Word class learning

Computational and cognitive aspects

Grzegorz Chrupała Afra Alishahi
Yevgen Matusevych

CLIN 2012

Word classes

go come fit try hang read say take see blow
bricks bits food things medicine cream
the your that this a my his some

Berlin Bangkok Tokyo Warsaw
Sarkozy Merkel Obama Berlusconi
Mr Ms President Dr

- Groups of words sharing syntax/semantics
- Useful for generalization and abstraction

Perspectives on class learning

- NLP
  - Efficiency
  - Performance on NLP tasks
- Cognitive modeling
  - Plausible cognitive constraints
  - Performance on simulations of human tasks

Goals

- Bring two perspectives closer together
- Analyze and improve 2 algorithms
  - ΔH - simulate online learning of word classes by humans (Chrupała and Alishahi 2010)
  - Word class LDA - efficiently learn soft word classes for NLP (Chrupała 2011)

Brief comparison

	ΔH	cLDA
Token level	✓	✓
Soft classes	✓	✓
Bayesian	✗	✓
Online	✓	✗
Parameters	✗	✓
Adaptive K	✓	✗
Fast	✗	✓

ΔH

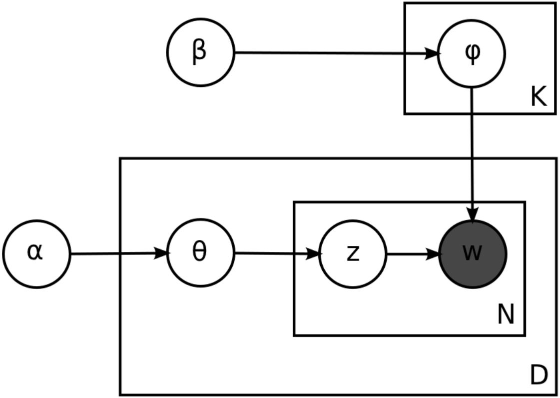
- Incrementally optimizes a joint entropy criterion:

$$H(X, Y) = H(X|Y) + H(Y)$$

    - Small class entropy - parsimony
    - Small conditional feature entropy - informativeness
- New classes are created as needed
- No free parameters

Word class LDA

- Generative model equivalent to LDA for topic models



Word class LDA

- Number of classes K is specified as a parameter
- α and β control sparsity of priors
- Inference using Gibbs sampler (batch)

Model evaluation

Evaluate

- **Parameterized** $\triangle$H
- **Online** Gibbs sampler for word class LDA

on the **same task** and the **same dataset**.

Dataset

- Manchester portion of CHILDES (mothers)
- Discard one-word sentences and punctuation

Data Set	Sessions	#Sent	#Words
Training	26–28	22,491	125,339
Development	29–30	15,193	85,361

Task: word prediction

- Relevant for cognitive modeling
- Used in NLP – language model evaluation

Word prediction

- (Soft)-assign classes from context
- Rank words based on predicted class

Reciprocal rank

want␣to | put | them␣on

Word prediction

- (Soft)-assign classes from context
- Rank words based on predicted class

Reciprocal rank

			y_{123}	make	
				take	
want␣to	put	them␣on		put	$rank^{-1} = \frac{1}{3}$
	y_{123}			get	
				sit	
				eat	
				let	

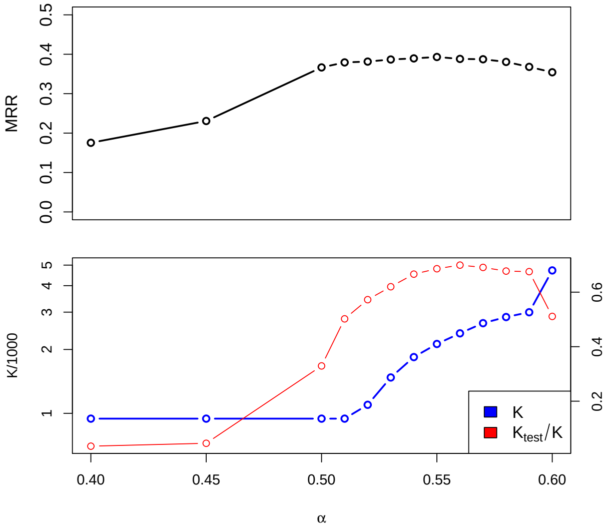
Parametrizing ΔH

- No free parameters in ΔH
  - ✓ No need to optimize them separately
  - ✗ Lack of flexibility
- If we force parametrization
  - Is the algorithm well-behaved?
  - Can we smoothly control the trade-off?

