The Security of Latent Dirichlet Allocation

Shike Mei **Xiaojin Zhu**
Department of Computer Sciences, University of Wisconsin-Madison, Madison WI 53706, USA
{mei, jerryzhu}@cs.wisc.edu

Abstract

Latent Dirichlet allocation (LDA) is an increasingly popular tool for data analysis in many domains. If LDA output affects decision making (especially when money is involved), there is an incentive for attackers to compromise it. We ask the question: how can an attacker minimally poison the corpus so that LDA produces topics that the attacker wants the LDA user to see? Answering this question is important to characterize such attacks, and to develop defenses in the future. We give a novel bilevel optimization formulation to identify the optimal poisoning attack. We present an efficient solution (up to local optima) using descent method and implicit functions. We demonstrate poisoning attacks on LDA with extensive experiments, and discuss possible defenses.

1 Introduction

The last few years have witnessed the wide adoption of latent topic modeling, exemplified by latent Dirichlet allocation (LDA), in science and art such as political analysis (Grimmer 2010), business intelligence (Mahajan, Dey & Haque 2008), music (Cai, Zhang, Wang, Zhang & Ma 2007) and even archaeology (Pratt, MacLean, Knutson & Ringger 2011). LDA is rapidly becoming the *modus operandi* for data mining practitioners to explore large data sets. Importantly, the recovered topics are increasingly driving data interpretation and decision making.

Whenever a machine learner drives decision making, one needs to consider its security vulnerabilities. Specifically, what if an attacker has the ability to maliciously poison the corpus with the goal to manipulate the topics produced by standard LDA? A user who runs standard LDA on the poisoned corpus will then see the manipulated topics, which may affect her decisions. There may be financial or political incentives for the attacker to mount such an attack.

Such security concerns are not unfounded. A similar attack on spam filters, where an attacker may feed specially designed emails to a spam filter in order to alter the filter's classification behavior, has been well-known, see e.g. (Nelson, Barreno, Chi, Joseph, Rubinstein, Saini, Sutton, Tygar & Xia 2009). Other examples of research on the security of machine learning include generic ways that a learning system might be compromised (Barreno, Nelson, Joseph & Tygar 2010), *ad hoc* attacking procedures against SVMs (Biggio, Nelson & Laskov 2012), network worm detectors (Newsome, Karp & Song 2006), HTTP requests (Chung & Mok 2007), malware detectors (Biggio, Corona, Maiorca, Nelson, Šrndić, Laskov, Giacinto & Roli 2013) and so on.

Although the security of some machine learners has been studied before, to the best of our knowledge the security risks to latent topic modeling, in particular LDA, remain unexplored. To what extent can such attacks be optimized to inflict the maximum damage? What are some ways to defend against such attacks? In this paper, we answer the first question by proposing a unified computational framework for attacking LDA under budget constraints. We formulate it as a bilevel optimization problem (Colson, Marcotte & Savard 2007). We develop an efficient descent method based on implicit functions for solving the bilevel optimization problem. Our method can be generalized easily to attacking other machine learning models that employ variational inference. Note that our ultimate goal is not to be the attacker but to understand the power and limits of such attacks, which is logically the first step towards designing defenses against such attacks in the future.

Appearing in Proceedings of the 18^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2015, San Diego, CA, USA. JMLR: W&CP volume 38.
Copyright 2015 by the authors.

2 The KKT Conditions for LDA Variational Inference

We first review the notation. Recall LDA is a generative model consisting of K topics. The k-th topic is a multinomial distribution φ_k over some vocabulary, and is drawn from a Dirichlet prior $\varphi_k \sim \mathbf{Dir}(\beta)$. Each document d has a topic proportion multinomial θ_d, which is drawn from another Dirichlet distribution $\theta_d \sim \mathbf{Dir}(\alpha)$. For the i-th word in document d, we draw a topic assignment z_{di} from the multinomial parametrized by θ_d: $p(z_{di} = k \mid \theta_d) = \theta_{dk}$, and then draw the word w_{di} from the selected topic $\varphi_{z_{di}}$: $p(w_{di} \mid z_{di}, \varphi) = \varphi_{z_{di},w_{di}}$. During inference, words $\mathbf{W} = \{w_{di}\}$ and hyperparameters α, β are observed, while topic assignments $\mathbf{Z} = \{z_{di}\}$, topic proportions $\boldsymbol{\theta} = \{\theta_{dk}\}$, and topics $\boldsymbol{\varphi} = \{\varphi_k\}$ are hidden. The posterior of interest is $p(\boldsymbol{\varphi}, \boldsymbol{\theta}, \mathbf{Z} \mid \mathbf{W}, \alpha, \beta)$. However, calculating this posterior $p(\boldsymbol{\varphi}, \boldsymbol{\theta}, \mathbf{Z} \mid \mathbf{W}, \alpha, \beta)$ exactly is intractable. Two common approximations are Markov chain Monte Carlo (MCMC) methods such as collapsed Gibbs sampling (Griffiths & Steyvers 2004), and variational methods (Blei, Ng & Jordan 2003). Our analysis is aimed at LDA with variational inference, although empirically our attacks are also effective on LDA with MCMC as discussed in Section 5.

LDA variational approximation typically employs a fully factorized variational distribution $q(\boldsymbol{\varphi}, \boldsymbol{\theta}, \mathbf{Z}) = \prod_k q(\varphi_k \mid \boldsymbol{\eta}_k) \prod_d (q(\theta_d \mid \boldsymbol{\gamma}_d) \prod_i q(z_{di} \mid \boldsymbol{\phi}_{di}))$, where $q(\varphi_k \mid \boldsymbol{\eta}_k)$, $q(\theta_d \mid \boldsymbol{\gamma}_d)$ and $q(z_{di} \mid \boldsymbol{\phi}_{di})$ are Dirichlet, Dirichlet and Multinomial distributions parametrized by variational parameters $\boldsymbol{\eta}_k$, $\boldsymbol{\gamma}_d$ and $\boldsymbol{\phi}_{di}$, respectively. Let $\boldsymbol{\mu} = \{\boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\phi}\}$ be all variational parameters and denote the space of $\boldsymbol{\mu}$ as $\boldsymbol{\Theta}$. The objective is to minimize the Kullback-Leibler (KL) divergence between the variational distribution and the true posterior distribution w.r.t. the variational parameters:

$$\hat{\boldsymbol{\mu}}(\mathbf{W}) \in \operatorname{argmin}_{\boldsymbol{\mu} \in \boldsymbol{\Theta}} \quad KL(q(\boldsymbol{\varphi}, \boldsymbol{\theta}, \mathbf{Z} \mid \boldsymbol{\mu}) \| p(\boldsymbol{\varphi}, \boldsymbol{\theta}, \mathbf{Z} \mid \mathbf{W}, \alpha, \beta)). \quad (1)$$

Variational inference on LDA is to solve the Karush-Kuhn-Tucker (KKT) conditions for Eq (1) to find a local optimum. The KKT conditions will be important to describe LDA attacks later. We state the main conditions here, and its complete derivation is in Appendix A in the supplementary material. To simplify notation, we note that $\boldsymbol{\phi}_{di}$ is the same for all word positions w_{di} sharing the same word v. We denote this shared value of $\boldsymbol{\phi}_{di}$ as $\boldsymbol{\phi}_{dv}$. Correspondingly, the number of times word v appears in document d is denoted as $m_{dv} \in \mathbb{Z}_{\geq 0}$. These elements m_{dv} form a size $D \times V$ document-word matrix $\mathbf{M}$, which is another representation of the input corpus. The KKT conditions consist of $K \times V$ equations in (2), $D \times K$ equations in (3), and $D \times V \times K$ equations in (4):

$$\eta_{kv} - \beta - \sum_d \phi_{dvk} m_{dv} = 0 \quad (2)$$

$$\gamma_{dk} - \alpha - \sum_v \phi_{dvk} m_{dv} = 0 \quad (3)$$

$$\phi_{dvk} - \frac{\exp(\Psi(\gamma_{dk}) + (\Psi(\eta_{kv}) - \Psi(\sum_{v'} \eta_{kv})))}{\sum_k \exp(\Psi(\gamma_{dk}) + (\Psi(\eta_{kv}) - \Psi(\sum_{v'} \eta_{kv'})))} = 0. \quad (4)$$

By solving the above KKT conditions for a given $\mathbf{M}$, one obtains a locally optimal set of variational parameters $\hat{\boldsymbol{\mu}}(\mathbf{M}) = \{\hat{\boldsymbol{\eta}}(\mathbf{M}), \hat{\boldsymbol{\gamma}}(\mathbf{M}), \hat{\boldsymbol{\phi}}(\mathbf{M})\}$. Note that $\hat{\boldsymbol{\mu}}(\mathbf{M})$ cannot be written in closed-form of $\mathbf{M}$. However, as we will see in Section 3, $\hat{\boldsymbol{\mu}}(\mathbf{M})$ is an implicit function of $\mathbf{M}$ and the KKT conditions are the corresponding implicit equations.

3 Optimal Attacks on LDA

In practice, LDA is usually used to learn the topics as a concise summary of a corpus. For example, in variational inference given an input corpus $\mathbf{M}$ the topics are defined by the optimal variational parameters $\hat{\boldsymbol{\mu}}(\mathbf{M})$ from Eq (1). As standard in variational inference, we use the expectation of $\boldsymbol{\varphi}$ in the variational distribution $q(\boldsymbol{\varphi} \mid \hat{\boldsymbol{\eta}}(\mathbf{M}))$ as the learned topics, denoted as $\hat{\boldsymbol{\varphi}}(\mathbf{M}) \equiv \hat{\boldsymbol{\varphi}}(\hat{\boldsymbol{\eta}}(\mathbf{M}))$.

We now consider the security risk when an attacker can poison the corpus, such that the topics learned by LDA will be guided toward some target multinomial distributions $\boldsymbol{\varphi}^*$ defined by the attacker. Intuitively, the closer the learned topics $\hat{\boldsymbol{\varphi}}(\mathbf{M})$ are to the attacker-defined target topics $\boldsymbol{\varphi}*$ the higher gain the attacker will get. Specifically, one goal of the attacker is to minimize an *attacker risk function* $R_A(\hat{\boldsymbol{\varphi}}(\mathbf{M}), \boldsymbol{\varphi}^*)$, which characterizes the distance between $\hat{\boldsymbol{\varphi}}(\mathbf{M})$ and $\boldsymbol{\varphi}^*$. Meanwhile, as a greatly altered corpus tends to attract attention, another goal of the attacker is to only make small changes to the corpus. From here on, we will use $\mathbf{M}_0$ to denote the document-word matrix of the original corpus, and $\mathbf{M}$ the poisoned corpus. Intuitively, the attacker wants to limit the danger of being detected by only considering $\mathbf{M}$ that is close to $\mathbf{M}_0$. We formally define the set of allowable poisoned corpus as a search space $\mathbb{M}$. Concrete definitions of $R_A()$ and $\mathbb{M}$ are task-dependent and we will give several instances later.

