A Framework for Incorporating General Domain Knowledge into Latent Dirichlet Allocation using First-Order Logic

David Andrzejewski[1] Xiaojin Zhu [2]
Mark Craven [3,2] Benjamin Recht [2]

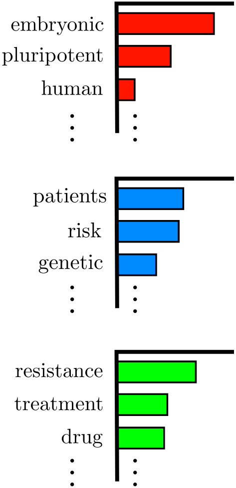
[1]Center for Applied Scientific Computing
Lawrence Livermore National Laboratory (USA)

[2]Department of Computer Sciences
[3]Department of Biostatistics
and Medical Informatics
University of Wisconsin–Madison (USA)

Topic modeling with Latent Dirichlet Allocation (LDA)

Blei et al, JMLR 2003

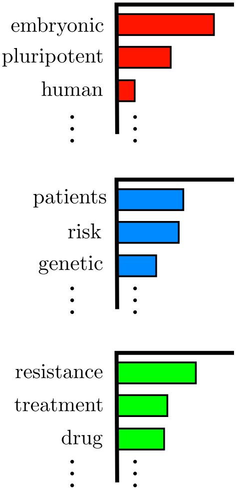
Topics ϕ



Topic modeling with Latent Dirichlet Allocation (LDA)

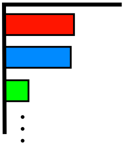
Blei et al, JMLR 2003

Topics ϕ

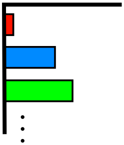


Document-topic weights θ

Human embryonic stem cell research may benefit patients with genetic risk factors...



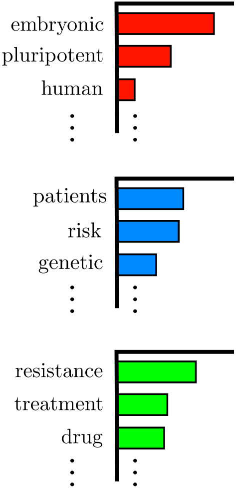
Patients at risk for drug-resistant infection...



Topic modeling with Latent Dirichlet Allocation (LDA)

Blei et al, JMLR 2003

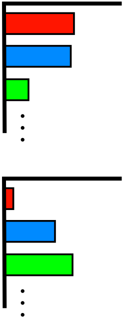
Topics ϕ



Document-topic weights θ

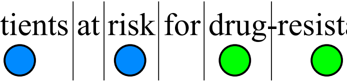
Human embryonic stem cell research may benefit patients with genetic risk factors...

Patients at risk for drug-resistant infection...



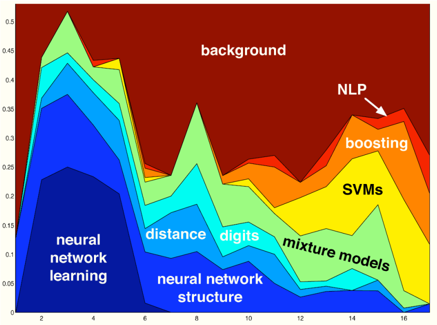
Observed $\mathbf{w}$ Patients | at | risk | for | drug-resistant

Latent $\mathbf{z}$



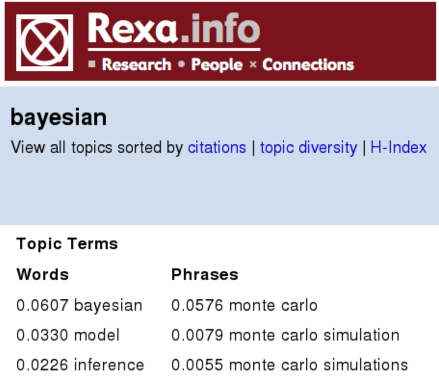
Topic modeling applications

- Research trends (Wang & McCallum, 2006)
- Info retrieval (UMass) (also KDD 2011!)
- Author/document profiling
  - Scientific impact/influence (Gerrish & Blei, 2009)
  - Match papers to reviewers (Mimno & McCallum, 2007)



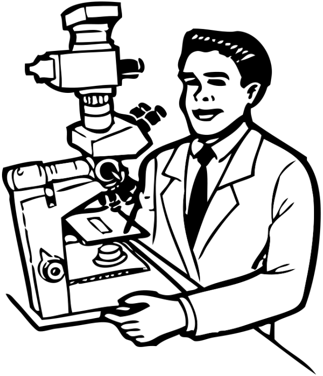
Topic modeling applications

- Research trends (Wang & McCallum, 2006)
- Info retrieval (UMass) (also KDD 2011!)
- Author/document profiling
  - Scientific impact/influence (Gerrish & Blei, 2009)
  - Match papers to reviewers (Mimno & McCallum, 2007)



Topic modeling applications

- Research trends
  (Wang & McCallum, 2006)
- Info retrieval (UMass)
  (also KDD 2011!)
- Author/document profiling
  - Scientific impact/influence
    (Gerrish & Blei, 2009)
  - Match papers to reviewers
    (Mimno & McCallum, 2007)



Unsupervised LDA

Extend the model? Add domain knowledge

- "These words words do (not) belong in the same topic"
- "I want a topic about X"
- "This topic is incompatible with this document"
- "These topics are incompatible - should not co-occur"

First-Order Logic latent Dirichlet Allocation (Fold·all)

- Weighted knowledge base (KB) of first-order logic (FOL) rules (Markov Logic Networks, Richardson and Domingos 2006)
- Learned topics ϕ influenced by *both*

Unsupervised LDA

~~Extend the model?~~ Add domain knowledge

- "These words words do (not) belong in the same topic"
- "I want a topic about *X*"
- "This topic is incompatible with this document"
- "These topics are incompatible - should not co-occur"

First-Order Logic latent Dirichlet Allocation (Fold·all)

- Weighted knowledge base (KB) of first-order logic (FOL) rules (Markov Logic Networks, Richardson and Domingos 2006)
- Learned topics ϕ influenced by *both*

Unsupervised LDA

~~Extend the model?~~ Add domain knowledge

- "These words words do (not) belong in the same topic"
- "I want a topic about *X*"
- "This topic is incompatible with this document"
- "These topics are incompatible - should not co-occur"

First-Order Logic latent Dirichlet Allocation (Fold·all)

- Weighted knowledge base (KB) of first-order logic (FOL) rules (Markov Logic Networks, Richardson and Domingos 2006)
- Learned topics ϕ influenced by *both*

Unsupervised LDA

~~Extend the model?~~ Add domain knowledge

- "These words words do (not) belong in the same topic"
- "I want a topic about *X*"
- "This topic is incompatible with this document"
- "These topics are incompatible - should not co-occur"

First-Order Logic latent Dirichlet Allocation (Fold·all)

- Weighted knowledge base (KB) of first-order logic (FOL) rules (Markov Logic Networks, Richardson and Domingos 2006)
- Learned topics ϕ influenced by *both*

Unsupervised LDA

~~Extend the model?~~ Add domain knowledge

- "These words words do (not) belong in the same topic"
- "I want a topic about *X*"
- "This topic is incompatible with this document"
- "These topics are incompatible - should not co-occur"

First-Order Logic latent Dirichlet Allocation (Fold·all)

- Weighted knowledge base (KB) of first-order logic (FOL) rules (Markov Logic Networks, Richardson and Domingos 2006)
- Learned topics ϕ influenced by *both*

Unsupervised LDA

~~Extend the model?~~ Add domain knowledge

- "These words words do (not) belong in the same topic"
- "I want a topic about *X*"
- "This topic is incompatible with this document"
- "These topics are incompatible - should not co-occur"

First-Order Logic latent Dirichlet Allocation (Fold·all)

- Weighted knowledge base (KB) of first-order logic (FOL) rules (Markov Logic Networks, Richardson and Domingos 2006)
- Learned topics ϕ influenced by *both*

Unsupervised LDA

~~Extend the model?~~ Add domain knowledge

- "These words words do (not) belong in the same topic"
- "I want a topic about *X*"
- "This topic is incompatible with this document"
- "These topics are incompatible - should not co-occur"

First-Order Logic latent Dirichlet Allocation (Fold·all)

- Weighted knowledge base (KB) of first-order logic (FOL) rules (Markov Logic Networks, Richardson and Domingos 2006)
- Learned topics ϕ influenced by *both*

Unsupervised LDA

~~Extend the model?~~ Add domain knowledge

- "These words words do (not) belong in the same topic"
- "I want a topic about *X*"
- "This topic is incompatible with this document"
- "These topics are incompatible - should not co-occur"

First-Order Logic latent Dirichlet Allocation (Fold·all)

- Weighted knowledge base (KB) of first-order logic (FOL) rules (Markov Logic Networks, Richardson and Domingos 2006)
- Learned topics ϕ influenced by *both*
  - Word-document statistics (as in LDA)
  - Domain knowledge rules (as in MLN)

