Analysis of Pegasos Algorithm

Wenxuan Zhou

1 Introduction

Support Vector Machine (SVM) is a supervised classification model. Given a set of labeled examples, the goal of SVM is to divide them by a gap which is as wide as possible.
Formally, consider a concept learning problem $\big(X = R^d, Y = \{\pm 1\}, \mathcal{P}, \mathcal{F}\big)$, where $\mathcal{P}$ is a set of probability distributions on $\mathcal{Z} = \mathcal{R}^d \times \{\pm 1\}$, $\mathcal{F}$ is the set of all half-space classifiers of the form $f_w(x) = \langle w, x\rangle$, the loss function is defined as $l_w(x, y) = \big(1 - y\langle w, x\rangle\big)_+$. The soft-margin SVM is a minimizer of

$$g(w, \mathcal{Z}^n) = \frac{1}{n}\sum_{i=1}^{n} l_i \ + \ \frac{\lambda}{2}||w||^2 \tag{1}$$

An alternative form is

$$\begin{aligned} g(w, \mathcal{Z}^n) = \frac{1}{n}\sum_{i=1}^{n} \xi_i \ + \ \frac{\lambda}{2}||w||^2 & \qquad (2)\\ s.t. \qquad y_i\langle w, x_i\rangle \ &\geq \ 1 - \xi_i \\ \xi_i \ &\geq \ 0\,,\ \forall i \end{aligned}$$

The classical methods of solving SVM is to convert it to a quadratic programming problem, but this kind of methods is usually very slow and cannot be used in large dataset. Pegasos is a popular SVM solving algorithm, one important property is the testing error is invariant w.r.t. the data size. In this report, we'll show and prove the error bound of Pegasos.

2 Algorithm

Pegasos is an online learning algorithm for SVM. It performs stochastic gradient algorithm on objective Eq. (1). The algorithm initializes $w_1 = 0$. Then on the i^{th} iteration, first choose a random example (x_{i_t}, y_{i_t}), then update the weight based on the sub-gradient

$$\nabla_t = \lambda w_t - 1_{\{y_{i_t}\langle w_t, x_{i_t}\rangle < 1\}} y_{i_t} x_{i_t} \tag{3}$$

After T iterations, output w_{T+1} as the result.

Algorithm 1 Pegasos

Input: S, λ, T
  Set $w_1 = 0$
  for $t = 1, 2, \ldots, T$ **do**
    Choose $i_t \in \{1, \ldots, |S|\}$ uniformly at random.
    Set $\eta_t = \frac{1}{\lambda t}$
    if $y_{i_t} \langle w_t, x_{i_t} \rangle < 1$ **then**
      $w_{t+1} = (1 - \eta_t \lambda) w_t + \eta_t y_{i_t} x_{i_t}$
    else
      $w_{t+1} = (1 - \eta_t \lambda) w_t$
    end if
  end for
Output: w_{T+1}

3 Performance Analysis

Consider $f : S \times Z \to [0, B]$, define $f_z(w) = f(w, z)$. Assume f_z is L-Lipschitz and λ-strongly convex. Let $L(w) = E_z[f(w, z)]$. Define

$$\begin{aligned} R_T &= \sum_{i=1}^{T} f(w_t, Z_t) - \sum_{t=1}^{T} f(w^*; Z_t) \\ D_T &= \sum_{i=1}^{T} (L(w_t) - L(w^*)) \\ \xi_t &= L(w_t) - L(w^*) - (f(w_t; Z_t) - f(w^*; Z_t)) \end{aligned} \tag{4}$$

Lemma 3.1. *Let ξ_t be the random sequence defined in Eq. (5). Then*

$$Var[\xi_t] \leq \frac{2L^2}{\lambda} (L(w_t) - L(w^*)).$$

Proof.

$$\begin{aligned} Var[\xi_t] &\leq E[(f(w_t; Z_t) - f(w^*; Z_t))^2] \\ &\leq E[L^2 ||w_t - w^*||^2] \\ &= L^2 ||w_t - w^*||^2 \end{aligned} \tag{5}$$