A rational attacker should minimize $R_A()$ while remaining in $\mathbb{M}$. We formulate the optimal LDA attacking problem as a bilevel programming prob-

lem: (Bard 1998, Colson et al. 2007)

$$\min_{\mathbf{M}\in\mathbb{M},\hat{\boldsymbol{\varphi}}(\mathbf{M})} \quad R_A(\hat{\boldsymbol{\varphi}}(\mathbf{M}), \boldsymbol{\varphi}^*) \tag{5}$$

$$\text{s.t.} \quad \hat{\boldsymbol{\varphi}}(\mathbf{M}) \text{ are LDA topics learned from } \mathbf{M}. \tag{6}$$

The optimization for $\mathbf{M}$ in Eq (5) is called the upper-level task, which is the attacker's optimization problem. Eq (6) is called the lower-level task, which is nothing but the LDA learner's optimization problem given the corpus $\mathbf{M}$. Our framework is similar to machine teaching (Zhu 2013, Zhu 2015, Mei & Zhu 2015, Patil, Zhu, Kopec & Love 2014) where the teacher plays the role of the attacker in our framework. Unfortunately, bilevel programming is in general difficult. Furthermore, it is well-known that the lower-level optimization Eq (6) does not admit a closed-form solution. In what follows, we focus on LDA variational inference algorithm to derive an efficient approximate solution to the bilevel programming problem.

Our framework and research on regularized topic model, e.g. (Newman, Bonilla & Buntine 2011), are quite different on the variables they optimize. Our work optimize the training data (corpus) while regularized topic model optimizes the topics given fixed training data. This difference leads to a difference in the optimization framework. Our work is necessarily a bilevel framework to combine attacker's risk and learner's KL-divergence. In contrast, regularized topic model is a single-level optimization problem combining KL-divergence and structured prior by modifying the loss function.

3.1 Attacking LDA with Variational Inference

In variational inference, each term $\hat{\varphi}(\hat{\boldsymbol{\eta}}(\mathbf{M}))_{kv}$ is defined as

$$\hat{\varphi}\left(\hat{\boldsymbol{\eta}}\left(\mathbf{M}\right)\right)_{kv} \triangleq \mathbb{E}_{q(\boldsymbol{\varphi}|\hat{\boldsymbol{\eta}}(\mathbf{M}))}\left[\varphi_{kv}\right] = \frac{\hat{\eta}(\mathbf{M})_{kv}}{(\sum_{v'}\hat{\eta}(\mathbf{M})_{kv'})}. \tag{7}$$

We need to make the upper-level problem continuous to solve it by projected gradient descent method in the following sections. Therefore, as a standard way in machine learning, we relax each element m_{dv} from integer to non-negative real values to make the upper bilevel problem continuous. We get our formulation for attacking variational LDA:

$$\min_{\mathbf{M}\in\mathbb{M},\hat{\boldsymbol{\mu}}(\mathbf{M})} \quad R_A\left(\hat{\boldsymbol{\varphi}}(\hat{\boldsymbol{\eta}}(\mathbf{M})), \boldsymbol{\varphi}^*\right) \tag{8}$$

$$\text{s.t.} \quad \hat{\boldsymbol{\mu}}(\mathbf{M}) \in \operatorname{argmin}_{\boldsymbol{\mu}\in\boldsymbol{\Theta}} \tag{9}$$
$$KL(q(\boldsymbol{\varphi}, \boldsymbol{\theta}, \mathbf{Z} \mid \boldsymbol{\mu}) \| p(\boldsymbol{\varphi}, \boldsymbol{\theta}, \mathbf{Z} \mid \mathbf{M}, \alpha, \beta)).$$

This intermediate bilevel optimization problem is still hard to solve. We convert it to a single-level optimization problem by the KKT conditions for the lower-level LDA variational inference problem:

$$\min_{\mathbf{M}\in\mathbb{M},\hat{\boldsymbol{\mu}}(\mathbf{M})} \quad R_A\left(\hat{\boldsymbol{\varphi}}(\hat{\boldsymbol{\eta}}(\mathbf{M})), \boldsymbol{\varphi}^*\right)$$
$$\text{s.t.} \quad \hat{\boldsymbol{\mu}}(\mathbf{M}) \text{ satisfies Eqs (2)(3)(4).} \tag{10}$$

3.2 Descent Method

We use the descent method (Savard & Gauvin 1994) to solve the relaxed problem Eq (10). The descent method is an iterative gradient method. In iteration t, we take a gradient step with stepsize λ_t in the opposite direction of the gradient of the upper-level objective with respect to $\mathbf{M}$, then project the updated corpus to the search space $\mathbb{M}$:

$$\mathbf{M}^{(t)} = \operatorname{Proj}_{\mathbb{M}}\Big[\mathbf{M}^{(t-1)} - \lambda_t \nabla_{\mathbf{M}} R_A\left(\hat{\boldsymbol{\varphi}}(\hat{\boldsymbol{\eta}}(\mathbf{M})), \boldsymbol{\varphi}^*\right) \Big|_{\mathbf{M}=\mathbf{M}^{(t-1)}}\Big]. \tag{11}$$

As explained later in Section 3.3, we actually perform projected gradient descent to ensure nonnegativity of $\mathbf{M}^{(t)}$. However, the main difficulty is in computing the gradient because $\hat{\boldsymbol{\varphi}}(\hat{\boldsymbol{\eta}}(\mathbf{M}))$ is an *implicit function* of $\mathbf{M}$. We denote the number of entries in $\boldsymbol{\eta}$, $\boldsymbol{\gamma}$, $\boldsymbol{\phi}$, $\boldsymbol{\mu}$ and $\mathbf{M}$ as $N_{\boldsymbol{\eta}}$,$N_{\boldsymbol{\gamma}}$,$N_{\boldsymbol{\phi}}$,$N_{\boldsymbol{\mu}}$ and $N_{\mathbf{M}}$, respectively. We calculate the gradient term in Eq (11) according to the chain rule

$$\nabla_{\mathbf{M}} R_A\left(\hat{\boldsymbol{\varphi}}(\hat{\boldsymbol{\eta}}(\mathbf{M})), \boldsymbol{\varphi}^*\right) \tag{12}$$
$$= \nabla_{\boldsymbol{\varphi}} R_A\left(\boldsymbol{\varphi}, \boldsymbol{\varphi}^*\right)\Big|_{\boldsymbol{\varphi}=\hat{\boldsymbol{\varphi}}(\hat{\boldsymbol{\eta}}(\mathbf{M}))} \frac{\partial\hat{\boldsymbol{\varphi}}(\boldsymbol{\eta})}{\partial\boldsymbol{\eta}}\Big|_{\boldsymbol{\eta}=\hat{\boldsymbol{\eta}}(\mathbf{M})} \frac{\partial\hat{\boldsymbol{\eta}}(\mathbf{M})}{\partial\mathbf{M}}$$

$\nabla_{\boldsymbol{\varphi}} R_A(\boldsymbol{\varphi}, \boldsymbol{\varphi}^*)\Big|_{\boldsymbol{\varphi}=\hat{\boldsymbol{\varphi}}(\hat{\boldsymbol{\eta}}(\mathbf{M}))}$ is easy to compute if we assume that $R_A(\boldsymbol{\varphi}, \boldsymbol{\varphi}^*)$ is differentiable with respect to $\boldsymbol{\varphi}$. It is a vector with length $N_{\boldsymbol{\eta}}$. We give specific forms in Section 3.3. The $\frac{\partial\hat{\boldsymbol{\varphi}}(\boldsymbol{\eta})}{\partial\boldsymbol{\eta}}\Big|_{\boldsymbol{\eta}=\hat{\boldsymbol{\eta}}(\mathbf{M})}$ term is also easy to compute. It is a size $N_{\boldsymbol{\eta}} \times N_{\boldsymbol{\eta}}$ Jacobian matrix and according to Eq (7), its element in the kv-th row and the $k'v'$-th column is

$$\left[\frac{\partial\hat{\boldsymbol{\varphi}}(\boldsymbol{\eta})}{\partial\boldsymbol{\eta}}\right]_{kv,k'v'} = \frac{(\sum_w \eta_{kw})\mathbb{I}_1(v'=v) - \eta_{kv'}}{(\sum_w \eta_{kw})^2}\mathbb{I}_1(k=k'), \tag{13}$$

where $\mathbb{I}_1(z) = 1$ if z is true, and 0 otherwise. The term $\frac{\partial\hat{\boldsymbol{\eta}}(\mathbf{M})}{\partial\mathbf{M}}$ is a size $N_{\boldsymbol{\eta}} \times N_{\mathbf{M}}$ Jacobian matrix. The element at kv-th row and dv'-th column is $\left[\frac{\partial\hat{\boldsymbol{\eta}}(\mathbf{M})}{\partial\mathbf{M}}\right]_{kv,dv'} = \frac{\partial\hat{\eta}_{kv}(\mathbf{M})}{\partial M_{dv'}}$. This Jacobian matrix is hard to compute because $\hat{\boldsymbol{\eta}}(\mathbf{M})$, as a component of $\hat{\boldsymbol{\mu}}(\mathbf{M})$, cannot be represented in closed-form with respect to $\mathbf{M}$. However, under the mild condition that the Jacobian matrix $\frac{\partial\mathbf{f}_{\boldsymbol{\mu}}}{\partial\boldsymbol{\mu}}$ is invertible, $\hat{\boldsymbol{\mu}}(\mathbf{M})$ is an implicit function of

$\mathbf{M}$ (Danilov 2001). Here, $\mathbf{f}_{\boldsymbol{\mu}}$ denotes the left-hand-side terms of the system of equations Eqs (2)(3)(4). Moreover, $\frac{\partial \hat{\boldsymbol{\mu}}(\mathbf{M})}{\partial \mathbf{M}}$ can be get as follows by the implicit function theorem:

$$\frac{\partial \hat{\boldsymbol{\mu}}(\mathbf{M})}{\partial \mathbf{M}} = -(\frac{\partial \mathbf{f}_{\boldsymbol{\mu}}}{\partial \boldsymbol{\mu}})^{-1}(\frac{\partial \mathbf{f}_{\boldsymbol{\mu}}}{\partial \mathbf{M}}), \tag{14}$$

where $\frac{\partial \mathbf{f}_{\boldsymbol{\mu}}}{\partial \boldsymbol{\mu}}$ is the $N_{\boldsymbol{\mu}} \times N_{\boldsymbol{\mu}}$ Jacobian matrix, $\frac{\partial \mathbf{f}_{\boldsymbol{\mu}}}{\partial \mathbf{M}}$ is the $N_{\boldsymbol{\mu}} \times N_{\mathbf{M}}$ Jacobian matrix. The technical detail is in supplemental material Appendix B.

