ECE543 Project Report: Regret Minimization for News Dissemination

Du Su

May 12, 2017

Abstract

In this report, we study the problem of disseminating a fresh news content, whose topic is yet unknown, to all interested users. The goal is to learn the real topic of a news content from the users' feedback, while spamming a minimum number of uninterested numbers. This problem can be formulated as an online stochastic optimization problem. One of the simplest and most popular stochastic optimization methods is Stochastic Gradient Descent (SGD). In this report We establish a suitably defined regret measure of news dissemination event, and analyze performance under several stochastic learning algorithms including SGD, Averaged-SGD(ASGD). The result shows $O(\sqrt{T})$ upper bound on training regret, and $O(log(T))$ upper bound on expected regret. We also use the training regret to bound of spamming loss caused by spamming event.

1 Introduction

It is a huge challenge to provide personalized, better-targeted information to users. When a online news company saw a piece of news, it needs to decide which users are most interested in it in order to improve information delivery efficiency . The way a user access news online can be generally categorized into pull mode, when users actively access news, or a push mode, when the news is proactively recommended to users by news company. Push systems has been drawing great attention from online social network companies because of its potential profitability. Differ from pull systems, push systems operate on fresh news, i.e. contents that are not fully characterized yet.

In a push system, an efficient way to characterize a piece of news is analyzing users' feedback to it: a news's category is likely to be similar to the preference of users who like the news. To be more specific, we assume known preference of all users, and each user will give positive or negative feedback to a news he saw, based on the news' feature and his preference. Given a set of users' preference together with their feedback, we can characterize the news based on certain estimation method such as maximum likelihood estimation.

During the process of exposing the news to users, some users may not be interested in the news, whom we call "spamming users", which may potentially cause benefit loss of the news delivery company. To stress this problem, one should improve his understanding of the news while receiving users' feedback. In another word, we disseminate the news to a sequence of users based on updating estimation of the news, and in return update estimation of the news accordingly. Intuitively if we can estimate the news with high accuracy in each step, the number of exposed spamming users can be reduced.

The estimation problem described above can be formulated as an online learning problem. [1] studies this problem of single-topic setting, where news is modeled as a τ-d index vector θ^*, which means one element of θ^* is 1 and the others are 0, denoting the one topic out of τ topics that this news is related to. Each user is also represented by a τ-d vector x, showing the user's interest in each topic. [1] studies the case that each news contains exactly one topic, and provide a so called Greedy-Bayes algorithm to extract the topic while

reducing the number of exposed spamming users. [1] provides a $O(\tau log(T))$ upper bound of spamming regret. While this algorithm works very well to single topic case, it cannot be extended to multiple topic case, because it always emphasizes on the most significant topic and neglect others.

In this report, we will focus on the setting of multi-topic news, where news θ^* is modeled as a τ-dimensional real vector, denoting how relevant this news is to each of the τ topics. Note that this vector is unknown at first due to properties of news push system. Other settings are the same as single-topic model. A user's feedback to a news depends on the similarity between his interest x and the news' topic vector θ^*. Based on this, we can learn the news vector by online stochastic learning methods, given a set of users and their feedback. Specifically, We use Stochastic Gradient Descent method and its variants with log likelihood as the loss function to update the estimation.

During the learning process, training loss occurs at each step, and they sum up to training regret. We will further study the performance of stochastic learning algorithms with respect to training regret. [3, 4, 5, 7] provide various bound on the training regret of online learning algorithms, based on necessary properties of the loss function. We thus prove certain properties of log likelihood function and provide bounds of the learning performance based on the result in [3, 4, 5, 7]. We further define a spamming regret similar to training regret and show it can be bounded by $O(\sqrt{T})$, or $O(log(T))$ in expectation, by making use training regret.

2 Online Stochastic Optimization Algorithms

2.1 Problem Description

Our aim is to disseminate a piece of fresh news to a set of interested users, while causing minimum spamming event. In another word, we want to learn the properties of a news while minimizing training regret. Given a news, we can define a τ-dimensional vector, called topic vector, to represent how relevant the news is to each topic. Similarly, we can also define a τ-dimensional vector for each user to represent his interest to each topic, and we call it interest vector. To be more specific, let $\mathcal{F} \subseteq \mathbb{R}^\tau$ be a Hilbert space. Given a news, let $\theta^* \in \mathcal{F}$ denote the topic vector for it, which is unknown at first due to the property of news push system. We try to approximate the topic vector with an estimation $\theta \in \mathcal{F}$. Given a user, let $x \in \mathcal{F}$ denote his interest vector. We further assume $\mathcal{F}$ is bounded and closed, $\mathcal{F} = \{f \in \mathbb{R}^\tau | \|f\| \leq H\}$ for some $H > 0$. And the users are uniformly distributed in $\mathcal{F}$

From now on, we name a news by its topic vector θ^* and name a user by his interest vector x. Given a news θ^* and a user x, we define their similarity η by their dot product, $\eta(\theta^*, x) = x^T\theta^*$. When a user x is exposed to θ^*, he will return a feedback $y \in \{0, 1\}$ depends on similarity η, where $y = 1$ means positive feedback and 0 means negative. Specifically, y is modeled as a Bernoulli distribution, with parameter $P(y = 1) = h(\eta) = h(x^T\theta^*)$, where $h(\eta) = \frac{1}{1+e^{-x^T\theta}}$ is logistic regression function which maps real numbers to the range $[0, 1]$. Intuitively if the similarity η between a user and a news is very high, then $h(\eta) \approx 1$, i.e. the user is likely to like the news; otherwise if it is a negative number with high absolute value, then $h(\eta) \approx 0$, i.e. the user is likely to dislike the news.

