{E}[\widehat{\Omega}])^+ \mathbf{E}[\widehat{\Omega}]^\top \\ &= A\Theta^\top(\Theta A^\top A \Theta^\top)^+ \Theta A^\top \\ &= A\Theta^\top(\Theta\Theta^\top)^+ \Theta A^\top \\ &= AA^\top \end{aligned}$$

Proof of Spectral Learning of O (Cont.)

If $U \in \mathbb{R}^{n\times m}$ is the top m left singular vectors of $\mathbf{E}[\widehat{\Omega}]$,

$$\begin{aligned} UU^\top &= \mathbf{E}[\widehat{\Omega}](\mathbf{E}[\widehat{\Omega}]^\top \mathbf{E}[\widehat{\Omega}])^+ \mathbf{E}[\widehat{\Omega}]^\top \\ &= A\Theta^\top(\Theta A^\top A\Theta^\top)^+\Theta A^\top \\ &= A\Theta^\top(\Theta\Theta^\top)^+\Theta A^\top \\ &= AA^\top \end{aligned}$$

$\Theta^\top(\Theta\Theta^\top)^+\Theta = I_{m\times m}$ since range$(\Theta) = \mathbb{R}^m$

Proof of Spectral Learning of O (Cont.)

If $U \in \mathbb{R}^{n\times m}$ is the top m left singular vectors of $\mathbf{E}[\widehat{\Omega}]$,

$$\begin{aligned} UU^\top &= \mathbf{E}[\widehat{\Omega}](\mathbf{E}[\widehat{\Omega}]^\top \mathbf{E}[\widehat{\Omega}])^+ \mathbf{E}[\widehat{\Omega}]^\top \\ &= A\Theta^\top(\Theta A^\top A\Theta^\top)^+\Theta A^\top \\ &= A\Theta^\top(\Theta\Theta^\top)^+\Theta A^\top \\ &= AA^\top \end{aligned}$$

$\Theta^\top(\Theta\Theta^\top)^+\Theta = I_{m\times m}$ since range$(\Theta) = \mathbb{R}^m$

So $UU^\top = AA^\top$, i.e., $\exists$ orthogonal $Q \in \mathbb{R}^{m\times m}$ such that

$$U = AQ^\top = \sqrt{O}Q^\top$$

$\square$

Variance Stabilization

A heuristic "proof": if $X \sim \text{Poisson}(\lambda)$ and

$$g(X) := \sqrt{X}$$

By the delta method:

$$\begin{aligned}
\text{Var}(g(X)) &\approx g'(\mathbf{E}[X])^2 \,\text{Var}(X) \\
&= \left(\frac{1}{2\sqrt{\lambda}}\right)^2 \lambda \\
&= \frac{1}{4}
\end{aligned}$$

Fast Agglomerative Clustering

Input: $\mu^{(1)} \dots \mu^{(n)} \in \mathbb{R}^d$ word vectors sorted in decreasing frequency, integer $m \leq n$
Output: hierarchical clustering of $\mu^{(1)} \dots \mu^{(n)}$
Tightening: $O(dm)$ subroutine tighten(c):

$$\text{nearest}(c) := \operatorname*{arg\,min}_{c' \in \mathcal{C}: c' \neq c} \triangle(c, c') \qquad \text{lb}(c) := \min_{c' \in \mathcal{C}: c' \neq c} \triangle(c, c') \qquad \text{tight}(c) := \texttt{True}$$

Main body:

1. $\mathcal{C} \leftarrow \{\{\mu^{(1)}\}, \dots, \{\mu^{(m)}\}\}$, call tighten$(c)$ for each $c \in \mathcal{C}$.
2. For $i = m + 1$ to $n + m - 1$:
   - 2.1 If $i \leq n$: let $c := \{\mu^{(i)}\}$, call tighten(c), and let $\mathcal{C} := \mathcal{C} \cup \{c\}$.
   - 2.2 Let $c^* := \arg\min_{c \in \mathcal{C}} \text{lb}(c)$.
   - 2.3 While tight(c^*) is False, call tighten(c^*) and let $c^* := \arg\min_{c \in \mathcal{C}}$
   - 2.4 Merge c^* and nearest(c^*) in $\mathcal{C}$.
   - 2.5 For each $c \in \mathcal{C}$: if nearest$(c) \in \{c^*, \text{nearest}(c^*)\}$, set tight$(c) := \texttt{False}$.

Instead of $O(dn^2m)$ (already using the fixed window trick), we have $O(dm^2 + \gamma dnm) = O(\gamma dnm)$ where empirically $\gamma \ll n$

