Regularized Winnow Methods

Tong Zhang
Mathematical Sciences Department
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598
tzhang@watson.ibm.com

Abstract

In theory, the Winnow multiplicative update has certain advantages over the Perceptron additive update when there are many irrelevant attributes. Recently, there has been much effort on enhancing the Perceptron algorithm by using regularization, leading to a class of linear classification methods called support vector machines. Similarly, it is also possible to apply the regularization idea to the Winnow algorithm, which gives methods we call regularized Winnows. We show that the resulting methods compare with the basic Winnows in a similar way that a support vector machine compares with the Perceptron. We investigate algorithmic issues and learning properties of the derived methods. Some experimental results will also be provided to illustrate different methods.

1 Introduction

In this paper, we consider the binary classification problem that is to determine a label $y \in \{-1, 1\}$ associated with an input vector x. A useful method for solving this problem is through linear discriminant functions, which consist of linear combinations of the components of the input variable. Specifically, we seek a weight vector w and a threshold θ such that $w^T x < \theta$ if its label $y = -1$ and $w^T x \geq \theta$ if its label $y = 1$. Given a training set of labeled data $(x^1, y^1), \ldots, (x^n, y^n)$, a number of approaches to finding linear discriminant functions have been advanced over the years. In this paper, we are especially interested in the following two families of online algorithms: Perceptron [14] and Winnow [11]. These algorithms typically fix the threshold θ and update the weight vector w by going through the training data repeatedly. They are mistake driven in the sense that the weight vector is updated only when the algorithm is not able to correctly classify an example.

For the Perceptron algorithm, the update rule is additive: if the linear discriminant function misclassifies an input training vector x^i with true label y^i, then we update each component j of the weight vector w as: $w_j \leftarrow w_j + \eta x^i_j y^i$, where $\eta > 0$ is a parameter called learning rate. The initial weight vector can be taken as $w = 0$.

For the Winnow algorithm (with positive weight), the update rule is multiplicative: if the linear discriminant function misclassifies an input training vector x^i with true label y^i, then we update each component j of the weight vector w as: $w_j \leftarrow w_j \exp(\eta x^i_j y^i)$, where $\eta > 0$ is the learning rate parameter, and the initial weight vector can be taken as $w_j = \mu_j > 0$. The Winnow algorithm belongs to a general family of algorithms called exponentiated

gradient descent with unnormalized weights (EGU) [10]. There can be several variants. One is called balanced Winnow, which is equivalent to an embedding of the input space into a higher dimensional space as: $\tilde{x} = [x, -x]$. This modification allows the positive weight Winnow algorithm for the augmented input $\tilde{x}$ to have the effect of both positive and negative weights for the original input x. Another modification is to normalize the one-norm of the weight w so that $\sum_j w_j = W$, leading to the normalized Winnow.

Theoretical properties of multiplicative update algorithms have been extensively studied since the introduction of Winnow. For linearly separable binary-classification problems, both Perceptron and Winnow are able to find a weight that separate the in-class vectors from the out-of-class vectors in the training set within a finite number of steps. However, the number of mistakes (updates) before finding a weight can be very different [11, 10]. This difference suggests that the two algorithms serve for different purposes.

For linearly separable problems, Vapnik proposed a method that optimizes the Perceptron mistake bound called "optimal hyperplane" method (see [17]). The same method has also appeared in the statistical mechanical learning literature (see [1, 9, 13]), and is referred to as achieving optimal stability. For non-separable problems, a generalization of optimal hyperplane was proposed in [2] by introducing a "soft-margin" loss term. In this paper, we derive regularized Winnow methods by constructing "optimal hyperplanes" that minimize the Winnow mistake bound (rather than the Perceptron mistake bound as in an SVM). We then derive a "soft-margin" version of the algorithm for non-separable problems.

For simplicity, we shall assume $\theta = 0$ in this paper. The restriction does not cause problems in practice since one can always append a constant feature to the input data x, which offset the effect of θ. The formulation with $\theta = 0$ can be more amenable to theoretical analysis. For an SVM, a fixed threshold also allows a simple Perceptron like numerical algorithm as described in chapter 12 of [15], as well as in [12] and [7]. However, although more complex, a non-fixed θ does not introduce any fundamental difficulty.

