Notes on Pegasos

Karl Stratos

1 Pegasos Algorithm

Given $x \in \mathbb{R}^d$ and $y \in \{\pm 1\}$, the per-example loss for a linear SVM and its gradient are

$$\begin{aligned} J_{x,y}(w) &= \frac{\lambda}{2} \left\|w\right\|^2 + \max(0, 1 - y \langle w, x \rangle) \\ \nabla J_{x,y}(w) &= \lambda w - [[y \langle w, x \rangle < 1]]\, yx \end{aligned}$$

where $\lambda > 0$ is a strictly positive regularization strength and $[[A]]$ is 1 if A is true and 0 otherwise. The Pegasos algorithm (Shalev-Shwartz *et al.*, 2011) is "just" a stochastic (sub)gradient descent on this loss with a paricular choice of learning rate: $\eta_t = 1/(\lambda t)$. With this choice, the t-th update on an example (x_{i_t}, y_{i_t}) becomes

$$\begin{aligned} w_{t+1} &= w_t - \eta_t \nabla J(w_t) \\ &= w_t - \eta_t \left(\lambda w_t - [[y_{i_t} \langle w_t, x_{i_t} \rangle < 1]]\, y_{i_t} x_{i_t}\right) \\ &= w_t - \frac{1}{t} w_t + \frac{1}{\lambda t} [[y_{i_t} \langle w_t, x_{i_t} \rangle < 1]]\, y_{i_t} x_{i_t} \\ &= \left(\frac{t-1}{t}\right) w_t + \frac{1}{\lambda t} v_t \end{aligned}$$

where we define $v_t = [[y_{i_t} \langle w_t, x_{i_t} \rangle < 1]]\, y_{i_t} x_{i_t}$. Unwinding the expression from an initial parameter of zeros reveals a strikingly simple form:

$$\begin{aligned} w_1 &= 0_d \\ w_2 &= \frac{1}{\lambda} v_1 \\ w_3 &= \frac{1}{2}\left(\frac{1}{\lambda} v_1\right) + \frac{1}{2\lambda} v_2 = \frac{1}{2\lambda}(v_1 + v_2) \\ w_4 &= \frac{2}{3}\left(\frac{1}{2\lambda}(v_1 + v_2)\right) + \frac{1}{3\lambda} v_3 = \frac{1}{3\lambda}(v_1 + v_2 + v_3) \\ w_{t+1} &= \frac{1}{t\lambda} \sum_{l=1}^{t} v_l \qquad \forall t \geq 1 \qquad (1) \end{aligned}$$

Now consider N training examples $(x_1, y_1) \ldots (x_N, y_N) \in \mathbb{R}^d \times \{\pm 1\}$ and for step $t = 1, 2, \ldots, T$ draw an example $i_t \in \{1 \ldots N\}$ arbitrarily. Then the update (1) is equivalent to

$$w_{t+1} = \frac{1}{t\lambda} \sum_{i=1}^{N} \alpha(i, t) y_i x_i \qquad (2)$$

where $\alpha(i, t) = \sum_{l=1}^{t} [[y_i \langle w_t, x_i \rangle < 1]]$ is the number of times the margin constraint has been violated on the i-th example at the t-th update.

1.1 Kernelized Pegasos

A highlight of the form of update (2) is that it allows for an alternative training scheme: instead of updating the parameter w_t, update the violation counts $\alpha(i, t)$ since this is sufficient to extract the final trained parameter. Formulating training as counting violations also yields a kernelized version. Let $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a Mercer kernel with an implicit feature function $\phi : \mathbb{R}^d \to \mathcal{F}$ such that $K(x,z) = \langle \phi(x), \phi(z) \rangle$. The *conceptual* parameter value is, for any $t \geq 1$,

$$w_{t+1} = \frac{1}{t\lambda} \sum_{i=1}^{N} \alpha(i,t) y_i \phi(x_i) \in \mathcal{F}$$

Its prediction on $x \in \mathbb{R}^d$ is the sign of

$$\langle w_{t+1}, \phi(x) \rangle = \frac{1}{t\lambda} \sum_{i=1}^{N} \alpha(i,t) y_i \langle \phi(x_i), \phi(x) \rangle = \frac{1}{t\lambda} \sum_{i=1}^{N} \alpha(i,t) y_i K(x_i, x) \in \mathbb{R}$$

Therefore we never have to store the parameter or the features in training or inference. More specifically:

Training. Given $(x_1, y_1) \ldots (x_N, y_N) \in \mathbb{R}^d \times \{\pm 1\}$, initialize $\alpha = 0_N$. For $t = 1 \ldots T$, draw $i_t \in \{1 \ldots N\}$ and set

$$\alpha_{i_t} \leftarrow \begin{cases} \alpha_{i_t} + 1 & \text{if } y_{i_t} \left(\frac{1}{t\lambda} \sum_{j=1}^{N} \alpha_j y_j K(x_j, x_{i_t}) \right) < 1 \\ \alpha_{i_t} & \text{otherwise} \end{cases}$$

Inference. Given $x \in \mathbb{R}^d$, predict

$$\hat{y} = \textbf{sign} \left(\sum_{i=1}^{N} \alpha_i y_i K(x_i, x) \right)$$

Note that we can ignore the scaling factor $1/((T+1)\lambda)$ since it does not affect the sign.

2 Convergence Analysis

Stochastic (sub)gradient descent with learning rate $\eta_t = 1/(\lambda t)$ turns out to be also amenable to convergence analysis. We will assume a general setting in which for $t = 1 \ldots T$, we are presented with a λ-strongly convex (sub-differentiable) function $f_t : \mathbb{R}^d \to \mathbb{R}$ to minimize. That is, given any $w \in \mathbb{R}^d$

$$f_t(u) \geq f_t(w) + \langle \nabla f_t(w), u - w \rangle + \frac{\lambda}{2} ||w - u||^2 \qquad \forall u \in \mathbb{R}^d \tag{3}$$

(i.e., the linear approximation of f_t around w is a *strict* lower bound).

Lemma 2.1. Pick $w^{(1)} \in \mathbb{R}^d$ arbitrarily and define $w_{t+1} = w_t - \eta_t \nabla f_t(w_t)$ for $t = 1 \ldots T$. If $\eta_t = 1/(\lambda t)$,

$$\frac{1}{T} \sum_{t=1}^{T} f_t(w_t) \leq \frac{1}{T} \sum_{t=1}^{T} f_t(u) + \frac{G_{\max}^2 (1 + \log T)}{2\lambda T} \qquad \forall u \in \mathbb{R}^d$$

where $G_{\max} = \max_{t=1}^{T} ||\nabla f_t(w_t)||$ is the largest norm of the gradient during training.

Proof. From (3) it follows that

$$\langle \nabla f_t(w_t), w_t - u \rangle \geq f_t(w_t) - f_t(u) + \frac{\lambda}{2} ||w_t - u||^2 \qquad \forall u \in \mathbb{R}^d \tag{4}$$

We will relate the LHS to the following expression, which is amenable to the telescoping sum.

$$\begin{aligned} ||w_t - u||^2 - ||w_{t+1} - u||^2 &= ||w_t - u||^2 - ||w_t - u - \eta_t \nabla f_t(w_t)||^2 \\ &= 2\eta_t \langle \nabla f_t(w_t), w_t - u \rangle - \eta_t^2 ||\nabla f_t(w_t)||^2 \\ \Leftrightarrow \quad \langle \nabla f_t(w_t), w_t - u \rangle &= \frac{||w_t - u||^2 - ||w_{t+1} - u||^2}{2\eta_t} + \frac{\eta_t ||\nabla f_t(w_t)||^2}{2} \end{aligned} \tag{5}$$

Combining (4) and (5) gives

$$f_t(w_t) - f_t(u) \leq \frac{||w_t - u||^2 - ||w_{t+1} - u||^2}{2\eta_t} - \frac{\lambda}{2} ||w_t - u||^2 + \frac{\eta_t ||\nabla f_t(w_t)||^2}{2} \qquad \forall u \in \mathbb{R}^d$$

Setting $\eta_t = 1/(\lambda t)$ gives

$$f_t(w_t) - f_t(u) \leq \frac{\lambda}{2} \left((t-1) ||w_t - u||^2 - t ||w_{t+1} - u||^2 \right) + \frac{||\nabla f_t(w_t)||^2}{2\lambda t} \qquad \forall u \in \mathbb{R}^d$$

Since this holds for every $t = 1 \dots T$, we can sum both sides over t. Note that

$$\begin{aligned}\sum_{t=1}^{T} (t-1) ||w_t - u||^2 - t ||w_{t+1} - u||^2 &= - ||w_2 - u||^2 + ||w_2 - u||^2 - 2 ||w_3 - u||^2 + \dots - T ||w_{T+1} - u||^2 \\ &= -T ||w_{T+1} - u||^2\end{aligned}$$

