Rademacher Penalty Optimization with the Generalized Ramp Loss

Christopher P. Diehl

February 9, 2009

Let us define the *Rademacher penalty* as

$$R_n(\phi \circ \mathcal{F} \mid \sigma) = \max_{h \in \mathcal{H}} \frac{2}{n} \sum_{i=1}^{n} \sigma_i \phi(y_i h(x_i)), \tag{1}$$

where $\mathcal{F} = \{yh(x) : \forall\ h \in \mathcal{H}\}$ and $\phi(h(x), y)$ is the loss function of interest. The labeled training sample $\mathcal{S} = \{(x_i, y_i) \mid i \in \{1, ..., n\}, x_i \in \mathbb{R}^m, y_i \in \{\pm 1\}\}$ is assumed to be generated from an independent, identically distributed (IID) random process. Given each Rademacher random variable $\sigma_i \in \{\pm 1\}$ with uniform odds, a realization of the Rademacher random variables σ amounts to a partition of the training set into two approximately equal-sized subsets with high probability. Reexpressing (1) to reflect this explicitly yields

$$R_n(\phi \circ \mathcal{F} \mid \sigma) = \max_{h \in \mathcal{H}} \frac{2}{n} \sum_{i, \sigma_i = 1} \phi(y_i h(x_i)) - \frac{2}{n} \sum_{j, \sigma_j = -1} \phi(y_j h(x_j)). \tag{2}$$

In the following, we will assume classifiers of the form $h(x) = w^{\mathrm{T}} \Phi(x)$ with $w = \sum_i \alpha_i \Phi(x_i)$. We will assume further the hypothesis class is the set $\mathcal{H} = \{h : ||w|| \leq W\}$. Let $K(x, y) = \Phi(x)^{\mathrm{T}} \Phi(y)$, $k_{ij} = K(x_i, x_j)$, $k_j = [k_{ij}\ \forall\ i \in \{1, \ldots, n\}]$ and matrix $K = [k_i\ \forall\ i \in \{1, \ldots, n\}]$. Therefore $h(x) = \sum_i \alpha_i K(x_i, x)$, $\mathcal{H} = \{h : \alpha^{\mathrm{T}} K \alpha \leq W\}$ and

$$R_n(\phi \circ \mathcal{F} \mid \sigma) = \max_{\substack{\alpha \\ \alpha^{\mathrm{T}} K \alpha \leq W}} \quad \frac{2}{n} \sum_{i, \sigma_i = 1} \phi(y_i k_i^{\mathrm{T}} \alpha) - \frac{2}{n} \sum_{j, \sigma_j = -1} \phi(y_j k_j^{\mathrm{T}} \alpha) \tag{3}$$

Since we are interested in a subset of the solutions along the trajectory parameterized by W, we address a corresponding set of unconstrained optimization problems to estimate the Rademacher penalties

$$\alpha_*(\beta) = \arg\max_{\alpha} \frac{2}{n} \sum_{i, \sigma_i = 1} \phi(y_i k_i^{\mathrm{T}} \alpha) - \frac{2}{n} \sum_{j, \sigma_j = -1} \phi(y_j k_j^{\mathrm{T}} \alpha) - \beta \alpha^{\mathrm{T}} K \alpha. \tag{4}$$

For a given β, there exists a corresponding $W(\beta) = \alpha_*(\beta)^{\mathrm{T}} K \alpha_*(\beta)$ for which β is the resulting Lagrange multiplier in the constrained optimization problem in

(3). This implies that β provides an alternate parameterization of the trajectory of solutions that we can trace without the burden of the norm constraint. By solving the maximization problem for a series of values $\{\beta_i\}$, we obtain samples $\{(W(\beta_i), R_n(\phi \circ \mathcal{F} \mid \sigma, W(\beta_i)))\}$ on the Rademacher penalty curve as a function of W supporting estimation of the needed Rademacher penalties through interpolation.

