Bounded Loss Classification[*]

Jason D. M. Rennie
jrennie@ai.mit.edu

February 25, 2003

1 Introduction

Consider the problem of classification. Modern-day solutions have looked toward problem formulations where the search space is convex. Such formulations guarantee that a minimization of the objective function is found. But, in order to achieve that guarantee, such formulations treat outliers somewhat overzealously. Many classification objectives can be viewed as minimizing a loss funciton. For example, the Support Vector Machine minimizes the hinge loss, $\sum_i l_h(w^T x_i + b)$, where $l_h(z) = (1-z)_+$. Logistic regression (LR) minimizes a similar loss, the logistic, $\sum_i l_l(w^T x_i + b)$, where $l_l(z) = -\log \frac{1}{1+e^{-x}}$. Many would not call these two losses "similar," but they share important properties. One is that they are convex, e.g. $l_h(\alpha z_1 + (1-\alpha)z_2) \geq \alpha l_h(z_1) + (1-\alpha) l_h(z_2)$. This is the reason that the objective has a unique minimum. Along with this property comes a less desirable property, that the loss for a single example is unbounded. In other words, an outlier can have an unbounded effect on the decision boundary. In practice, regularization is used to temper this effect, but it can produce negative effects. In summary, state-of-the-art classifiers utilize a convex loss function to achieve an objective with a unique minimum, but as a result, outliers can significantly effect the decision boundary.

Motivation for current, state-of-the-art techniques was a long history of classification algorithms that used non-convex objective functions. For classification, one wishes to learn a decision boundary that will minimize the zero-one loss ($l_z(x) = \theta(-w^T x + b)$, θ is the heavaside function) on unseen examples drawn from the same distribution as the training examples. Many objectives minimized this or something very similar. Of course, such an optimization is riddled with local minima. Various techniques were developed to work around this problem, but none was as effective as the convex objectives that have been recently brought forward. We believe there is still hope in a more traditional objective, one that minimizes the zero-one loss. There are issues to be addressed—since the zero-one loss is not convex, how can we find a good solution? But, we feel that there are ways to sufficiently address this issue. Additionally, with the

[*] Joint work with Tommi Jaakkola, tommi@ai.mit.edu

zero-one loss, we can properly handle outliers. Whereas convex objectives may significantly alter their decision boundaries to handle outliers, a zero-one objective can incur a unit penalty and effective ignore points that clearly cannot be classified correctly.

2 Bounded Loss Classification

We initially make very restrictve assumptions on the way labels are generated, but we later relax those assumptions.

Consider a distribution that governs how labels are assigned to examples, $p(y|\mathbf{x})$. We assume that an example is assigned a label in a probabilistic manner. Let $\mathbf{w}$, b be parameters of the distribution; let $z(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$; let the distribution be defined by

$$p(y=+1|\mathbf{x};\mathbf{w},b) = 1 - p(y=-1|\mathbf{x};\mathbf{w},b) = \frac{1}{1+e^{-z(\mathbf{x})}}. \tag{1}$$

We wish to learn a decision boundary ($\mathbf{w}$ and b) such that the number of classification errors is minimized. In other words, we want to maximize the expected number of correct predictions

$$J(\mathbf{w}, b|\{(x_i, y_i)\}) = \sum_i \frac{1}{1+e^{-y_i z(\mathbf{x}_i)}}. \tag{2}$$

Let $\sigma_i = \frac{1}{1+e^{-y_i z(\mathbf{x}_i)}}$; then we can write $J = \sum_i \sigma_i$. It is equivalent to maximize a monotonic transform of J. Let $\theta = (\mathbf{w}, b)$. Let

$$\begin{aligned} A(\theta) &:= \log \sum_i \sigma_i, && (3)\\ &= \log \sum_i q_i \frac{\sigma_i}{q_i}, && (4)\\ &\geq \sum_i q_i \log \frac{\sigma_i}{q_i}, && (5)\\ &:= B(q, \theta). && (6) \end{aligned}$$

This inequality motivates an approach for maximizing A. We can find a local maximum of A through a two-step alternating minimization proceedure,

$$\textbf{(E step)} \qquad q^{(t+1)} = \arg\max_q B(q, \theta^{(t)}) \tag{7}$$

$$\textbf{(M step)} \qquad \theta^{(t+1)} = \arg\max_\theta B(q^{(t+1)}, \theta) \tag{8}$$

Clearly, $B(q^{(t+1)}, \theta^{(t+1)}) \geq B(q^{(t)}, \theta^{(t)})$. We must show $B(q^{(t+1)}, \theta^{(t+1)}) \geq A(\theta^{(t)})$ in order that this procedure be valid.