Although $\frac{\partial \hat{\boldsymbol{\eta}}(\mathbf{M})}{\partial \mathbf{M}}$ (as a component of $\frac{\partial \hat{\boldsymbol{\mu}}(\mathbf{M})}{\partial \mathbf{M}}$) can be computed by Eq (14), it requires inverting a large Jacobian matrix $\frac{\partial \mathbf{f}_{\boldsymbol{\mu}}}{\partial \boldsymbol{\mu}}$ and could be impractical. We propose to approximate the computation efficiently as follows:

$$\left[\frac{\partial \hat{\boldsymbol{\eta}}(\mathbf{M})}{\partial \mathbf{M}}\right]_{kv,dv'} \approx \phi_{dvk}\mathbb{I}_1(v = v') \tag{15}$$

This approximation is discussed in supplemental material Appendix C. Empirically, it works well in our experiments.

Summary of the descent method. The gradient of the upper-level objective w.r.t. $\mathbf{M}$ allows us to do gradient descent on the relaxed attack objective in Eq (10), where $\mathbf{M}$ was relaxed to a real-valued matrix. In the end, we project $\mathbf{M}$ back to the space of nonnegative integer matrices $\mathbb{Z}_{\geq 0}^{D\times V}$. Let $c_v \triangleq \text{round}(\sum_d m_{dv})$ be the rounded column sum in $\mathbf{M}$. The projection is defined as the nearest integer-value matrix with $\mathbf{M}$ (measured in L_1 distance) which maintains the column sum as c_v:

$$\tilde{\mathbf{M}} \triangleq \text{argmin}_{\mathbf{M}' \in \mathbb{Z}_{\geq 0}^{D\times V}, \sum_d m'_{dv} = c_v} \sum_d \sum_v |m'_{dv} - m_{dv}|. \tag{16}$$

The complete optimal LDA attack algorithm is summarized in Algorithm 1.

3.3 Attack Solution for Specific $R_A()$ Functions and $\mathbb{M}$ Sets

The key to Algorithm 1 is computing the gradient, which depends on specific forms of the attacker risk functions and search spaces. Note that there can be many choices on the attacker risk functions and search spaces. For example, if we use the topic proportions of each document as features for downstream document classification, then classification accuracy based on the learned topics can be the attacker risk function. However, in this paper we restrict ourselves to LDA alone without a downstream task. We discuss two $R_A()$ functions and one $\mathbb{M}$ set below that are suitable for attacking LDA.

Algorithm 1 Descent Method for the Optimal Attack on LDA

Require: $\mathbf{M}^{(0)}$, $\boldsymbol{\varphi}^*$, α, β, $\{\lambda_t\}$
  $t = 0$
  while $\mathbf{M}_t$ not converged **do**
    Use standard LDA variational inference procedure (e.g. the software in (Blei et al. 2003)) to compute the variational parameters $\hat{\boldsymbol{\mu}}(\mathbf{M}^{(t)})$ satisfying Eq (1).
    Descent step: Update $\mathbf{M}^{(t+1)}$ from $\mathbf{M}^{(t)}$ by Eq (11).
    $t = t + 1$
  end while
  Project $\mathbf{M}^{(t)}$ to an integer-valued matrix $\tilde{\mathbf{M}}$ by Eq (16).
  return $\tilde{\mathbf{M}}$

The ℓ_2 attacker risk function is the sum of squares of the difference of the topics, that is $R_{A,\ell_2}(\hat{\boldsymbol{\varphi}}(\hat{\boldsymbol{\eta}}(\mathbf{M})), \boldsymbol{\varphi}^*) = \frac{1}{2}\sum_k \sum_v (\hat{\varphi}(\hat{\boldsymbol{\eta}}(\mathbf{M}))_{kv} - \varphi^*_{kv})^2$. We get the gradient by Eqs (12), (13) and (15): $\nabla_{m_{dv}} R_{A,\ell_2}(\hat{\boldsymbol{\varphi}}(\hat{\boldsymbol{\eta}}(\mathbf{M})), \boldsymbol{\varphi}^*) = \sum_k (\hat{\varphi}_{kv}(\hat{\boldsymbol{\eta}}(\mathbf{M})) - \varphi^*_{kv}) \frac{(\sum_{v'} \hat{\eta}_{kv'}(\mathbf{M})) - \hat{\eta}_{kv}(\mathbf{M})}{(\sum_{v'} \hat{\eta}_{kv'}(\mathbf{M}))^2} \hat{\phi}_{dvk}$. The $\epsilon-$insensitive ℓ_2 attacker risk function is defined as:

$$\begin{aligned} & R_{A,\epsilon-\ell_2}(\hat{\boldsymbol{\varphi}}(\hat{\boldsymbol{\eta}}(\mathbf{M})), \boldsymbol{\varphi}^*) \\ = & \frac{1}{2}\sum_k \sum_v ((|\hat{\varphi}(\hat{\boldsymbol{\eta}}(\mathbf{M}))_{kv} - \varphi^*_{kv}| - \epsilon)_+)^2, \end{aligned} \tag{17}$$

where $(x)_+ = \max\{0, x\}$. The gradient is $\nabla_{m_{dv}} R_{A,\epsilon-\ell_2}(\hat{\boldsymbol{\varphi}}(\hat{\boldsymbol{\eta}}(\mathbf{M})), \boldsymbol{\varphi}^*) =$

$$\sum_k \text{sign}(\hat{\varphi}_{kv}(\hat{\boldsymbol{\eta}}(\mathbf{M})) - \varphi^*_{kv})(|\hat{\varphi}_{kv}(\hat{\boldsymbol{\eta}}(\mathbf{M})) - \varphi^*_{kv}| - \epsilon)_+ \frac{(\sum_{v'} \hat{\eta}_{kv'}(\mathbf{M})) - \hat{\eta}_{kv}(\mathbf{M})}{(\sum_{v'} \hat{\eta}_{kv'}(\mathbf{M}))^2} \hat{\phi}_{dvk}. \tag{18}$$

We define a $\mathbb{M}$ set in which the ℓ_1 distance between the original matrix $\mathbf{M}_0$ and the manipulated $\mathbf{M}$ is within a total change limit L, and the ℓ_1 distance of each row is within a per-document change limit L_d (Nelson, Rubinstein, Huang, Joseph, Lee, Rao & Tygar 2012). In other words, the "small attacks" are:

$$\mathbb{M} = \{\mathbf{M} \in \mathbf{M} \in \mathbb{R}_{\geq 0}^{D\times V} : \|\mathbf{M}_0 - \mathbf{M}\|_1 \leq L \bigwedge \forall d : \|\mathbf{M}_{0,d,\cdot} - \mathbf{M}_{d,\cdot}\|_1 \leq L_d\} \tag{19}$$

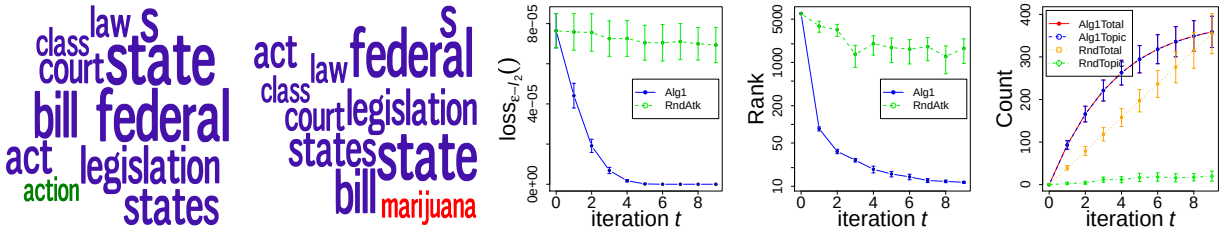
4 Experiments

We perform a variety of attacks on LDA to demonstrate the effectiveness of Algorithm 1. These attacks are for illustration purpose only. We use three disparate corpora which have been studied by the topic modeling research community before: CONG

Table 1: Corpus statistics, LDA parameters, and attack settings

corpus	#documents	#vocabulary	avg doc length	#topics	α	β
CONG	2740	6157	160	10	0.1	0.01
WISH	89533	23594	5	12	0.1	0.10
AP	2246	10473	134	15	0.1	0.01

attack goal φ^*	corpus to poison	L	$\frac{\lvert\text{budget}\rvert}{\lvert\text{corpus}\rvert}$	$\frac{\lvert\text{attack}\rvert}{\lvert\text{corpus}\rvert}$	L_d
promote "marijuana" to top 10 in the *legislation* topic	CONG	600	0.13%	0.08%	10
promote "debt" and "ceiling" to top 10 in the *market* topic	AP	600	0.20%	0.17%	10
demote "iraq" from top 10 in the *war* topic	CONG	300	0.06%	0.05%	10
replace "Paul" with "Weasley" in the *president* topic	WISH	800	0.17%	0.16%	2
promote "marijuana" but with Part-of-Speech constraints	CONG	600	0.13%	0.13%	10
move "president" from top 10 in the *president* topic to top 10 in the *peace* topic with sentence-level modification	CONG	500 sentences	0.55%	0.38%	1 sentence



(a) before attack (b) after attack (c) objective (d) rank (e) contribution

Figure 1: Promote-word attack on word "marijuana" in the *legislation* topic from CONG

consists of floor-debate transcripts from the United States House of Representatives in 2005 (Thomas, Pang & Lee 2006); WISH contains online new year's wishes (Goldberg, Fillmore, Andrzejewski, Xu, Gibson & Zhu 2009); AP is a subset of TREC AP newswire articles collected around 1990 (`http://www.cs.princeton.edu/blei/lda-c/`). We employ a standard implementation of variational LDA (Blei et al. 2003). Table 1 lists the corpus statistics and LDA parameters. The attacks are informally described in Table 1 and will be precisely defined later in the section. We use the ϵ-insensitive $R_{A,\epsilon-\ell_2}()$ as in Eq (17). ϵ is set to a very small value 0.005. We define $\mathbb{M}$ as in Eq (19) with parameters L and L_d specified in Table 1.