The paper is organized as follows. In Section 2, we review mistake bounds for Perceptron and Winnow. Based on the bounds, we show how regularized Winnow methods can be derived by mimicking the optimal stability method (and SVM) for Perceptron. We also discuss the relationship of the newly derived methods with related methods. In Section 3, we investigate learning aspects of the newly proposed methods in a context similar to some known SVM results. Finally an example will be given in Section 4 to illustrate these methods.

2 SVM and regularized Winnow

2.1 From Perceptron to SVM

We review the derivation of SVM from Perceptron, which serves as a reference for our derivation of regularized Winnow. Consider linearly separable problems and let w be a weight that separates the in-class vectors from the out-of-class vectors in the training set. It is well known that the Perceptron algorithm computes a weight that correctly classifies all training data after at most M updates (a proof can be found in [17]) where $M = \|w\|_2^2 \max_i \|x^i\|_2^2 / (\min_i w^T x^i)^2$. The weight vector w_* that minimizes the right hand side of the bound is called the optimal hyperplane in [17] or the optimal stability hyperplane in [1, 9, 13]. This optimal hyperplane is the solution to the following quadratic programming problem:

$$\min_w \frac{1}{2} w^2 \qquad \text{s.t.} \quad w^T x^i y^i \geq 1 \quad \text{for } i = 1, \ldots, n.$$

For non-separable problems, we can introduce a slack variable ξ^i for each data (x^i, y^i) $(i = 1, \ldots, n)$, and compute a weight vector $w_*(C)$ that solves

$$\min_{w,\xi} \frac{1}{2} w^T w + C \sum_i \xi^i \qquad \text{s.t.} \quad w^T x^i y^i \geq 1 - \xi^i, \quad \xi^i \geq 0 \quad \text{for } i = 1, \ldots, n.$$

Where $C > 0$ is a given parameter [17]. It is known that when $C \to \infty$, then $\xi_i \to 0$ and $w_*(C)$ converges to the weight vector w_* of the optimal hyperplane. We can write down the KKT condition for the above optimization problem, and let α^i be the Lagrangian multiplier for $w^T x^i y^i \geq 1 - \xi^i$. After elimination of w and ξ, we obtain the following dual optimization problem of the dual variable α (see [17], chapter 10 for details):

$$\max_{\alpha} \sum_i \alpha^i - \frac{1}{2} (\sum_i \alpha^i x^i y^i)^2 \qquad \text{s.t.} \quad \alpha^i \in [0, C] \quad \text{for } i = 1, \ldots, n.$$

The weight $w_*(C)$ is given by $w_*(C) = \sum_i \alpha^i x^i y^i$ at the optimal solution. To solve this problem, one can use the following modification of the Perceptron update algorithm (see [12, 7] and chapter 12 of [15]): at each data (x^i, y^i), we fix all α_k with $k \neq i$, and update α_i to maximize the dual objective functional, which gives:

$$\alpha^i \to \max(\min(C, \alpha^i + \eta(1 - w^T x^i y^i), 0),$$

where $w = \sum_i \alpha^i x^i y^i$. The learning rate η can be set as $\eta = 1/x^{iT} x^i$ which corresponds to the exact maximization of the dual objective functional.

2.2 From Winnow to regularized Winnow

Similar to Perceptron, if a problem is linearly separable with a positive weight w, then Winnow computes a solution that correctly classifies all training data after at most M updates with $M = 2W(\sum_j w_j \ln \frac{w_j \|\mu\|_1}{\mu_j \|w\|_1}) \max_i \|x^i\|_\infty^2 / \delta^2$, where $0 < \delta \leq \min_i w^T x^i y^i$, $W \geq \|w\|_1$ and the learning rate is $\eta = \delta / (W \max_i \|x^i\|_\infty^2)$. The technique for deriving the above bound was developed in [5] (also see [11] for earlier results). The detailed proof of this specific bound can be found in [18] which employed techniques in [5]. Note that unlike the Perceptron mistake bound, the above bound is learning rate dependent. It also depends on the prior $\mu_j > 0$ which is the initial value of w.