Unsupervised LDA

~~Extend the model?~~ Add domain knowledge

- "These words words do (not) belong in the same topic"
- "I want a topic about *X*"
- "This topic is incompatible with this document"
- "These topics are incompatible - should not co-occur"

First-Order Logic latent Dirichlet Allocation (Fold·all)

- Weighted knowledge base (KB) of first-order logic (FOL) rules (Markov Logic Networks, Richardson and Domingos 2006)
- Learned topics ϕ influenced by *both*
  - Word-document statistics (as in LDA)
  - Domain knowledge rules (as in MLN)

Unsupervised LDA

~~Extend the model?~~ Add domain knowledge

- "These words words do (not) belong in the same topic"
- "I want a topic about *X*"
- "This topic is incompatible with this document"
- "These topics are incompatible - should not co-occur"

First-Order Logic latent Dirichlet Allocation (Fold·all)

- Weighted knowledge base (KB) of first-order logic (FOL) rules (Markov Logic Networks, Richardson and Domingos 2006)
- Learned topics ϕ influenced by *both*
  - Word-document statistics (as in LDA)
  - Domain knowledge rules (as in MLN)

Unsupervised LDA

~~Extend the model?~~ Add domain knowledge

- "These words words do (not) belong in the same topic"
- "I want a topic about *X*"
- "This topic is incompatible with this document"
- "These topics are incompatible - should not co-occur"

First-Order Logic latent Dirichlet Allocation (Fold·all)

- Weighted knowledge base (KB) of first-order logic (FOL) rules (Markov Logic Networks, Richardson and Domingos 2006)
- Learned topics ϕ influenced by *both*
  - Word-document statistics (as in LDA)
  - Domain knowledge rules (as in MLN)

Representing LDA with logical predicates

	Value	Logical Predicate	Description
	$z_i = t$	$\mathtt{Z}(i, t)$	Latent topic
LDA	$w_i = v$	$\mathtt{W}(i, v)$	Observed word
	$d_i = j$	$\mathtt{D}(i, j)$	Observed document

Unified way to capture metadata / annotations

Representing LDA with logical predicates

	Value	Logical Predicate	Description
LDA	$z_i = t$	$\texttt{Z}(i, t)$	Latent topic
	$w_i = v$	$\texttt{W}(i, v)$	Observed word
	$d_i = j$	$\texttt{D}(i, j)$	Observed document
Attributes		$\texttt{HasLabel}(j, \ell)$	Document label
		$\texttt{S}(i, k)$	Observed sentence

Unified way to capture metadata / annotations

Representing LDA with logical predicates

	Value	Logical Predicate	Description
LDA	$z_i = t$	$\mathtt{Z}(i, t)$	Latent topic
	$w_i = v$	$\mathtt{W}(i, v)$	Observed word
	$d_i = j$	$\mathtt{D}(i, j)$	Observed document
Attributes		$\mathtt{HasLabel}(j, \ell)$	Document label
		$\mathtt{S}(i, k)$	Observed sentence

Unified way to capture metadata / annotations

Encoding domain knowledge in First-Order Logic

- CNF Knowledge Base $KB = \{(\lambda_1, \psi_1), \ldots, (\lambda_L, \psi_L)\}$
  - Rule ψ_k
  - Weight $\lambda_k > 0$ ("strength" of rule)

Example *KB*

Rule	λ_k	$\forall$	ψ_k
Seed	5	i	$\mathtt{W}(i, embryo) \Rightarrow \mathtt{Z}(i, 3)$
Doc label	500	i, j	$\mathtt{D}(i, j) \wedge \mathtt{HasLabel}(j, +) \Rightarrow \neg\mathtt{Z}(i, 3)$

Can specify "contradictory" domain knowledge!

Encoding domain knowledge in First-Order Logic

- CNF Knowledge Base $KB = \{(\lambda_1, \psi_1), \ldots, (\lambda_L, \psi_L)\}$
  - Rule ψ_k
  - Weight $\lambda_k > 0$ ("strength" of rule)

Example KB

Rule	λ_k	$\forall$	ψ_k
Seed	5	i	$\mathtt{W}(i, embryo) \Rightarrow \mathtt{Z}(i, 3)$
Doc label	500	i, j	$\mathtt{D}(i, j) \wedge \mathtt{HasLabel}(j, +) \Rightarrow \neg \mathtt{Z}(i, 3)$

Can specify "contradictory" domain knowledge!

Encoding domain knowledge in First-Order Logic

- CNF Knowledge Base $KB = \{(\lambda_1, \psi_1), \ldots, (\lambda_L, \psi_L)\}$
  - Rule ψ_k
  - Weight $\lambda_k > 0$ ("strength" of rule)

Example *KB*

Rule	λ_k	$\forall$	ψ_k
Seed	5	i	$\mathtt{W}(i, embryo) \Rightarrow \mathtt{Z}(i, 3)$
Doc label	500	i, j	$\mathtt{D}(i, j) \wedge \mathtt{HasLabel}(j, +) \Rightarrow \neg\mathtt{Z}(i, 3)$

Can specify "contradictory" domain knowledge!

Encoding domain knowledge in First-Order Logic

- CNF Knowledge Base $KB = \{(\lambda_1, \psi_1), \ldots, (\lambda_L, \psi_L)\}$
  - Rule ψ_k
  - Weight $\lambda_k > 0$ ("strength" of rule)

Example *KB*

Rule	λ_k	$\forall$	ψ_k
Seed	5	i	$\texttt{W}(i, embryo) \Rightarrow \texttt{Z}(i, 3)$
Doc label	500	i, j	$\texttt{D}(i, j) \wedge \texttt{HasLabel}(j, +) \Rightarrow \neg\texttt{Z}(i, 3)$

Can specify "contradictory" domain knowledge!

Encoding domain knowledge in First-Order Logic

- CNF Knowledge Base $KB = \{(\lambda_1, \psi_1), \ldots, (\lambda_L, \psi_L)\}$
  - Rule ψ_k
  - Weight $\lambda_k > 0$ ("strength" of rule)

Example *KB*

Rule	λ_k	$\forall$	ψ_k
Seed	5	i	$\texttt{W}(i, embryo) \Rightarrow \texttt{Z}(i, 3)$
Doc label	500	i, j	$\texttt{D}(i, j) \wedge \texttt{HasLabel}(j, +) \Rightarrow \neg\texttt{Z}(i, 3)$

Can specify "contradictory" domain knowledge!

Encoding domain knowledge in First-Order Logic

- CNF Knowledge Base $KB = \{(\lambda_1, \psi_1), \ldots, (\lambda_L, \psi_L)\}$
  - Rule ψ_k
  - Weight $\lambda_k > 0$ ("strength" of rule)

Example *KB*

Rule	λ_k	$\forall$	ψ_k
Seed	5	i	$\mathtt{W}(i, embryo) \Rightarrow \mathtt{Z}(i, 3)$
Doc label	500	i, j	$\mathtt{D}(i, j) \wedge \mathtt{HasLabel}(j, +) \Rightarrow \neg \mathtt{Z}(i, 3)$

Can specify "contradictory" domain knowledge!

Propositionalization / Grounding

Example Cannot-Link rule ψ_{CL}

λ_k	$\forall$	ψ_k
5	i, j, t	$\mathtt{W}(i, \mathit{neural}) \wedge \mathtt{W}(j, \mathit{disorder}) \Rightarrow \neg\mathtt{Z}(i, t) \vee \neg\mathtt{Z}(j, t)$

- $G(\psi_{CL})$ = set of ground formulas g for **EVERY** (i, j, t)
  - $i, j \in \{1, 2, \ldots, N\}$
  - $t \in \{1, 2, \ldots, T\}$
- $\mathbb{1}_g(\mathbf{z}) = \begin{cases} 1 & \text{if } g \text{ true under } \mathbf{z} \\ 0 & \text{else} \end{cases}$
- Each $g \in G(\psi_{CL}) \rightarrow \lambda \mathbb{1}_g(\mathbf{z})$ term (as in MLN)
- **Combinatorial explosion!**

Propositionalization / Grounding

Example Cannot-Link rule ψ_{CL}

λ_k	$\forall$	ψ_k
5	i, j, t	$\mathtt{W}(i, neural) \wedge \mathtt{W}(j, disorder) \Rightarrow \neg \mathtt{Z}(i, t) \vee \neg \mathtt{Z}(j, t)$

- $G(\psi_{CL})$ = set of ground formulas g for **EVERY** (i, j, t)
  - $i, j \in \{1, 2, \ldots, N\}$
  - $t \in \{1, 2, \ldots, T\}$
- $\mathbb{1}_g(\mathbf{z}) = \begin{cases} 1 & \text{if } g \text{ true under } \mathbf{z} \\ 0 & \text{else} \end{cases}$
- Each $g \in G(\psi_{CL}) \rightarrow \lambda \mathbb{1}_g(\mathbf{z})$ term (as in MLN)
- **Combinatorial explosion!**

Propositionalization / Grounding

Example Cannot-Link rule ψ_{CL}

λ_k	$\forall$	ψ_k
5	i, j, t	$\mathtt{W}(i, neural) \wedge \mathtt{W}(j, disorder) \Rightarrow \neg\mathtt{Z}(i, t) \vee \neg\mathtt{Z}(j, t)$