In this document, we address the specific maximization problem induced when $\phi(x)$ is a generalized ramp loss

$$\phi(x) = \begin{cases} 1 - \gamma x & \text{if } x < 0 \\ 1 - x & \text{if } 0 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $0 < \gamma \leq 1$. The key will be to exploit the fact that the loss can be decomposed into the sum of a convex and a concave function. Specifically the generalized ramp loss can be decomposed into a difference of shifted hinge losses

$$\begin{aligned} \phi(x) &= \phi_v(x) + \phi_c(x) \\ &= [1 - x]_+ - [-(1 - \gamma)x]_+. \end{aligned} \tag{6}$$

As we shall see, such structure admits a similar decomposition for the entire objective function.

The approach we will employ to maximize the Rademacher penalty is the ConCave-Convex Procedure (CCCP) [1]. CCCP identifies a local maximum by solving a sequence of concave maximization problems where the convex component is lower bounded by a first-order approximation about the current parameters. After each convex maximization problem is solved, the first-order approximation is recomputed and the process continues until a local maximum is reached.

Given our objective is to maximize

$$\max_\alpha \frac{2}{n} \sum_{i, \sigma_i = 1} \phi(y_i k_i^{\mathrm{T}} \alpha) - \frac{2}{n} \sum_{j, \sigma_j = -1} \phi(y_j k_j^{\mathrm{T}} \alpha) - \beta \alpha^{\mathrm{T}} K \alpha, \tag{7}$$

we first decompose the loss and group convex and concave terms yielding

$$\begin{aligned} \max_\alpha & \overbrace{\frac{2}{n} \sum_{i, \sigma_i = 1} \phi_c(y_i k_i^{\mathrm{T}} \alpha) - \frac{2}{n} \sum_{j, \sigma_j = -1} \phi_v(y_j k_j^{\mathrm{T}} \alpha) - \beta \alpha^{\mathrm{T}} K \alpha}^{g_c(\alpha)} + \\ & \underbrace{\frac{2}{n} \sum_{i, \sigma_i = 1} \phi_v(y_i k_i^{\mathrm{T}} \alpha) - \frac{2}{n} \sum_{j, \sigma_j = -1} \phi_c(y_j k_j^{\mathrm{T}} \alpha)}_{g_v(\alpha)} \end{aligned} \tag{8}$$

$$\max_\alpha g_c(\alpha) + g_v(\alpha). \tag{9}$$

Beginning with $\alpha_0 = 0$, we want to maximize a concave surrogate

$$\max_{\alpha_{k+1}} g_c(\alpha_{k+1}) + \nabla_\alpha g_v(\alpha)|^{\mathrm{T}}_{\alpha=\alpha_k} \alpha_{k+1} \tag{10}$$

during each iteration, where $\nabla_\alpha g_v(\alpha)$ is the subgradient of g_v, yielding the parameters α_{k+1} from the concave approximation centered about the parameters α_k from the previous iteration. The core optimization problem we must address is therefore

$$\max_{\alpha} g_c(\alpha) + \nabla_\alpha g_v(\alpha)|^{\mathrm{T}}_{\alpha=\alpha_0} \alpha. \tag{11}$$

Substituting in for ϕ_c and ϕ_v, we obtain the following for the concave and convex components

$$g_c(\alpha) = -\frac{2}{n} \sum_{i,\sigma_i=1} [-(1-\gamma) y_i k_i^{\mathrm{T}} \alpha]_+ - \frac{2}{n} \sum_{j,\sigma_j=-1} [1 - y_j k_j^{\mathrm{T}} \alpha]_+ - \beta \alpha^{\mathrm{T}} K \alpha \tag{12}$$

$$g_v(\alpha) = \frac{2}{n} \sum_{i,\sigma_i=1} [1 - y_i k_i^{\mathrm{T}} \alpha]_+ + \frac{2}{n} \sum_{j,\sigma_j=-1} [-(1-\gamma) y_j k_j^{\mathrm{T}} \alpha]_+. \tag{13}$$

Although the subderivative of the hinge loss $[\cdot]_+$ is not unique at the origin, any valid subderivative will suffice to define the lower bound for g_v. Therefore we will define the subderivative as

$$\frac{\partial}{\partial x}[x]_+ = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}. \tag{14}$$