With a user x and his feedback y, we can define log likelihood function to measure a news estimation θ by $LLF(\theta, (x, y)) = ylog(h(\theta, x)) + (1 - y)log(1 - h(\theta, x))$, i.e. when the user likes the news, the function means how likely the user will give positive feedback to θ; when the user dislike like the news, the function means how likely the user will return negative feedback to θ. In general, it measures how likely the estimation is the real news vector. In expectation, the real news vector θ^* should maximize the likelihood function.

Given a piece of fresh news with unknow θ^*, the news dissemination process is to continuously do the following: expose the news to a user, get the feedback, update the estimation and then expose to the next user. At the beginning of news dissemination, our initial estimation of θ^* based on our prior knowledge is θ_1. At a disseminate step t, we update our estimation θ_t based on the feedback of the user. After update, we choose the next user that is most likely to return positive feedback. As the user x that is exactly the

same with θ_t is not always available, we ended up select a user that is nearest to θ_t. Formally, we choose a user uniformly from a ball centered at θ_t, with a small diameter ϵ.

We are solving the news dissemination process by online stochastic algorithms, where we need to minimize a loss function based on our observation and current estimation at each step. We define loss function as negative log likelihood function $L(\theta,(x,y)) = -Likelihood(\theta,(x,y)) = -ylog(h(\theta,x)) - (1-y)log(1-h(\theta,x))$, i.e. by minimizing the loss function, we are maximizing the likelihood that the estimation is real. We can calculate the gradient of $L(\theta,(x,y))$ at θ, we represent it as $g(\theta,(x,y)) = \nabla L(\theta,(x,y))$. At step t, we observe the user x_t's feedback y_t, and update our estimation of a news from θ_t to θ_{t+1}. For example, SGD update schema is: $\theta_{t+1} = \theta_t + \alpha_t g(\theta_t,(x_t,y_y))$. With the definitions above, we define training loss at step T as $J_T((\theta_T)) = \sum_{t=1}^{T} L(\theta_t,(x_t,y_y))$, and we can define regret as $R_T = J_T(\theta_t) - min_{\hat{\theta}\epsilon\mathcal{F}} J_T((\hat{\theta}))$.

Our true purpose is to measure how many uninterested users are exposed to the news, and how much loss it may cause. We define a spamming event such that the probability that a user more likeli to return negative feedback, i.e. he likes a news with probability less than 1/2. Based on this definition, we define spamming loss as $L_s(theta^*,x) = max(0,1-2*P(y=1))$. We can then define the total spamming loss at step T as $K((\theta_t)) = \sum_{t=1}^{T} L_s(\theta_t,x_t)$.

The result in this report is to prove upper bound for regret and total spamming loss, and prove almost sure convergence of the distance between estimation and real value.

3 Properties of Loss Function

In the previous session we have defined log likelihood function for a user-feedback pair x,y and an estimation of the news θ, and loss function as negative log likelihood function.

$$LLF(\theta,(x,y)) = ylog(h(\theta,x)) + (1-y)log(1-h(\theta,x)) \tag{1}$$

$$L(\theta,(x,y)) = -LLF(\theta,(x,y)) = -ylog(h(\theta,x)) - (1-y)log(1-h(\theta,x)) \tag{2}$$

In this section we study several properties of the loss function, include convexity, Lipschitz continuous and smoothness, which will be used by latter sessions to provide performance bound.

3.1 Convexity

Lemma 1. *The loss function $L(\theta,(x,y)) = -ylog(h(\theta,x)) - (1-y)log(1-h(\theta,x))$ is a convex function.*

Proof.

$$\begin{aligned} L(\theta,(x,y)) &= -y*log(h(\theta,x)) - (1-y)log(1-h(\theta,x)) \\ &= y*log(\frac{1-h(\theta,x)}{h(\theta,x)}) - log(1-h(\theta,x)) \\ &= -y*\eta + log(1+e^{\eta}) \\ &= L(\eta) \end{aligned} \tag{3}$$

,where $\eta = x^T\theta$.

$$\text{Note} \qquad \frac{1-h(\theta,x)}{h(\theta,x)} = \frac{\frac{e^{-x^T\theta}}{1+e^{-x^\theta}}}{\frac{1}{1+e^{-x^\theta}}} = e^{-x^T\theta}$$

$$1-h(\theta,x) = \frac{e^{-x^T\theta}}{1+e^{-x^T\theta}} = \frac{1}{e^{x^T\theta}+1}$$

Let $A(\eta) = log(1 + e^{\eta})$

$$\frac{dA}{d\eta} = \frac{e^{\eta}}{1 + e^{\eta}} = \frac{1}{1 + e^{-\eta}} \tag{4}$$

$$\frac{d^2A}{d\eta^2} = \frac{e^{-\eta}}{(1 + e^{-\eta})^2} \geq 0 \tag{5}$$

Hence $A(\eta)$ is a convex function of η, and $L(\eta) = -y * \eta + A(\eta)$ is a convex function of η. $L(\theta, (x, y))$ is then convex in θ as η is an affine function of θ □

Note the loss function is not necessarily strong convex because with $x = \vec{0}$, $L(\theta, (x, y)$ is constant to θ. However later on we will prove strong convexity of expected loss function with respect to assumption to x.

