The Security of Latent Dirichlet Allocation

Shike Mei and Jerry Zhu*

Department of Computer Sciences
University of Wisconsin-Madison

AISTATS 2015

My topics are buggy!

My topics are buggy!

- !@#$ in top words

My topics are buggy!

- !@#$ in top words
- duplicate documents?

My topics are buggy!

- !@#$ in top words
- duplicate documents?
- clean up, save the day

But what if it is intentional?

But what if it is intentional?

- real people use LDA

But what if it is intentional?

- real people use LDA
- attacker wants them to see !@#$

But what if it is intentional?

- real people use LDA
- attacker wants them to see !@#$
    - to influence their decision

But what if it is intentional?

- real people use LDA
- attacker wants them to see !@#$
  - to influence their decision
  - to profit financially and politically

But what if it is intentional?

- real people use LDA
- attacker wants them to see !@#$
  - to influence their decision
  - to profit financially and politically
- data poisoning attack

But what if it is intentional?

- real people use LDA
- attacker wants them to see !@#$
  - to influence their decision
  - to profit financially and politically
- data poisoning attack
  - attacker can modify the corpus

But what if it is intentional?

- real people use LDA
- attacker wants them to see !@#$
  - to influence their decision
  - to profit financially and politically
- data poisoning attack
  - attacker can modify the corpus
  - but not the LDA code

But what if it is intentional?

- real people use LDA
- attacker wants them to see !@#$
  - to influence their decision
  - to profit financially and politically
- data poisoning attack
  - attacker can modify the corpus
  - but not the LDA code
  - prefers small modifications

But what if it is intentional?

- real people use LDA
- attacker wants them to see !@#$
  - to influence their decision
  - to profit financially and politically
- data poisoning attack
  - attacker can modify the corpus
  - but not the LDA code
  - prefers small modifications
  - user runs vanilla LDA on poisoned corpus, sees planted topics

But what if it is intentional?

- real people use LDA
- attacker wants them to see !@#$
  - to influence their decision
  - to profit financially and politically
- data poisoning attack
  - attacker can modify the corpus
  - but not the LDA code
  - prefers small modifications
  - user runs vanilla LDA on poisoned corpus, sees planted topics
- our paper shows how the attacker may do so optimally

Latent Dirichlet allocation

$$\begin{aligned}
\color{red}{\psi_1 \ldots \psi_k} &\sim Dir(\beta) \\
\theta_1 \ldots \theta_n &\sim Dir(\alpha) \\
z_{di} &\sim \theta_d \\
w_{di} &\sim \psi_{z_{di}}
\end{aligned}$$

Using LDA

- The user: (without attack)

Using LDA

- The user: (without attack)
  - receives corpus W

Using LDA

- The user: (without attack)
  - receives corpus W
  - runs off-the-shelf LDA and gets $\hat{\psi} \mid W$

Using LDA

- The user: (without attack)
  - receives corpus W
  - runs off-the-shelf LDA and gets $\hat{\psi} \mid W$
  - under the hood: $\hat{\psi} = \operatorname{argmax} p(\psi \mid W, \alpha, \beta)$, variational or MCMC

Using LDA

- The user: (without attack)
  - receives corpus W
  - runs off-the-shelf LDA and gets $\hat{\psi} \mid W$
  - under the hood: $\hat{\psi} = \operatorname{argmax} p(\psi \mid W, \alpha, \beta)$, variational or MCMC
  - stares at top words in $\hat{\psi}_1 \ldots \hat{\psi}_k$

Data poisoning attack on LDA

- The attacker:

Data poisoning attack on LDA

- The attacker:
  - has target topics ψ^* in mind

Data poisoning attack on LDA

- The attacker:
  - has target topics ψ^* in mind
  - example: $\psi^*_{1,\text{!@\#\$}} \leftarrow 9\max(\hat{\psi}_{1,1} \ldots \hat{\psi}_{1,v})$, renormalize ψ^*_1

Data poisoning attack on LDA

- The attacker:
  - has target topics ψ^* in mind
  - example: $\psi^*_{1,\text{!@\#\$}} \leftarrow 9\max(\hat{\psi}_{1,1} \ldots \hat{\psi}_{1,v})$, renormalize ψ^*_1
  - changes W to $\tilde{W}$ so that $\left(\hat{\psi} \mid \tilde{W}\right) \approx \psi^*$