The subgradient $\nabla_\alpha g_v(\alpha)$ is then

$$\begin{aligned} \nabla_\alpha g_v(\alpha) &= -\frac{2}{n} \sum_{i,\sigma_i=1} y_i k_i p(\alpha)_i - \frac{2}{n} \sum_{j,\sigma_j=-1} (1-\gamma) y_j k_j q(\alpha,\gamma)_j & (15) \\ &= -\frac{2}{n} K \left(\frac{1}{2}(\sigma+1) \bullet y \bullet p(\alpha) + \frac{1-\gamma}{2}(1-\sigma) \bullet y \bullet q(\alpha,\gamma) \right) & (16) \\ &= -\frac{1}{n} K \left(((\sigma+1) \bullet p(\alpha) + (1-\gamma)(1-\sigma) \bullet q(\alpha,\gamma)) \bullet y \right) & (17) \end{aligned}$$

where the element-wise or Hadamard product $a \bullet b = [a_1 b_1 \ldots a_n b_n]^{\mathrm{T}}$, $p(\alpha)_i = \mathcal{I}_{y_i k_i^{\mathrm{T}} \alpha < 1}$, $q(\alpha,\gamma)_i = \mathcal{I}_{(1-\gamma) y_i k_i^{\mathrm{T}} \alpha < 0}$, $p(\alpha) = [p(\alpha)_1, \ldots, p(\alpha)_n]$ and $q(\alpha,\gamma) = [q(\alpha,\gamma)_1, \ldots, q(\alpha,\gamma)_n]$. Let us define

$$\begin{aligned} & s(\alpha_0) = \nabla_\alpha g_v(\alpha)|_{\alpha=\alpha_0} = \\ & \quad -\frac{1}{n} K \left(((\sigma+1) \bullet p(\alpha_0) + (1-\gamma)(1-\sigma) \bullet q(\alpha_0,\gamma)) \bullet y \right) \end{aligned} \tag{18}$$

We can now reexpress (11) as

$$\min_{\alpha} \beta \alpha^{\mathrm{T}} K \alpha - s(\alpha_0)^{\mathrm{T}} \alpha + \frac{2}{n} \sum_{i,\sigma_i=1} [-(1-\gamma) y_i k_i^{\mathrm{T}} \alpha]_+ + \frac{2}{n} \sum_{j,\sigma_j=-1} [1 - y_j k_j^{\mathrm{T}} \alpha]_+. \tag{19}$$

By introducing slack variables for the hinge losses, we obtain the following quadratic program

$$\begin{array}{ll} \min_{\alpha,\epsilon} & \beta\alpha^{\mathrm{T}}K\alpha - s(\alpha_0)^{\mathrm{T}}\alpha + \frac{2}{n}\sum_i \epsilon_i \\ \text{subject to} & (1-\gamma)y_i k_i^{\mathrm{T}}\alpha \geq -\epsilon_i,\ \sigma_i = 1 \\ & y_i k_i^{\mathrm{T}}\alpha \geq 1-\epsilon_i,\ \sigma_i = -1 \\ & \epsilon_i \geq 0 \ \ \forall\ i \in \{1,\ldots,n\}. \end{array}$$

The corresponding Lagrangian for this quadratic program is

$$\begin{aligned} \max_{\lambda,r}\min_{\alpha,\epsilon} \qquad & L(\alpha,\epsilon,\lambda,r) = \beta\alpha^{\mathrm{T}}K\alpha - s(\alpha_0)^{\mathrm{T}}\alpha + \frac{2}{n}\sum_i \epsilon_i \\ & - \sum_{i,\sigma_i=1}\lambda_i((1-\gamma)y_i k_i^{\mathrm{T}}\alpha + \epsilon_i) \\ & - \sum_{i,\sigma_i=-1}\lambda_i(y_i k_i^{\mathrm{T}}\alpha - 1 + \epsilon_i) - \sum_i r_i\epsilon_i \end{aligned} \tag{20}$$

subject to $\lambda, r \geq 0$. Our next objective is to derive the dual optimization problem from the above primal formulation. Setting the partial derivatives with respect to ϵ_i equal to zero, we find

$$\frac{\partial L}{\partial \epsilon_i} = \frac{2}{n} - \lambda_i - r_i = 0 \longrightarrow \frac{2}{n} - r_i = \lambda_i. \tag{21}$$