3.2 Lipschitz Continuity

Lemma 2. *The loss function $L(\theta, (x, y)) = -ylog(h(\theta, x)) - (1 - y)log(1 - h(\theta, x))$ is H-Lipschitz continuous where H is the maximum length of a vector in $\mathcal{F}$.*

Proof. It is easy to see that $L(\theta, (x, y))$ is everywhere differentiable in $\mathcal{F}$. So it is sufficient to prove $\|\nabla_\theta L(\theta, (x, y))\| \leq H$.

We can get the derivative from (3) and (4) that

$$\nabla_\theta L(\theta, (x, y)) = \frac{dL}{d\eta} * \nabla\eta = x * (\frac{1}{1 + e^{-x^T\theta}} - y) \tag{6}$$

Since $y = 0$ or 1 and $\frac{1}{1+e^{-x^T\theta}} \in (0, 1)$

$$\|\nabla_\theta L(\theta, (x, y))\| \leq \|x\| * 1 = H \tag{7}$$

□

3.3 Smoothness

Lemma 3. *The loss function $L(\theta, (x, y)) = -ylog(h(\theta, x)) - (1 - y)log(1 - h(\theta, x))$ is $\frac{H^2}{4}$-smooth where H is the maximum length of a vector in $\mathcal{F}$.*

Proof. We need to prove $\|\nabla L(\theta, (x, y)) - \nabla L(\theta', (x, y))\| \leq \frac{H^2}{4}\|\theta - \theta'\|$.

$$\begin{aligned}
& \|\nabla L(\theta, (x, y)) - \nabla L(\theta', (x, y))\| \\
= \; & \|x * \frac{1}{1 + e^{-x^T\theta}} - x * \frac{1}{1 + e^{-x^T\theta'}}\| \\
\leq \; & \|x\| * \|\frac{1}{1 + e^{-x^T\theta}} - \frac{1}{1 + e^{-x^T\theta'}}\| \\
\leq \; & \|x\| * max_{\hat{\theta}}\|\nabla\frac{1}{1 + e^{-x^T\hat{\theta}}}\| * \|\theta - \theta'\| \\
\leq \; & \|x\|^2 * max_{\hat{\theta}}\frac{e^{-x^T\hat{\theta}}}{(1 + e^{-x^T\hat{\theta}})^2} * \|\theta - \theta'\|
\end{aligned} \tag{8}$$

$\frac{e^{-x^T\hat{\theta}}}{(1+e^{-x^T\hat{\theta}})^2} = \frac{1}{(1+e^{-x^T\hat{\theta}})(1+e^{x^T\hat{\theta}})}$ is maximized at $x^T\hat{\theta} = 0$, when $\frac{e^{-x^T\hat{\theta}}}{(1+e^{-x^T\hat{\theta}})^2} = 1/4$. Thus we have:

$$
\begin{aligned}
& \|\nabla L(\theta,(x,y)) - \nabla L(\theta',(x,y))\| \\
\leq\ & \|x\|^2 * \frac{1}{4} * \|\theta - \theta'\| \\
\leq\ & \frac{H^2}{4}\|\theta - \theta'\| \qquad (9)
\end{aligned}
$$

□

3.4 Convexity for Expected Log Likelihood Function

We have analyzed properties of negative log likelihood function $L(\theta,(x,y))$. The convexity of $L(\theta,(x,y))$ is somehow week, due to two reasons. First, log logistic regression function $A(\eta) = log(1+e^{\eta})$ itself is not strong convex. $A(\eta)$ is only strong convex in a compact set. Second, even when $A(\eta)$ is convex, $\nabla\eta = x$ vanishes when $\|x\| = 0$. We can solve the first problem because our hypothesis space, $\mathcal{F}$, is bounded. To address the second reason, we can instead exam $E[L(\theta,(x,y))]$. Note that at each step, given θ we are free to choose the next user x. And we hereby assume x is uniformly distributed in the ball centered at θ, with radius ϵ, $\epsilon > 0$. We are able to select the user because we know θ and all users' vectors. This mimic the realistic case that we always want to the user who aligns with θ. However, this user is not always available given a finite set of users. So we use a uniform distribution to approximate the case that we choose the best available user. Also note that the expectation is on the randomness of user selection, and user's feedback, i.e.

$$
\begin{aligned}
E[L(\theta,(x,y))] &= E_x[E_y[L(\theta,(x,y))]] \\
&= E_x[E_y[-y * x^T\theta + log(1+e^{x^T\theta})]] \\
&= E_x[-\frac{1}{1+e^{-x^T\theta^*}} * x^T\theta + log(1+e^{x^T\theta})]
\end{aligned} \qquad (10)
$$

Lemma 4. *The expected loss function* $E[L(\theta,(x,y))] = E_x[E_y[-ylog(h(\theta,x)) - (1-y)log(1-h(\theta,x))]]$ *is* σ*-strongly convex function with respect to* θ. $\sigma = \frac{1}{1+e^{-H^2}} * \int_{\alpha=0}^{\epsilon} \frac{\Gamma(\frac{\tau}{2}+1)*(\epsilon^2-\alpha^2)^{\frac{\tau-1}{2}}*\alpha}{\sqrt{\pi}*\Gamma(\frac{\tau+1}{2})*\epsilon^{\tau}}$