Data poisoning attack on LDA

- The attacker:
  - has target topics ψ^* in mind
  - example: $\psi^*_{1,\text{!@\#\$}} \leftarrow 9\max(\hat{\psi}_{1,1} \ldots \hat{\psi}_{1,v})$, renormalize ψ^*_1
  - changes W to $\tilde{W}$ so that $\left(\hat{\psi} \mid \tilde{W}\right) \approx \psi^*$
  - gives $\tilde{W}$ to the user

Data poisoning attack on LDA

- The attacker:
  - has target topics ψ^* in mind
  - example: $\psi^*_{1,\text{!@\#\$}} \leftarrow 9\max(\hat{\psi}_{1,1} \ldots \hat{\psi}_{1,v})$, renormalize ψ^*_1
  - changes W to $\tilde{W}$ so that $\left(\hat{\psi} \mid \tilde{W}\right) \approx \psi^*$
  - gives $\tilde{W}$ to the user
- The user:

Data poisoning attack on LDA

- The attacker:
  - has target topics ψ^* in mind
  - example: $\psi^*_{1,\text{!@\#\$}} \leftarrow 9\max(\hat{\psi}_{1,1} \ldots \hat{\psi}_{1,v})$, renormalize ψ^*_1
  - changes W to $\tilde{W}$ so that $\left(\hat{\psi} \mid \tilde{W}\right) \approx \psi^*$
  - gives $\tilde{W}$ to the user
- The user:
  - runs off-the-shelf LDA and gets $\hat{\psi} \mid \tilde{W}$

Data poisoning attack on LDA

- The attacker:
  - has target topics ψ^* in mind
  - example: $\psi^*_{1,!@\#\$} \leftarrow 9\max(\hat{\psi}_{1,1} \dots \hat{\psi}_{1,v})$, renormalize ψ^*_1
  - changes W to $\tilde{W}$ so that $\left(\hat{\psi} \mid \tilde{W}\right) \approx \psi^*$
  - gives $\tilde{W}$ to the user
- The user:
  - runs off-the-shelf LDA and gets $\hat{\psi} \mid \tilde{W}$
  - stares at top words in $\hat{\psi}_1 \dots \hat{\psi}_k$ and sees !@#$ in topic 1

Formulating the attack

$$\begin{aligned} \min_{\tilde{W},\hat{\psi}} \quad & \|\psi^* - \hat{\psi}\|_\epsilon^2 \\ \text{s.t.} \quad & \hat{\psi} = \operatorname{argmax} p(\psi \mid \tilde{W}, \alpha, \beta) \\ & \tilde{W} \geq 0 \\ & \|\tilde{W} - W\|_1 \leq L \end{aligned}$$

$\tilde{W}$: doc-word count matrix, relaxed to real
L: attack budget

How come there is optimization in the constraint?

$$
\begin{aligned}
\min_{\tilde{W},\hat{\psi}} \quad & \|\psi^* - \hat{\psi}\|_\epsilon^2 \\
\text{s.t.} \quad & \hat{\psi} = \operatorname{argmax} p(\psi \mid \tilde{W}, \alpha, \beta) \\
& \tilde{W} \geq 0 \\
& \|\tilde{W} - W\|_1 \leq L
\end{aligned}
$$

- bilevel optimization (Stackelberg game)

How come there is optimization in the constraint?

$$
\begin{aligned}
\min_{\tilde{W},\hat{\psi}} \quad & \|\psi^* - \hat{\psi}\|_\epsilon^2 \\
\text{s.t.} \quad & \hat{\psi} = \operatorname{argmax} p(\psi \mid \tilde{W}, \alpha, \beta) \\
& \tilde{W} \geq 0 \\
& \|\tilde{W} - W\|_1 \leq L
\end{aligned}
$$

- bilevel optimization (Stackelberg game)
- hard

KKT conditions to the rescue

Replace the lower problem ...

$$
\begin{aligned}
\min_{\tilde{W},\hat{\psi}} \quad & \|\psi^* - \hat{\psi}\|_\epsilon^2 \\
\text{s.t.} \quad & \hat{\psi} = \operatorname{argmax} p(\psi \mid \tilde{W}, \alpha, \beta) \\
& \tilde{W} \geq 0 \\
& \|\tilde{W} - W\|_1 \leq L
\end{aligned}
$$