4.1 Promote-Word Attacks

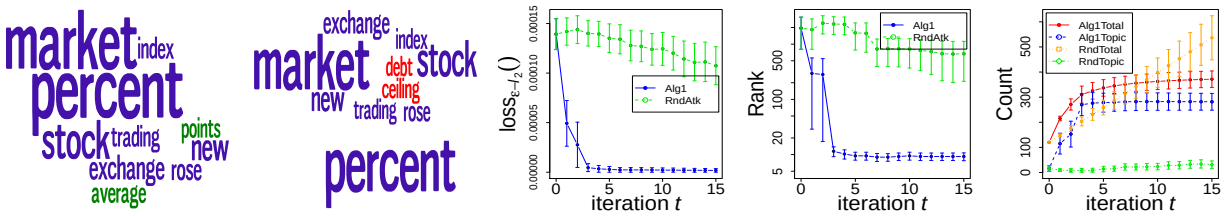
The first kind of attack aims to promote the topic-probability $\varphi^*_{k,w} \equiv p(w \mid k)$ of attack word w in topic k. This attack is motivated by the fact that often a user interprets LDA output by examining the top few words in each topic. Therefore, with sufficiently large $\varphi^*_{k,w}$ the attack word w will be seen by the user as if it is important in topic k. We present two example promote-word attacks. The first attack promotes the word "marijuana" into the top 10 words of the legislation topic in the CONG corpus.[1] The second promotes two words "debt" and "ceiling" into the top 10 words of the market topic in the AP corpus. These attack words describe "hot button issues" that *emerge after* the corpora (CONG was collected in 2005, AP in 1990). Therefore, no LDA topic on the original corpus assigns high probability to those attack words. This provides a valid setting to demonstrate our LDA attacks.

For the first attack, we define the attack target $\boldsymbol{\varphi}^*$ so that "marijuana" is the 10th word in the legislation topic (denoted as the kth topic). Specifically, given LDA topics $\boldsymbol{\varphi}$ obtained from the original CONG corpus, we denote the word ranked 9th in topic k as w_9. We promote the attack word $\boldsymbol{\varphi}^*_{k,\text{marijuana}} = \varphi_{k,w_9}$ while keeping all other words the same $\boldsymbol{\varphi}^*_{k,w} = \varphi_{k,w}$ for $w \neq$marijuana, then normalize $\boldsymbol{\varphi}^*_k$ so it sums to one. For other topics $k' \neq k$, $\boldsymbol{\varphi}^*_{k'}$ are exactly the same as $\boldsymbol{\varphi}_{k'}$. We note that LDA is known to have a "topic switching" problem in that topic indices can be permuted due to non-identifiability (Griffiths & Steyvers 2004). To deal with this issue, we matched each learned topic to the topic in $\boldsymbol{\varphi}^*$ with the minimum ℓ_1 distance.

Figure 1 shows that Algorithm 1 (Alg1) effectively attacked LDA. Panels (a,b) show the word cloud of the top 10 words in the legislation topic for one run. The attack word "marijuana" (red) became the 10-th word and forced "action" (green) out of top 10 ("action" ranked 11-th after attack). The other words (blue) remained in top 10 after attack.

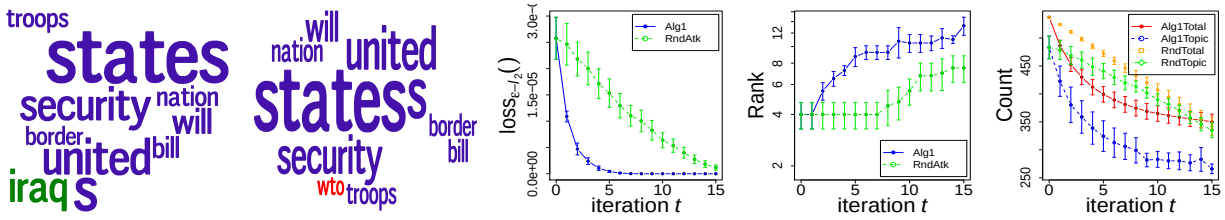
Panel (c) shows that Alg1 rapidly optimized the attack objective function $R_{A,\epsilon-\ell_2}()$. The error bars in all figures are $\pm$ standard error on 5 runs. Different

[1] As customary in LDA, the topic names are manually assigned to reflect the main words in that topic.



(a) before attack (b) after attack (c) objective (d) debt rank (e) debt contribution

Figure 2: Promote-word attack on words "debt" and "ceiling" in the *market* topic from AP



(a) before attack (b) after attack (c) objective (d) rank (e) contribution

Figure 3: Demote-word attack on word "Iraq" in the *war* topic from CONG

runs only differs by the random seeds of LDA. Since we are not aware of prior work on attacking LDA, we implemented a baseline attack (RndAtk) which adds the attack word(s) to randomly selected documents in the corpus subject to the same constraint encoded in the $\mathbb{M}$ set. RndAtk modified a fixed amount of words in each iteration, and the total amount of modification matches the amount by Alg1. Panel (c) shows that RndAtk only very slowly decreased the objective function $RA, \epsilon - \ell_2()$. Similarly, Panel (d) shows the rank of the word "marijuana" in the target topic as attack progresses. Alg1 promoted the attack word much more rapidly than RndAtk. Note the y-axis has logrithmic spacing.

Panel (e) sheds some light on why Alg1 is effective by showing how much of the attack words contributed to the target topic. Let $c_{kv} \triangleq \sum_d \phi_{dvk}$ be the variational contribution of word v to topic k. Let c_v be the total count of word v in the corpus. Panel (e) shows these quantities for "marijuana". Alg1 is effective because almost all the added attack words are assigned to the target topic.

The majority of Alg1's attacks consist of selecting 320 documents with high target-topic proportions and adding about 350 tokens of "marijuana" into them. This is necessary to boost the count of the attack word, which was small in the original corpus, in order for the attack word to enter top 10. However, Alg1 is more nuanced: it also added some top words in the target topic (i.e. "states", "s", "state" in the *legislation* topic), in conjunction with "marijuana", to selected documents with relatively low target topic proportion. This behavior made these documents more target topic heavy and potentially improved marijuana's contribution to the target topic. The detailed statistics of attack behaviors is in the supplemental material Appendix D.

Alg1 effectively attacked LDA on the second attack, too, where the target probability of "debt" and "ceiling" are set to the same as the rank 8-th word in the *market* topic. Figure 2 shows the curves for "debt" due to space limit; Other figures and the attack behaviors are similar to the first attack, and are left in supplementary material Appendix D.

4.2 Demote- and Replace-Word Attacks

Another kind of attack demotes the probability of a specific word w in the target topic, making w invisible to LDA users who examine only the top topic words. For example, LDA on the original CONG corpus consistently produced a *war* topic with "Iraq" in its top 10 words, see Figure 3(a). We demonstrate an attack that demotes "Iraq." Let k be the index of the war topic and w_{11} the 11-th word in that topic. We define the attack target probability to reduce the probability of "Iraq": $\varphi^*_{k,\text{Iraq}} = \varphi_{k,w_{11}}$, and then renormalize φ^*_k. For other topics $k' \neq k$, $\varphi^*_{k'} = \varphi_{k'}$. We then run Alg1 with this target φ^*. "Iraq" disappeared from the topic's top-10 (it ranked 12th after the attack), replace by "WTO" which ranked 11-th before attack (panel b). All other top words' rank did not change. For comparison, we let the RndAtk baseline delete "Iraq" from randomly selected documents that contain the word, subject to the constraint encoded in the $\mathbb{M}$ set. Alg1 optimized the objective function much more rapidly than RndAtk (panel c), and demoted the rank of "Iraq" more rapidly (panel d). Panel (e) shows that Alg1 removed "Iraq" in a selective way that more than half of the reduction is contributed to the war topic. This was not the case for RndAtk. Alg1 chose 150 doc-

uments with the highest target topic proportions and deleted about 170 tokens of "Iraq" from these documents. However, Alg1 also deleted other top words in the target topic (e.g. "united", "states", "s" in the *war* topic) from other documents containing "Iraq", to make those documents (and the "Iraq" in them) less associated with the target topic. More details are discussed in Appendix D.

Similarly, replace-word attacks (substitute one top word with another) can be achieved by a combination of promote and demote attacks. An example on the WISH corpus can be found in Appendix D.5.

4.3 Part-of-Speech (POS) Attacks

So far, the attacks modify the document-word matrix $\mathbf{M}$ entry-wise (i.e. adding or removing word tokens) without considering how the poisoned corpus may read. Simply appending L_d tokens of "marijuana" to a document is rather suspicious and prone to detection. Suppose the attacker wishes to introduce attack words into documents by only *replacing* existing words with the same POS. This ensures grammaticality of the poisoned corpus and may evade parser-based automatic corpus checking, for instance. Our framework can encode this POS constraint in the $\mathbb{M}$ set (19) in the obvious way. We add this constraint to the promote-word "marijuana" attack. The attack effects are very similar to Figure 1. Alg1's attack behavior is different, though: it mainly replaced the top words in the target topic (i.e. "bill", "legislation" and "state") in selected documents with "marijuana". All four words have the same POS (uncountable noun). This behavior soon made "marijuana" the top word in the target topic. Details are in Appendix D.