Proof. In section 3.1, we see that $\frac{d^2L}{d\eta^2} = \frac{d^2A}{d\eta^2} = \frac{1}{1+e^{-\eta}}$. Since we assume bounded x and θ, $\eta = x^T\theta \leq H^2$. So $\frac{d^2L}{d\eta^2} = \frac{1}{1+e^{-\eta}} \geq \frac{1}{1+e^{-H^2}}$. Thus $L(\eta)$ is $\frac{1}{1+e^{-H^2}}$-strong convex on η. It follows $E_y[L(\eta)]$ is $\frac{1}{1+e^{-H^2}}$-strong convex on η. Let $\sigma_1 = \frac{1}{1+e^{-H^2}}$

Then we have $E_y[L(\eta_1)] \geq E_y[L(\eta_0)] + \frac{dE_y[L(\eta_0)]}{d\eta+0} * (\eta_1 - \eta_0) + \frac{\sigma_1}{2}|\eta_1 - \eta_0|$. where $\eta_0 = x_0^{\theta}$, $\eta_1 = x^T\theta_1$, we have

$$
\begin{aligned}
E[L(\theta_1,(x,y))] &= E_x[E_y[L(\theta_1,(x,y))]] \\
&\geq E_x[E_y[L(\theta_1,(x,y))] + \frac{dE_y[L(\eta_0)]}{d\eta_0} * (x^T\theta_1 - x\theta_0) + \frac{\sigma_1}{2}|x^T\theta_1 - x^T\theta_0|] \\
&= E_x[E_y[L(\theta_1,(x,y))] + (\nabla E_y[L(\theta_0,(x,y))])^T(\theta_1 - \theta_0) + \frac{\sigma_1}{2}|x^T\theta_1 - x^T\theta_0|] \\
&= E[L(\theta_1,(x,y))] + (\nabla_{\theta_0} E[L(\theta_0,(x,y))])^T(\theta_1 - \theta_0) + \frac{\sigma_1}{2}E_x[|x^T\theta_1 - x^T\theta_0|]
\end{aligned} \qquad (11)
$$

To prove the σ-strong convexity of $E[L(\theta,(x,y))]$, we need to prove that $E[L(\theta_1,(x,y))] \geq E[L(\theta_0,(x,y))] +$

$(\nabla_{\theta_0} E[L(\theta_1,(x,y))])^T(\theta_1-\theta_0)+\frac{\sigma}{2}\|\theta_1-\theta_0\|$. It is sufficient to prove $E_x[|x^T\theta_1 - x^T\theta_0|] \geq \frac{\sigma}{\sigma_1}\|\theta_1-\theta_0\|$. We can represent $x = \theta + \delta$ where δ is uniform in the ball with radius ϵ centered at 0.

$$\begin{aligned}
E_x[|x^T\theta_1 - x^T\theta_0|] &= E_\delta[|(\theta+\delta)^T\theta_1 - (\theta+\delta)^T\theta_0|] \\
&= E_\delta[|\theta^T(\theta_1-\theta_0)+\delta^T(\theta_1-\theta_0)|] \\
&= \int_{\delta\epsilon Uniform\{Ball(0,\epsilon)\}} |\theta^T(\theta_1-\theta_0)+\delta^T(\theta_1-\theta_0)| * p(\delta) \\
&= \int_{\alpha=-\epsilon}^{\epsilon} |\theta^T(\theta_1-\theta_0)+\alpha\|\theta_1-\theta_0\|| * p(\alpha) \\
&= \int_{\alpha=0}^{\epsilon} (|\theta^T(\theta_1-\theta_0)+\alpha\|\theta_1-\theta_0\|| + |\theta^T(\theta_1-\theta_0)-\alpha\|\theta_1-\theta_0\||) * p(\alpha) \\
&\geq \|\theta_1-\theta_0\| \int_{\alpha=0}^{\epsilon} \alpha * p(\alpha)
\end{aligned} \tag{12}$$

The transforming from δ to α is: $\alpha = \delta^T(\theta_0-\theta_1)$. Given $p(\delta) = \frac{1}{V(Ball_\tau(\epsilon))}$, the volume of a τ-dimensional ball with radius equal to ϵ, since this value is the same for all δ project to $\theta_0-\theta_1$, $p(\alpha) = \frac{V(Ball_{\tau-1}(\sqrt{\epsilon^2-\alpha^2}))}{V(Ball_\tau(\epsilon))} = \frac{\Gamma(\frac{\tau}{2}+1)*(\epsilon^2-\alpha^2)^{\frac{\tau-1}{2}}}{\sqrt{\pi}*\Gamma(\frac{\tau+1}{2})*\epsilon^\tau}$

Let $\sigma_2 = \int_{\alpha=0}^{\epsilon} \frac{\Gamma(\frac{\tau}{2}+1)*(\epsilon^2-\alpha^2)^{\frac{\tau-1}{2}}*\alpha}{\sqrt{\pi}*\Gamma(\frac{\tau+1}{2})*\epsilon^\tau}$. We have $E_x[|x^T\theta_1 - x^T\theta_0|] \geq \sigma_2\|\theta_0-\theta_1\|$. Thus we have $\sigma(H,\epsilon) = \sigma_1 * \sigma_2$. Note that σ_1 is a function of H and σ_2 is a function of ϵ □

4 Performance of Online Stochastic Optimization Algorithms

Stochastic gradient descent (SGD) and Averaged stochastic gradient descent (ASGD) are two commonly used and classic online stochastic optimization algorithms. Roughly speaking, both start with an initial estimation of optimum, and then update the estimation in a step-by-step manner.In each step, SGD observe a data, and construct a loss function based on the data and current estimation, and calculate the gradient descent of the loss function with respect to estimation. It update the estimation by a step size factor times the gradient descent. ASGD uses the same updating schema as SGD, but it will use a weighted average of all history estimations to be the final estimation. In ASGD, the training loss is calculated from the weighted estimation instead of the estimation at the last step. For SGD and ASGD, one may use different step size strategy for different loss functions. For ASGD, one may further choose different weighting schema to weight estimations.