KKT conditions to the rescue

... with its KKT conditions (variational approximation)

$$\min_{\tilde{W},\hat{\psi}} \quad \|\psi^* - \hat{\psi}\|_\epsilon^2$$

$$\text{s.t.} \quad \eta_{kv} - \beta - \sum_d \phi_{dvk} m_{dv} = 0$$

$$\gamma_{dk} - \alpha - \sum_v \phi_{dvk} m_{dv} = 0$$

$$\phi_{dvk} - \frac{\exp(\Psi(\gamma_{dk}) + (\Psi(\eta_{kv}) - \Psi(\sum_{v'} \eta_{kv})))}{\sum_k \exp(\Psi(\gamma_{dk}) + (\Psi(\eta_{kv}) - \Psi(\sum_{v'} \eta_{kv'})))} = 0$$

$$\tilde{W} \geq 0$$

$$\|\tilde{W} - W\|_1 \leq L$$

- nonlinear constraints, but single level optimization

KKT conditions to the rescue

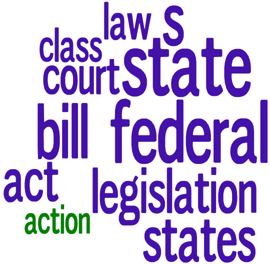
... with its KKT conditions (variational approximation)

$$
\begin{aligned}
\min_{\tilde{W},\hat{\psi}} \quad & \|\psi^* - \hat{\psi}\|_\epsilon^2 \\
\text{s.t.} \quad & \eta_{kv} - \beta - \sum_d \phi_{dvk} m_{dv} = 0 \\
& \gamma_{dk} - \alpha - \sum_v \phi_{dvk} m_{dv} = 0 \\
& \phi_{dvk} - \frac{\exp(\Psi(\gamma_{dk}) + (\Psi(\eta_{kv}) - \Psi(\sum_{v'} \eta_{kv})))}{\sum_k \exp(\Psi(\gamma_{dk}) + (\Psi(\eta_{kv}) - \Psi(\sum_{v'} \eta_{kv'})))} = 0 \\
& \tilde{W} \geq 0 \\
& \|\tilde{W} - W\|_1 \leq L
\end{aligned}
$$

- nonlinear constraints, but single level optimization
- gradient descent

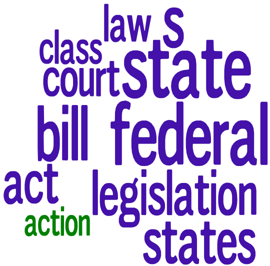
Let's pretend to be the attacker

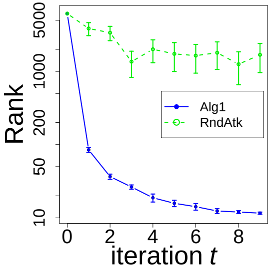
Promote "marijuana" to top-10 in this topic:



Let's pretend to be the attacker

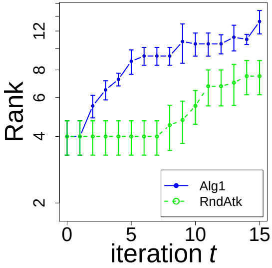
Promote "marijuana" to top-10 in this topic:





Can demote words, too

Can demote words, too



Isn't word-based attack easy to detect?

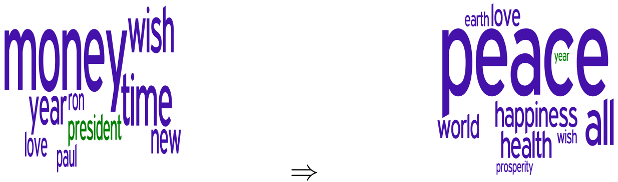
Isn't word-based attack easy to detect?

Can attack by adding / removing sentences

Isn't word-based attack easy to detect?

Can attack by adding / removing sentences

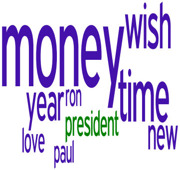
goal: move "president" to another topic



Isn't word-based attack easy to detect?