Because $f_z(w)$ is λ strongly-convex

$$\begin{aligned} f_z(w_t) &\geq f_z(w_t^*) + \frac{\lambda}{2} ||w_t - w_t^*||^2 \\ f_z(w^*) &\geq f_z(w_t^*) + \frac{\lambda}{2} ||w^* - w_t^*||^2 \end{aligned}$$

Then

$$\begin{aligned} f_z(w^*) + f_z(w_t) &\geq \frac{\lambda}{2}||w_t - w_t^*||^2 \\ L(w^*) + L(w_t) &\geq \frac{\lambda}{2}||w_t - w_t^*||^2 \end{aligned} \tag{6}$$

Combining Eq. (6) and (7) derives the result.

□

Lemma 3.2. *Suppose $X_1, \ldots, X_T$ is a martingale difference sequence with $|X_t| \leq b$. Let*

$$Var_t X_t = Var(X_t | X_1, \ldots, X_{t-1}).$$

Let $\sigma = \sqrt{\sum_{t=1}^{T} Var_t X_t}$. Then we have, for any $\delta < \frac{1}{e}$ and $T \geq 3$,

$$Prob\Big(\sum_{t=1}^{T} X_t > max \left\{ 2\sigma, 3b\sqrt{ln(\frac{1}{\delta})} \right\} \Big) \;\leq\; 4ln(T)\delta$$

This lemma is an inference from Freedman's inequality. Then we have the following theorem.

Theorem 3.3. *With probability at least $1 - 4ln(T)\delta$.*

$$\frac{D_T}{T} \leq \frac{R_T}{T} + 2\sqrt{\frac{2L^2 ln(\frac{1}{\delta})}{\lambda}} \frac{\sqrt{R_T}}{T} + max \left\{ \frac{8L^2}{\lambda}, 6B \right\} \frac{ln(\frac{1}{\delta})}{T}$$

Proof. By Lemma 2.1, we have $\sigma \leq \sqrt{\frac{2L^2}{\lambda} D_T}$. $|\xi_t| \leq 2B$. Then by Lemma 2.2, with probability $1 - 4ln(T)\delta$,

$$\sum_{t=1}^{T} \xi_i \;\leq\; \max \left\{ 2\sigma, 6B\sqrt{ln(\frac{1}{\delta})} \right\} \sqrt{ln(\frac{1}{\delta})}.$$

Therefore, with probability $1 - 4ln(T)\delta$

$$D_T - R_T \;\leq\; \max \left\{ 2\sqrt{\frac{2L^2}{\lambda} D_T}, 6B\sqrt{ln(\frac{1}{\delta})} \right\} \sqrt{ln(\frac{1}{\delta})}.$$

Solving for D_T derives the result. □

Theorem 3.4. *If a projected gradient descent algorithm on f runs with step size $a_t = \frac{1}{\lambda t}$ for $t \geq 1$, then*

$$R_T((f_t)) \leq \frac{L^2(1 + logT)}{2\lambda}$$

Lemma 3.5. *Assume* $||x||_2 \leq R$. *Let* $v_t = 1_{\{y_t\langle w_t, x_t\rangle < 1\}} y_t x_t$, *then*

$$\begin{aligned} w_{t+1} &= -\frac{1}{\lambda t}\sum_{i=1}^{t} v_i \\ ||w_{t+1}|| &\leq \frac{R}{\lambda} \end{aligned}$$

From Lemme 2.5, we can know $||\nabla_t|| \leq 2R$. So the object function $g(w; Z_n)$ is λ-strongly convex and $2R$-Lipschitz, the upper bound is $B = \frac{3R^2}{2\lambda} + 1$. So we have the following theorem.