Specifically, the SGD and ASGD algorithms for our news dissemination case are as follows

Algorithm 1 Standard SGD

Initialize θ_1.
for $t = 1, 2, \ldots$ **do**
  Draw (x_t, y_t) randomly from $Ball(\theta, \epsilon)$
  Update θ_t as
  $\theta_{t+1} = \Pi(\theta_t - \alpha_t * \nabla_{\theta_t} L(\theta_t, (x_t, y_t)))$
end for
return θ_T.

Averaged-SGD algorithm outperforms standard SGD algorithm with respect training loss, i.e. $L(\theta_T)$ is the loss function has certain convex properties. But in general, the performance of training regret of the two algorithms are in the same order: if the loss function is convex, then the training regret is bounded by $O(\sqrt{T})$; if the loss function is strongly convex, then the training regret is bounded by $O(log(T))$.

Algorithm 2 Averaged SGD

Initialize θ_1.
for $t = 1, 2, \ldots$ **do**
  Draw (x_t, y_t) randomly from $Ball(\theta, \epsilon)$
  Update θ_t as
  $\theta_{t+1} = \Pi(\theta_t - \alpha_t * \nabla_{\theta_t} L(\theta_t, (x_t, y_t)))$
end for
return $\frac{1}{\gamma T} \sum_{t=(1-\gamma)T+1}^{T} \theta_t$.

Specifically, in the case of SGD and convex loss function, [5] shows $R_T \leq H^2\sqrt{T}$, [7] shows $R_T \leq 2H^2\sqrt{2T}$ and [4] shows $R_T \leq 4H^2\sqrt{2T}$; if further the loss function is σ-strongly convex, [7] shows $R_T \leq \frac{1+log(T)}{\sigma}$ and [2] shows $R_T \leq \frac{H^2(1+log(T))}{2\sigma}$. In the case for ASGD, [5] shows $R_T \leq O(\sqrt{T})$ for convex loss function and [4] shows $R_T \leq \frac{17H^2}{\sigma} log(T)$

5 Strong Convexity in Expected Regret Proof

When considering the regret at step T, one may use the definition $J_T((\theta_T)) = \sum_{t=1}^{T} L(\theta_t, (x_t, y_y))$, and we can define regret as $R_T((\theta_t)) = J_T((\theta_t)) - min_{\hat{\theta} \epsilon \mathcal{F}} J_T((\hat{\theta}))$. To prove a bound for R_T, one usually need to assume convexity of loss function L. As we can see above, sometime L only has very weak convexity, but the expectation loss $E[L]$ has strong convexity. We define expected regret $ER_T((\theta_t)) = E[J_T((\theta_t))] - min_{\hat{\theta} \epsilon \mathcal{F}} E[J_T((\hat{\theta}))]$. This is different from the expectation of regret, which is $E[R_T((\theta_t))] = E[J_T((\theta_t))] - E[min_{\hat{\theta} \epsilon \mathcal{F}} J_T((\hat{\theta}))]$. Given the expectation of regret, we cannot do better than the previous proof. But when considering the expected regret, we can see later that we can get a tighter bound by discovering the strong convexity of each step. Actually, the difference between ER_t and $E[R_T]$ is from the fixed strategy term. In ER_T it is the minimized value of expected loss; while in $E[R_T]$, it is the expected minimized loss. It is obvious that the expected minimization is less than minimized expected value, this is where the difference come from. One may wonder why expected is valid: let's go to our news-user case, we can see later that the minimizer for the expected loss is θ^*, which is the real news. But in the origin definition of $E[min_{\hat{\theta} \epsilon \mathcal{F}} J_T((\hat{\theta}))]$, the minimizer for each random observation is not necessarily $\theta*$.

More specifically: $E[J_T((\hat{\theta}))] = \sum_{t=1}^{T} E[L(\hat{\theta}, (x_t, y_t))] = \sum_{t=1}^{T} E_x[-\frac{1}{1+e^{-x^T\theta^*}} * x^T\hat{\theta} + log(1 + e^{x^T\hat{\theta}})]$ for each term in the summation, the derivative is $(\frac{1}{1+e^{-x^T\hat{\theta}}} - \frac{1}{1+e^{-x^T\theta^*}}) * x$ which is zero at $\hat{\theta} = \theta^*$ if $x \neq 0$. So $argmin_{\hat{\theta}} E[J_T((\hat{\theta}))] = \theta^*$. Let's then consider J_T with $T = 1$ $J_1((\hat{\theta})) = L(\hat{\theta}, (x_1, y_1)) = -y_1 * log(h(\theta, x_1)) - (1 - y_1) log(1 - h(\theta, x_1))$. When $y_1 = 1$, it is minimized at $\hat{\theta} = x_1 * \frac{H}{\|x_1\|}$, when $y_1 = 0$, it is minimized at $\hat{\theta} = -x_1 * \frac{H}{\|x_1\|}$.