4.4 Sentence Attacks

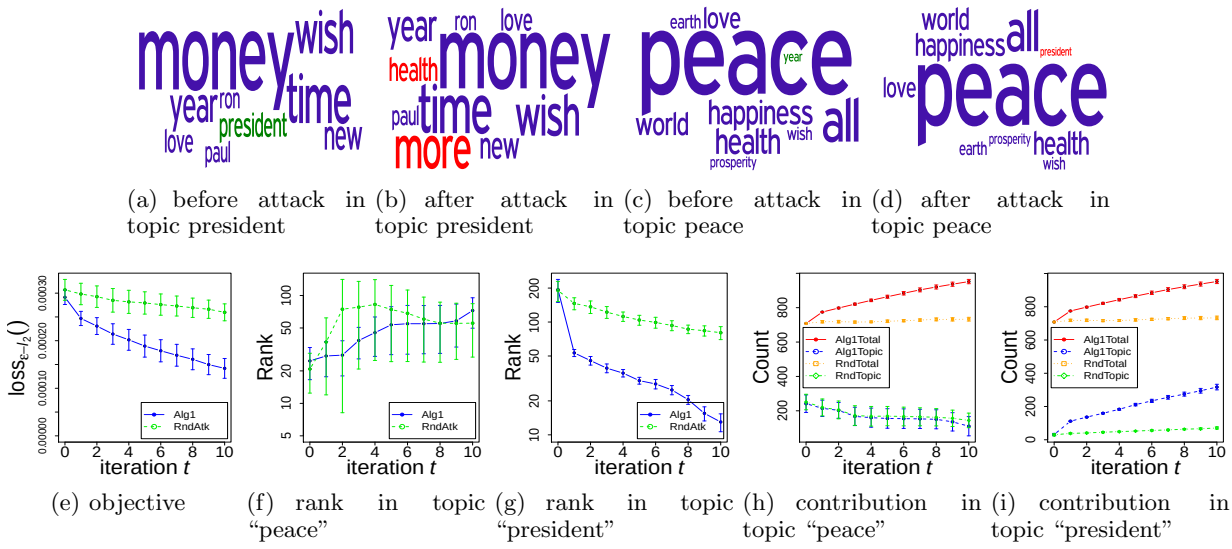
Perhaps a more practical, harder to detect attack is for an attacker to only add sentences (from a candidate corpus) to or remove existing sentences from any document. Again, this can be easily incorporated into the $\mathbb{M}$ set (19). We demonstrate such an attack on the WISH corpus, where the attack goal is to move a specific word from the top word list of a source topic to the top word list of a target topic. Specifically, vanilla LDA on the WISH corpus consistently returns a *president* topic with a top word *president*. Our attack goal is to move the word "president" from the source topic *president* to the target topic *peace*. This goal is encoded as a combination of demote-word attack in the source topic (denote as $k1$) and a promote-word attack in the target topic (denote as $k2$): we set $\varphi^*_{k1,\text{president}} = \varphi_{k1,w_{100}}$ and $\varphi^*_{k2,\text{president}} = \varphi_{k2,w_9}$. Changes must be whole sentences, where the candidate corpus is the original WISH corpus itself. In other words, the attacker can copy any sentence in the WISH corpus and add the copy to any document, or remove any sentence from any document. The RndAtk baseline is also a combination which randomly adds sentences containing "president"s and deletes sentences containing "president"s and "ron paul"s [2], subject to the constraint encoded in $\mathbb{M}$.

Figure 4 shows Alg1's effectiveness in the sentence attack. Panel (a,b) and Panel(c,d) show the top 10 words in the source topic and target topic, respectively. In the source topic, the word "president" (green) disappeared from top-10 after the attack (now ranked 80th) while "health" and "more" entered top-10. We note that the word "ron" was also forced out of top-10 (ranked 11th after the attack) because it frequently co-occurs with "president" in the same sentences, and is affected under sentence attacks. In the target topic, Alg1 promoted the word "president" (red) to the 10th and expelled the word "year" (green) from the top-10 (ranked 11th after attack). In summary, Alg1 successfully moved the word "president" from the source topic to the target topic. Alg1 optimized objective function rapidly as shown in Panel (e). The rank and contribution of "president" in the source topic shown in Panel (f,h) are similar to the previous demote-word attack, while those in the target topic shown in Panel (g,i) are similar to the previous promote-word attack.

Alg1's sentence attack behaviors, some shown in Table 2 and detailed statistics shown in Appendix D.7, were a trade-off between promote-word attack for word "president" in the target topic and demote-word attack for the same word in the source topic. It inserted about 300 sentences containing "president" into documents with high target topic proportion, and deleted about 50 instances of the sentence "president ron paul 2008" from documents with high source topic proportion, among other changes.

We also experimented with two extreme strategies, where the attacker is allowed to either only inserting sentences or only deleting sentences. None of the extreme strategies performed as well as Alg1. When only inserting sentences, the attacker needed to insert 400 sentences (50 more than Alg1) containing "president" into documents to promote "president" up to top-10 in the target topic, and could not demote "president" out of top-10 in the source topic. When only deleting sentences, the attacker needed to delete about 100 sentences (almost all containing "president ron paul") to demote "president" in the source topic, but could not promote "president" in the target topic. In summary, Alg1 was able to trade-off between inserting or deleting sentences intelligently.

[2] We only consider deleting sentences containing words "president" and "ron paul" because we observe that in Alg1, the optimal attack only deletes such sentences.



(a) before attack in topic president (b) after attack in topic president (c) before attack in topic peace (d) after attack in topic peace

(e) objective (f) rank in topic "peace" (g) rank in topic "president" (h) contribution in topic "peace" (i) contribution in topic "president"

Figure 4: Sentence attack: moving the word "president" from the source topic *president* to the target topic *peace* in the WISH corpus

Table 2: Sampled attack behaviors for sentence attack on WISH

Document before attack	Attack behavior
all needs love each other	+smarter president rudy
prosperity bring 2008 world happiness	+president romney
may new year peace	+president kokopelli
may everyone always love	+president zappa
peace, friend and family health	+president hillery
peace, all us health please	+president democratic obama
peace, hope forever love more	+president obama
peace world, new home and car	+president obama
president 2008 ron paul	Removed
president ron paul	Removed

5 Discussions

Alg1 takes about 20 minutes to complete a run of the promote-word attack on CONG on a 2-core 3.6 GHz 8 GB memory Linux desktop. The time for other attacks are similar. We note that should real attacks happen, the attacker would likely be an entity with access to much more powerful computation resources. Besides, corpus poisoning can be done offline. Thus the speed will not be an issue that deters such attacks.

Our experiment suggests that the attacker doesn't need to know exactly what LDA implementation is used. Even though the attacks in this paper are derived for LDA with variational methods, they work equally well on LDA with MCMC methods. We simply take the poisoned corpus by Alg1 and give it to LDA with collapsed Gibbs sampling (Griffiths & Steyvers 2004). On the promote-word "marijuana" attack in Section 4.1, the results are nearly identical to Figure 1. The only difference is minor fluctuations in the rank of other top words in the target topic (e.g. word with rank 8 moves to rank 11). This suggests that the attacker can compute/design his attack against some common, public available implementation of LDA and the attack could be effective against other LDA implementations.

Our ultimate goal is to defend LDA against corpus poisoning attacks. Within the machine learning community, there are two lines of research on general defense strategies: robust learning and optimal attack. The two lines are distinct but complement each other. Robust learning designs desensitized models that are robust under attacks in a minimax sense (e.g. (Globerson & Roweis 2006, Torkamani & Lowd 2013, Barreno et al. 2010, Laskov & Lippmann 2010)). Optimal attack quantifies the attacker: the harm done to a (potential vanilla) model when the attacker performs optimally (e.g. (Biggio et al. 2012)). Both lines of research are important. Our research on optimal attacks characterizes the worst-case attacks and offers a novel angle for defenses. For instance, in the experiments we showed that injecting several hundred words (about 0.1% change) into the corpus will modify the top topic words. However, the optimal attack needs to be highly selective on which documents to poison. A potential defense derived from our result is to alert human analysts to carefully inspect documents with high target topic proportion. This type of defense complements robust learning, while the latter may sacrifice model performance due to the need to desensitize the learner (Barreno et al. 2010).

Acknowledgments We thank Daniel Lowd for helpful discussions. Support for this research was provided in part by NSF grant IIS 0953219 and by the University of Wisconsin–Madison Graduate School with funding from the Wisconsin Alumni Research Foundation.

References

Bard, J. F. (1998), *Practical Bilevel Optimization: Algorithms And Applications*, Kluwer Academic Publishers.

Barreno, M., Nelson, B., Joseph, A. D. & Tygar, J. (2010), 'The security of machine learning', *Machine Learning* **81**(2), 121–148.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G. & Roli, F. (2013), Evasion attacks against machine learning at test time, *in* 'ECML-PKDD'.

Biggio, B., Nelson, B. & Laskov, P. (2012), Poisoning attacks against support vector machines, *in* 'ICML'.

Blei, D., Ng, A. & Jordan, M. (2003), 'Latent Dirichlet allocation', *Journal of Machine Learning Research* **3**, 993–1022.

Cai, R., Zhang, C., Wang, C., Zhang, L. & Ma, W. Y. (2007), Musicsense: contextual music recommendation using emotional allocation modeling, *in* 'Proceedings of the 15th international conference on Multimedia'.

Chung, S. P. & Mok, A. K. (2007), Advanced allergy attacks: Does a corpus really help, *in* 'Recent advances in intrusion detection (RAID)'.

Colson, B., Marcotte, B. & Savard, G. (2007), 'An overview of bilevel optimization', *Annals of operations research* **153**(1), 235–256.

Danilov, V. I. (2001), *Implicit function (in algebraic geometry)*, Encyclopedia of Mathematics, Springer.

Globerson, A. & Roweis, S. T. (2006), Nightmare at test time: robust learning by feature deletion, *in* 'ICML'.

Goldberg, A., Fillmore, N., Andrzejewski, D., Xu, Z. T., Gibson, B. & Zhu, X. J. (2009), May all your wishes come true: A study of wishes and how to recognize them, *in* 'NAACL HLT'.

Griffiths, T. L. & Steyvers, M. (2004), 'Finding scientific topics', *PNAS* **101**, 5228–5235.

Grimmer, J. (2010), 'A bayesian hierarchical topic model for political texts: Measuring expressed agendas in senate press releases', *Political Analysis* **18**(1), 1–35.

Laskov, P. & Lippmann, R. (2010), 'Machine learning in adversarial environments', *Machine Learning* **81**(2), 115–119.

Mahajan, A., Dey, L. & Haque, S. M. (2008), Mining financial news for major events and their impacts on the market, *in* 'Web Intelligence and Intelligent Agent Technology'.

Mei, S. & Zhu, X. (2015), Using machine teaching to identify optimal training-set attacks on machine learners, *in* 'The Twenty-Ninth AAAI Conference on Artificial Intelligence'.

Nelson, B., Barreno, M., Chi, F., Joseph, A., Rubinstein, B., Saini, U., Sutton, C., Tygar, J. & Xia, K. (2009), 'Misleading learners: Co-opting your spam filter'.