- $G(\psi_{CL})$ = set of ground formulas g for **EVERY** (i, j, t)
  - $i, j \in \{1, 2, \ldots, N\}$
  - $t \in \{1, 2, \ldots, T\}$
- $\mathbb{1}_g(\mathbf{z}) = \begin{cases} 1 & \text{if } g \text{ true under } \mathbf{z} \\ 0 & \text{else} \end{cases}$
- Each $g \in G(\psi_{CL}) \rightarrow \lambda \mathbb{1}_g(\mathbf{z})$ term (as in MLN)
- **Combinatorial explosion!**

Propositionalization / Grounding

Example Cannot-Link rule ψ_{CL}

λ_k	$\forall$	ψ_k
5	i, j, t	$\mathtt{W}(i, neural) \wedge \mathtt{W}(j, disorder) \Rightarrow \neg \mathtt{Z}(i, t) \vee \neg \mathtt{Z}(j, t)$

- $G(\psi_{CL})$ = set of ground formulas g for **EVERY** (i, j, t)
  - $i, j \in \{1, 2, \ldots, N\}$
  - $t \in \{1, 2, \ldots, T\}$
- $\mathbb{1}_g(\mathbf{z}) = \begin{cases} 1 & \text{if } g \text{ true under } \mathbf{z} \\ 0 & \text{else} \end{cases}$
- Each $g \in G(\psi_{CL}) \rightarrow \lambda \mathbb{1}_g(\mathbf{z})$ term (as in MLN)
- **Combinatorial explosion!**

Propositionalization / Grounding

Example Cannot-Link rule ψ_{CL}

λ_k	$\forall$	ψ_k
5	i, j, t	$\mathtt{W}(i, neural) \wedge \mathtt{W}(j, disorder) \Rightarrow \neg\mathtt{Z}(i, t) \vee \neg\mathtt{Z}(j, t)$

- $G(\psi_{CL})$ = set of ground formulas g for **EVERY** (i, j, t)
  - $i, j \in \{1, 2, \ldots, N\}$
  - $t \in \{1, 2, \ldots, T\}$
- $\mathbb{1}_g(\mathbf{z}) = \begin{cases} 1 & \text{if } g \text{ true under } \mathbf{z} \\ 0 & \text{else} \end{cases}$
- Each $g \in G(\psi_{CL}) \rightarrow \lambda \mathbb{1}_g(\mathbf{z})$ term (as in MLN)
- **Combinatorial explosion!**

Propositionalization / Grounding

Example Cannot-Link rule ψ_{CL}

λ_k	$\forall$	ψ_k
5	i, j, t	$\mathtt{W}(i, neural) \wedge \mathtt{W}(j, disorder) \Rightarrow \neg\mathtt{Z}(i, t) \vee \neg\mathtt{Z}(j, t)$

- $G(\psi_{CL})$ = set of ground formulas g for **EVERY** (i, j, t)
  - $i, j \in \{1, 2, \ldots, N\}$
  - $t \in \{1, 2, \ldots, T\}$
- $\mathbb{1}_g(\mathbf{z}) = \begin{cases} 1 & \text{if } g \text{ true under } \mathbf{z} \\ 0 & \text{else} \end{cases}$
- Each $g \in G(\psi_{CL}) \rightarrow \lambda \mathbb{1}_g(\mathbf{z})$ term (as in MLN)
- **Combinatorial explosion!**

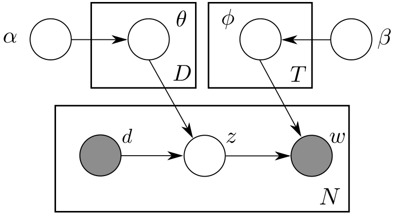
Propositionalization / Grounding

Example Cannot-Link rule ψ_{CL}

λ_k	$\forall$	ψ_k
5	i, j, t	$\mathtt{W}(i, neural) \land \mathtt{W}(j, disorder) \Rightarrow \neg\mathtt{Z}(i, t) \lor \neg\mathtt{Z}(j, t)$

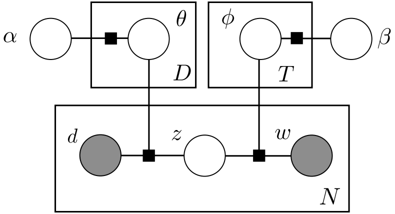
- $G(\psi_{CL})$ = set of ground formulas g for **EVERY** (i, j, t)
  - $i, j \in \{1, 2, \ldots, N\}$
  - $t \in \{1, 2, \ldots, T\}$
- $\mathbb{1}_g(\mathbf{z}) = \begin{cases} 1 & \text{if } g \text{ true under } \mathbf{z} \\ 0 & \text{else} \end{cases}$
- Each $g \in G(\psi_{CL}) \rightarrow \lambda \mathbb{1}_g(\mathbf{z})$ term (as in MLN)
- **Combinatorial explosion!**

LDA graphical model



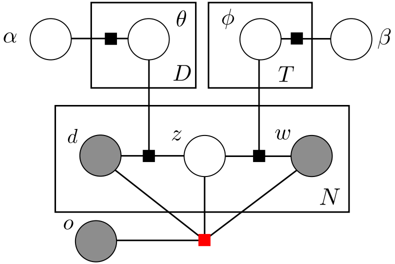
$$P \propto \left(\prod_{t}^{T} p(\phi_t | \beta) \right) \left(\prod_{j}^{D} p(\theta_j | \alpha) \right) \left(\prod_{i}^{N} \phi_{z_i}(w_i) \theta_{d_i}(z_i) \right)$$

LDA graphical model



$$P \propto \left(\prod_{t}^{T} p(\phi_t|\beta)\right) \left(\prod_{j}^{D} p(\theta_j|\alpha)\right) \left(\prod_{i}^{N} \phi_{z_i}(w_i)\theta_{d_i}(z_i)\right)$$

LDA graphical model → Fold·all



$$P \propto \left(\prod_{t}^{T} p(\phi_t|\beta)\right)\left(\prod_{j}^{D} p(\theta_j|\alpha)\right)\left(\prod_{i}^{N} \phi_{z_i}(w_i)\theta_{d_i}(z_i)\right)$$

$$\times \exp\left[\sum_{k}^{L} \sum_{g \in G(\psi_k)} \lambda_k \mathbb{1}_g(\mathbf{z}, \mathbf{w}, \mathbf{d}, \mathbf{o})\right]$$

MAP inference - $Q(\mathbf{z}, \phi, \theta)$

Alternating Optimization with Mirror Descent

For each step

1. $(\phi, \theta) \leftarrow \underset{\phi,\theta}{\text{argmax}}\ Q(\mathbf{z}, \phi, \theta)$ — $\mathbf{z}$ fixed
2. $\mathbf{z} \leftarrow \underset{\mathbf{z}}{\text{argmax}}\ Q(\mathbf{z}, \phi, \theta)$ — (ϕ, θ) fixed
   - $\mathbf{z} \setminus \mathbf{z}_{KB} \leftarrow$ argmax with respect to (ϕ, θ) — TRIVIAL
   - $\mathbf{z}_{KB} \leftarrow$ mirror descent — HARD

Scalable approach to optimize $\mathbf{z}_{KB}$

1. Relax discrete problem to continuous
2. Optimize relaxed problem with stochastic gradient descent
3. Round relaxed $\mathbf{z}$ to recover final assignment

MAP inference - $Q(\mathbf{z}, \phi, \theta)$

Alternating Optimization with Mirror Descent

For each step

1. $(\phi, \theta) \leftarrow \underset{\phi,\theta}{\text{argmax}}\ Q(\mathbf{z}, \phi, \theta)$ — $\mathbf{z}$ fixed
2. $\mathbf{z} \leftarrow \underset{\mathbf{z}}{\text{argmax}}\ Q(\mathbf{z}, \phi, \theta)$ — (ϕ, θ) fixed
   - $\mathbf{z} \setminus \mathbf{z}_{KB} \leftarrow$ argmax with respect to (ϕ, θ) — **TRIVIAL**
   - $\mathbf{z}_{KB} \leftarrow$ mirror descent — **HARD**

Scalable approach to optimize $\mathbf{z}_{KB}$

1. Relax discrete problem to continuous
2. Optimize relaxed problem with stochastic gradient descent
3. Round relaxed $\mathbf{z}$ to recover final assignment

MAP inference - $Q(\mathbf{z}, \phi, \theta)$

Alternating Optimization with Mirror Descent

For each step

1. $(\phi, \theta) \leftarrow \underset{\phi,\theta}{\text{argmax}}\; Q(\mathbf{z}, \phi, \theta)$ — $\mathbf{z}$ fixed
2. $\mathbf{z} \leftarrow \underset{\mathbf{z}}{\text{argmax}}\; Q(\mathbf{z}, \phi, \theta)$ — (ϕ, θ) fixed
   - $\mathbf{z} \setminus \mathbf{z}_{KB} \leftarrow$ argmax with respect to (ϕ, θ) — **TRIVIAL**
   - $\mathbf{z}_{KB} \leftarrow$ mirror descent — **HARD**