Can attack by adding / removing sentences

goal: move "president" to another topic



$\Rightarrow$

after attack

What can I do to protect LDA?

What can I do to protect LDA?

- protect your corpus

What can I do to protect LDA?

- protect your corpus
- inspect docs with large “suspicious topic” proportion $\theta_{d,k}$

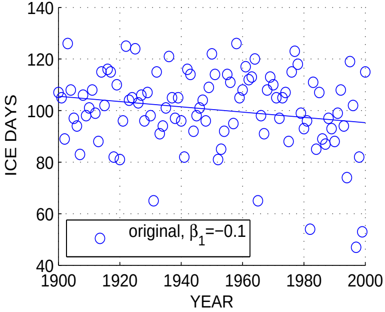
What can I do to protect LDA?

- protect your corpus
- inspect docs with large "suspicious topic" proportion $\theta_{d,k}$
- adversarial classification [Li Vorobeychik AISTATS'15]

I don't care about LDA

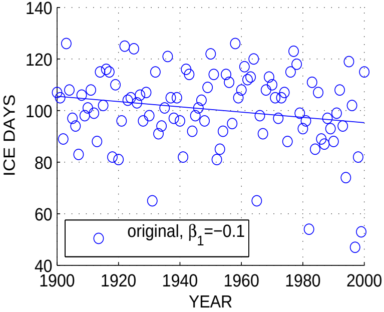
I don't care about LDA

Data poisoning attack can happen to any learner



I don't care about LDA

Data poisoning attack can happen to any learner

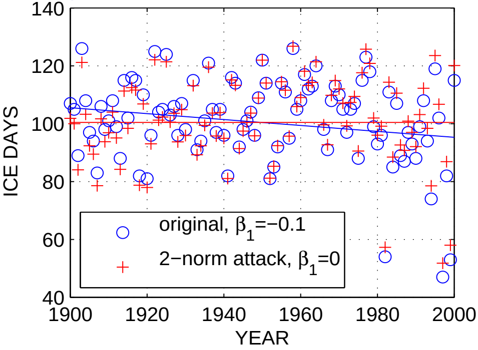


$$\min_{\mathbf{y}\in\mathbb{R}^n,\hat{\beta}\in\mathbb{R}^2} \quad \|\mathbf{y}-\mathbf{y}_0\|_p$$ small modifications

s.t. $$\hat{\beta} = \min_{\beta\in\mathbb{R}^2} \|\mathbf{y}-X\beta\|^2$$

$$\hat{\beta}_1 \geq 0$$ attack goal: nonnegative slope

Attacking linear regression, 2-norm

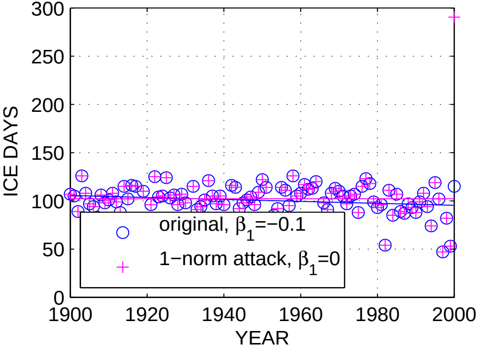


$$\min_{\mathbf{y}\in\mathbb{R}^n,\hat{\beta}\in\mathbb{R}^2} \quad \|\mathbf{y}-\mathbf{y}_0\|_{2}$$

$$\text{s.t.} \quad \hat{\beta} = \min_{\beta\in\mathbb{R}^2} \|\mathbf{y}-X\beta\|^2$$

$$\hat{\beta}_1 \geq 0$$

Attacking linear regression, 1-norm



$$
\begin{aligned}
\min_{\mathbf{y}\in\mathbb{R}^n, \hat\beta\in\mathbb{R}^2} \quad & \|\mathbf{y}-\mathbf{y}_0\|_{\color{red}{1}} \\
\text{s.t.} \quad & \hat\beta = \min_{\beta\in\mathbb{R}^2} \|\mathbf{y}-X\beta\|^2 \\
& \hat\beta_1 \geq 0
\end{aligned}
$$