Nelson, B., Rubinstein, B. I., Huang, L., Joseph, A. D., Lee, S., Rao, S. & Tygar, J. (2012), 'Query strategies for evading convex-inducing classifiers', *The Journal of Machine Learning Research* pp. 1293–1332.

Newman, D., Bonilla, E. V. & Buntine, W. (2011), Improving topic coherence with regularized topic models, *in* 'Advances in Neural Information Processing Systems', pp. 496–504.

Newsome, J., Karp, B. & Song, D. (2006), Paragraph: Thwarting signature learning by training maliciously, *in* 'Recent advances in intrusion detection (RAID)'.

Patil, K., Zhu, X., Kopec, L. & Love, B. (2014), Optimal teaching for limited-capacity human learners, *in* 'Advances in Neural Information Processing Systems (NIPS)'.

Pratt, L. J., MacLean, A. C., Knutson, C. D. & Ringger, E. K. (2011), Cliff walls: An analysis of monolithic commits using latent dirichlet allocation, *in* 'Open Source Systems: Grounding Research', Springer, pp. 282–298.

Savard, G. & Gauvin, J. (1994), 'The steepest descent direction for the nonlinear bilevel programming problem', *Operations Research Letters* **15**(5), 265–272.

Thomas, M., Pang, B. & Lee, L. (2006), Get out the vote: Determining support or opposition from congressional floor-debate transcripts, *in* 'EMNLP'.

Torkamani, M. & Lowd, D. (2013), Convex adversarial collective classification, *in* 'ICML'.

Zhu, X. (2013), Machine teaching for Bayesian learners in the exponential family, *in* 'NIPS'.

Zhu, X. (2015), Machine teaching: an inverse problem to machine learning and an approach toward optimal education, *in* 'The Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI "Blue Sky" Senior Member Presentation Track)'.

A The KKT Conditions

The optimization problem for variational inference on LDA is:

$$\begin{aligned} \min_{\boldsymbol{\eta},\boldsymbol{\gamma},\boldsymbol{\phi}} \quad & KL(q(\boldsymbol{\varphi},\boldsymbol{\theta},\mathbf{Z} \mid \boldsymbol{\eta},\boldsymbol{\gamma},\boldsymbol{\phi}) \| p(\boldsymbol{\varphi},\boldsymbol{\theta},\mathbf{Z} \mid \mathbf{W},\alpha,\beta)) \\ \text{s.t.} \quad & -\eta_{kv} \leq 0, \quad \forall k, v; \\ & -\gamma_{dk} \leq 0, \quad \forall d, k; \\ & -\phi_{dik} \leq 0, \quad \forall d, i, k; \\ & \sum_k \phi_{dik} = 1, \quad \forall d, i. \end{aligned} \tag{20}$$

To derive the KKT conditions, we first introduce KKT multipliers $\lambda_{\eta_{kv}}, \lambda_{\gamma_{dk}}, \lambda_{\phi_{dik}}$ and $\rho_{\phi_{di}}$ to each constraint in Eq (20). The KKT conditions have four parts:

Stationarity

$$\begin{aligned} \eta_{kv} - \beta - \sum_d \sum_i \phi_{dik} \mathbb{I}_1(w_{di} = v) - \lambda_{\eta_{kv}} &= 0 \\ \gamma_{dk} - \alpha - \sum_i \phi_{dik} - \lambda_{\gamma_{dk}} &= 0 \\ \log \phi_{dik} - (\Psi(\gamma_{dk}) - \sum_{k'} \Psi(\gamma_{dk'}) + \Psi(\eta_{kw_{di}}) - \Psi(\sum_{v'} \eta_{kv})) - 1 - \lambda_{\phi_{dik}} + \rho_{\phi_{di}} &= 0. \end{aligned} \tag{21}$$

Complementary Slackness

$$\begin{aligned} -\lambda_{\eta_{kv}} \eta_{kv} &= 0 \\ -\lambda_{\gamma_{dk}} \gamma_{dk} &= 0 \\ -\lambda_{\phi_{dik}} \phi_{dik} &= 0. \end{aligned} \tag{22}$$

Primal Feasibility

$$\begin{aligned} -\eta_{kv} &\leq 0 \\ -\gamma_{dk} &\leq 0 \\ -\phi_{dik} &\leq 0 \\ \sum_k \phi_{dik} - 1 &= 0. \end{aligned} \tag{23}$$

Dual Feasibility

$$\begin{aligned} \lambda_{\eta_{kv}} &\geq 0 \\ \lambda_{\gamma_{dk}} &\geq 0 \\ \lambda_{\phi_{dik}} &\geq 0. \end{aligned} \tag{24}$$

First, we observe that Eq (21) implies that $-\eta_{kv}$, $-\gamma_{dk}$ and $-\phi_{dik}$ are both strictly negative because:

$$\begin{aligned} -\eta_{kv} \leq -\beta &< 0 \\ -\gamma_{dk} \leq -\alpha &< 0 \\ -\phi_{dik} = -\exp((\Psi(\gamma_{dk}) - \sum_{k'} \Psi(\gamma_{dk'}) + \Psi(\eta_{kw_{di}}) - \Psi(\sum_{v'} \eta_{kv})) - 1 - \lambda_{\phi_{dik}} + \rho_{\phi_{di}}) &< 0. \end{aligned}$$

We combine the above result with the complementary slackness Eq (22):

$$\begin{aligned} \lambda_{\eta_{kv}} &= 0 \\ \lambda_{\gamma_{dk}} &= 0 \\ \lambda_{\phi_{dik}} &= 0. \end{aligned} \tag{25}$$

We plug Eq (25) into Eqs (21) and (23):

$$\begin{aligned}
\eta_{kv} - \beta - \sum_d \sum_i \phi_{dik} \mathbb{I}_1(w_{di} = v) &= 0 \\
\gamma_{dk} - \alpha - \sum_i \phi_{dik} &= 0 \\
\log \phi_{dik} - (\Psi(\gamma_{dk}) - \sum_{k'} \Psi(\gamma_{dk'}) + \Psi(\eta_{kw_{di}}) - \Psi(\sum_{v'} \eta_{kv})) - 1 + \rho_{\phi_{di}} &= 0 \\
\rho_{\phi_{di}} - \log[\sum_k \exp(\Psi(\gamma_{dk}) - \sum_{k'} \Psi(\gamma_{dk'}) + \Psi(\eta_{kw_{di}}) - \Psi(\sum_{v'} \eta_{kv}) + 1)] &= 0. \qquad (26)
\end{aligned}$$

Eqs (26) and (25) are equivalent with the KKT conditions in Eqs (21),(23),(24) and (22). We focus on the conditions on primal variables and further simplify Eq (26) to get the equivalent form of KKT condition:

$$\begin{aligned}
\eta_{kv} - \beta - \sum_d \sum_i \phi_{dik} \mathbb{I}_1(w_{di} = v) &= 0 \\
\gamma_{dk} - \alpha - \sum_i \phi_{dik} &= 0 \\
\phi_{dik} - \frac{\exp(\Psi(\gamma_{dk}) + (\Psi(\eta_{kw_{di}}) - \Psi(\sum_{v'} \eta_{kv})))}{\sum_k \exp(\Psi(\gamma_{dk}) + (\Psi(\eta_{kw_{di}}) - \Psi(\sum_{v'} \eta_{kv'})))} &= 0. \qquad (27)
\end{aligned}$$

These are exactly the variational inference formulas in (Blei et al. 2003). After changing the notation (discussed in the main paper), we get Eq (4).

B Implicit Functions

We review the definition of implicit functions. We denote the $\epsilon-$ball of $\mathbf{x} \in \mathbb{R}^d$ as $N(\mathbf{x}, \epsilon) = \{\mathbf{y} : \|\mathbf{y} - \mathbf{x}\|_2 < \epsilon\}$. We call $\hat{\mathbf{y}}(\mathbf{x}) \in \mathbb{R}^m$ an implicit function of $\mathbf{x} \in \mathbb{R}^n$ defined by implicit equation $\mathbf{f}(\mathbf{x}, \mathbf{y}) = \mathbf{0}$, where $\mathbf{f}(\mathbf{x}, \mathbf{y}) : \mathbb{R}^{n+m} \mapsto \mathbb{R}^m$, if for any $\mathbf{x}$, $\hat{\mathbf{y}}(\mathbf{x})$ satisfies $\mathbf{f}(\mathbf{x}, \mathbf{y}) = \mathbf{0}$ and there exists $\epsilon > 0$ such that for any $\mathbf{x} + \delta \in N(\mathbf{x}, \epsilon)$, only $\hat{\mathbf{y}}(\mathbf{x} + \delta)$ both satisfies $\mathbf{f}(\mathbf{x}, \mathbf{y}) = \mathbf{0}$ and belongs to $N(\hat{\mathbf{y}}(\mathbf{x}), \epsilon)$.

Then we show $\hat{\boldsymbol{\mu}}(\mathbf{M})$ is an implicit function of $\boldsymbol{\mu}$ defined by implicit equations $\mathbf{f}_{\boldsymbol{\mu}} = \mathbf{0}$. First, $\mathbf{f}_{\boldsymbol{\mu}}$ is a (multivariate) continuous differential function of $\boldsymbol{\mu}$ because each component of $\mathbf{f}_{\boldsymbol{\mu}}$ consists of continuous differential terms. Second, by the assumption in the theorem, $\mathbf{f}_{\boldsymbol{\mu}}$ has an invertible Jacobian matrix $\frac{\partial \mathbf{f}_{\boldsymbol{\mu}}}{\partial \boldsymbol{\mu}}$. Therefore, according to implicit function theorem (Danilov 2001), $\hat{\boldsymbol{\mu}}(\mathbf{M})$ is an implicit function of $\mathbf{M}$ and $\hat{\boldsymbol{\mu}}(\mathbf{M})$ is continuously differentiable with respect to $\mathbf{M}$. The gradient is as in Eq (14).