Theorem 3.6 (The Generalization Bound for Pegasos Algorithm)**.** *For the sequence* $w_1, \ldots, w_n$ *generated by the Pegasos Algorithm, with probability at least* $1 - 4ln(T)\delta$,

$$\frac{D_T}{T} \leq \frac{2R^2(1+lnT)}{\lambda T} + 8R^2\frac{\sqrt{1+lnT}}{\lambda T}\sqrt{ln(\frac{1}{\delta})} + max\left\{\frac{32R^2}{\lambda}, \frac{9R^2}{\lambda} + 6\right\}\frac{ln(\frac{1}{\delta})}{T}$$

Corollary 3.6.1. *Assume* $R = 1$. *For* λ *small enough, with probability at least* $1 - \delta$,

$$\frac{D_T}{T} = O(\frac{ln(\frac{T}{\delta})}{\lambda T})$$

Corollary 3.6.2 (Extension from Jensen's inequality)**.** *Assume* $R = 1$. *Let* $\bar{w} = \frac{1}{T}\sum_{i=1}^{T} w_i$. *For* λ *small enough, with probability at least* $1 - \delta$,

$$L(\bar{w}) - L(w^*) = O(\frac{ln(\frac{T}{\delta})}{\lambda T})$$

Corollary 3.6.3 (Extension from Markov's inequality)**.** *Assume* $R = 1$ *If* t *is randomly selected from* $[T]$. *For* λ *small enough, with probability at least* $\frac{1}{2}$,

$$L(w_t) - L(w^*) = O(\frac{ln(\frac{T}{\delta})}{\lambda T})$$

Remark. The corollary implies that if we run the Pegasos Algorithm at stop at a random position, with probability at least $\frac{1}{2}$, the generalization error will be small.

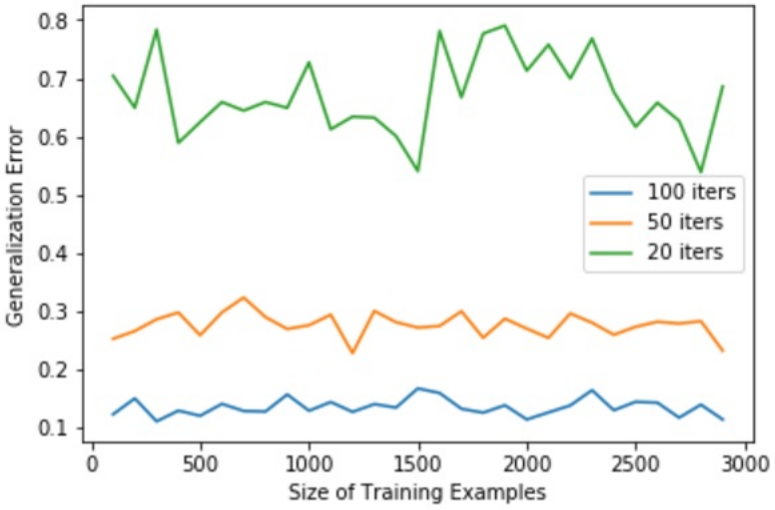
The above bounds are irrelevant with the data size n. So for Pegasos Algorithm, the number of iterations needed does not scale with the data size.
Because $L(w_t)$ is up-bounded, we have the following theorem.

Theorem 3.7. *Assume* $R = 1$ *and* λ *is small enough, then*

$$E[\frac{D_T}{T}] = O(\frac{lnT}{\lambda T})$$

4 Experiment

In this section we'll check the generalization ability of Pegasos on nearly linearly separable data. The data is generated by a Gaussian distribution and the noise rate is 10%. The result is shown is the following graph.



From the graph, we can see the generalization error is invariant with the data size. And the error is inverse proportional to the number of iterations. When the number of iterations is 100, the algorithm achieves the optimal performance, which is very efficient.

5 Conclusion

In this report we prove and examine the error bound of Pegasos algorithm is $O(\frac{lnT}{\lambda T})$ with probability at least $\frac{1}{2}$. In practice it almost ensures a low error, which needs further study to give a tighter bound.

References

[1] Shalev-Shwartz, Shai, Yoram Singer, and Nathan Srebro. "Pegasos: Primal estimated sub-gradient solver for svm." Proceedings of the 24th international conference on Machine learning. ACM, 2007.

[2] Kakade, Sham M., and Ambuj Tewari. "On the generalization ability of online strongly convex programming algorithms." Advances in Neural Information Processing Systems. 2009.

[3] Shalev-Shwartz, Shai, and Nathan Srebro. "SVM optimization: inverse dependence on training set size." Proceedings of the 25th international conference on Machine learning. ACM, 2008.