Parametrized ΔH

$$H_\alpha(X,Y) = \alpha H(X|Y) + (1-\alpha)H(Y)$$

Results



MRR
0.0
0.1
0.2
0.3
0.4
0.5
K/1000
1
2
3
4
5
0.2
0.4
0.6
K
K_{test}/K
0.40
0.45
0.50
0.55
0.60
α

Interpretation

- K increases with α
- Word prediction performance changes smoothly with α
- Values of α slightly > 0.5
  - Give best MRR
  - Best ratio of K_{test}/K
- Some degree of trade-off tuning possible α
- Parameterless ΔH close to optimal

Running word class LDA online

- Common LDA inference algorithm: Batch collapsed Gibbs sampler
- Online extensions compared by Canini et al 2005 for topic modeling
- Only one, oLDA, strictly online
- oLDA did not work very well for inferring document topic

Word classes with online LDA (CoLaDA)

- d - word type
- w - context feature
- z - class
- Replicate incoming sentence j times
  - For each w_i in the sentence, sample:

$$P(z_i|\mathbf{z}_{i-1}, \mathbf{w}_i, \mathbf{d}_i) \propto \frac{(n_{z,d} + \alpha) \times (n_{z,w} + \beta)}{n_{z,\bullet} + V\beta}$$

and update the counts.

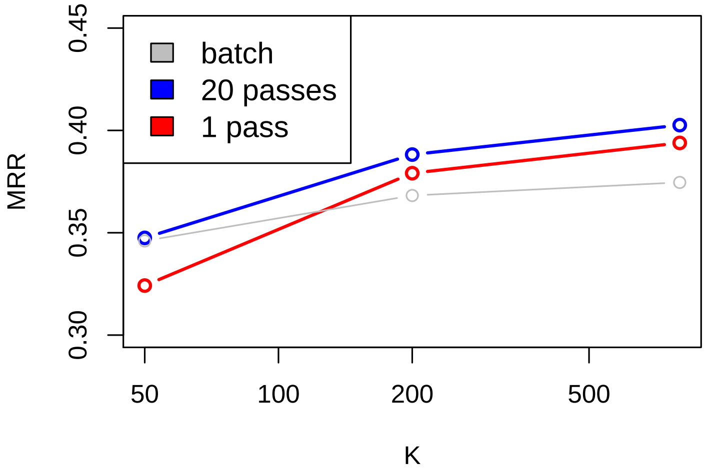
CoLaDA

- oLDA did not work for inferring topics
- Key difference: word types d recur

CoLaDA

- oLDA did not work for inferring topics
- Key difference: word types d recur
  - Classes for common word types will be **frequently resampled**
  - **Without any special arrangements**

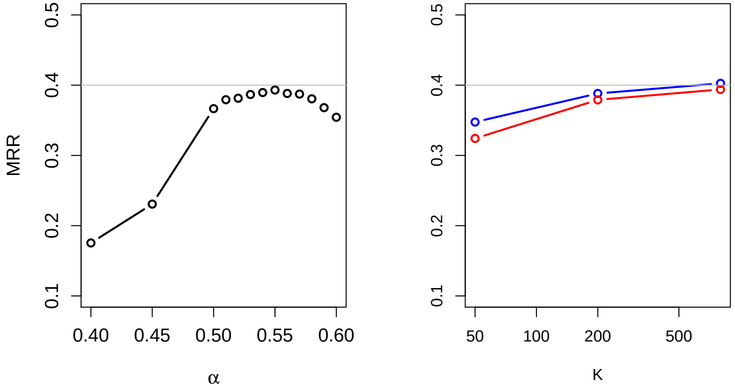
CoLaDA results



CoLaDA discussion

- Word prediction for $K \in \{200, 800\}$ similar to ΔH
- Multiple passes help a bit
- Best parameters
  - 1 pass: $K\alpha = 0.1$, $\beta = 0.01$
  - 20 passes: $K\alpha = 10$, $\beta = 0.1$
- Clusters don't always "look" as coherent as with batch LDA

ΔH vs CoLaDA



Conclusion

Look at models from complementary perspectives:

- Make the cognitive model more flexible
  - Learn more about it
  - Make it tweakable
- Impose cognitive plausibility on practical model
  - Improve memory efficiency
  - Learn from data streams

Future

- Nonparametric version of CoLaDA
  - Adaptive K
- Other tasks, including large-scale NLP
  - Speed up (especially ΔH)

Thank you

Word prediction: variants

- $\Delta H_{\max}$

$$P(w|h) = P(w|\operatorname*{argmax}_i R(y_i|h)^{-1})$$

- ΔH_{Σ}

$$P(w|h) = \sum_{i=1}^{N} P(w|y_i) \frac{\mathrm{R}(y_i|h)^{-1}}{\sum_{i=1}^{N} \mathrm{R}(y_i|h)^{-1}}$$