Data poisoning attack on any learner

$$\min_{D,\hat{\theta}} \quad d_1(\hat{\theta}, \theta^*) + d_2(D, D_0) \quad \text{attacker's problem}$$

$$\text{s.t.} \quad \hat{\theta} = \operatorname*{argmin}_{\theta \in \Theta} \frac{1}{|D|} \sum_{z_i \in D} \ell(z_i, \theta) + \Omega(\theta) \quad \text{learner's problem}$$

Attack linear regression, logistic regression, SVM [Mei Zhu AAAI'15]

I don't care about attacks, either

I don't care about attacks, either

How about education?

$$\min_{D,\hat{\theta}} \quad d_1(\hat{\theta}, \theta^*) + \|D\|_0 \quad \text{teacher finding optimal lesson } D$$

$$\text{s.t.} \quad \hat{\theta} = \operatorname*{argmin}_{\theta \in \Theta} \frac{1}{|D|} \sum_{z_i \in D} \ell(z_i, \theta) + \Omega(\theta) \quad \text{student's cognitive model}$$

I don't care about attacks, either

How about education?

$$\min_{D,\hat{\theta}} \quad d_1(\hat{\theta}, \theta^*) + \|D\|_0$$ teacher finding optimal lesson D

$$\text{s.t.} \quad \hat{\theta} = \operatorname*{argmin}_{\theta \in \Theta} \frac{1}{|D|} \sum_{z_i \in D} \ell(z_i, \theta) + \Omega(\theta)$$ student's cognitive model

Human categorization [PZKB NIPS'14, Zhu AAAI'15]

human trained on	human test accuracy
optimal lesson D	72.5%
iid	69.8%

(statistically significant)

This whole thing doesn't look like machine learning

This whole thing doesn't look like machine learning

It is not.

This whole thing doesn't look like machine learning

It is not.
We call it machine teaching.

Example one

- The student runs a linear SVM:

$$\text{Given a training set with } n \text{ items } \mathbf{x}_i \in \mathbb{R}^d, y_i \in \{-1, 1\}$$

$$\text{student learns } \mathbf{w} \in \mathbb{R}^d$$

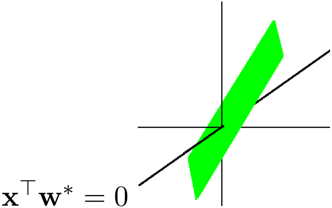
Example one

- The student runs a linear SVM:

$$\text{Given a training set with } n \text{ items } \mathbf{x}_i \in \mathbb{R}^d, y_i \in \{-1, 1\}$$

$$\text{student learns } \mathbf{w} \in \mathbb{R}^d$$

- The teacher wants to teach a target $\mathbf{w}^*$



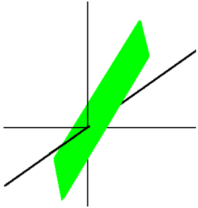
Example one

- The student runs a linear SVM:

$$\text{Given a training set with } n \text{ items } \mathbf{x}_i \in \mathbb{R}^d, y_i \in \{-1, 1\}$$

$$\text{student learns } \mathbf{w} \in \mathbb{R}^d$$

- The teacher wants to teach a target $\mathbf{w}^*$

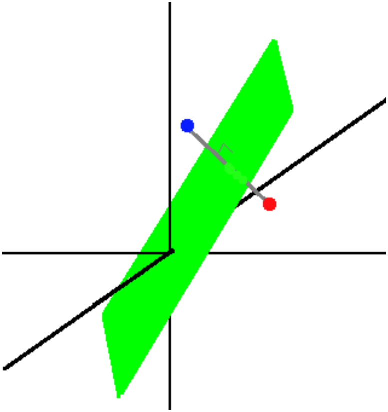


$\mathbf{x}^\top \mathbf{w}^* = 0$

- What is the smallest training set the teacher can construct?