C Fast Approximation for Computing $\frac{\partial \hat{\boldsymbol{\eta}}(\mathbf{M})}{\partial \mathbf{M}}$

Theorem 1 *If $\frac{\partial \mathbf{f}_{\boldsymbol{\eta}}}{\partial \boldsymbol{\phi}} = \mathbf{0}$, the element at kv-th row and dv′-th column in the Jacobian matrix $\frac{\partial \hat{\boldsymbol{\eta}}(\mathbf{M})}{\partial \mathbf{M}}$ is*

$$[\frac{\partial \hat{\boldsymbol{\eta}}(\mathbf{M})}{\partial \mathbf{M}}]_{kv,dv'} = \phi_{dvk} \mathbb{I}_1(v = v'). \qquad (28)$$

Proof: First, $\frac{\partial \hat{\boldsymbol{\eta}}(\mathbf{M})}{\partial \mathbf{M}}$ is the first $N_{\boldsymbol{\eta}}$ rows of the size $(N_{\boldsymbol{\eta}} + N_{\boldsymbol{\gamma}} + N_{\boldsymbol{\phi}}) \times N_{\mathbf{M}}$ Jacobian matrix $\frac{\partial \hat{\boldsymbol{\mu}}(\mathbf{M})}{\partial \mathbf{M}}$,

$$\frac{\partial \hat{\boldsymbol{\mu}}(\mathbf{M})}{\partial \mathbf{M}} = \begin{bmatrix} \frac{\partial \hat{\boldsymbol{\eta}}(\mathbf{M})}{\partial \mathbf{M}} \\ \frac{\partial (\hat{\boldsymbol{\gamma}}, \hat{\boldsymbol{\phi}})(\mathbf{M})}{\partial \mathbf{M}} \end{bmatrix},$$

where $\frac{\partial \hat{\boldsymbol{\eta}}(\mathbf{M})}{\partial \mathbf{M}}$ is a $N_{\boldsymbol{\eta}} \times N_{\mathbf{M}}$ Jacobian matrix, $\frac{\partial (\hat{\boldsymbol{\gamma}}, \hat{\boldsymbol{\phi}})(\mathbf{M})}{\partial \mathbf{M}}$ is a $(N_{\boldsymbol{\gamma}} + N_{\boldsymbol{\phi}}) \times N_{\mathbf{M}}$ Jacobian matrix.

We define a selection matrix $\mathbf{P} \triangleq [\mathbf{I}\ \mathbf{0}]$ (size of $N_{\boldsymbol{\eta}} \times (N_{\boldsymbol{\eta}} + N_{\mathbf{M}} + N_{\boldsymbol{\phi}})$) and $\frac{\partial \hat{\boldsymbol{\eta}}(\mathbf{M})}{\partial \mathbf{M}}$ is selected from $\frac{\partial \hat{\boldsymbol{\mu}}(\mathbf{M})}{\partial \mathbf{M}}$ by multiplying $\mathbf{P}$ on the left:

$$\frac{\partial \hat{\boldsymbol{\eta}}(\mathbf{M})}{\partial \mathbf{M}} = \mathbf{P} \frac{\partial \hat{\boldsymbol{\mu}}(\mathbf{M})}{\partial \mathbf{M}}. \qquad (29)$$

$\frac{\partial \hat{\boldsymbol{\mu}}(\mathbf{M})}{\partial \mathbf{M}}$ is computed as in Eq (14). We introduce the two terms on the right side of Eq (14) respectively.

The first term, $(\frac{\partial \mathbf{f}_{\boldsymbol{\mu}}}{\partial \boldsymbol{\mu}})^{-1}$ is the inversion of a Jacobian matrix with size of $(N_{\boldsymbol{\eta}} + N_{\boldsymbol{\gamma}} + N_{\boldsymbol{\phi}}) \times (N_{\boldsymbol{\eta}} + N_{\boldsymbol{\gamma}} + N_{\boldsymbol{\phi}})$. Similar to the divide of $\frac{\partial \hat{\boldsymbol{\mu}}(\mathbf{M})}{\partial \mathbf{M}}$, we write $\frac{\partial \mathbf{f}_{\boldsymbol{\mu}}}{\partial \boldsymbol{\mu}}$ (and correspondingly its inversion) as 4 blocks:

$$(\frac{\partial \mathbf{f}_{\boldsymbol{\mu}}}{\partial \boldsymbol{\mu}})^{-1} = \begin{bmatrix} \frac{\partial \mathbf{f}_{\boldsymbol{\eta}}}{\partial \boldsymbol{\eta}} & \frac{\partial \mathbf{f}_{\boldsymbol{\eta}}}{\partial (\boldsymbol{\gamma}, \boldsymbol{\phi})} \\ \frac{\partial (\mathbf{f}_{\boldsymbol{\gamma}}, \mathbf{f}_{\boldsymbol{\phi}})}{\partial \boldsymbol{\eta}} & \frac{\partial (\mathbf{f}_{\boldsymbol{\gamma}}, \mathbf{f}_{\boldsymbol{\phi}})}{\partial (\boldsymbol{\gamma}, \boldsymbol{\phi})} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}, \tag{30}$$

where $\frac{\partial \mathbf{f}_{\boldsymbol{\eta}}}{\partial \boldsymbol{\eta}}$ and $\mathbf{A}$ have size $N_{\boldsymbol{\eta}} \times N_{\boldsymbol{\eta}}$, $\frac{\partial \mathbf{f}_{\boldsymbol{\eta}}}{\partial (\boldsymbol{\gamma}, \boldsymbol{\phi})}$ and $\mathbf{B}$ have size $N_{\boldsymbol{\eta}} \times (N_{\boldsymbol{\gamma}} + N_{\boldsymbol{\phi}})$, $\frac{\partial (\mathbf{f}_{\boldsymbol{\gamma}}, \mathbf{f}_{\boldsymbol{\phi}})}{\partial \boldsymbol{\eta}}$ and $\mathbf{C}$ have size $(N_{\boldsymbol{\gamma}} + N_{\boldsymbol{\phi}}) \times N_{\boldsymbol{\eta}}$, and $\frac{\partial (\mathbf{f}_{\boldsymbol{\gamma}}, \mathbf{f}_{\boldsymbol{\phi}})}{\partial (\boldsymbol{\gamma}, \boldsymbol{\phi})}$ and $\mathbf{D}$ have size $(N_{\boldsymbol{\gamma}} + N_{\boldsymbol{\phi}}) \times (N_{\boldsymbol{\gamma}} + N_{\boldsymbol{\phi}})$.

The second term on the right side $\frac{\partial \mathbf{f}_{\boldsymbol{\mu}}}{\partial \mathbf{M}}$ has $(N_{\boldsymbol{\eta}} + N_{\boldsymbol{\gamma}} + N_{\boldsymbol{\phi}})$ rows and $N_{\mathbf{M}}$ columns. We write it as two blocks according to the division of $\frac{\partial \hat{\boldsymbol{\mu}}(\mathbf{M})}{\partial \mathbf{M}}$,

$$\frac{\partial \mathbf{f}_{\boldsymbol{\mu}}}{\partial \mathbf{M}} = \begin{bmatrix} \frac{\partial \mathbf{f}_{\boldsymbol{\eta}}}{\partial \mathbf{M}} \\ \frac{\partial (\mathbf{f}_{\boldsymbol{\gamma}}, \mathbf{f}_{\boldsymbol{\phi}})}{\partial \mathbf{M}} \end{bmatrix}, \tag{31}$$

where $\frac{\partial \mathbf{f}_{\boldsymbol{\eta}}}{\partial \mathbf{M}}$ has size $N_{\boldsymbol{\eta}} \times N_{\mathbf{M}}$ and $\frac{\partial (\mathbf{f}_{\boldsymbol{\gamma}}, \mathbf{f}_{\boldsymbol{\phi}})}{\partial \mathbf{M}}$ has size $(N_{\boldsymbol{\gamma}} + N_{\boldsymbol{\phi}}) \times N_{\mathbf{M}}$.

We plug the block form of matrices in Eqs (29), (30) and (31) into Eq (14):

$$\frac{\partial \hat{\boldsymbol{\eta}}(\mathbf{M})}{\partial \mathbf{M}} = -\mathbf{P} \left(\frac{\partial \mathbf{f}_{\boldsymbol{\mu}}}{\partial \boldsymbol{\mu}} \right)^{-1} \frac{\partial \mathbf{f}_{\boldsymbol{\mu}}}{\partial \mathbf{M}} = -\begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{f}_{\boldsymbol{\eta}}}{\partial \mathbf{M}} \\ \frac{\partial (\mathbf{f}_{\boldsymbol{\gamma}}, \mathbf{f}_{\boldsymbol{\phi}})}{\partial \mathbf{M}} \end{bmatrix} = -\left(\mathbf{A} \frac{\partial \mathbf{f}_{\boldsymbol{\eta}}}{\partial \mathbf{M}} + \mathbf{B} \frac{\partial (\mathbf{f}_{\boldsymbol{\gamma}}, \mathbf{f}_{\boldsymbol{\phi}})}{\partial \mathbf{M}} \right). \tag{32}$$

Now we need to calculate the two blocks $\mathbf{A}$ and $\mathbf{B}$ of the inverted Jacobian matrix $(\frac{\partial \mathbf{f}_{\boldsymbol{\mu}}}{\partial \boldsymbol{\mu}})^{-1}$. In the assumption of theorem, $\frac{\partial \mathbf{f}_{\boldsymbol{\eta}}}{\partial \boldsymbol{\phi}} = \mathbf{0}$. Note that $\frac{\partial \mathbf{f}_{\boldsymbol{\eta}}}{\partial \boldsymbol{\phi}} = \mathbf{0}$. According to Eq (2), we have $\frac{\partial \mathbf{f}_{\boldsymbol{\eta}}}{\partial \boldsymbol{\gamma}} = \mathbf{0}$ and $\frac{\partial \mathbf{f}_{\boldsymbol{\eta}}}{\partial (\boldsymbol{\gamma}, \boldsymbol{\phi})} = \mathbf{0}$. Therefore, $\frac{\partial \mathbf{f}_{\boldsymbol{\mu}}}{\partial \boldsymbol{\mu}}$ is a blockwise lower triangle matrix. Based on the property of blockwise inversion of matrix, we get

$$\mathbf{A} = \mathbf{I}, \mathbf{B} = \mathbf{0}.$$

We put the values of $\mathbf{A}$ and $\mathbf{B}$ into Eq (32) and get

$$\frac{\partial \hat{\boldsymbol{\eta}}(\mathbf{M})}{\partial \mathbf{M}} = -\frac{\partial \mathbf{f}_{\boldsymbol{\eta}}}{\partial \mathbf{M}}, \tag{33}$$

where each element in $\frac{\partial \mathbf{f}_{\boldsymbol{\eta}}}{\partial \mathbf{M}}$ is calculated by Eq (2):

$$\frac{\partial f_{\eta_{kv}}(\boldsymbol{\mu}, \mathbf{M})}{\partial m_{dv'}} = -\phi_{dvk} \mathbb{I}_1(v = v'). \tag{34}$$

We combine Eq (33) and Eq (34) and get

$$\nabla_{m_{dv'}} \hat{\eta}_{kv}(\mathbf{M}) = -(-\phi_{dvk} \mathbb{I}_1(v = v')) = \phi_{dvk} \mathbb{I}_1(v = v'). \tag{35}$$

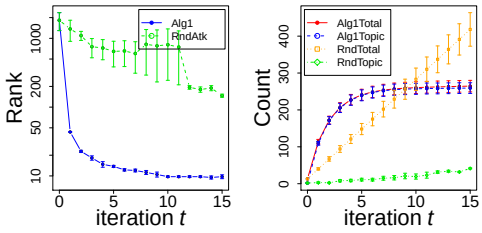
■

We note that in practice Theorem 1's condition does not hold. Nonetheless, Theorem 1 provides an approximation to $\nabla_{\mathbf{M}} \hat{\eta}_{kv}(\mathbf{M})$. In our experiments, this approximation works well.