Also,

$$\sum_{t=1}^{T} \frac{||\nabla f_t(w_t)||^2}{2\lambda t} \leq \frac{G_{\max}^2}{2\lambda} \sum_{t=1}^{T} \frac{1}{t} \leq \frac{G_{\max}^2 (1 + \log T)}{2\lambda}$$

where we use the fact that the partial sums of the harmonic series have logarithmic growth (Wikipedia). Thus for all $u \in \mathbb{R}^d$:

$$\begin{aligned}\sum_{t=1}^{T} f_t(w_t) - f_t(u) &\leq -\frac{\lambda T}{2} ||w_{T+1} - u||^2 + \frac{G_{\max}^2 (1 + \log T)}{2\lambda} \\ &\leq \frac{G_{\max}^2 (1 + \log T)}{2\lambda} \\ \Leftrightarrow \quad \frac{1}{T} \sum_{t=1}^{T} f_t(w_t) &\leq \frac{1}{T} \sum_{t=1}^{T} f_t(u) + \frac{G_{\max}^2 (1 + \log T)}{2\lambda T}\end{aligned}$$

□

2.1 Application

Let us apply Lemma 2.1 to optimizing the SVM loss with Pegasos. We can in fact consider a more general minibatch version: for $t = 1 \dots T$ draw some $B_t \subseteq \{1 \dots N\}$ and take a subgradient step on

$$J_{B_t}(w) = \frac{\lambda}{2} ||w||^2 + \frac{1}{|B_t|} \sum_{i \in B_t} \max(0, 1 - y_i \langle w, x_i \rangle)$$

with learning rate $1/(\lambda t)$. J_{B_t} is λ-strongly convex. Let $R = \max_{i=1}^N ||x_i||$ and note that

$$\nabla J_{B_t}(w_t) = \lambda w_t - v_t \qquad v_t = \frac{1}{|B_t|} \sum_{i \in B_t} [[y_i \langle w_t, x_i \rangle < 1]] \, y_i x_i$$

$$||\nabla J_{B_t}(w_t)|| \leq \lambda ||w_t|| + ||v_t|| \leq \lambda \left(\frac{R}{\lambda} \right) + R \leq 2R$$

where the first inequality is the triangle inequality $||u + v|| \leq ||u|| + ||v||$ and the second inequality follows from the fact that $w_{t+1} = \frac{1}{t\lambda} \sum_{l=1}^{t} v_l$ (minibatch version of (1)). So by Lemma 2.1,

$$\frac{1}{T} \sum_{t=1}^{T} J_{B_t}(w_t) \leq \frac{1}{T} \sum_{t=1}^{T} J_{B_t}(u) + \frac{2R^2(1 + \log T)}{\lambda T} \tag{6}$$

for *any* $u \in \mathbb{R}^d$. In particular, we can set u to be the optimal parameter of the full loss $w^* = \arg\min_{w \in \mathbb{R}^d} J(w)$ where $J(w) = J_{\{1 \dots N\}}(w)$. But this analysis is a bit unsatisfying because it only bounds the average loss across updates rather than the loss of a single model. One quick fix is to set $B_t = \{1 \dots N\}$ (i.e., full subgradient descent) so that $J(w) = J_{B_t}(w)$ for all t. Let $\bar{w} = (1/T) \sum_{t=1}^{T} w_t$ denote the average parameter. Then

$$\begin{aligned}
J(\bar{w}) &\leq \frac{1}{T} \sum_{t=1}^{T} J(w_t) && \text{(by the convexity of } J\text{)} \\
&= \frac{1}{T} \sum_{t=1}^{T} J_{B_t}(w_t) && \text{(since } B_t = \{1 \dots N\} \text{ for all } t\text{)} \\
&\leq \frac{1}{T} \sum_{t=1}^{T} J_{B_t}(w^*) + \frac{2R^2(1 + \log T)}{\lambda T} && \text{(by (6))} \\
&= J(w^*) + \frac{2R^2(1 + \log T)}{\lambda T} && \text{(since } B_t = \{1 \dots N\} \text{ for all } t\text{)}
\end{aligned}$$

If $B_t \neq \{1 \dots N\}$ this argument does not hold in general. However, it is possible to show that even in this case it holds if B_t is sampled uniformly at random from $\{1 \dots N\}$ (with or without replacement) with high probability with slightly worse constants (Kakade and Tewari, 2009).

References

Kakade, S. M. and Tewari, A. (2009). On the generalization ability of online strongly convex programming algorithms. In *Advances in Neural Information Processing Systems*, pages 801–808.

Shalev-Shwartz, S., Singer, Y., Srebro, N., and Cotter, A. (2011). Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, **127**(1), 3–30.