Scalable approach to optimize $\mathbf{z}_{KB}$

1. Relax discrete problem to continuous
2. Optimize relaxed problem with stochastic gradient descent
3. Round relaxed $\mathbf{z}$ to recover final assignment

For each step

1. $(\phi, \theta) \leftarrow \operatorname{argmax}_{\phi,\theta} Q(\mathbf{z}, \phi, \theta)$ — **z** fixed
2. $\mathbf{z} \leftarrow \operatorname{argmax}_{\mathbf{z}} Q(\mathbf{z}, \phi, \theta)$ — (ϕ, θ) fixed
   - $\mathbf{z} \setminus \mathbf{z}_{KB} \leftarrow$ argmax with respect to (ϕ, θ) — **TRIVIAL**
   - $\mathbf{z}_{KB} \leftarrow$ mirror descent — **HARD**

Scalable approach to optimize $\mathbf{z}_{KB}$

1. Relax discrete problem to continuous
2. Optimize relaxed problem with stochastic gradient descent
3. Round relaxed **z** to recover final assignment

MAP inference - $Q(\mathbf{z}, \phi, \theta)$

Alternating Optimization with Mirror Descent

For each step

1. $(\phi, \theta) \leftarrow \underset{\phi,\theta}{\text{argmax}}\ Q(\mathbf{z}, \phi, \theta)$ — $\mathbf{z}$ fixed
2. $\mathbf{z} \leftarrow \underset{\mathbf{z}}{\text{argmax}}\ Q(\mathbf{z}, \phi, \theta)$ — (ϕ, θ) fixed
   - $\mathbf{z} \setminus \mathbf{z}_{KB} \leftarrow$ argmax with respect to (ϕ, θ) — **TRIVIAL**
   - $\mathbf{z}_{KB} \leftarrow$ mirror descent — **HARD**

Scalable approach to optimize $\mathbf{z}_{KB}$

1. Relax discrete problem to continuous
2. Optimize relaxed problem with stochastic gradient descent
3. Round relaxed $\mathbf{z}$ to recover final assignment

MAP inference - $Q(\mathbf{z}, \phi, \theta)$

Alternating Optimization with Mirror Descent

For each step

1. $(\phi, \theta) \leftarrow \underset{\phi,\theta}{\text{argmax}}\ Q(\mathbf{z}, \phi, \theta)$ — **z** fixed
2. $\mathbf{z} \leftarrow \underset{\mathbf{z}}{\text{argmax}}\ Q(\mathbf{z}, \phi, \theta)$ — (ϕ, θ) fixed
   - $\mathbf{z} \setminus \mathbf{z}_{KB} \leftarrow$ argmax with respect to (ϕ, θ) — **TRIVIAL**
   - $\mathbf{z}_{KB} \leftarrow$ mirror descent — **HARD**

Scalable approach to optimize $\mathbf{z}_{KB}$

1. Relax discrete problem to continuous
2. Optimize relaxed problem with stochastic gradient descent
3. Round relaxed **z** to recover final assignment

MAP inference - $Q(\mathbf{z}, \phi, \theta)$

Alternating Optimization with Mirror Descent

For each step

1. $(\phi, \theta) \leftarrow \underset{\phi,\theta}{\text{argmax}}\ Q(\mathbf{z}, \phi, \theta)$ — $\mathbf{z}$ fixed
2. $\mathbf{z} \leftarrow \underset{\mathbf{z}}{\text{argmax}}\ Q(\mathbf{z}, \phi, \theta)$ — (ϕ, θ) fixed
   - $\mathbf{z} \setminus \mathbf{z}_{KB} \leftarrow$ argmax with respect to (ϕ, θ) — **TRIVIAL**
   - $\mathbf{z}_{KB} \leftarrow$ mirror descent — **HARD**

Scalable approach to optimize $\mathbf{z}_{KB}$

1. Relax discrete problem to continuous
2. Optimize relaxed problem with stochastic gradient descent
3. Round relaxed $\mathbf{z}$ to recover final assignment

MAP inference - $Q(\mathbf{z}, \phi, \theta)$

Alternating Optimization with Mirror Descent

For each step

1. $(\phi, \theta) \leftarrow \operatorname{argmax}_{\phi,\theta} Q(\mathbf{z}, \phi, \theta)$ — $\mathbf{z}$ fixed
2. $\mathbf{z} \leftarrow \operatorname{argmax}_{\mathbf{z}} Q(\mathbf{z}, \phi, \theta)$ — (ϕ, θ) fixed
   - $\mathbf{z} \setminus \mathbf{z}_{KB} \leftarrow$ argmax with respect to (ϕ, θ) — **TRIVIAL**
   - $\mathbf{z}_{KB} \leftarrow$ mirror descent — **HARD**

Scalable approach to optimize $\mathbf{z}_{KB}$

1. Relax discrete problem to continuous
2. Optimize relaxed problem with stochastic gradient descent
3. Round relaxed $\mathbf{z}$ to recover final assignment

Scalable $\mathbf{z}_{KB}$ inference

$$\underset{\mathbf{z}}{\text{argmax}} \sum_{i}^{N} \sum_{t}^{T} z_{it} \log \phi_t(w_i)\theta_{d_i}(t) + \sum_{k}^{L} \sum_{g \in G(\psi_k)} \lambda_k \mathbb{1}_g(\mathbf{z})$$

1. Continuous relaxation
   - $z_i = t$
   - Represent indicator function $\mathbb{1}_g(\mathbf{z})$ as *polynomial* in z_{it}
   - Can calculate ∇Q
2. Stochastic gradient - sample a term from objective function Q
   - Logic: single ground formula g
   - LDA: single corpus index i
3. Entropic Mirror Descent (Beck & Teboulle, 2003)

$$z_{it} \leftarrow \frac{z_{it} \exp\left(\eta \nabla_{z_{it}} f\right)}{\sum_{t'} z_{it'} \exp\left(\eta \nabla_{z_{it'}} f\right)}$$

4. Recover discrete $\mathbf{z}$: $z_i = \underset{t}{\text{argmax}}\, z_{it}$ for $i = 1, \ldots, N$

Scalable $\mathbf{z}_{KB}$ inference

$$\operatorname*{argmax}_{\mathbf{z}} \sum_{i}^{N} \sum_{t}^{T} z_{it} \log \phi_t(w_i)\theta_{d_i}(t) + \sum_{k}^{L} \sum_{g \in G(\psi_k)} \lambda_k \mathbb{1}_g(\mathbf{z})$$

1. Continuous relaxation
   - $z_i = t \rightarrow z_{it} \in \{0, 1\}$
   - Represent indicator function $\mathbb{1}_g(\mathbf{z})$ as *polynomial* in z_{it}
   - Can calculate ∇Q
2. Stochastic gradient - sample a term from objective function Q
   - Logic: single ground formula g
   - LDA: single corpus index i
3. Entropic Mirror Descent (Beck & Teboulle, 2003)

$$z_{it} \leftarrow \frac{z_{it} \exp\left(\eta \nabla_{z_{it}} f\right)}{\sum_{t'} z_{it'} \exp\left(\eta \nabla_{z_{it'}} f\right)}$$

4. Recover discrete $\mathbf{z}$: $z_i = \operatorname*{argmax}_t z_{it}$ for $i = 1, \ldots, N$

Scalable $\mathbf{z}_{KB}$ inference

$$\underset{\mathbf{z}}{\text{argmax}} \sum_{i}^{N} \sum_{t}^{T} z_{it} \log \phi_t(w_i)\theta_{d_i}(t) + \sum_{k}^{L} \sum_{g \in G(\psi_k)} \lambda_k \mathbb{1}_g(\mathbf{z})$$

1. Continuous relaxation
   - $z_i = t \rightarrow z_{it} \in \{0, 1\} \rightarrow z_{it} \in [0, 1]$
   - Represent indicator function $\mathbb{1}_g(\mathbf{z})$ as *polynomial* in z_{it}
   - Can calculate ∇Q
2. Stochastic gradient - sample a term from objective function Q
   - Logic: single ground formula g
   - LDA: single corpus index i
3. Entropic Mirror Descent (Beck & Teboulle, 2003)

$$z_{it} \leftarrow \frac{z_{it} \exp\left(\eta \nabla_{z_{it}} f\right)}{\sum_{t'} z_{it'} \exp\left(\eta \nabla_{z_{it'}} f\right)}$$

4. Recover discrete $\mathbf{z}$: $z_i = \underset{t}{\text{argmax}}\, z_{it}$ for $i = 1, \ldots, N$

Scalable $\mathbf{z}_{KB}$ inference

$$\underset{\mathbf{z}}{\text{argmax}} \sum_{i}^{N} \sum_{t}^{T} z_{it} \log \phi_t(w_i)\theta_{d_i}(t) + \sum_{k}^{L} \sum_{g \in G(\psi_k)} \lambda_k \mathbb{1}_g(\mathbf{z})$$

