Improving Topic Models with Latent Feature Word Representations

Mark Johnson

Joint work with
Dat Quoc Nguyen, Richard Billingsley and Lan Du

Dept of Computing
Macquarie University
Sydney
Australia

July 2015

Outline

Introduction

Latent-feature topic models

Experimental evaluation

Conclusions and future work

High-level overview

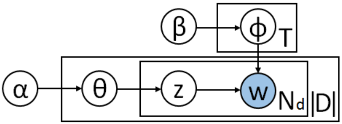
- *Topic models* take a corpus of documents as input, and jointly cluster:
    - *words* by the documents that they occur in, and
    - *documents* by the words that they contain
- If the corpus is small and/or the documents are short, these clusters will be noisy
- *Latent feature representations* of words learnt from large external corpora (e.g., word2vec, Glove) capture various aspects of word meanings
- Here we use latent feature representations learnt on a large external corpus to improve the topic-word distributions in a topic model
    - we combine the Dirichlet-Multinomial models of Latent Dirichlet Allocation (LDA) with the distributed representations used in neural networks
    - the improvement is greatest on small corpora with short documents, e.g., Twitter data

Related work

- Phan et al. (2011) assumed that the small corpus is a sample of topics from a larger corpus like Wikipedia, and use the topics discovered in the larger corpus to help shape the topic representations in the small corpus
  - if the larger corpus has many irrelevant topics, this will “use up” the topic space of the model
- Petterson et al. (2010) proposed an extension of LDA that uses external information about word similarity, such as thesauri and dictionaries, to smooth the topic-to-word distribution
- Sahami and Heilman (2006) employed web search results to improve the information in short texts
- *Neural network topic models* of a single corpus have also been proposed (Salakhutdinov and Hinton, 2009; Srivastava et al., 2013; Cao et al., 2015).

Latent Dirichlet Allocation (LDA)

$$\boldsymbol{\theta}_d \sim \mathrm{Dir}(\alpha) \qquad z_{d_i} \sim \mathrm{Cat}(\boldsymbol{\theta}_d)$$
$$\boldsymbol{\phi}_z \sim \mathrm{Dir}(\beta) \qquad w_{d_i} \sim \mathrm{Cat}(\boldsymbol{\phi}_{z_{d_i}})$$



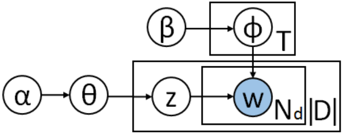
- Latent Dirichlet Allocation (LDA) is an *admixture model*, i.e., each document d is associated with a *distribution over topics* θ_d
- Inference is typically performed with a *Gibbs sampler* over the $z_{d,i}$, integrating out $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ (Griffiths et al., 2004)

$$\mathrm{P}(z_{d_i}{=}t \mid \boldsymbol{Z}_{\neg d_i}) \quad \propto \quad (N^t_{d_{\neg i}} + \alpha)\frac{N^{t,w_{d_i}}_{\neg d_i} + \beta}{N^t_{\neg d_i} + V\beta}$$

The Dirichlet Multinomial Mixture (DMM) model

$\boldsymbol{\theta} \sim \mathrm{Dir}(\alpha)$ $\quad z_d \sim \mathrm{Cat}(\boldsymbol{\theta})$

$\boldsymbol{\phi}_z \sim \mathrm{Dir}(\beta)$ $\quad w_{d_i} \sim \mathrm{Cat}(\boldsymbol{\phi}_{z_d})$



- The Dirichlet Multinomial Mixture (DMM) model is a *mixture model*, i.e., each document d is associated with a single topic z_d (Nigam et al., 2000)
- Inference can also be performed using a collapsed Gibbs sampler in which θ and ϕ_z are integrated out (Yin and Wang, 2014)

$$\mathrm{P}(z_d = t \mid \mathbf{Z}_{\neg d}) \quad \propto \quad (M^t_{\neg d} + \alpha) \frac{\Gamma(N^t_{\neg d} + V\beta)}{\Gamma(N^t_{\neg d} + N_d + V\beta)} \prod_{w \in W} \frac{\Gamma(N^{t,w}_{\neg d} + N^w_d + \beta)}{\Gamma(N^{t,w}_{\neg d} + \beta)}$$