If we assume that at each step. an oracle gives the gradient of expected loss function, then our regret bound can be tighter due to strong convexity. This oracle is not available in practice, but we will see this is not a problem. Here I will go through the convergence proof in [2, 7] to demonstrate this point.

Let $\theta_{t+1}^b = \theta_t - \alpha_t \nabla L(\theta_t, (x_t, y_t))$, and $\theta_{t+1} = \Pi(\theta_{t+1}^b)$. Let θ^* be the real value of the news, i.e. the fixed strategy that minimize $E[L(\theta, (x, y))]$.

by contraction property of Π, and H-Lipschitz continuity of L, we have

$$\|\theta_{t+1} - \theta^*\|^2 \leq \|\theta_t - \theta^*\|^2 - 2\alpha_t * \nabla_{\theta_t} L(\theta_t, (x_t, y_t))^T (\theta_t - \theta^*) + \alpha_t^2 H^2$$

. or equivalently:

$$2\nabla_{\theta_t} L(\theta_t, (x_t, y_t))^T (\theta_t - \theta^*) \leq \frac{\|\theta_t - \theta^*\|^2 - \|\theta_{t+1} - \theta^*\|^2}{\alpha_t} + \alpha_t H^2$$

.

The strong convexity of $E[L(\theta,(x,y))]$ implies that

$$\begin{aligned} 2(E[L(\theta_t)]-E[L(\theta^*)]) &\leq E_{\theta_t}[2\nabla_{\theta_t}E_{x_t}[L(\theta_t,(x_t,y_t))|\theta_t]^T(\theta_t-\theta^*)-\sigma\|\theta_t-\theta^*\|] \\ &\leq \frac{E[\|\theta_t-\theta^*\|^2]-E[\|\theta_{t+1}-\theta^*\|^2]}{\alpha_t}+\alpha_t H^2-\sigma E[\|\theta_t-\theta^*\|] \end{aligned} \tag{13}$$

Note that $\mathcal{F}$ is a ball with radius H, thus its diameter is $D=2H$. Summing each side of (13) from $t=1$ to T yields:

$$\begin{aligned} 2(E[J_T((\theta_t))]-E[J_T(\theta^*)]) &\leq (\frac{1}{\alpha_1}-\sigma)E[\|\theta_1-\theta^*\|]-\frac{1}{\alpha_T}E[\|\theta_{T+1}-\theta^*\|]+\sum_{t=1}^{T-1}(\frac{1}{\alpha_{t+1}}-\frac{1}{\alpha_t}-\sigma)E[\|\theta_{t+1}-\theta^*\|]+H^2\sum_{t=1}^{T}\alpha_t \\ &\leq D^2(\frac{1}{\alpha_T}-\sigma T)+H^2\sum_{t=1}^{T}\alpha_t \\ &= 4H^2(\frac{1}{\alpha_T}-\sigma T)+H^2\sum_{t=1}^{T}\alpha_t \end{aligned} \tag{14}$$

Let $\alpha_t=\frac{1}{\sigma T}$, we have

$$\begin{aligned} 2(E[J_T((\theta_t))]-E[J_T(\theta^*)]) &\leq 4H^2(\frac{1}{\alpha_T}-\sigma T)+H^2\sum_{t=1}^{T}\alpha_t \\ &= H^2\sum_{t=1}^{T}\alpha_t \\ &\leq \frac{1+log(T)}{\sigma} \end{aligned} \tag{15}$$

6 Bound Spamming Regret by Training Regret

In section 2.1, we have define training regret as $R_T=J_T(\theta_t)-min_{\hat{\theta}\epsilon\mathcal{F}}J_T((\hat{\theta}))$. And also spamming loss at step T as $K((\theta_t))=\sum_{t=1}^{T}L_s(\theta_t)$. The training loss measures how well our estimator performs on the sampled data compared to the best fixed strategy. As our sampled data depends on current estimation, an inaccurate estimation may lead to a user that is unlikely to like the news. So we want to measure at each step, how likely the user which is equal to our estimation may cause a spamming event. At step t, our estimation ls θ_t. If a user $x=\theta_t$, the Bernoulli feedback parameter $p=h(\theta^*,\theta_t)$. We hope to see that $p>1/2$, which means the user will likely to give positive feedback. Thus we use $1/2$ as a threshold and define $L_s(\theta)=max(0,1-2*h(\theta^*,\theta))$.

We will bound $L_s(\theta_t)$ by $E[L(\theta_t,(x_t,y_t))-L(\theta^*,(x_t,y_t))]$ in order to bound the total spamming loss K by expected training regret ER_t which is in the order of $O(log(T))$.

$L_s(\theta)=max(0,1-2*\frac{1}{1+e^{-\theta^T\theta^*}})$

$E[L(\theta,(x,y))-L(\theta^*,(x_t,y_t))]=E_x[\frac{1}{1+e^{-x^T\theta^*}}x^T(\theta^*-\theta)+log(\frac{1+e^{x^T\theta}}{1+e^{x^T\theta^*}})]$.