1. Continuous relaxation
   - $z_i = t \rightarrow z_{it} \in \{0, 1\} \rightarrow z_{it} \in [0, 1]$
   - Represent indicator function $\mathbb{1}_g(\mathbf{z})$ as *polynomial* in z_{it}
   - Can calculate ∇Q
2. Stochastic gradient - sample a term from objective function Q
   - Logic: single ground formula g
   - LDA: single corpus index i
3. Entropic Mirror Descent (Beck & Teboulle, 2003)

$$z_{it} \leftarrow \frac{z_{it} \exp\left(\eta \nabla_{z_{it}} f\right)}{\sum_{t'} z_{it'} \exp\left(\eta \nabla_{z_{it'}} f\right)}$$

4. Recover discrete $\mathbf{z}$: $z_i = \underset{t}{\text{argmax}}\, z_{it}$ for $i = 1, \ldots, N$

Scalable $\mathbf{z}_{KB}$ inference

$$\underset{\mathbf{z}}{\text{argmax}} \sum_{i}^{N} \sum_{t}^{T} z_{it} \log \phi_t(w_i)\theta_{d_i}(t) + \sum_{k}^{L} \sum_{g \in G(\psi_k)} \lambda_k \mathbb{1}_g(\mathbf{z})$$

1. Continuous relaxation
   - $z_i = t \rightarrow z_{it} \in \{0,1\} \rightarrow z_{it} \in [0,1]$
   - Represent indicator function $\mathbb{1}_g(\mathbf{z})$ as *polynomial* in z_{it}
   - Can calculate ∇Q
2. Stochastic gradient - sample a term from objective function Q
   - Logic: single ground formula g
   - LDA: single corpus index i
3. Entropic Mirror Descent (Beck & Teboulle, 2003)

$$z_{it} \leftarrow \frac{z_{it} \exp\left(\eta \nabla_{z_{it}} f\right)}{\sum_{t'} z_{it'} \exp\left(\eta \nabla_{z_{it'}} f\right)}$$

4. Recover discrete $\mathbf{z}$: $z_i = \underset{t}{\text{argmax}}\, z_{it}$ for $i = 1, \ldots, N$

Scalable $\mathbf{z}_{KB}$ inference

$$\operatorname*{argmax}_{\mathbf{z}} \sum_{i}^{N} \sum_{t}^{T} z_{it} \log \phi_t(w_i)\theta_{d_i}(t) + \sum_{k}^{L} \sum_{g \in G(\psi_k)} \lambda_k \mathbb{1}_g(\mathbf{z})$$

1. Continuous relaxation
   - $z_i = t \rightarrow z_{it} \in \{0,1\} \rightarrow z_{it} \in [0,1]$
   - Represent indicator function $\mathbb{1}_g(\mathbf{z})$ as *polynomial* in z_{it}
   - Can calculate ∇Q...but $G(\psi_k)$ may be very large
2. Stochastic gradient - sample a term from objective function Q
   - Logic: single ground formula g
   - LDA: single corpus index i
3. Entropic Mirror Descent (Beck & Teboulle, 2003)

$$z_{it} \leftarrow \frac{z_{it} \exp\left(\eta \nabla_{z_{it}} f\right)}{\sum_{t'} z_{it'} \exp\left(\eta \nabla_{z_{it'}} f\right)}$$

4. Recover discrete $\mathbf{z}$: $z_i = \operatorname*{argmax}_{t} z_{it}$ for $i = 1, \ldots, N$

Scalable $\mathbf{z}_{KB}$ inference

$$\underset{\mathbf{z}}{\operatorname{argmax}} \sum_{i}^{N} \sum_{t}^{T} z_{it} \log \phi_t(w_i)\theta_{d_i}(t) + \sum_{k}^{L} \sum_{g \in G(\psi_k)} \lambda_k \mathbb{1}_g(\mathbf{z})$$

1. Continuous relaxation
   - $z_i = t \rightarrow z_{it} \in \{0, 1\} \rightarrow z_{it} \in [0, 1]$
   - Represent indicator function $\mathbb{1}_g(\mathbf{z})$ as *polynomial* in z_{it}
   - Can calculate ∇Q...but $G(\psi_k)$ may be very large
2. Stochastic gradient - sample a term from objective function Q
   - Logic: single ground formula g
   - LDA: single corpus index i
3. Entropic Mirror Descent (Beck & Teboulle, 2003)

$$z_{it} \leftarrow \frac{z_{it} \exp\left(\eta \nabla_{z_{it}} f\right)}{\sum_{t'} z_{it'} \exp\left(\eta \nabla_{z_{it'}} f\right)}$$

4. Recover discrete $\mathbf{z}$: $z_i = \operatorname{argmax}_t z_{it}$ for $i = 1, \ldots, N$

Scalable $\mathbf{z}_{KB}$ inference

$$\underset{\mathbf{z}}{\text{argmax}} \sum_i^N \sum_t^T z_{it} \log \phi_t(w_i)\theta_{d_i}(t) + \sum_k^L \sum_{g \in G(\psi_k)} \lambda_k \mathbb{1}_g(\mathbf{z})$$

1. Continuous relaxation
   - $z_i = t \rightarrow z_{it} \in \{0,1\} \rightarrow z_{it} \in [0,1]$
   - Represent indicator function $\mathbb{1}_g(\mathbf{z})$ as *polynomial* in z_{it}
   - Can calculate ∇Q...but $G(\psi_k)$ may be very large
2. Stochastic gradient - sample a term from objective function Q
   - Logic: single ground formula g
   - LDA: single corpus index i
3. Entropic Mirror Descent (Beck & Teboulle, 2003)

$$z_{it} \leftarrow \frac{z_{it} \exp\left(\eta \nabla_{z_{it}} f\right)}{\sum_{t'} z_{it'} \exp\left(\eta \nabla_{z_{it'}} f\right)}$$

4. Recover discrete $\mathbf{z}$: $z_i = \underset{t}{\text{argmax}}\, z_{it}$ for $i = 1, \ldots, N$

Scalable $\mathbf{z}_{KB}$ inference

$$\underset{\mathbf{z}}{\operatorname{argmax}} \sum_{i}^{N} \sum_{t}^{T} z_{it} \log \phi_t(w_i)\theta_{d_i}(t) + \sum_{k}^{L} \sum_{g \in G(\psi_k)} \lambda_k \mathbb{1}_g(\mathbf{z})$$

1. Continuous relaxation
   - $z_i = t \rightarrow z_{it} \in \{0,1\} \rightarrow z_{it} \in [0,1]$
   - Represent indicator function $\mathbb{1}_g(\mathbf{z})$ as *polynomial* in z_{it}
   - Can calculate ∇Q...but $G(\psi_k)$ may be very large
2. Stochastic gradient - sample a term from objective function Q
   - Logic: single ground formula g
   - LDA: single corpus index i
3. Entropic Mirror Descent (Beck & Teboulle, 2003)

$$z_{it} \leftarrow \frac{z_{it} \exp\left(\eta \nabla_{z_{it}} f\right)}{\sum_{t'} z_{it'} \exp\left(\eta \nabla_{z_{it'}} f\right)}$$

4. Recover discrete $\mathbf{z}$: $z_i = \operatorname{argmax}_t z_{it}$ for $i = 1, \ldots, N$

Scalable $\mathbf{z}_{KB}$ inference

$$\underset{\mathbf{z}}{\text{argmax}} \sum_{i}^{N} \sum_{t}^{T} z_{it} \log \phi_t(w_i)\theta_{d_i}(t) + \sum_{k}^{L} \sum_{g \in G(\psi_k)} \lambda_k \mathbb{1}_g(\mathbf{z})$$

1. Continuous relaxation
   - $z_i = t \rightarrow z_{it} \in \{0, 1\} \rightarrow z_{it} \in [0, 1]$
   - Represent indicator function $\mathbb{1}_g(\mathbf{z})$ as *polynomial* in z_{it}
   - Can calculate ∇Q...but $G(\psi_k)$ may be very large
2. Stochastic gradient - sample a term from objective function Q
   - Logic: single ground formula g
   - LDA: single corpus index i
3. Entropic Mirror Descent (Beck & Teboulle, 2003)

$$z_{it} \leftarrow \frac{z_{it} \exp\left(\eta \nabla_{z_{it}} f\right)}{\sum_{t'} z_{it'} \exp\left(\eta \nabla_{z_{it'}} f\right)}$$

4. Recover discrete $\mathbf{z}$: $z_i = \underset{t}{\text{argmax}}\, z_{it}$ for $i = 1, \ldots, N$

Scalable $\mathbf{z}_{KB}$ inference

$$\underset{\mathbf{z}}{\operatorname{argmax}} \sum_{i}^{N} \sum_{t}^{T} z_{it} \log \phi_t(w_i)\theta_{d_i}(t) + \sum_{k}^{L} \sum_{g \in G(\psi_k)} \lambda_k \mathbb{1}_g(\mathbf{z})$$

1. Continuous relaxation
   - $z_i = t \rightarrow z_{it} \in \{0, 1\} \rightarrow z_{it} \in [0, 1]$
   - Represent indicator function $\mathbb{1}_g(\mathbf{z})$ as *polynomial* in z_{it}
   - Can calculate ∇Q...but $G(\psi_k)$ may be very large
2. Stochastic gradient - sample a term from objective function Q
   - Logic: single ground formula g
   - LDA: single corpus index i
3. Entropic Mirror Descent (Beck & Teboulle, 2003)

$$z_{it} \leftarrow \frac{z_{it} \exp\left(\eta \nabla_{z_{it}} f\right)}{\sum_{t'} z_{it'} \exp\left(\eta \nabla_{z_{it'}} f\right)}$$

4. Recover discrete $\mathbf{z}$: $z_i = \underset{t}{\operatorname{argmax}}\, z_{it}$ for $i = 1, \ldots, N$

Experimental questions

- Can logic *KBs* *generalize* to unseen documents?
- Can Alternating Optimization with Mirror Descent scale?
- Can we recover useful logic-influenced topics?