Coupling this constraint with $\lambda, r \geq 0$, we now have the optimization problem

$$\max_{\lambda}\min_{\alpha} L(\alpha,\lambda) = \beta\alpha^{\mathrm{T}}K\alpha - s(\alpha_0)^{\mathrm{T}}\alpha - \sum_{i,\sigma_i=1}\lambda_i(1-\gamma)y_i k_i^{\mathrm{T}}\alpha - \sum_{i,\sigma_i=-1}\lambda_i(y_i k_i^{\mathrm{T}}\alpha - 1) \tag{22}$$

subject to $0 \leq \lambda \leq \frac{2}{n}$. Setting the partial derivatives with respect to α equal to zero, we find

$$\frac{\partial L}{\partial \alpha} = 2\beta K\alpha_* - s(\alpha_0) - \sum_{i,\sigma_i=1}\lambda_i(1-\gamma)y_i k_i - \sum_{i,\sigma_i=-1}\lambda_i y_i k_i = 0 \tag{23}$$

$$2\beta K\alpha_* = s(\alpha_0) + \sum_{i,\sigma_i=1}\lambda_i(1-\gamma)y_i k_i + \sum_{i,\sigma_i=-1}\lambda_i y_i k_i \tag{24}$$

This implies

$$2\beta\alpha_*^{\mathrm{T}}K\alpha_* = s(\alpha_0)^{\mathrm{T}}\alpha_* + \sum_{i,\sigma_i=1}\lambda_i(1-\gamma)y_i k_i^{\mathrm{T}}\alpha_* + \sum_{i,\sigma_i=-1}\lambda_i y_i k_i^{\mathrm{T}}\alpha_*. \tag{25}$$

This leads to the optimization problem

$$\begin{aligned} \min_{\lambda} L'(\alpha_*,\lambda) &= -\beta\alpha_*^{\mathrm{T}}K\alpha_* + s(\alpha_0)^{\mathrm{T}}\alpha_* + \sum_{i,\sigma_i=1}\lambda_i(1-\gamma)y_i k_i^{\mathrm{T}}\alpha_* \\ &\quad + \sum_{i,\sigma_i=-1}\lambda_i y_i k_i^{\mathrm{T}}\alpha_* - \sum_{i,\sigma_i=-1}\lambda_i \\ &= \beta\alpha_*^{\mathrm{T}}K\alpha_* - \sum_{i,\sigma_i=-1}\lambda_i \end{aligned} \tag{26}$$

subject to $0 \le \lambda \le \frac{2}{n}$. If we assume the kernel matrix K results from a positive definite kernel such that K is invertible, we can solve for α_* yielding

$$\begin{aligned}
\alpha_* &= \frac{1}{2\beta}K^{-1}\left(s(\alpha_0) + \sum_{i,\sigma_i=1}\lambda_i(1-\gamma)y_i k_i + \sum_{i,\sigma_i=-1}\lambda_i y_i k_i\right) & (27)\\
&= \frac{1}{2\beta}\left(K^{-1}s(\alpha_0) + \left(\frac{1-\gamma}{2}(\sigma+1) + \frac{1}{2}(1-\sigma)\right)\bullet\lambda\bullet y\right) & (28)\\
&= \frac{1}{2\beta}\left(v(\alpha_0) + w(\gamma,\sigma)\bullet\lambda\bullet y\right) & (29)
\end{aligned}$$

where $v(\alpha_0) = K^{-1}s(\alpha_0)$ and $w(\gamma,\sigma) = \frac{1-\gamma}{2}(\sigma+1) + \frac{1}{2}(1-\sigma)$. The final task is to expand $\beta\alpha_*^{\mathrm{T}}K\alpha_*$ which yields