D Detailed Experiment Results

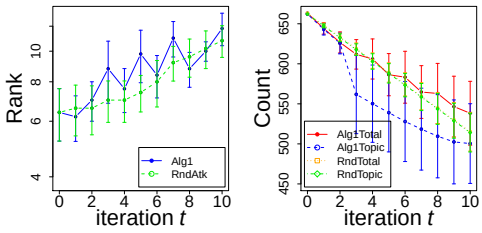
D.1 Rank and Contribution of "ceiling" in Promote-Word Attack

The rank and contribution of word "ceiling" in promote-word attack on AP are shown in Figure 5. The results are very similar as results of "marijuana" and "debt" in promote-word attack shown in main paper.



(a) ceiling rank (b) ceiling contribution

Figure 5: (second-part of) Promote-word attack on word "debt" and "ceiling" in the *market* topic from AP.



(a) Paul rank (b) Paul contribution

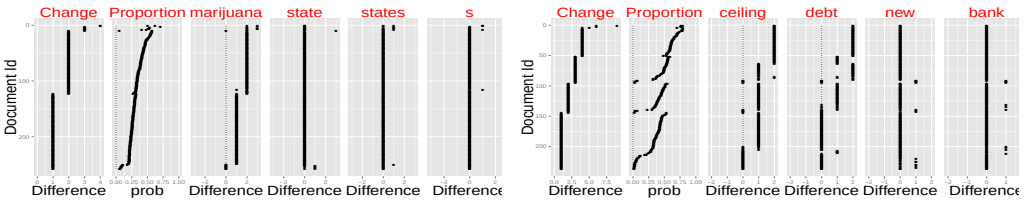
Figure 6: (second-part of) Replace-word attack to replace word "Paul" with "Weasley" in the *president* topic from WISH.

D.2 Rank and Contribution of "Paul" in Replace-Word Attack

We show the rank and contribution of word "Paul" in replace-word attack on WISH in Figure 6. The results are very similar as results of "Iraq" in demote-word attack shown in main paper.

D.3 Detailed Attack Behavior of Promote-Word Attacks

We show the detailed attack behavior of Alg1 in promote-word attack on CONG and AP on documents in Figure 7. The documents are shown from up to down sorted by the decreasing amount of changes defined in Eq (19). For each document d, we show the target topic proportion $\hat{\theta}_{dk} \triangleq \gamma_{dk} / \sum_k \gamma_{dk}$ and the count changes (in the document) of 4 words which have the largest changes on the whole corpus.

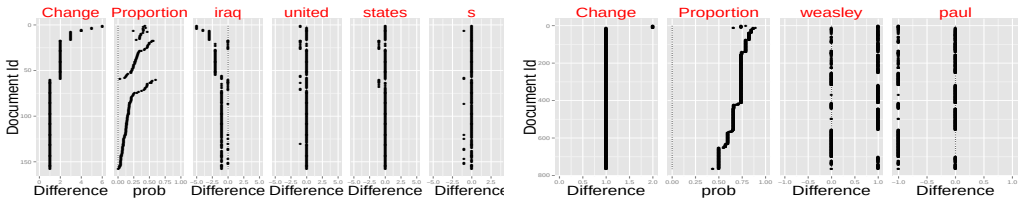


(a) Promote "marijuana" Attack (b) Promote "debt" and "ceiling" Attack

Figure 7: Statistics of attack behavior of promote-word attacks

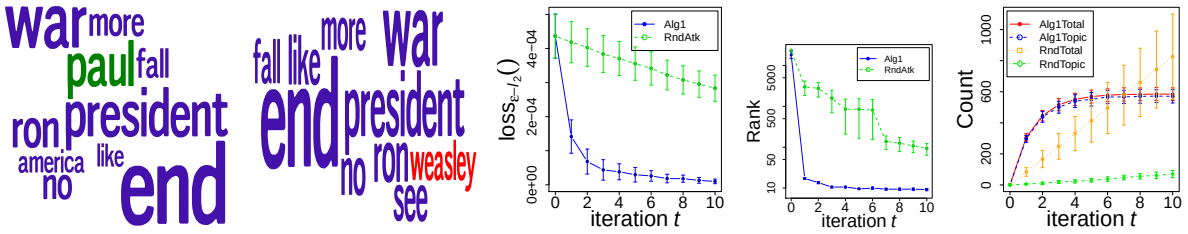
D.4 Detailed Attack Behavior of Demote-Word Attack

We plot the attack behavior of the Alg1 in demote-word attack on CONG in Figure 8(a). Things we show are exactly the same as in promote-word attack.



(a) Demote "Iraq" Attack (b) Replace "Paul" with "Weasley" Attack

Figure 8: Statistics of attack behavior of demote "Iraq" attack and replace "Paul" with "Weasley" attack



(a) before attack (b) after attack (c) objective (d) Weasley rank (e) Weasley contribution

Figure 9: Replace-word attack on "Paul" and "Weasley" in the *president* topic from WISH

D.5 Replace-Word Attack

This kind of attack replaces a top-10 word in the target topic with another word. LDA on the original WISH corpus consistently produced a *president* topic with "Paul" (as in Ron Paul) as a top word. To demonstrate replace-word attack, we replace "Paul" with "Weasley" (as in Ron Weasley of Harry Potter fame). The target encoding is a combination of promotion and demotion with $\varphi^*_{k,\text{paul}} = \varphi_{k,w_{11}}$ and $\varphi^*_{k,\text{weasley}} = \varphi_{k,w_9}$, then renormalize φ^*_k. The RndAtk baseline is also a combination which randomly adds "Weasley" and deletes "Paul", subject to the constraint encoded in $\mathbb{M}$. Similar to previous attacks, Figure 9 shows Alg1's effectiveness in the replace-word attack. In Panel (a,b) Alg1 successfully replaced word "Paul" with "Weasley" in top-10 words. "Paul" ranked 11th after attack. The objective function are optimized rapidly in Panel (c). The rank and contribution of "Weasley" in Panel (d,e) are similar to the promote-word attack, and those of "Paul" are similar to the demote-word attack. Alg1's attack behavior was a combination of promote-word attack and demote-word attack. It mainly replaced "Paul" with "Weasley" in selected documents with high target topic proportion. We plot the attack behavior of the Alg1 in replace-word attack on WISH in Figure 8(b). The settings of things we show are exactly the same as in previous attacks. Modification on only two words "Paul" and "Weasley" is shown because no modification exists on other words.

D.6 Detailed Attack Behavior of Attack with POS Constraint

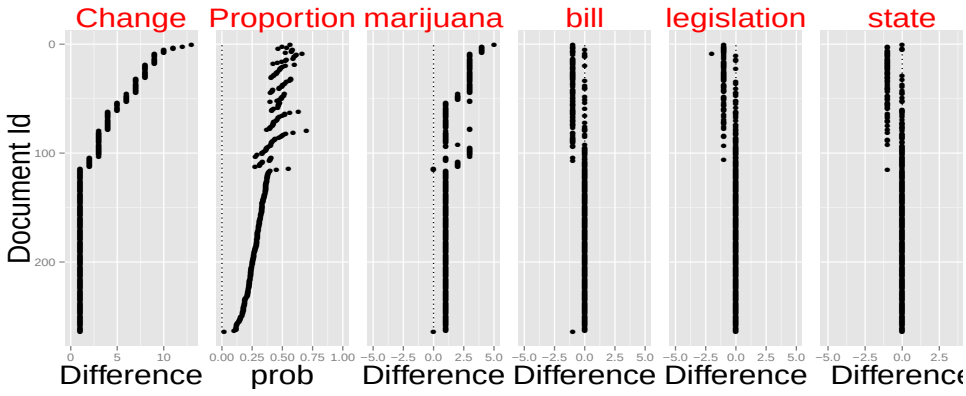


Figure 10: Statistics of attack behavior of attack with POS constraint

We plot the attack behavior of the Alg1 in promote-word attack with POS constraint on CONG in Figure 10. The things we show are exactly the same as in previous attacks.

D.7 Detailed Attack Behavior of Attack with POS Constraint

We plot the attack behavior of the Alg1 in sentence attack on WISH in Figure 11. The things we show are exactly the same as in previous attacks.

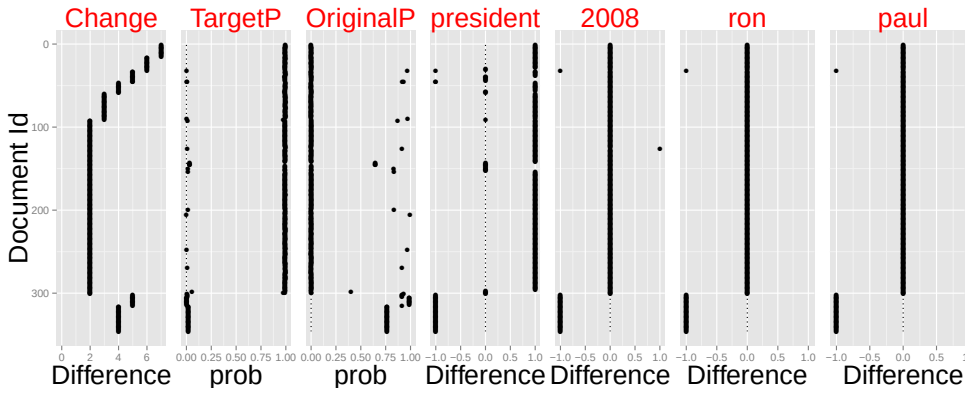


Figure 11: Statistics of attack behavior of attack with sentence attack