Latent feature word representations

- Traditional count-based methods (Deerwester et al., 1990; Lund and Burgess, 1996; Bullinaria and Levy, 2007) for learning real-valued latent feature (LF) vectors rely on co-occurrence counts
- Recent approaches based on deep neural networks learn vectors by predicting words given their window-based context (Collobert and Weston, 2008; Mikolov et al., 2013; Pennington et al., 2014; Liu et al., 2015)
- We downloaded the pre-trained vectors for word2vec and Glove for this paper

Outline

Introduction

Latent-feature topic models

Experimental evaluation

Conclusions and future work

Latent-feature topic-to-word distributions

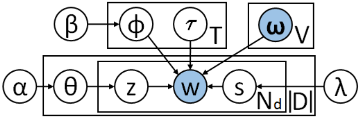
- We assume that each word w is associated with a *word vector* $\boldsymbol{\omega}_w$
- We learn a *topic vector* $\boldsymbol{\tau}_t$ for each topic t
- We use these to define a distribution $\mathrm{CatE}(w)$ over words:

$$\mathrm{CatE}(w \mid \boldsymbol{\tau}_t \boldsymbol{\omega}^\top) \quad \propto \quad \exp(\boldsymbol{\tau}_t \cdot \boldsymbol{\omega}_w)$$

  - $\boldsymbol{\tau}_t \boldsymbol{\omega}^\top$ is a vector of unnormalised scores, one per word
- In our topic models, we *mix the* CatE *distribution* with a multinomial distribution over words, so we can capture ideosyncratic properties of the corpus (e.g., words not seen in the external corpus)
  - we use a Boolean *indicator variable* that records whether a word is generated from CatE or the multinomial distribution

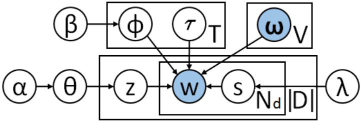
The Latent Feature LDA model

$$\boldsymbol{\theta}_d \sim \mathrm{Dir}(\alpha) \qquad z_{d_i} \sim \mathrm{Cat}(\boldsymbol{\theta}_d)$$
$$\boldsymbol{\phi}_z \sim \mathrm{Dir}(\beta) \qquad s_{d_i} \sim \mathrm{Ber}(\lambda)$$
$$w_{d_i} \sim (1 - s_{d_i})\mathrm{Cat}(\boldsymbol{\phi}_{z_{d_i}}) + s_{d_i}\mathrm{CatE}(\boldsymbol{\tau}_{z_{d_i}}\,\boldsymbol{\omega}^\top)$$



- s_{d_i} is the Boolean indicator variable indicating whether word d_i is generated from CatE
- λ is a user-specified hyper-parameter determining how often words are generated from the CatE distribution
  - if we estimated λ from data, we expect it would never generate through CatE

The Latent Feature DMM model



$$\boldsymbol{\theta}_d \sim \text{Dir}(\alpha) \qquad z_{d_i} \sim \text{Cat}(\boldsymbol{\theta}_d)$$
$$\boldsymbol{\phi}_z \sim \text{Dir}(\beta) \qquad s_{d_i} \sim \text{Ber}(\lambda)$$
$$w_{d_i} \sim (1 - s_{d_i})\text{Cat}(\boldsymbol{\phi}_{z_{d_i}}) + s_{d_i}\text{CatE}(\boldsymbol{\tau}_{z_{d_i}}\, \boldsymbol{\omega}^\top)$$

- s_{d_i} is the Boolean indicator variable indicating whether word d_i is generated from CatE
- λ is a user-specified hyper-parameter determining how often words are generated from the CatE distribution

Inference for the LF-LDA model

- We integrate out $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ as in the Griffiths et al. (2004) sampler, and *interleave MAP estimation for $\boldsymbol{\tau}$ with Gibbs sweeps for the other variables*
- **Algorithm outline:**
  initialise the word-topic variables z_{d_i} using the LDA sampler
  repeat:

  for each topic t:

  $\boldsymbol{\tau}_t = \arg\max_{\boldsymbol{\tau}_t} \mathrm{P}(\boldsymbol{\tau}_t \mid \boldsymbol{z}, \boldsymbol{s})$

  for each document d and each word location i:

  sample z_{d_i} from $\mathrm{P}(z_{d_i} \mid \boldsymbol{z}_{\neg d_i}, \boldsymbol{s}_{\neg d_i}, \boldsymbol{\tau})$ sample s_{d_i} from $\mathrm{P}(s_{d_i} \mid \boldsymbol{z}, \boldsymbol{s}_{\neg d_i}, \boldsymbol{\tau})$

Inference for the LF-DMM model (1)

- We integrate out $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ as in the Yin and Wang (2014) sampler, and *interleave MAP estimation for $\boldsymbol{\tau}$ with Gibbs sweeps*
- **Algorithm outline:**
  initialise the word-topic variables z_{d_i} using the DMM sampler
  repeat:

  for each topic t:

  $\boldsymbol{\tau}_t = \arg\max_{\boldsymbol{\tau}_t} \mathrm{P}(\boldsymbol{\tau}_t \mid \boldsymbol{z}, \boldsymbol{s})$

  for each document d:

  sample z_d from $\mathrm{P}(z_d \mid \boldsymbol{z}_{\neg d}, \boldsymbol{s}_{\neg d_i}, \boldsymbol{\tau})$

  for each word location i:

  sample s_{d_i} from $\mathrm{P}(s_{d_i} \mid \boldsymbol{z}, \boldsymbol{s}_{\neg d_i}, \boldsymbol{\tau})$
- Note: $\mathrm{P}(z_d \mid \boldsymbol{z}_{\neg d}, \boldsymbol{s}_{\neg d_i}, \boldsymbol{\tau})$ is *computationally expensive* to compute exactly, as it requires *summing over all possible values for $\boldsymbol{s}_d$*

Inference for the LF-DMM model (2)

- The computational problems stem from the fact that all the words in a document have the same topic
  - have to jointly sample *document topic* z_t and *indicator variables* $\boldsymbol{s}_d$
  - the sampling probability is a product of *ascending factorials*
- We approximate these probabilities by *assuming that the topic-word counts are "frozen"*, i.e., they don't increase within a document
  - the DMM is mainly used on *short documents* (e.g., Twitter), where the "one topic per document" assumption is accurate

  $\Rightarrow$ "freezing" the counts should have less impact
  - could correct this with a *Metropolis-Hastings accept-reject step*

$$\mathrm{P}(z_d, \boldsymbol{s}_d \mid \boldsymbol{z}_{\neg d}, \boldsymbol{s}_{\neg d}, \boldsymbol{\tau}) \quad \propto \quad \lambda^{K_d}\,(1-\lambda)^{N_d}\,(M^t_{\neg d}+\alpha)\frac{\Gamma(N^t_{\neg d}+V\beta)}{\Gamma(N^t_{\neg d}+N_d+V\beta)}$$
$$\left(\prod_{w\in W}\frac{\Gamma(N^{t,w}_{\neg d}+N^w_d+\beta)}{\Gamma(N^{t,w}_{\neg d}+\beta)}\right)\left(\prod_{w\in W}\mathrm{CatE}(w\mid \boldsymbol{\tau}_t\,\boldsymbol{\omega}^\top)^{K^w_d}\right)$$

Estimating the topic vectors $\boldsymbol{\tau}_t$

- Both the LF-LDA and LF-DMM associate each topic t with a *topic vector* $\boldsymbol{\tau}_t$, which must be learnt from the training corpus
- After each Gibbs sweep:
  - the topic variables $\boldsymbol{z}$ identify which topic each word is generated from
  - the indicator variables $\boldsymbol{s}$ identify which words are generated from the latent feature distributions CatE

  $\Rightarrow$ we can use a supervised estimation procedure to find $\boldsymbol{\tau}$
- We use LBFGS to optimise the L2-regularised log-loss (MAP estimation)

$$L_t = -\sum_{w \in W} K^{t,w} \left(\boldsymbol{\tau}_t \cdot \boldsymbol{\omega}_w - \log\left(\sum_{w' \in W} \exp(\boldsymbol{\tau}_t \cdot \boldsymbol{\omega}_{w'}) \right) \right) + \mu \parallel \boldsymbol{\tau}_t \parallel_2^2$$