$$\begin{aligned}
\beta\alpha_*^{\mathrm{T}}K\alpha_* &= \frac{1}{4\beta}\left(w(\gamma,\sigma)\bullet\lambda\bullet y + v(\alpha_0)\right)^{\mathrm{T}}K\left(w(\gamma,\sigma)\bullet\lambda\bullet y + v(\alpha_0)\right) & (30)\\
&= \frac{1}{4\beta}((w(\gamma,\sigma)\bullet\lambda\bullet y)^{\mathrm{T}}K(w(\gamma,\sigma)\bullet\lambda\bullet y) + \\
&\quad 2v(\alpha_0)^{\mathrm{T}}K(w(\gamma,\sigma)\bullet\lambda\bullet y) + \\
&\quad v(\alpha_0)^{\mathrm{T}}Kv(\alpha_0)) & (31)\\
&= \frac{1}{4\beta}((\lambda\bullet y)^{\mathrm{T}}(w(\gamma,\sigma)w(\gamma,\sigma)^{\mathrm{T}}\bullet K)(\lambda\bullet y) + \\
&\quad 2v(\alpha_0)^{\mathrm{T}}(\mathbf{1}w(\gamma,\sigma)^{\mathrm{T}}\bullet K)(\lambda\bullet y) + \\
&\quad v(\alpha_0)^{\mathrm{T}}Kv(\alpha_0)) \\
&= \frac{1}{4\beta}(\lambda^{\mathrm{T}}((w(\gamma,\sigma)\bullet y)(w(\gamma,\sigma)\bullet y)^{\mathrm{T}}\bullet K)\lambda + \\
&\quad \frac{1}{2\beta}v(\alpha_0)^{\mathrm{T}}(\mathbf{1}(w(\gamma,\sigma)\bullet y)^{\mathrm{T}}\bullet K)\lambda + \\
&\quad v(\alpha_0)^{\mathrm{T}}Kv(\alpha_0)) & (32)
\end{aligned}$$

where $\mathbf{1}$ is a vector of ones. Substituting this result into equation (26) without the constant term yields

$$\begin{aligned}
\min_{\lambda} L'(\lambda) &= \frac{1}{4\beta}\lambda^{\mathrm{T}}((w(\gamma,\sigma)\bullet y)(w(\gamma,\sigma)\bullet y)^{\mathrm{T}}\bullet K)\lambda + \\
&\quad \frac{1}{2\beta}v(\alpha_0)^{\mathrm{T}}(\mathbf{1}(w(\gamma,\sigma)\bullet y)^{\mathrm{T}}\bullet K)\lambda - \\
&\quad \frac{1}{2}(1-\sigma)^{\mathrm{T}}\lambda & (33)
\end{aligned}$$

subject to $0 \le \lambda \le \frac{2}{n}$.

A Identities

We claim that the following equivalence holds

$$(a\bullet b)^{\mathrm{T}}K(c\bullet d) = a^{\mathrm{T}}(bc^{\mathrm{T}}\bullet K)d. \qquad (34)$$

Let $v = K(c \bullet d)$ and $v_i = \sum_{j=1}^{N} k_{ij} c_j d_j$. Therefore

$$(a \bullet b)^{\mathrm{T}} K(c \bullet d) = (a \bullet b)^{\mathrm{T}} v = \sum_{i=1}^{N} \sum_{j=1}^{N} a_i b_i k_{ij} c_j d_j. \tag{35}$$

Let $M = bc^{\mathrm{T}} \bullet K$ and $m_{ij} = b_i k_{ij} c_j$. Therefore

$$a^{\mathrm{T}}(bc^{\mathrm{T}} \bullet K)d = a^{\mathrm{T}} M d = \sum_{i=1}^{N} \sum_{j=1}^{N} a_i m_{ij} d_j = \sum_{i=1}^{N} \sum_{j=1}^{N} a_i b_i k_{ij} c_j d_j. \tag{36}$$

This implies that

$$a^{\mathrm{T}} K(b \bullet c) = (a \bullet \mathbf{1})^{\mathrm{T}} K(b \bullet c) = a^{\mathrm{T}}(\mathbf{1} b^{\mathrm{T}} \bullet K)c \tag{37}$$

where **1** is a vector of ones.

References

[1] A. L. Yuille and A. Rangarajan. The concave-convex procedure (cccp). In *Advances in Neural Information Processing Systems 14*. MIT Press, 2002.