Consider the E step. The Lagrangian can be written

$$J_E = \sum_i q_i \log \frac{\sigma_i}{q_i} - \lambda \left(\sum_i q_i - 1 \right) \tag{9}$$

Taking the partial with respect to q_k, we find that the maximizing distribution is $q_k = \frac{\sigma_k}{\sum_i \sigma_i}$,

$$\frac{\partial J_E}{\partial q_k} = \log \sigma_k - 1 - \log q_k - \lambda = 0, \tag{10}$$

$$\Rightarrow \frac{q_i}{q_j} = \frac{\sigma_i}{\sigma_j}. \tag{11}$$

Note that $B(q^{(t+1)}, \theta^{(t)}) = A(\theta^{(t)})$.

$$B(q^{(t+1)}, \theta^{(t)}) = \sum_i \frac{\sigma_i}{\sum_j \sigma_j} \log \sum_j \sigma_j, \tag{12}$$

$$= \sum_j \sigma_j, \tag{13}$$

$$= A(\theta^{(t)}). \tag{14}$$

Hence, our minimization procedure is valid.

Next, we consider the M step. This is simply a generalization of LR, where the loss associated with each point is weighted,

$$\max_{\mathbf{w},b} \sum_i q_i \log \sigma_i. \tag{15}$$

The solution must be found numerically, but it is not difficult and the maximum is guaranteed to be found.

Worth some discussion is the effect of the q_i. The logistic loss function, $l_l(z) = -\log \frac{1}{1+e^{-x}}$, is unbounded and the slope is approximately -1 for $z < 0$. i.e. there is no way to forget about “outliers.” The q_i in our optimization mitigate this and effectively put a bound on the loss of an outlier. Consider what happens when we are close to a solution: θ changes very little so we are approximately maximizing

$$\sum_i \frac{\sigma_i}{\sum_j \sigma_j} \log \sum_j \sigma_j = \sum_j \sigma_j, \tag{16}$$

which is the noisy zero-one loss objective that we have discussed. When we are close to a solution, the derivative of the loss with respect to a small change in the position of an example is not only bounded, but highest near the decision boundary. Examples that are far from the decision boundary (both easy-to-classify points and outliers) have little effect on the boundary. We find this property satisfying and possibly the most appropriate realization of margin maximization.

3 Extensions

The algorithm we have described to this point is both ill-posed and makes severe assumptions on the distribution of example lables. This section is concerned with making the problem well posed and lifting some of the restrictive assumptions.

3.1 Regularization

To this point, we have intentionally avoided the issue of regularization for the purposes of clarity. But, the objective function we have discussed, $A(\theta)$, is ill posed. It becomes well posed if we add a penalty for the magnitude of the parameters. A common choice is the quadratic penalty, $\|\mathbf{w}\|^2$. Though it is not strictly necessary for objectives with a convex loss function when the data is not separable (such as with unregularized SVM and logistic regression), a regularization penalty empirically tends to improve generalization. For these reasons, we re-pose our objective as

$$\max_{\theta} A(\theta) := \max_{\mathbf{w},b} \log \sum_i \sigma_i - \lambda \|\mathbf{w}\|^2. \tag{17}$$

Manipulations like those already described yield a similar alternating minimization procedure, with the only difference being that the M step is a generalization of regularized LR (instead of non-regularized LR).

3.1.1 Implementation

The objective for the M step of Regularized Bounded Loss Classification is

$$f(\theta) = \max_{\theta} \sum_i q_i \log \sigma_i - \frac{C}{2} \|w\|^2. \tag{18}$$

We use Newton's method to find a minimum. This involves an iterative procedure where

$$\theta^{t+1} = \theta^t - \left(f''(\theta^t)\right)^{-1} f'(\theta^t). \tag{19}$$