Outline

Introduction

Latent-feature topic models

Experimental evaluation

Conclusions and future work

Goals of evaluation

- A topic model learns document-topic and topic-word distributions:
  - *topic coherence* evaluates the topic-word distributions
  - *document clustering* and *document classification* evaluate the document-topic distribution
    - the latent feature component only directly changes the topic-word distributions, so these are challenging evaluations
- Do the word2vec and Glove word vectors behave differently in topic modelling?
- We expect that the latent feature component will have *the greatest impact on small corpora*, so our evaluation focuses on them:

Dataset		# labels	# docs	words/doc	# types
N20	20 newsgroups	20	18,820	103.3	19,572
N20short	$\leq$ 20 words	20	1,794	13.6	6,377
N20small	400 docs	20	400	88.0	8,157
TMN	TagMyNews	7	32,597	18.3	13,428
TMNtitle	TMN titles	7	32,503	4.9	6,347
Twitter		4	2,520	5.0	1,390

Word2vec-DMM on TagMyNews titles corpus (1)

Topic 1								
Initdmm	Iter=1	Iter=2	Iter=5	Iter=10	Iter=20	Iter=50	Iter=100	Iter=500
japan	japan	japan	japan	japan	japan	japan	japan	japan
nuclear	nuclear	nuclear	nuclear	nuclear	nuclear	nuclear	nuclear	nuclear
u.s.	u.s.	u.s.	u.s.	u.s.	u.s.	plant	u.s.	u.s.
crisis	**russia**	crisis	plant	plant	plant	u.s.	plant	plant
plant	radiation	china	crisis	radiation	**quake**	**quake**	**quake**	**quake**
china	**nuke**	**russia**	radiation	crisis	radiation	radiation	radiation	radiation
libya	**iran**	plant	china	china	crisis	**earthquake**	**earthquake**	**earthquake**
radiation	crisis	radiation	**russia**	**nuke**	**nuke**	**tsunami**	**tsunami**	**tsunami**
u.n.	china	**nuke**	**nuke**	**russia**	china	**nuke**	**nuke**	**nuke**
vote	libya	libya	**power**	**power**	**tsunami**	crisis	crisis	crisis
korea	plant	**iran**	u.n.	u.n.	**earthquake**	**disaster**	**disaster**	**disaster**
europe	u.n.	u.n.	**iran**	**iran**	**disaster**	**plants**	**oil**	**power**
government	**mideast**	**power**	**reactor**	**earthquake**	**power**	**power**	**plants**	**oil**
election	**pakistan**	**pakistan**	**earthquake**	**reactor**	**reactor**	**oil**	**power**	**japanese**
deal	**talks**	**talks**	libya	**quake**	**japanese**	**japanese**	**tepco**	**plants**

- Table shows the 15 most probable topical words in Topic 1 found by 20-topic word2vec-DMM on the TMN titles corpus
- Words found by DMM but not by word2vec-DMM are underlined
- Words found by word2vec-DMM but not DMM are in bold

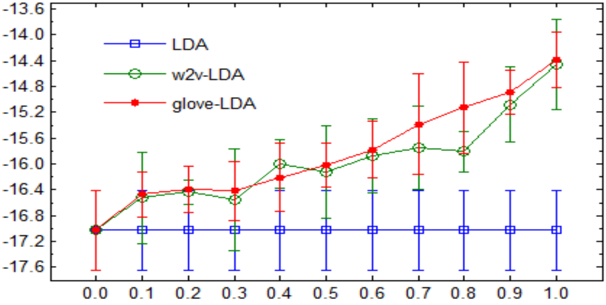
Word2Vec-DMM on TagMyNews titles corpus (2)