Experimental questions

- Can logic *KB*s *generalize* to unseen documents?
- Can Alternating Optimization with Mirror Descent scale?
- Can we recover useful logic-influenced topics?

Experimental questions

- Can logic *KB*s *generalize* to unseen documents?
- Can Alternating Optimization with Mirror Descent scale?
- Can we recover useful logic-influenced topics?

Generalization and scalability

- Example datasets and *KB*s (see paper)
- *k*-fold cross-validation
  - Training: do Fold·all MAP inference to estimate $(\hat{\phi}, \hat{\theta})$
  - Testing: use trainset $\hat{\phi}$ *only* to estimate testset $\hat{\mathbf{z}}$
  - Evaluation: testset objective function Q (unnormalized probability)

Generalization and scalability

- Example datasets and *KB*s (see paper)
- *k*-fold cross-validation
  - Training: do Fold·all MAP inference to estimate $(\hat{\phi}, \hat{\theta})$
  - Testing: use trainset $\hat{\phi}$ *only* to estimate testset $\hat{\mathbf{z}}$
  - Evaluation: testset objective function Q (unnormalized probability)

Generalization and scalability

- Example datasets and *KB*s (see paper)
- *k*-fold cross-validation
  - Training: do Fold·all MAP inference to estimate $(\hat{\phi}, \hat{\theta})$
  - Testing: use trainset $\hat{\phi}$ *only* to estimate testset $\hat{\mathbf{z}}$
  - Evaluation: testset objective function Q (unnormalized probability)

Generalization and scalability

- Example datasets and *KB*s (see paper)
- *k*-fold cross-validation
  - Training: do Fold·all MAP inference to estimate $(\hat{\phi}, \hat{\theta})$
  - Testing: use trainset $\hat{\phi}$ *only* to estimate testset $\hat{\mathbf{z}}$
  - Evaluation: testset objective function *Q* (unnormalized probability)

Generalization and scalability

- Example datasets and *KB*s (see paper)
- *k*-fold cross-validation
  - Training: do Fold·all MAP inference to estimate $(\hat{\phi}, \hat{\theta})$
  - Testing: use trainset $\hat{\phi}$ *only* to estimate testset $\hat{\mathbf{z}}$
  - Evaluation: testset objective function *Q* (unnormalized probability)

Generalization and scalability

- Example datasets and *KB*s (see paper)
- *k*-fold cross-validation
  - Training: do Fold·all MAP inference to estimate $(\hat{\phi}, \hat{\theta})$
  - Testing: use trainset $\hat{\phi}$ *only* to estimate testset $\hat{\mathbf{z}}$
  - Evaluation: testset objective function Q (unnormalized probability)

	Fold·all		Baselines		
	Mir	M+L	LDA	Alchemy	$\lvert \cup_k G(\psi_k) \rvert$
Synth	**9.86**	**11.13**	-2.18	-1.73	1.2×10^5
Comp	**2.40**	**2.45**	1.19	–	6.3×10^3
Con	**2.51**	**2.56**	1.09	–	2.9×10^3
Pol	**5.67**	–	**5.67**	–	9.6×10^8
HDG	**10.66**	–	3.59	–	2.3×10^8

Generalization and scalability

- Example datasets and *KB*s (see paper)
- *k*-fold cross-validation
  - Training: do Fold·all MAP inference to estimate $(\hat{\phi}, \hat{\theta})$
  - Testing: use trainset $\hat{\phi}$ *only* to estimate testset $\hat{\mathbf{z}}$
  - Evaluation: testset objective function Q (unnormalized probability)

	Fold·all		Baselines		
	Mir	M+L	LDA	Alchemy	$\lvert \cup_k G(\psi_k) \rvert$
Synth	**9.86**	**11.13**	−2.18	−1.73	1.2×10^5
Comp	**2.40**	**2.45**	1.19	–	6.3×10^3
Con	**2.51**	**2.56**	1.09	–	2.9×10^3
Pol	**5.67**	–	**5.67**	–	9.6×10^8
HDG	**10.66**	–	3.59	–	2.3×10^8

Generalization and scalability

- Example datasets and *KB*s (see paper)
- *k*-fold cross-validation
  - Training: do Fold·all MAP inference to estimate $(\hat{\phi}, \hat{\theta})$
  - Testing: use trainset $\hat{\phi}$ *only* to estimate testset $\hat{\mathbf{z}}$
  - Evaluation: testset objective function Q (unnormalized probability)

	Fold·all		Baselines		
	Mir	M+L	LDA	Alchemy	$\lvert \cup_k G(\psi_k) \rvert$
Synth	**9.86**	**11.13**	−2.18	−1.73	1.2×10^5
Comp	**2.40**	**2.45**	1.19	–	6.3×10^3
Con	**2.51**	**2.56**	1.09	–	2.9×10^3
Pol	**5.67**	–	**5.67**	–	9.6×10^8
HDG	**10.66**	–	3.59	–	2.3×10^8

Biological concept expansion
Human Development Genes (HDG)

Given “seed” terms for each concept
Do discover other related terms

Concept	Provided terms
Neural	neur dendro(cyte), glia, synapse, neural crest
Embryo	human embryonic stem cell, inner cell mass, pluripotent
Blood	hematopoietic, blood, endothel(ium)
Gastrulation	organizer, gastru(late)
Cardiac	heart, ventricle, auricle, aorta
Limb	limb, blastema, zeugopod, autopod, stylopod

Biological concept expansion

Human Development Genes (HDG)

Given "seed" terms for each concept
Do discover other related terms

Concept	Provided terms
Neural	neur dendro(cyte), glia, synapse, neural crest
Embryo	human embryonic stem cell, inner cell mass, pluripotent
Blood	hematopoietic, blood, endothel(ium)
Gastrulation	organizer, gastru(late)
Cardiac	heart, ventricle, auricle, aorta
Limb	limb, blastema, zeugopod, autopod, stylopod

Seed and n-gram rules

$Neural \rightarrow$ "synapse" $\rightarrow$ Topic 0

$$\mathtt{W}(i, \text{synapse}) \Rightarrow \mathtt{Z}(i, 0)$$

$Embryo \rightarrow$ "inner cell mass" $\rightarrow$ Topic 1

$$\mathtt{W}(i, \text{inner}) \wedge \mathtt{W}(i+1, \text{cell}) \wedge \mathtt{W}(i+2, \text{mass}) \Rightarrow \mathtt{Z}(i, 1)$$
$$\mathtt{W}(i-1, \text{inner}) \wedge \mathtt{W}(i, \text{cell}) \wedge \mathtt{W}(i+1, \text{mass}) \Rightarrow \mathtt{Z}(i, 1)$$
$$\mathtt{W}(i-2, \text{inner}) \wedge \mathtt{W}(i-1, \text{cell}) \wedge \mathtt{W}(i, \text{mass}) \Rightarrow \mathtt{Z}(i, 1)$$

Seed and *n*-gram rules

Neural $\rightarrow$ "synapse" $\rightarrow$ Topic 0

$$\mathtt{W}(i, \text{synapse}) \Rightarrow \mathtt{Z}(i, 0)$$

Embryo $\rightarrow$ "inner cell mass" $\rightarrow$ Topic 1

$$\mathtt{W}(i, \text{inner}) \wedge \mathtt{W}(i+1, \text{cell}) \wedge \mathtt{W}(i+2, \text{mass}) \Rightarrow \mathtt{Z}(i, 1)$$
$$\mathtt{W}(i-1, \text{inner}) \wedge \mathtt{W}(i, \text{cell}) \wedge \mathtt{W}(i+1, \text{mass}) \Rightarrow \mathtt{Z}(i, 1)$$
$$\mathtt{W}(i-2, \text{inner}) \wedge \mathtt{W}(i-1, \text{cell}) \wedge \mathtt{W}(i, \text{mass}) \Rightarrow \mathtt{Z}(i, 1)$$

Sentence rules

Sentence inclusion

- New *development* Topic 6
  {differentiation, maturation, develops, formation, differentiates}
- Development Topic 6 allows each seed Topic t in sentence

$$\texttt{Sentence}(i, i_1, \ldots, i_{S_k}) \wedge \neg \texttt{Z}(i_1, 6) \wedge \ldots \wedge \neg \texttt{Z}(i_{S_k}, 6) \Rightarrow \neg \texttt{Z}(i, 0)$$