Let $z_i = w_i^x + b$; let $y_i^0 = 2y_i - 1$. The derivatives are

$$\begin{aligned} \frac{\partial f}{\partial w_j} &= \sum_i q_i(1 - \sigma(y_i z_i)) y_i x_{ij} - C w_j \qquad (20) \\ &= \sum_i (y_i^0 - \sigma(z_i)) q_i x_{ij} - C w_j, \qquad (21) \end{aligned}$$

$$\frac{\partial f}{\partial b} = \sum_i q_i(1 - \sigma(y_i z_i)) y_i = \sum_i (y_i^0 - \sigma(z_i)) q_i, \tag{22}$$

$$\frac{\partial^2 f}{\partial w_j \partial w_k} = -\sum_i q_i^2 y_i^2 x_{ij} x_{ik} \sigma(y_i z_i)(1-\sigma(y_i z_i)) - C\delta_{j-k} \tag{23}$$

$$= -\sum_i q_i^2 x_{ij} x_{ik} \sigma(z_i)(1-\sigma(z_i)) - C\delta_{j-k}, \text{ and} \tag{24}$$

$$\frac{\partial^2 f}{\partial b} = -\sum_i q_i^2 y_i^2 \sigma(y_i z_i)(1-\sigma(y_i z_i)) \tag{25}$$

$$= -\sum_i q_i^2 \sigma(z_i)(1-\sigma(z_i)). \tag{26}$$

Substituting back into equation 19 gives us one step of the iterative procedure. We continue in this manner until $\|\theta^{t+1} - \theta^t\|$ becomes very small (e.g. $< 10^{-6}$).

3.2 Zero-One Loss

To this point, we have described an algorithm that approximates the zero-one loss. We maximize the average log probability of the data points using the assumption that the logistic reliably converts output values to probabilities. This is a smoothed version of the zero-one loss. For $\lambda > 0$, the regularization term ensures relatively small weights. Thus, the approximate nature of the logistic is observable and we cannot claim to be optimizing for the zero-one loss. However, as $\lambda \rightarrow 0$, there is less encouragement for small weights. Larger weights make the approximation look better. If we take the limit of $\lambda \rightarrow 0$, we end up minimizing the zero-one loss.

We take this approach in an annealing fashion. In other words, we begin with $\lambda^{(1)} = \overline{\lambda} > 0$ (say, $\overline{\lambda} = 100$). This yields a fairly smooth problem, but one that poorly approximates the zero-one loss. After having solved for the appropriate $\theta^{(1)} := \theta$, we decrement λ, e.g. $\lambda^{(2)} = \lambda^{(1)}/2$, and again conduct the minimization, this time using $\theta^{(1)}$ as a starting point for the minimization. We continue in this fashion until λ is small and we observe little change in successive $\theta^{(t)}$'s. The resulting θ is local a minimum of the zero-one loss problem.

3.3 Non-linear Decision Boundary

Thus far we have assumed a linear decision boundary. However, many problems require a decision bounday that is not linear. We extend our framework to non-linear decision boundaries via a technique that has seen recent popularity.

Instead of directly parameterizing a non-linear decision boundary, we take the reverse approach of projecting our data into a higher dimensional space. We then find a linear decision boundary in that space. When this boundary is projected back into the original data space, it is no longer linear—much more complex boundaries are possible. Let $\mathbb{R}^n$ be the low-dimensional space and $\mathcal{H}$ be the high-dimensional space. We can summarize this projection through a kernel function, $K : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathcal{H}$, which returns the value of a dot-product between two points in $\mathcal{H}$. For example, take $K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})^2$, where $\cdot$ indicates linear dot-product. Let $\Phi : \mathbb{R}^n \to \mathcal{H}$ be the projective function, then we can see that $\Phi(x_1, x_2) = (x_1^2, 2x_1x_2, x_2^2)$ corresponds to our example kernel function, i.e. $\Phi(\mathbf{x}) \cdot \Phi(\mathbf{y}) = K(\mathbf{x}, \mathbf{y})$. An advantage to using a kernel function, K, over a projection, Φ, is that we can project into an infinite-dimensional space, for which there is no realizable projection function. For example, $K(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{1}{\sigma}\|\mathbf{x} - \mathbf{y}\|^2\right)$.

We return to our classification objective, with the regularizer added,

$$\min_{\mathbf{w},b} J = \min_{\mathbf{w},b} \frac{\|\mathbf{w}\|^2}{2} - C \log \sum_i \sigma_i, \tag{27}$$

$$\geq \min_{\mathbf{w},b} \frac{\|\mathbf{w}\|^2}{2} - C \sum_i q_i \log \frac{\sigma_i}{q_i}, \tag{28}$$

leading us to the two-stage, EM-like minimization that we have already described. In the M step, we minimize with respect to $\theta = (\mathbf{w}, b)$. We devlop the dual form of our objective by substituting for $\log \sigma_i$. Recall that $z_i = y_i(\mathbf{w}^T x_i + b)$. We use $H(p)$ to denote the binary entropy function.