For problems separable with positive weights, to obtain an optimal stability hyperplane associated with the Winnow mistake bound, we consider fixing $\|w\|_1$ such that $\|w\|_1 = W > 0$. It is then natural to define the optimal hyperplane as the (positive weight) solution to the following convex programming problem:

$$\min_{w} \sum_j w_j \ln \frac{w_j}{e \mu_j} \qquad \text{s.t.} \quad w^T x^i y^i \geq 1 \quad \text{for } i = 1, \ldots, n.$$

Similar to the derivation of SVM, for non-separable problems, we can introduce a slack variable ξ^i for each data (x^i, y^i), and compute a weight vector $w_*(C)$ that solves

$$\min_{w,\xi} \sum_j w_j \ln \frac{w_j}{e \mu_j} + C \sum_i \xi^i \qquad \text{s.t.} \quad w^T x^i y^i \geq 1 - \xi^i, \quad \xi^i \geq 0 \quad \text{for } i = 1, \ldots, n.$$

Where $C > 0$ is a given parameter. Note that to derive the above methods, we have assumed that $\|w\|_1$ is fixed at $\|w\|_1 = \|\mu\|_1 = W$, where W is a given parameter. This implies that the derived methods are in fact regularized versions of the normalized Winnow. However in practice, one can also ignore this normalization constraint so that the derived methods correspond to regularized versions of the unnormalized Winnow. It is also worth mentioning that the appearance of the entropy regularization condition is natural to all exponentiated gradient methods investigated in [10], which can be readily observed from the

theoretical results in [10]. The regularized normalized Winnow is closely related to the maximum entropy discrimination studied in [6] (the two methods are almost identical for linearly separable problems). However, in the framework of maximum entropy discrimination, the Winnow connection is quite non-obvious. Their derivation was also not motivated from minimizing the Winnow mistake bound — as we shall show later, it is possible to derive some interesting learning bounds for our formulations that are connected with the Winnow mistake bound.

Similar to the SVM formulation, the non-separable formulation of regularized Winnow approaches the separable formulation as $C \to \infty$. We shall thus only focus on the non-separable case below. Also similar to an SVM, we can write down the KKT condition and let α^i be the Lagrangian multiplier for $w^T x^i y^i \geq 1 - \xi^i$. After elimination of w and ξ, we obtain (the algebra resembles that of [17], chapter 10, which we shall skip due to the limitation of space) the following dual formulation for regularized unnormalized Winnow:

$$\max_\alpha \sum_i \alpha^i - \sum_j \mu_j \exp(\sum_i \alpha^i x_j^i y^i) \qquad \text{s.t.} \quad \alpha^i \in [0, C] \quad \text{for } i = 1, \ldots, n.$$

The j-th component of weight $w_*(C)$ is given by $w_*(C)_j = \mu_j \exp(\sum_i \alpha^i x_j^i y^i)$ at the optimal solution. For regularized normalized Winnow with $\|w\|_1 = W > 0$, we obtain

$$\max_\alpha \sum_i \alpha^i - W \ln(\sum_j \mu_j \exp(\sum_i \alpha^i x_j^i y^i)) \qquad \text{s.t.} \quad \alpha^i \in [0, C] \quad \text{for } i = 1, \ldots, n.$$

The weight $w_*(C)$ is given by $w_*(C)_j = W \mu_j \exp(\sum_i \alpha^i x_j^i y^i) / \sum_j \mu_j \exp(\sum_i \alpha^i x_j^i y^i)$ at the optimal solution.

Similar to the Perceptron-like update rule for the dual SVM formulation, it is possible to derive Winnow-like update rules for the regularized Winnow formulations. At each data (x^i, y^i), we fix all α_k with $k \neq i$, and update α_i to maximize the dual objective functionals. We shall not try to derive an analytical solution, but rather use a gradient ascent method with a learning rate η: $\alpha_i \to \alpha_i + \eta \frac{\partial}{\partial \alpha_i} L_D(\alpha_i)$, where we use L_D to denote the dual objective function to be maximized. η can be either fixed as a small number or computed by the Newton's method. It is not hard to verify that we obtain the following update rule for regularized unnormalized Winnow:

$$\alpha^i \to \max(\min(C, \alpha^i + \eta(1 - w^T x^i y^i), 0),$$

where $w_j = \mu_j \exp(\sum_i \alpha^i x_j^i y^i)$. This gradient ascent on the dual variable gives an EGU rule as in [10]. Compared with the SVM dual update rule which is a soft-margin version of the Perceptron update rule, this method naturally corresponds to a soft-margin version of unnormalized Winnow update. Similarly, we obtain the following dual update rule for regularized normalized Winnow:

$$\alpha^i \to \max(\min(C, \alpha^i + \eta(1 - w^T x^i y^i), 0),$$

where $w_j = W \mu_j \exp(\sum_i \alpha^i x_j^i y^i) / \sum_j \mu_j \exp(\sum_i \alpha^i x_j^i y^i)$. Again, this rule (which is an EG rule in [10]) can be naturally regarded as the soft-margin version of the normalized Winnow update. Note that for regularized normalized Winnow, the normalization constant W needs to be carefully chosen based on data. For example, if data is infinity-norm bounded by 1, then it does not seem to be appropriate if we choose $W \leq 1$ since $|w^T x| \leq 1$: a hyperplane with $\|w\|_1 \leq 1$ does not achieve reasonable margin. This problem is less crucial for unnormalized Winnow, but the norm of the initial weight μ_j still affects the solution.

Besides maximum entropy discrimination which is closely related to regularized normalized Winnow, there has been an earlier attempt to derive a large margin version of unnormalized Winnow in [3] based on some heuristics. However, their algorithm was purely mistake driven without dual variables α^i (the algorithm does not compute an optimal stability hyperplane for the Winnow mistake bound). In addition, they did not include a regularization parameter C which in practice can be important for non-separable problems.

3 Some statistical properties of regularized Winnows

In this section, we show that similar to the case of SVM, we are able to derive learning bounds based on our formulations that minimize the Winnow mistake bound. There are many possible bounds which can be of interests. The following result is an analogy of a leave-one-out cross-validation bound for separable SVMs — Theorem 10.7 in [17].

Theorem 3.1 *The expected misclassification error err_n with the true distribution by using hyperplane w obtained from the linearly separable ($C = \infty$) unnormalized regularized Winnow algorithm with n training samples is bounded by $err_n \leq \frac{1}{n+1} E \min(K, 1.5W(\sum_j w_j \ln \frac{w_j}{\mu_j}) \max_i \|x^i\|_\infty^2)$, where the right-hand side expectation is taken with $n+1$ random samples $(x^1, y^1), \ldots, (x^{n+1}, y^{n+1})$. K is the number of support vectors of the solution. Let w be the optimal solution using all the samples with dual α^i for $i = 1, \ldots, n+1$. Let w^k be the weight obtained from setting $\alpha^k = 0$, then $W = \max(\|w\|_1, \|w^1\|_1, \ldots, \|w^{n+1}\|_1)$.*

Proof Sketch. We only describe the major steps due to the limitation of space. Denote by $\tilde{w}^k$ the weight obtained from the optimal solution by removing (x^k, y^k) from the training sample. Similar to the proof of Theorem 10.7 in [17], we need to bound the leave-one-out cross-validation error, which is at most K. Also note that the leave-one-out cross-validation error is at most $|\{k : \|\tilde{w}^k - w\|_1 \|x^k\|_\infty \geq 1\}|$. We then use the following two inequalities: $\|\tilde{w}^k - w\|_1^2 \leq 2W(\sum_j \tilde{w}_j^k - w_j - w_j \ln(\tilde{w}_j^k / w_j))$; and $\sum_j \tilde{w}_j^k - w_j - w_j \ln(\tilde{w}_j^k / w_j) \leq \sum_j w_j^k - w_j - w_j \ln(w_j^k / w_j)$ — the latter inequality can be obtained by comparing the dual objective functionals and by using the corresponding KKT condition of the dual problem. The remaining problem is now reduced to proving that $|\{k : \sum_j w_j^k - w_j - w_j \ln(w_j^k / w_j) \geq 1/(2W\|x^k\|_\infty^2)\}| \leq \sqrt{2} W \sum_j w_j \ln \frac{w_j}{\mu_j}$. For the dual formulation, by summing over index k of the KKT first order condition with respective to the dual α^k, multiplied by α^k, one obtains $\sum_k \alpha^k = \sum_j w_j \ln \frac{w_j}{\mu_j}$. We thus only need to show that if $\sum_j w_j^k - w_j - w_j \ln(w_j^k / w_j) \geq 1/(2W\|x^k\|_\infty^2)$, then $\alpha^k \geq 2/(3W\|x^k\|_\infty^2)$. This can be checked directly through Taylor expansion. □

By using the same technique, we can also obtain a bound for regularized normalized Winnow, which we shall skip to save space. One disadvantage of the above bound is that it is the expectation of a random estimator that is no better than the leave-one-out cross-validation error based on observed data (it is also difficult to generalize this specific technique to non-separable problems since $\sum_k \alpha^k$ can be unbounded). However, the bound does convey some useful information: for example, we can observe that the expected misclassification error (learning curve) converges at a rate of $O(1/n)$ as long as $W(\sum_j w_j \ln \frac{w_j}{\mu_j})$ and $\sup \|x\|_\infty$ are reasonably bounded.