Topic 4			Topic 5			Topic 19			Topic 14		
Initdmm	Iter=50	Iter=500	Initdmm	Iter=50	Iter=500	Initdmm	Iter=50	Iter=500	Initdmm	Iter=50	Iter=500
egypt	libya	libya	critic	**dies**	star	nfl	nfl	nfl	nfl	law	law
china	egypt	egypt	corner	star	sheen	idol	draft	**sports**	court	bill	texas
u.s.	**mideast**	**iran**	office	**broadway**	**idol**	draft	**lockout**	draft	law	governor	bill
mubarak	**iran**	**mideast**	video	**american**	**broadway**	american	players	players	bill	texas	governor
bin	**opposition**	**opposition**	game	**idol**	show	show	**coach**	**lockout**	wisconsin	senate	senate
libya	**leader**	**protests**	star	lady	**american**	film	**nba**	**football**	players	union	union
laden	u.n.	**leader**	lady	gaga	gaga	season	**player**	**league**	judge	**obama**	**obama**
france	**protests**	**syria**	gaga	show	**tour**	sheen	sheen	n.f.l.	governor	wisconsin	budget
bahrain	**syria**	u.n.	show	**news**	**cbs**	n.f.l.	**league**	**player**	union	budget	wisconsin
air	**tunisia**	**tunisia**	weekend	critic	**hollywood**	back	n.f.l.	**baseball**	house	**state**	**immigration**
report	**protesters**	**chief**	sheen	**film**	**mtv**	top	**coaches**	**court**	texas	**immigration**	**state**
rights	**chief**	**protesters**	box	**hollywood**	lady	star	**football**	**coaches**	lockout	**arizona**	**vote**
court	**asia**	mubarak	park	**fame**	**wins**	charlie	**judge**	**nflpa**	budget	**california**	**washington**
u.n.	**russia**	**crackdown**	takes	**actor**	**charlie**	players	**nflpa**	**basketball**	peru	**vote**	**arizona**
war	**arab**	bahrain	man	**movie**	**stars**	men	**court**	**game**	senate	**federal**	**california**

- Table shows 15 most probable topical words in several topics found by 20-topic word2vec-DMM on the TMN titles corpus
- Words found by DMM but not by w2v-DMM are underlined
- Words found by w2v-DMM but not DMM are in bold

Topic coherence evaluation

- Lau et al. (2014) showed that *human scores on a word intrusion task* are highly correlated with the *normalised pointwise mutual information* (NPMI) against a large external corpus (we used English Wikipedia)
- We found latent feature vectors produced a *significant improvement of NPMI scores on all models and corpora*
  - greatest improvement when $\lambda = 1$ (unsurprisingly)



NPMI scores on the N20 short dataset with 20 topics, as the mixture weight λ varies from 0 to 1

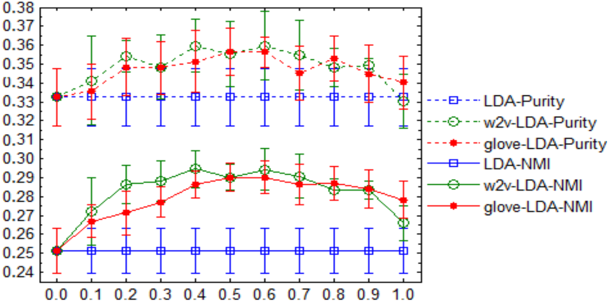
Topic coherence on Twitter corpus

Data	Method	$\lambda = 1.0$			
		T=4	T=20	T=40	T=80
Twitter	lda	-8.5 $\pm$ 1.1	-14.5 $\pm$ 0.4	-15.1 $\pm$ 0.4	-15.9 $\pm$ 0.2
	w2v-lda	-7.3 $\pm$ 1.0	**-13.2** $\pm$ 0.6	**-14.0** $\pm$ 0.3	**-14.1** $\pm$ 0.3
	glove-lda	**-6.2** $\pm$ 1.6	-13.9 $\pm$ 0.6	-14.2 $\pm$ 0.4	-14.2 $\pm$ 0.2
	Improve.	2.3	1.3	1.1	1.8
Twitter	dmm	-5.9 $\pm$ 1.1	-10.4 $\pm$ 0.7	-12.0 $\pm$ 0.3	-13.3 $\pm$ 0.3
	w2v-dmm	-5.5 $\pm$ 0.7	-10.5 $\pm$ 0.5	-11.2 $\pm$ 0.5	**-12.5** $\pm$ 0.1
	glove-dmm	**-5.1** $\pm$ 1.2	**-9.9** $\pm$ 0.6	**-11.1** $\pm$ 0.3	**-12.5** $\pm$ 0.4
	Improve.	0.8	0.5	0.9	0.8

- The normalised pointwise mutual information score improves for both LDA and DMM on the Twitter corpus, across a wide range of number of topics