Example one

Teacher's non-iid training set with $n = 2$ items



Example two

- The student estimates a Gaussian density:

$$\text{Given } \mathbf{x}_1 \ldots \mathbf{x}_n \in \mathbb{R}^d$$

$$\text{Steve learns } \hat{\mu} = \frac{1}{n} \sum \mathbf{x}_i, \quad \hat{\Sigma} = \frac{1}{n-1} \sum (\mathbf{x}_i - \hat{\mu})(\mathbf{x}_i - \hat{\mu})^\top$$

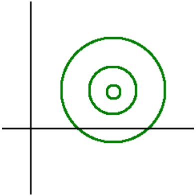
Example two

- The student estimates a Gaussian density:

$$\text{Given } \mathbf{x}_1 \ldots \mathbf{x}_n \in \mathbb{R}^d$$

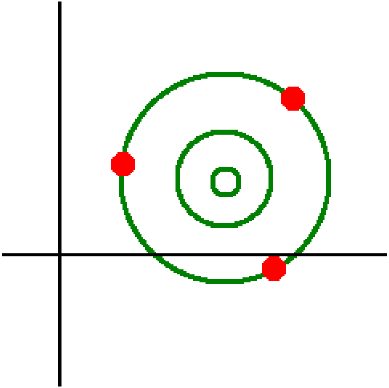
$$\text{Steve learns } \hat{\mu} = \frac{1}{n} \sum \mathbf{x}_i, \quad \hat{\Sigma} = \frac{1}{n-1} \sum (\mathbf{x}_i - \hat{\mu})(\mathbf{x}_i - \hat{\mu})^\top$$

- The teacher wants to teach a target Gaussian with (μ^*, Σ^*)

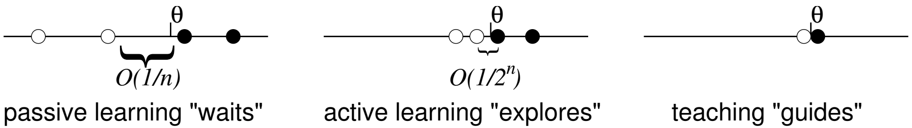


Example two

Teacher's minimal training set: $n = d + 1$ tetrahedron vertices



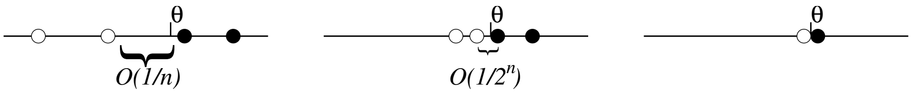
Machine teaching is stronger than active learning



Sample complexity to achieve ϵ error

- passive learning $1/\epsilon$

Machine teaching is stronger than active learning

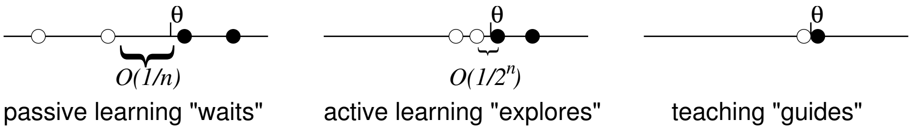


passive learning "waits" | active learning "explores" | teaching "guides"

Sample complexity to achieve ϵ error

- passive learning $1/\epsilon$
- active learning $\log(1/\epsilon)$

Machine teaching is stronger than active learning



Sample complexity to achieve ϵ error

- passive learning $1/\epsilon$
- active learning $\log(1/\epsilon)$
- machine teaching 2: the teacher knows θ

Machine teaching

- teacher knows the learning algorithm

Machine teaching

- teacher knows the learning algorithm
- teacher has a target model

Machine teaching

- teacher knows the learning algorithm
- teacher has a target model
- teacher constructs the smallest training set (Teaching Dimension [Goldman Kearns 1995])

Machine teaching

- teacher knows the learning algorithm
- teacher has a target model
- teacher constructs the smallest training set (Teaching Dimension [Goldman Kearns 1995])
- applications in education and security

Machine teaching

- teacher knows the learning algorithm
- teacher has a target model
- teacher constructs the smallest training set (Teaching Dimension [Goldman Kearns 1995])
- applications in education and security
- many open problems in optimization and theory

Machine teaching

- teacher knows the learning algorithm
- teacher has a target model
- teacher constructs the smallest training set (Teaching Dimension [Goldman Kearns 1995])
- applications in education and security
- many open problems in optimization and theory

Machine teaching

- teacher knows the learning algorithm
- teacher has a target model
- teacher constructs the smallest training set (Teaching Dimension [Goldman Kearns 1995])
- applications in education and security
- many open problems in optimization and theory

References:

`http://pages.cs.wisc.edu/~jerryzhu/machineteaching/`

Thank you