As is shown above, $E[L(\theta,(x,y))]$ is convex and is minimized at θ^*. Thus $E[L(\theta,(x,y))-L(\theta^*,(x_t,y_t))]\geq 0$. L_s is always 0 when $\theta^T\theta^*>0$, thus it is suffice to prove bound for θ such that $\theta^T\theta^*<0$. Let $\theta=\theta_\parallel+\theta_\perp$ where $\theta_\parallel^T\theta^*=\|\theta_\parallel\|\|\theta^*\|$ and $\theta_\perp^T\theta^*=0$.

Without loss of generality, we can fix the case that $\|\theta^*\|=1$. Thus $L_s(\theta)=1-2*\frac{1}{1+e^{-\|\theta_\parallel\|}}=L_s(\|\theta_\parallel\|)$. $\frac{dL_s(\|\theta_\parallel\|)}{d\|\theta_\parallel\|}=-2*\frac{e^{-\|\theta_\parallel\|}}{(1+e^{-\|\theta_\parallel\|})^2}$. $|\frac{dL_s(\|\theta_\parallel\|)}{d\|\theta_\parallel\|}|=2|\frac{1}{(1+e^{-\|\theta_\parallel\|})(1+e^{\|\theta_\parallel\|})}|\leq\frac{1}{2}$. $L_s(\theta_\parallel)$ is equal to zero if $\|\theta_\parallel\|=0$, and it is decreasing with respect to $\|\theta_\parallel\|$ with bounded derivative.

We then exam $E[L(\theta,(x,y)) - L(\theta^*,(x,y))]$. It is know that $E[L(\theta,(x,y)) - L(\theta^*,(x,y))] \geq 0$ when $\|\theta_\parallel\| = 0$. So it is sufficient to prove $E[L(\theta,(x,y)) - L(\theta^*,(x,y))]$ is decreasing on $\|\theta_\parallel\|$ with the norm of derivative lower bounded. Indeed:

$$\begin{aligned}
\nabla E[L(\theta,(x,y)) - L(\theta^*,(x,y))] &= E[\nabla L(\theta,(x,y))] \\
&= E[\nabla - yx^T\theta + log(1+e^{x^T\theta})] \\
&= E[x(-y + \frac{1}{1+e^{-x^T\theta}})] \\
&= E_x[x(\frac{1}{1+e^{-x^T\theta}} - \frac{1}{1+e^{-x^T\theta^*}})] \\
&= \int_{\delta \epsilon Unifom\{Ball(0,\epsilon)\}} (\theta+\delta)(\frac{1}{1+e^{-(\theta+\delta)^T\theta}} - \frac{1}{1+e^{-(\theta+\delta)^T\theta^*}})
\end{aligned} \tag{16}$$

We now exam the gradient along the reverse direction of θ^* over θ. Since $\theta^T\theta^* < 0$, it means the gradient along the direction of $\theta_\parallel$. Let $\delta = \delta_\parallel + \delta_\perp$ where $\delta_\parallel^T\delta^* = \|\delta_\parallel\|\|\delta^*\|$ and $\delta_\perp^T\delta^* = 0$.

$$\begin{aligned}
&\frac{-\theta^*}{\|\theta^*\|}^T \nabla E[L(\theta,(x,y)) - L(\theta^*,(x,y))] \\
= &\int_{\delta \epsilon Unifom\{Ball(0,\epsilon)\}} \frac{-\theta^*}{\|\theta^*\|}^T (\theta_\parallel+\delta)(\frac{1}{1+e^{-(\theta+\delta)^T\theta}} - \frac{1}{1+e^{-(\theta+\delta)^T\theta^*}}) \\
= &\int_{\delta \epsilon Unifom\{Ball(0,\epsilon)\}} \frac{-\theta^*}{\|\theta^*\|}^T (\theta_\parallel+\delta_\parallel)(\frac{1}{1+e^{-(\theta_\parallel+\delta_\parallel)^T\theta_\parallel-(\theta_\perp+\delta_\perp)^T\theta_\perp}} - \frac{1}{1+e^{-(\theta_\parallel+\delta_\parallel)^T\theta^*}}) \\
= &\int_{\delta_\parallel}\int_{\delta_\perp} \frac{-\theta^*}{\|\theta^*\|}^T (\theta_\parallel+\delta_\parallel)(\frac{1}{1+e^{-(\theta_\parallel+\delta_\parallel)^T\theta_\parallel-(\theta_\perp+\delta_\perp)^T\theta_\perp}} - \frac{1}{1+e^{-(\theta_\parallel+\delta_\parallel)^T\theta^*}}) \\
= &\int_{\delta_\parallel}\int_{\delta_\perp} \frac{-\theta^*}{\|\theta^*\|}^T (\theta_\parallel+\delta_\parallel)(\frac{1}{2}(\frac{1}{1+e^{-(\theta_\parallel+\delta_\parallel)^T\theta_\parallel-(\theta_\perp+\delta_\perp)^T\theta_\perp}} + \frac{1}{1+e^{-(\theta_\parallel+\delta_\parallel)^T\theta_\parallel-(\theta_\perp+-\delta_\perp)^T\theta_\perp}}) - \frac{1}{1+e^{-(\theta_\parallel+\delta_\parallel)^T\theta^*}}) \\
= &\int_{\delta_\parallel}\int_{\delta_\perp} \frac{-\theta^*}{\|\theta^*\|}^T (\theta_\parallel+\delta_\parallel)(\frac{1}{4}(\frac{1}{1+e^{-(\theta_\parallel+\delta_\parallel)^T\theta_\parallel-(\theta_\perp+\delta_\perp)^T\theta_\perp}} + \frac{1}{1+e^{-(\theta_\parallel+\delta_\parallel)^T\theta_\parallel-(\theta_\perp-\delta_\perp)^T\theta_\perp}} \\
&+\frac{1}{1+e^{-(\theta_\parallel-\delta_\parallel)^T\theta_\parallel-(\theta_\perp+\delta_\perp)^T\theta_\perp}} + \frac{1}{1+e^{-(\theta_\parallel-\delta_\parallel)^T\theta_\parallel-(\theta_\perp-\delta_\perp)^T\theta_\perp}}) - \frac{1}{2}(\frac{1}{1+e^{-(\theta_\parallel+\delta_\parallel)^T\theta^*}} + \frac{1}{1+e^{-(\theta_\parallel-\delta_\parallel)^T\theta^*}}))) \\
> &\int_{\delta_\parallel}\int_{\delta_\perp} \frac{-\theta^*}{\|\theta^*\|}^T (\theta_\parallel+\delta_\parallel)(1/2 - \frac{1}{1+e^{-(\theta_\parallel+\delta_\parallel)^T\theta^*}}) \\
= &\ c
\end{aligned} \tag{17}$$