Document clustering evaluation

- Cluster documents by assigning them to the *highest probability topic*
- Evaluate clusterings by *purity* and *normalised mutual information* (NMI) (Manning et al., 2008)



Evaluation of 20-topic LDA on the N20 short corpus, as mixture weight λ varies from 0 to 1

- In general, best results with $\lambda = 0.6$

$\Rightarrow$ Set $\lambda = 0.6$ in all further experiments

Document clustering of Twitter data

Data	Method	Purity				NMI			
		T=4	T=20	T=40	T=80	T=4	T=20	T=40	T=80
Twitter	lda	0.559 ± 0.020	0.614 ± 0.016	0.626 ± 0.011	0.631 ± 0.008	0.196 ± 0.018	0.174 ± 0.008	0.170 ± 0.007	0.160 ± 0.004
	w2v-lda	**0.598** ± 0.023	**0.635** ± 0.016	**0.638** ± 0.009	**0.637** ± 0.012	**0.249** ± 0.021	**0.191** ± 0.011	0.176 ± 0.003	**0.167** ± 0.006
	glove-lda	0.597 ± 0.016	**0.635** ± 0.014	0.637 ± 0.010	**0.637** ± 0.007	0.242 ± 0.013	**0.191** ± 0.007	**0.177** ± 0.007	0.165 ± 0.005
	Improve.	0.039	0.021	0.012	0.006	0.053	0.017	0.007	0.007
Twitter	dmm	0.552 ± 0.020	0.624 ± 0.010	0.647 ± 0.009	0.675 ± 0.009	0.194 ± 0.017	0.186 ± 0.006	0.184 ± 0.005	0.190 ± 0.003
	w2v-dmm	**0.581** ± 0.019	0.641 ± 0.013	**0.660** ± 0.010	**0.687** ± 0.007	0.230 ± 0.015	0.195 ± 0.007	**0.193** ± 0.004	**0.199** ± 0.005
	glove-dmm	0.580 ± 0.013	**0.644** ± 0.016	0.657 ± 0.008	0.684 ± 0.006	**0.232** ± 0.010	**0.201** ± 0.010	0.191 ± 0.006	0.195 ± 0.005
	Improve.	0.029	0.02	0.013	0.012	0.038	0.015	0.009	0.009

- On the short, small Twitter dataset our models obtain better clustering results than the baseline models with small T.
  - with $T = 4$ we obtain 3.9% purity and 5.3% NMI improvements
- For small $T \leq 7$, on the large datasets of N20, TMN and TMNtitle, our models and baseline models obtain similar clustering results.
- With larger T our models perform better than baselines on the short TMN and TMNtitle datasets
- On the N20 dataset, the baseline LDA model obtains better clustering results than ours
- No reliable difference between word2vec and Glove vectors

Document classification of N20 and N20short corpora

- Train a SVM to predict document label based on topic(s) assigned to document

Data	Model	$\lambda = 0.6$			
		T=6	T=20	T=40	T=80
N20	lda	0.312 $\pm$ 0.013	0.635 $\pm$ 0.016	**0.742** $\pm$ 0.014	0.763 $\pm$ 0.005
	w2v-lda	**0.316** $\pm$ 0.013	0.641 $\pm$ 0.019	0.730 $\pm$ 0.017	**0.768** $\pm$ 0.004
	glove-lda	0.288 $\pm$ 0.013	**0.650** $\pm$ 0.024	0.733 $\pm$ 0.011	0.762 $\pm$ 0.006
	Improve.	0.004	0.015	-0.009	0.005
N20small	lda	0.204 $\pm$ 0.020	0.392 $\pm$ 0.029	0.459 $\pm$ 0.030	0.477 $\pm$ 0.025
	w2v-lda	**0.213** $\pm$ 0.018	**0.442** $\pm$ 0.025	**0.502** $\pm$ 0.031	**0.509** $\pm$ 0.022
	glove-lda	0.181 $\pm$ 0.011	0.420 $\pm$ 0.025	0.474 $\pm$ 0.029	0.498 $\pm$ 0.012
	Improve.	0.009	0.05	0.043	0.032