$$\min_{\mathbf{w},b} J = \min_{\mathbf{w},b} \frac{\|\mathbf{w}\|^2}{2} - C \sum_i q_i \max_{\alpha_i \in [0,1]} (\alpha_i z_i - H(\alpha_i)) \tag{29}$$

$$= \max_{\{\alpha_i \in [0,1]\}} \min_{\mathbf{w},b} \frac{\|\mathbf{w}\|^2}{2} - C \sum_i q_i(\alpha_i z_i + H(\alpha_i)) \tag{30}$$

Certain conditions allow us to reverse the order of the min and max in the objective. Next, we solve for the minimizing $\mathbf{w}$ and b,

$$\frac{\partial J}{\partial w_k} = w_k - C \sum_i q_i \alpha_i y_i x_{ik} = 0 \Rightarrow w_k = C \sum_i q_i \alpha_i y_i x_{ik} \tag{31}$$

$$\frac{\partial J}{\partial b} = -C \sum_i q_i \alpha_i y_i = 0 \Rightarrow \sum_i q_i \alpha_i y_i = 0 \tag{32}$$

Substituting back into Equation 30, we end up with a form that only requires us to evaluate dot-products of data points,

$$\min_{\mathbf{w},b} J = \max_{\{\alpha_i \in [0,1]\}} \sum_i q_i H(\alpha_i) - \frac{C}{2} \sum_{i,j} q_i q_j \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j). \tag{33}$$

Substituting $K(\mathbf{x}_i, \mathbf{x}_j)$ for $(\mathbf{x}_i \cdot \mathbf{x}_j)$ gives us a non-linear decision boundary without the requirement that we directly project any data point into a higher-dimensional space.

What remains for us to show is that we can similiarly replace any instances of data-point computations with the chosen kernel function. We proceded to describe the M step first because our ability to show that the E step is kernelizable relies on the $w_k = C \sum_i \alpha_i y_i x_{ik}$ result from the M step. We avoid the chicken-and-egg problem by using $q^{(1)} = \frac{1}{n}\ \forall i$. The E step minimizes the objective with respect to the q_i, yielding

$$q_i^{(t+1)} = \frac{\sigma_i^{(t)}}{\sum_j \sigma_j^{(t)}}. \tag{34}$$

Note that

$$\sigma_i^{(t)} = \frac{1}{1+e^{-y_i(\mathbf{w}^{(t)}\cdot\mathbf{x}_i+b^{(t)})}} = \frac{1}{1+e^{-y_i(\sum_j \alpha_j y_j(\mathbf{x}_i\cdot\mathbf{x}_j)+b^{(t)})}}, \tag{35}$$

and as before, we can replace $(\mathbf{x}_i \cdot \mathbf{x}_j)$ with $K(\mathbf{x}_i, \mathbf{x}_j)$ to achieve a non-linear decision boundary without having to compute the projection of any x_i.

4 Analysis

We have already given some reason as to why we believe this to be a reasonable technique for classification. In this section, we further refine those reasons and reveal properties of the classifier that may be appealing.

4.1 Convergence in Expectation

One important aspect of a classifier is for the training loss to have a strong connection with the generalization loss. Let $1-f(\{(\mathbf{x}_i, y_i)\};\theta) = 1-\frac{1}{n}\sum_{i=1}^n y_i(\mathbf{w}^T\mathbf{x}_i+b)$ be the average loss on the data points $\{(\mathbf{x}_k, y_k)\}$. Now, consider replacing $(\mathbf{x}_k, y_k)$ with $(\mathbf{x}'_k, y_k)$. The new loss, say $1-f'(\{(\mathbf{x}_i, y_i)\};\theta)$, is bounded by $\frac{1}{n} + 1 - f(\{(\mathbf{x}_i, y_i)\};\theta)$. As a result, we use McDiarmid's theorem [2, 1] to bound the difference between the average loss and the expected loss over the underlying distribution,

$$\Pr\{E[f(\{(\mathbf{x}_i, y_i)\};\theta)] - f(\{(\mathbf{x}_i, y_i)\};\theta) \geq \epsilon\} \leq e^{-n\epsilon^2} \tag{36}$$

This is a powerful statement since it tightly bounds the difference between the average loss on the training data and that of the underlying distribution. Note that if we use the annealing approach described in section 3.2 to approach the zero-one loss, then this is an even more powerful statement—the chance of significant difference between training error and generalization error drops off exponentially as a function of the number of training examples.