Note that we use the fact that $\frac{1}{1+e^{-(\theta_\parallel+\delta_\parallel)^T\theta_\parallel-(\theta_\perp+\delta_\perp)^T\theta_\perp}} + \frac{1}{1+e^{-(\theta_\parallel+\delta_\parallel)^T\theta_\parallel-(\theta_\perp-\delta_\perp)^T\theta_\perp}} + \frac{1}{1+e^{-(\theta_\parallel-\delta_\parallel)^T\theta_\parallel-(\theta_\perp+\delta_\perp)^T\theta_\perp}} + \frac{1}{1+e^{-(\theta_\parallel-\delta_\parallel)^T\theta_\parallel-(\theta_\perp-\delta_\perp)^T\theta_\perp}} > 2$. Let $\alpha = e^{-\theta_\parallel^T\theta_\parallel-\theta_\perp^T\theta_\perp}$, $\beta = e^{-\delta_\parallel^T\theta_\parallel}$, $\gamma = e^{-\delta_\perp^T\theta_\perp}$. Then the above can be written as $\frac{1}{1+\alpha\beta\gamma} + \frac{1}{1+\alpha\beta^{-1}\gamma^{-1}} + \frac{1}{1+\alpha\beta^{-1}\gamma} + \frac{1}{1+\alpha\beta\gamma^{-1}}$. When $a < 1$, $\frac{1}{1+ab} + \frac{1}{1+ab^{-1}} > 1$, thus $\frac{1}{1+\alpha\beta\gamma} + \frac{1}{1+\alpha\beta^{-1}\gamma^{-1}} + \frac{1}{1+\alpha\beta^{-1}\gamma} + \frac{1}{1+\alpha\beta\gamma^{-1}} > 2$. We also use the fact that $\frac{1}{1+e^{-(\theta_\parallel+\delta_\parallel)^T\theta^*}} + \frac{1}{1+e^{-(\theta_\parallel-\delta_\parallel)^T\theta^*}} < 2$ Based on similar analysis.

Thus we can assume an lower bound c of $\frac{-\theta^*}{\|\theta^*\|}^T \nabla E[L(\theta,(x,y)) - L(\theta^*,(x,y))]$ that only depends on ϵ and is larger than 0.

Thus we have $E[L(\theta,(x,y)) - L(\theta^*,(x,y))] \geq 2c * L_s(\theta)$. Which follows $K_T((\theta_t)) = \sum_{t=1}^T L_s(\theta_t) \leq \frac{1}{2c}R_T((\theta_t))$

7 Conclusion

In this project, we studied the news dissemination problem. We used stochastic gradient descent as learning algorithm and use negative log likelihood function as training loss function. By analyzing properties of this loss function and surveying works on training regret bound, we proved that the training regret under such loss function is bounded by $O(\sqrt{T})$, and the expected training regret is bounded by $O(log(T))$. By selecting the nearest user based on estimated news, we can prove that the expected spamming loss is bounded by the order $O(logT)$

References

[1] Massouli, Laurent, Mesrob I. Ohannessian, and Alexandre Proutiere. "Greedy-Bayes for targeted news dissemination." ACM SIGMETRICS Performance Evaluation Review. Vol. 43. No. 1. ACM, 2015.

[2] Hazan, Elad, Amit Agarwal, and Satyen Kale. "Logarithmic regret algorithms for online convex optimization." Machine Learning 69.2 (2007): 169-192.

[3] Hazan, Elad, and Satyen Kale. "Beyond the regret minimization barrier: optimal algorithms for stochastic strongly-convex optimization." Journal of Machine Learning Research 15.1 (2014): 2489-2512.

[4] Shamir, Ohad, and Tong Zhang. "Stochastic Gradient Descent for Non-smooth Optimization: Convergence Results and Optimal Averaging Schemes." ICML (1). 2013.

[5] Zhang, Tong. "Solving large scale linear prediction problems using stochastic gradient descent algorithms." Proceedings of the twenty-first international conference on Machine learning. ACM, 2004.

[6] ECE543 Spring'17 5th course note

[7] ECE543 Spring'17 6th course note