- F_1 scores (mean and standard deviation) for N20 and N20small corpora

Document classification of TMN and TMN title corpora

Data	Model	$\lambda = 0.6$			
		T=7	T=20	T=40	T=80
TMN	lda	0.658 ± 0.026	0.754 ± 0.009	0.768 ± 0.004	0.778 ± 0.004
	w2v-lda	0.663 ± 0.021	0.758 ± 0.009	**0.769** ± 0.005	**0.780** ± 0.004
	glove-lda	**0.664** ± 0.025	**0.760** ± 0.006	0.767 ± 0.003	0.779 ± 0.004
	Improve.	0.006	0.006	0.001	0.002
TMN	dmm	0.605 ± 0.023	0.724 ± 0.016	0.738 ± 0.008	0.741 ± 0.005
	w2v-dmm	0.619 ± 0.033	0.744 ± 0.009	0.759 ± 0.005	**0.777** ± 0.005
	glove-dmm	**0.624** ± 0.025	**0.757** ± 0.009	**0.761** ± 0.005	0.774 ± 0.010
	Improve.	0.019	0.033	0.023	0.036
TMNtitle	lda	0.564 ± 0.015	0.625 ± 0.011	0.626 ± 0.010	0.624 ± 0.006
	w2v-lda	0.563 ± 0.029	**0.644** ± 0.010	**0.643** ± 0.007	0.640 ± 0.004
	glove-lda	**0.568** ± 0.028	**0.644** ± 0.010	0.632 ± 0.008	**0.642** ± 0.005
	Improve.	0.004	0.019	0.017	0.018
TMNtitle	dmm	0.570 ± 0.022	0.650 ± 0.011	0.654 ± 0.008	0.646 ± 0.008
	w2v-dmm	0.562 ± 0.022	0.670 ± 0.012	0.677 ± 0.006	**0.680** ± 0.003
	glove-dmm	**0.592** ± 0.017	**0.674** ± 0.016	**0.683** ± 0.006	0.679 ± 0.009
	Improve.	0.022	0.024	0.029	0.034

Document classification of Twitter corpus

Data	Method	$\lambda = 0.6$			
		T=4	T=20	T=40	T=80
Twitter	lda	0.526 ± 0.021	0.636 ± 0.011	0.650 ± 0.014	0.653 ± 0.008
	w2v-lda	**0.578** ± 0.047	0.651 ± 0.015	0.661 ± 0.011	**0.664** ± 0.010
	glove-lda	0.569 ± 0.037	**0.656** ± 0.011	**0.662** ± 0.008	0.662 ± 0.006
	Improve.	0.052	0.02	0.012	0.011
Twitter	dmm	0.505 ± 0.023	0.614 ± 0.012	0.634 ± 0.013	0.656 ± 0.011
	w2v-dmm	**0.541** ± 0.035	0.636 ± 0.015	**0.648** ± 0.011	**0.670** ± 0.010
	glove-dmm	0.539 ± 0.024	**0.638** ± 0.017	0.645 ± 0.012	0.666 ± 0.009
	Improve.	0.036	0.024	0.014	0.014

- For document classification the latent feature models generally perform better than the baseline models
  - On the small N20small and Twitter datasets, when the number of topics T is equal to number of ground truth labels (i.e. 20 and 4 correspondingly) our W2V-LDA model obtains $5^{+}\%$ higher F_1 score than the LDA model
  - Our W2V-DMM model achieves 3.6% and 3.4% higher F_1 score than the DMM model on the TMN and TMNtitle datasets with $T = 80$, respectively.

Outline

Introduction

Latent-feature topic models

Experimental evaluation

Conclusions and future work

Conclusions

- Latent feature vectors induced from large external corpora can be used to improve topic modelling
  - latent features significantly improve topic coherence across a range of corpora with both the LDA and DMM models
  - document clustering and document classification also significantly improve, even though these depend directly only on the document-topic distribution
- The improvements were greatest for small document collections and/or for short documents
  - with enough training data there is sufficient information in the corpus to accurately estimate topic-word distributions
  - the improvement in the topic-word distributions also improves the document-topic distribution
- We did not detect any reliable difference between word2vec and Glove vectors

Future directions

- Retrain the word vectors to fit the training corpus
  - how do we avoid losing information from external corpus?
- More sophisticated latent-feature models of topic-word distributions
- More efficient training procedures (e.g., using SGD)
- Extend this approach to a richer class of topic models