Sentence exclusion

- New *disease* Topic 7
  {patient, disease, parasite, . . ., condition, disorder, symptom}
- Disease Topic 7 prevents each seed Topic t in sentence

$$\texttt{S}(i, s) \wedge \texttt{S}(j, s) \wedge \texttt{Z}(i, 7) \Rightarrow \neg \texttt{Z}(j, 0)$$

Sentence rules

Sentence inclusion

- New *development* Topic 6
  {differentiation, maturation, develops, formation, differentiates}
- Development Topic 6 allows each seed Topic t in sentence

$$\texttt{Sentence}(i, i_1, \ldots, i_{S_k}) \wedge \neg \texttt{Z}(i_1, 6) \wedge \ldots \wedge \neg \texttt{Z}(i_{S_k}, 6) \Rightarrow \neg \texttt{Z}(i, 0)$$

Sentence exclusion

- New *disease* Topic 7
  {patient, disease, parasite, ..., condition, disorder, symptom}
- Disease Topic 7 prevents each seed Topic t in sentence

$$\texttt{S}(i, s) \wedge \texttt{S}(j, s) \wedge \texttt{Z}(i, 7) \Rightarrow \neg \texttt{Z}(j, 0)$$

Accuracy at Top 50 threshold

(means over 10 randomized runs)

	Fold·all *KBs*				
	ALL	INCL	EXCL	SEED	LDA
Neural	**0.59**	**0.57**	**0.54**	**0.54**	0.31
Embryo	**0.24**	**0.24**	**0.23**	**0.23**	0.07
Blood	**0.46**	**0.47**	0.40	0.39	0.13
Gast.	**0.18**	**0.18**	**0.16**	**0.16**	0.00
Cardiac	**0.36**	**0.37**	**0.34**	**0.35**	0.08
Limb	**0.18**	**0.18**	**0.15**	**0.14**	0.09

Novel terms discovered for *neural*

{dendritic, forebrain, hindbrain, microglial, motoneurons, neuroblasts, neurogenesis, retinal}

Accuracy at Top 50 threshold

(means over 10 randomized runs)

	Fold·all *KBs*				
	ALL	INCL	EXCL	SEED	LDA
Neural	**0.59**	**0.57**	**0.54**	**0.54**	0.31
Embryo	**0.24**	**0.24**	**0.23**	**0.23**	0.07
Blood	**0.46**	**0.47**	0.40	0.39	0.13
Gast.	**0.18**	**0.18**	**0.16**	**0.16**	0.00
Cardiac	**0.36**	**0.37**	**0.34**	**0.35**	0.08
Limb	**0.18**	**0.18**	**0.15**	**0.14**	0.09

Novel terms discovered for *neural*

{dendritic, forebrain, hindbrain, microglial, motoneurons, neuroblasts, neurogenesis, retinal}

Accuracy at Top 50 threshold

(means over 10 randomized runs)

	Fold·all *KBs*				
	ALL	INCL	EXCL	SEED	LDA
Neural	**0.59**	**0.57**	**0.54**	**0.54**	0.31
Embryo	**0.24**	**0.24**	**0.23**	**0.23**	0.07
Blood	**0.46**	**0.47**	0.40	0.39	0.13
Gast.	**0.18**	**0.18**	**0.16**	**0.16**	0.00
Cardiac	**0.36**	**0.37**	**0.34**	**0.35**	0.08
Limb	**0.18**	**0.18**	**0.15**	**0.14**	0.09

Novel terms discovered for *neural*

{dendritic, forebrain, hindbrain, microglial, motoneurons, neuroblasts, neurogenesis, retinal}

Accuracy at Top 50 threshold

(means over 10 randomized runs)

	Fold·all *KB*s				
	ALL	INCL	EXCL	SEED	LDA
Neural	**0.59**	**0.57**	**0.54**	**0.54**	0.31
Embryo	**0.24**	**0.24**	**0.23**	**0.23**	0.07
Blood	**0.46**	**0.47**	0.40	0.39	0.13
Gast.	**0.18**	**0.18**	**0.16**	**0.16**	0.00
Cardiac	**0.36**	**0.37**	**0.34**	**0.35**	0.08
Limb	**0.18**	**0.18**	**0.15**	**0.14**	0.09

Novel terms discovered for *neural*

{dendritic, forebrain, hindbrain, microglial, motoneurons, neuroblasts, neurogenesis, retinal}

Conclusion

- Fold·all topic modeling with domain knowledge
  - user-specified constraints
  - side information
- Scalable inference
- Experimental results
  - Logic *KB*s generalize to unseen documents
  - Inference scales to realistic datasets and *KB*s
  - Topics reflect domain knowledge in interesting ways

Acknowledgements

- This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. LLNL-PRES-489752
- Additional support
  - NSF IIS-0953219
  - AFOSR FA9550-09-1-0313
  - NIH/NLM R01 LM07050
- HDG experiments: Ron Stewart (Thomson Lab, UW–Madison)

Source code

https://github.com/davidandrzej/LogicLDA

MAP inference

Find most probable $(\mathbf{z}, \phi, \theta)$

$$Q(\mathbf{z}, \phi, \theta) = \sum_{t}^{T} \log p(\phi_t | \beta) + \sum_{j}^{D} \log p(\theta_j | \alpha)$$
$$+ \sum_{i}^{N} \log \phi_{z_i}(w_i) \theta_{d_i}(z_i) + \sum_{k}^{L} \sum_{g \in G(\psi_k)} \lambda_k \mathbb{1}_g(\mathbf{z}, \mathbf{w}, \mathbf{d}, \mathbf{o})$$

- LDA terms
- Logic terms

MAP inference

Find most probable $(\mathbf{z}, \phi, \theta)$

$$Q(\mathbf{z}, \phi, \theta) = \sum_t^T \log p(\phi_t|\beta) + \sum_j^D \log p(\theta_j|\alpha) + \sum_i^N \log \phi_{z_i}(w_i)\theta_{d_i}(z_i) + \sum_k^L \sum_{g \in G(\psi_k)} \lambda_k \mathbb{1}_g(\mathbf{z}, \mathbf{w}, \mathbf{d}, \mathbf{o})$$

- LDA terms
- Logic terms

MAP inference

Find most probable $(\mathbf{z}, \phi, \theta)$

$$Q(\mathbf{z}, \phi, \theta) = \sum_{t}^{T} \log p(\phi_t|\beta) + \sum_{j}^{D} \log p(\theta_j|\alpha) + \sum_{i}^{N} \log \phi_{z_i}(w_i)\theta_{d_i}(z_i) + \sum_{k}^{L} \sum_{g \in G(\psi_k)} \lambda_k \mathbb{1}_g(\mathbf{z}, \mathbf{w}, \mathbf{d}, \mathbf{o})$$

- LDA terms
- Logic terms

Ignore trivial rule groundings

Shavlik & Natarajan, 2009

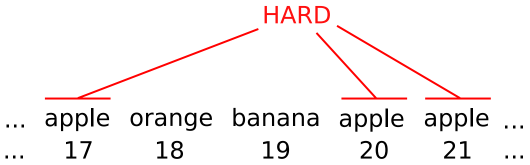
λ_k	$\forall$	ψ_k
5	i	$\mathtt{W}(i, apple) \Rightarrow \mathtt{Z}(i, 3)$

...	apple	orange	banana	apple	apple	...
...	17	18	19	20	21	...

Ignore trivial rule groundings

Shavlik & Natarajan, 2009

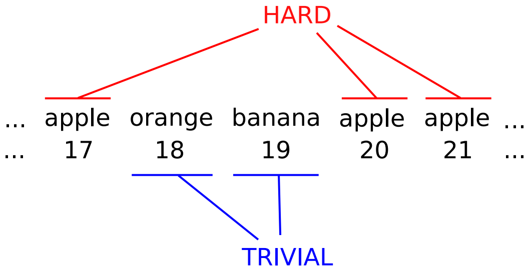
λ_k	$\forall$	ψ_k
5	i	$\mathtt{W}(i, apple) \Rightarrow \mathtt{Z}(i, 3)$



Ignore trivial rule groundings

Shavlik & Natarajan, 2009

λ_k	$\forall$	ψ_k
5	i	$\mathtt{W}(i, apple) \Rightarrow \mathtt{Z}(i, 3)$



Represent $\mathbb{1}$ as a polynomial

$g = \mathtt{Z}(i,1) \vee \neg\mathtt{Z}(j,2)$, and $t \in \{1,2,3\}$

1. Take complement $\neg g$ — $\neg\mathtt{Z}(i,1) \wedge \mathtt{Z}(j,2)$
2. Remove negations $(\neg g)_+$ — $(\mathtt{Z}(i,2) \vee \mathtt{Z}(i,3)) \wedge \mathtt{Z}(j,2)$
3. Numeric $z_{it} \in \{0,1\}$ — $(z_{i2} + z_{i3})z_{j2}$
4. Polynomial $\mathbb{1}_g(\mathbf{z})$ — $1 - (z_{i2} + z_{i3})z_{j2}$
5. Relax discrete z_{it} — $z_{it} \in \{0,1\} \rightarrow z_{it} \in [0,1]$

$$\mathbb{1}_g(\mathbf{z}) = 1 - \prod_{g_i \neq \emptyset} \left(\sum_{\mathtt{Z}(i,t) \in (\neg g_i)_+} z_{it} \right)$$