Why the Brown Clustering Algorithm is Slow

$$
\begin{aligned}
&\texttt{computeL2usingOld}(s,t,u,v,w) = \texttt{L2}[v][w] \\
&- \texttt{q2}[v][s] - \texttt{q2}[s][v] - \texttt{q2}[w][s] - \texttt{q2}[s][w] \\
&- \texttt{q2}[v][t] - \texttt{q2}[t][v] - \texttt{q2}[w][t] - \texttt{q2}[t][w] \\
&+ (\texttt{p2}[v][s] + \texttt{p2}[w][s]) * \log((\texttt{p2}[v][s] + \texttt{p2}[w][s])/((\texttt{p1}[v] + \texttt{p1}[w]) * \texttt{p1}[s])) \\
&+ (\texttt{p2}[s][v] + \texttt{p2}[s][w]) * \log((\texttt{p2}[s][v] + \texttt{p2}[s][w])/((\texttt{p1}[v] + \texttt{p1}[w]) * \texttt{p1}[s])) \\
&+ (\texttt{p2}[v][t] + \texttt{p2}[w][t]) * \log((\texttt{p2}[v][t] + \texttt{p2}[w][t])/((\texttt{p1}[v] + \texttt{p1}[w]) * \texttt{p1}[t])) \\
&+ (\texttt{p2}[t][v] + \texttt{p2}[t][w]) * \log((\texttt{p2}[t][v] + \texttt{p2}[t][w])/((\texttt{p1}[v] + \texttt{p1}[w]) * \texttt{p1}[t])) \\
&+ \texttt{q2}[v][u] + \texttt{q2}[u][v] + \texttt{q2}[w][u] + \texttt{q2}[u][w] \\
&- (\texttt{p2}[v][u] + \texttt{p2}[w][u]) * \log((\texttt{p2}[v][u] + \texttt{p2}[w][u])/((\texttt{p1}[v] + \texttt{p1}[w]) * \texttt{p1}[u])) \\
&- (\texttt{p2}[u][v] + \texttt{p2}[u][w]) * \log((\texttt{p2}[u][v] + \texttt{p2}[u][w])/((\texttt{p1}[v] + \texttt{p1}[w]) * \texttt{p1}[u]))
\end{aligned}
$$

A $O(1)$ function that is called $O(nm^2)$ times in Liang's implementation of the Brown algorithm, accounting for over 40% of the runtime.

Template

Input: count(w, c), dimension m, transform t, scaling s

- count$(w) := \sum_c$ count(w, c)
- count$(c) := \sum_w$ count(w, c)

Output: embedding $v(w) \in \mathbb{R}^m$ for each word w

---

1. Transform counts

2. Scale counts to construct matrix $\hat{\Omega}$

3. Do **rank**-m **SVD** on $\hat{\Omega} \approx \hat{U}\hat{\Sigma}\hat{V}^\top$ and let $v(w) = \hat{U}_w / \left|\left|\hat{U}_w\right|\right|$

Template: No Scaling (Pennington et al., 2014)

Input: count(w, c), dimension m, $t =$ log, $s =$ —

- count$(w) := \sum_c$ count(w, c)
- count$(c) := \sum_w$ count(w, c)

Output: embedding $v(w) \in \mathbb{R}^m$ for each word w

---

1. Transform counts

   count$(w, c) \leftarrow \log(1 + \text{count}(w, c))$

2. Scale counts to construct matrix $\hat{\Omega}$

$$\hat{\Omega}_{w,c} = \text{count}(w, c)$$

3. Do **rank-m SVD** on $\hat{\Omega} \approx \hat{U}\hat{\Sigma}\hat{V}^\top$ and let $v(w) = \hat{U}_w / \left|\left|\hat{U}_w\right|\right|$

Template: PPMI (Levy and Goldberg, 2014)

Input: count(w, c), dimension m, $t =$ —, $s =$ ppmi

- count$(w) := \sum_c$ count(w, c)
- count$(c) := \sum_w$ count(w, c)

Output: embedding $v(w) \in \mathbb{R}^m$ for each word w

---

1. Transform counts

$$\text{count}(w,c) \leftarrow \text{count}(w,c) \qquad \text{count}(w) \leftarrow \text{count}(w)$$
$$\text{count}(c) \leftarrow \text{count}(c)$$

2. Scale counts to construct matrix $\hat{\Omega}$

$$\hat{\Omega}_{w,c} = \max\left(0, \log \frac{\text{count}(w,c) \times \sum_{w,c} \text{count}(w,c)}{\text{count}(w) \times \text{count}(c)}\right)$$

3. Do **rank-m SVD** on $\hat{\Omega} \approx \hat{U}\hat{\Sigma}\hat{V}^\top$ and let $v(w) = \hat{U}_w / \left|\left|\hat{U}_w\right|\right|$

Template: CCA with Square-Root (this work)

Input: count(w, c), dimension m, $t =$ sqrt, $s =$ cca

- count$(w) := \sum_c$ count(w, c)
- count$(c) := \sum_w$ count(w, c)

Output: embedding $v(w) \in \mathbb{R}^m$ for each word w

---

1. Transform counts

$$\text{count}(w,c) \leftarrow \sqrt{\text{count}(w,c)} \qquad \text{count}(w) \leftarrow \sqrt{\text{count}(w)}$$

$$\text{count}(c) \leftarrow \sqrt{\text{count}(c)}$$

2. Scale counts to construct matrix $\hat{\Omega}$

$$\hat{\Omega}_{w,c} = \frac{\text{count}(w,c)}{\sqrt{\text{count}(w) \times \text{count}(c)}}$$

3. Do **rank-m SVD** on $\hat{\Omega} \approx \hat{U}\hat{\Sigma}\hat{V}^\top$ and let $v(w) = \hat{U}_w / \left|\left|\hat{U}_w\right|\right|$

Some Nearest Neighbor Examples

rochester	seattle	yahoo	starbucks	lol
binghamton	tacoma	linkedin	dunkin	yeah
albany	portland	msn	mcdonalds	heh
hartford	washington	facebook	mcdonald's	kidding
utica	denver	digg	domino's	thats
syracuse	oakland	aol	applebee's	damn
elmira	baltimore	google	7-eleven	ahh
bridgeport	chicago	friendster	kfc	gosh
newark	cleveland	orkut	walmart	kinda

smile	frown	1	1945	second
smiles	frowns	2	1944	third
smiling	frowned	3	1943	fourth
grin	disapprove	4	1942	fifth
wide-eyed	cringe	5	1941	first
laugh	discourages	6	1946	sixth
cheerful	overreact	8	1940	seventh
eyes	detest	7	1939	eighth
grinning	forbid	9	1947	ninth