The above result holds if we replace $f(\{(\mathbf{x}_i, y_i)\};\theta)$ by $\tilde{f}(\{(\mathbf{x}_i, y_i)\}) = \min_\theta f(\{(\mathbf{x}_i, y_i)\};\theta)$. $1-\tilde{f}(\{(\mathbf{x}_i, y_i)\})$ is the average training loss for the learned decision boundary. What remains is to relate this to the generalization error on the optimal decision boundary. We would like to say something about

$$E[1-\tilde{f}] - \min_\theta E[f(\{(\mathbf{x}_i, y_i)\};\theta), \tag{37}$$

the difference between generalization error on the decision boundary optimized based on the training data and generalization error on the optimal decision boundary.

4.2 Gaussian Classification

Consider the problem of classifying data points generated from two 1-D Gaussian distributions with idential variances. Logistic regression does very well in this case because the logistic function is the log-posterior ratio.

Let $X^+ \sim N(1,1)$ and $X_- \sim N(-1,1)$. Let $\mathbf{x}^+ = (x_1, \dots, x_n)$ and $\mathbf{x}^- = (x_{n+1}, \dots, x_{2n})$ each be a set of independent draws from X^+ and X^-, respectively. Conventional wisdom dictates that when the form of the underlying distribution is known, you should classify using the best-fit parameters (e.g. Maximum a posteriori). For two equal-variance Gaussians, the decision boundary corresponding to the best fit parameters minimizes the logistic loss,

$$L(b) = \sum_{i=1}^{n} \log \sigma(y_i(x_i^+ + b)), \tag{38}$$

where $y_i = +1$ for $i = \{1, \dots, n\}$ and $y_i = -1$ for $i = \{n+1, \dots, 2n\}$. The loss for each data point is convex, so the sum of the losses is convex and the total loss, $L(b)$ has a unique minimum. That minimum occurs where the derivative with respect to b is zero,

$$\frac{\partial L}{\partial b} = \sum_i \frac{(1-\sigma_i)\sigma_i}{\sigma_i} y_i = 0. \tag{39}$$

Equivalently, it occurs when $\sum_i \sigma_i = -2n$.

$$\sum_i \sigma_i = \sum_{i=1}^{n} \frac{1}{1+e^{-(x_i+b)}} + \sum_{i=n+1}^{2n} \frac{1}{1+e^{x_i+b}} \tag{40}$$

$$= \sum_{i=1}^{n} \frac{e^{x_i}}{e^{x_i}+e^{-b}} + \sum_{i=n+1}^{2n} \frac{e^{-b}}{e^{-b}+e^{x_i}} \tag{41}$$

$$\tag{42}$$

5 Conclusion

We have introduced a new class of classification algorithms that use a bounded loss function. By doing this, our optimization problem is no longer convex, but the solution is more closely tied to the classification objective (the zero-one loss). We introduced an algorithm for finding a local minimum of the bounded loss function that begins with the solution of a regularized, convex optimization (such as the Support Vector Machine or regularized Logistic Regression). The algorithm proceeds by reweighting examples and solving a convex optimization at each stage. We believe that this proceedure is advantageous to gradient descent algorithms that are normally used with unbounded loss functions (such as those used with neural networks). We have shown how our algorithm can be generalized to non-linear decision boundaries. Also, using McDiarmid's thorem, we showed the importance of their being a strong connection between the training and generalization loss functions. We believe that current techniques will remain the most effective for problems where noise is low or where there are relatively few outliers; when noise is high or there are many outliers, it is more important that the loss function used in optimization be closely tied to the generalization loss. It is here that we believe our new algorithm may prove beneficial.

Appendix

Proof of Dual Equality

Here we prove that

$$\log \frac{1}{1+e^{-z}} = \min_{a \in [0,1]} az + a \log a + (1-a)\log(1-a) \tag{43}$$

Let $J = az + a\log a + (1-a)\log(1-a)$. Then, $\frac{\partial J}{\partial a} = z + \log \frac{a}{1-a}$. Setting $\frac{\partial J}{\partial a} = 0$, we get

$$a = \frac{1}{1+e^{z}} \tag{44}$$

Substituting back into J, we get

$$J = \log \frac{1}{1+e^{-z}}, \tag{45}$$

which is what we wanted to show.

References

[1] Gábor Lugosi. Concentration-of-measure inequalities. http://www.econ.upf.es/∼lugosi/anu.ps, 2003.

[2] Colin McDiarmid. On the method of bounded differences. In *Surveys in Combinatorics 1989*, pages 148–188. Cambridge University Press, 1989.