Represent $\mathbb{1}$ as a polynomial

$g = \mathtt{z}(i,1) \vee \neg\mathtt{z}(j,2)$, and $t \in \{1,2,3\}$

1. Take complement $\neg g$ — $\neg\mathtt{z}(i,1) \wedge \mathtt{z}(j,2)$
2. Remove negations $(\neg g)_+$ — $(\mathtt{z}(i,2) \vee \mathtt{z}(i,3)) \wedge \mathtt{z}(j,2)$
3. Numeric $z_{it} \in \{0,1\}$ — $(z_{i2} + z_{i3})z_{j2}$
4. Polynomial $\mathbb{1}_g(\mathbf{z})$ — $1 - (z_{i2} + z_{i3})z_{j2}$
5. Relax discrete z_{it} — $z_{it} \in \{0,1\} \to z_{it} \in [0,1]$

$$\mathbb{1}_g(\mathbf{z}) = 1 - \prod_{g_i \neq \emptyset} \left(\sum_{\mathtt{z}(i,t) \in (\neg g_i)_+} z_{it} \right)$$

Represent $\mathbb{1}$ as a polynomial

$g = \mathtt{Z}(i, 1) \vee \neg\mathtt{Z}(j, 2)$, and $t \in \{1, 2, 3\}$

1. Take complement $\neg g$ — $\neg\mathtt{Z}(i, 1) \wedge \mathtt{Z}(j, 2)$
2. Remove negations $(\neg g)_+$ — $(\mathtt{Z}(i, 2) \vee \mathtt{Z}(i, 3)) \wedge \mathtt{Z}(j, 2)$
3. Numeric $z_{it} \in \{0, 1\}$ — $(z_{i2} + z_{i3})z_{j2}$
4. Polynomial $\mathbb{1}_g(\mathbf{z})$ — $1 - (z_{i2} + z_{i3})z_{j2}$
5. Relax discrete z_{it} — $z_{it} \in \{0, 1\} \rightarrow z_{it} \in [0, 1]$

$$\mathbb{1}_g(\mathbf{z}) = 1 - \prod_{g_i \neq \emptyset} \left(\sum_{\mathtt{Z}(i,t) \in (\neg g_i)_+} z_{it} \right)$$

Represent $\mathbb{1}$ as a polynomial

$g = \mathtt{z}(i,1) \vee \neg\mathtt{z}(j,2)$, and $t \in \{1,2,3\}$

1. Take complement $\neg g$ — $\neg\mathtt{z}(i,1) \wedge \mathtt{z}(j,2)$
2. Remove negations $(\neg g)_+$ — $(\mathtt{z}(i,2) \vee \mathtt{z}(i,3)) \wedge \mathtt{z}(j,2)$
3. Numeric $z_{it} \in \{0,1\}$ — $(z_{i2} + z_{i3})z_{j2}$
4. Polynomial $\mathbb{1}_g(\mathbf{z})$ — $1 - (z_{i2} + z_{i3})z_{j2}$
5. Relax discrete z_{it} — $z_{it} \in \{0,1\} \to z_{it} \in [0,1]$

$$\mathbb{1}_g(\mathbf{z}) = 1 - \prod_{g_i \neq \emptyset} \left(\sum_{\mathtt{z}(i,t) \in (\neg g_i)_+} z_{it} \right)$$

Represent $\mathbb{1}$ as a polynomial

$g = \mathtt{z}(i, 1) \vee \neg \mathtt{z}(j, 2)$, and $t \in \{1, 2, 3\}$

1. Take complement $\neg g$ — $\neg \mathtt{z}(i, 1) \wedge \mathtt{z}(j, 2)$
2. Remove negations $(\neg g)_+$ — $(\mathtt{z}(i, 2) \vee \mathtt{z}(i, 3)) \wedge \mathtt{z}(j, 2)$
3. Numeric $z_{it} \in \{0, 1\}$ — $(z_{i2} + z_{i3})z_{j2}$
4. Polynomial $\mathbb{1}_g(\mathbf{z})$ — $1 - (z_{i2} + z_{i3})z_{j2}$
5. Relax discrete z_{it} — $z_{it} \in \{0, 1\} \rightarrow z_{it} \in [0, 1]$

$$\mathbb{1}_g(\mathbf{z}) = 1 - \prod_{g_i \neq \emptyset} \left(\sum_{\mathtt{z}(i,t) \in (\neg g_i)_+} z_{it} \right)$$

Represent $\mathbb{1}$ as a polynomial

$g = \mathbb{Z}(i,1) \vee \neg\mathbb{Z}(j,2)$, and $t \in \{1,2,3\}$

1. Take complement $\neg g$ — $\neg\mathbb{Z}(i,1) \wedge \mathbb{Z}(j,2)$
2. Remove negations $(\neg g)_+$ — $(\mathbb{Z}(i,2) \vee \mathbb{Z}(i,3)) \wedge \mathbb{Z}(j,2)$
3. Numeric $z_{it} \in \{0,1\}$ — $(z_{i2} + z_{i3})z_{j2}$
4. Polynomial $\mathbb{1}_g(\mathbf{z})$ — $1 - (z_{i2} + z_{i3})z_{j2}$
5. Relax discrete z_{it} — $z_{it} \in \{0,1\} \to z_{it} \in [0,1]$

$$\mathbb{1}_g(\mathbf{z}) = 1 - \prod_{g_i \neq \emptyset} \left(\sum_{\mathbb{Z}(i,t) \in (\neg g_i)_+} z_{it} \right)$$

Represent $\mathbb{1}$ as a polynomial

$g = \mathtt{Z}(i, 1) \vee \neg \mathtt{Z}(j, 2)$, and $t \in \{1, 2, 3\}$

1. Take complement $\neg g$ — $\neg \mathtt{Z}(i, 1) \wedge \mathtt{Z}(j, 2)$
2. Remove negations $(\neg g)_+$ — $(\mathtt{Z}(i, 2) \vee \mathtt{Z}(i, 3)) \wedge \mathtt{Z}(j, 2)$
3. Numeric $z_{it} \in \{0, 1\}$ — $(z_{i2} + z_{i3}) z_{j2}$
4. Polynomial $\mathbb{1}_g(\mathbf{z})$ — $1 - (z_{i2} + z_{i3}) z_{j2}$
5. Relax discrete z_{it} — $z_{it} \in \{0, 1\} \rightarrow z_{it} \in [0, 1]$

$$\mathbb{1}_g(\mathbf{z}) = 1 - \prod_{g_i \neq \emptyset} \left(\sum_{\mathtt{Z}(i,t) \in (\neg g_i)_+} z_{it} \right)$$

Standard LDA: *Neural* concept

Do standard LDA, then find topics containing seed terms in Top 50

brain system nervous neurons neuronal central development 's neural human gene disease function cortex spinal disorders developing motor cerebral glial peripheral cortical cord disorder astrocytes nerve neurological regions suggest schizophrenia including syndrome neurodegenerative mental involved retardation behavior cerebellum migration behavioral abnormal cerebellar found precursor results amyloid hippocampus sclerosis neurotrophic present

Standard LDA: *Neural* concept

Do standard LDA, then find topics containing seed terms in Top 50

brain system nervous neurons neuronal central development 's neural human gene disease function cortex spinal disorders developing motor cerebral glial peripheral cortical cord disorder astrocytes nerve neurological regions suggest schizophrenia including syndrome neurodegenerative mental involved retardation behavior cerebellum migration behavioral abnormal cerebellar found precursor results amyloid hippocampus sclerosis neurotrophic present

Fold·all can encode existing LDA variants

Example: Hidden Topic Markov Model (HTMM) - Gruber et al, 2007

- Each sentence uses only *one* topic
- Topic transitions possible between sentences with probability ϵ

FOL encoding of HTMM

λ_k	$\forall$	ψ_k
∞	i, j, s, t	$\texttt{S}(i, s) \wedge \texttt{S}(j, s) \wedge \texttt{Z}(i, t) \Rightarrow \texttt{Z}(j, t)$

Fold·all can encode existing LDA variants

Example: Hidden Topic Markov Model (HTMM) - Gruber et al, 2007

- Each sentence uses only *one* topic
- Topic transitions possible between sentences with probability ϵ

FOL encoding of HTMM

λ_k	$\forall$	ψ_k
∞	i, j, s, t	$\mathtt{S}(i, s) \wedge \mathtt{S}(j, s) \wedge \mathtt{Z}(i, t) \Rightarrow \mathtt{Z}(j, t)$
$-\log \epsilon$	i, s, t	$\mathtt{S}(i, s) \wedge \neg \mathtt{S}(i+1, s) \wedge \mathtt{Z}(i, t) \Rightarrow \mathtt{Z}(i+1, t)$