It is also not difficult to obtain interesting PAC style bounds by using the covering number result for entropy regularization in [18] and ideas in [16]. Although the PAC analysis would imply a slightly suboptimal learning curve of $O(\log n/n)$ for linearly separable problems, the bound itself provides a probability confidence and can be generalized to non-separable problems. We state below an example for non-separable problems, which justifies the entropy regularization. The bound itself is a direct consequence of Theorem 2.2 and Theorem 2.8 in [18]. Note that as in [16], the square root can be removed if $k_\gamma = 0$ and γ can be made data-dependent.

Theorem 3.2 *If the data is infinity-norm bounded as $\|x\|_\infty \leq b$, then consider the family Γ of hyperplanes w such that $\|w\|_1 \leq a$ and $\sum_j w_j \ln(\frac{w_j \|\mu\|_1}{\mu_j \|w\|_1}) \leq c$. Denote by $err(w)$ the misclassification error of w with the true distribution. Then there is a constant C such*

that for any $\gamma > 0$, with probability $1-\eta$ over n random samples, any $w \in \Gamma$ satisfies:

$$err(w) \leq \frac{k_\gamma}{n} + \sqrt{\frac{C}{\gamma^2 n} b^2 (a^2 + ac) \ln(\frac{nab}{\gamma} + 2) + \ln \frac{1}{\eta}},$$

where $k_\gamma = |\{i : w^T x^i y^i < \gamma\}|$ is the number of samples with margin less than γ.

4 Text categorization example

We use a text-categorization application to show that a regularized Winnow can enhance a Winnow just like an SVM can enhance a Perceptron. The text-categorization example is quite interesting for this comparison since SVM is regarded as a state of the art method [4, 8], and unnormalized Winnow has also been applied to the same problem with good results [3]. There has also been some arguments concerning whether there are many irrelevant features (used to support Winnow [3]) or there are many relevant features (used to support SVM) [8]. Our comparison implies that this argument seems less relevant: it is the regularization that truly helps — no matter the underlying algorithm is Perceptron or Winnow. Although in theory, there are differences between the two approaches (for example, see [10]), the practical differences remain to be more carefully investigated.

The standard test set for text-categorization is the Mod-Apte split of the Reuters-21578 data set available from *http://www.research.att.com/∼lewis/reuters21578.html*. There are 9603 training data and 3299 test data. We only report results of the largest 10 categories (each is a binary-classification problem) which can be compared with those of [4]. The performance is usually measured by precision and recall rather than classification error. Precision is the percentage of correctly classified data among all data that are classified to be in-class; recall is the percentage of correctly classified data among all data that are truly in-class. Since we can adjust the threshold of a linear classifier after training to trade-off its precision and recall, a popular performance metric is the break-even point where the threshold is chosen so that precision equals recall (also see [4]).

We use binary word occurrences in the documents as features, and append a constant feature of 1 to offset the effect of θ. We shall only report the balanced versions of the unnormalized Winnows to save space (the performances of normalized versions are similar with appropriate normalization conditions). For consistency and comparison purposes, we fix the learning rates as 0.001 and use 200 iterations over the training data for all algorithms. The initial values for the Winnows are $\mu_j = 0.01$. The "average" in Table 1 is micro-average as in [4]. Our results are comparable with an enhanced SVM result in [4], where the micro-averaged break-even over top 10 categories was 92.0. Note that both SVM and regularized Winnow consistently outperform Perceptron and Winnow, with compatible results among themselves. This implies that although regularization helps, there does not seem to be any practical difference between Perceptron and Winnow in this application.

References

[1] J.K. Anlauf and M. Biehl. The AdaTron: an adaptive perceptron algorithm. *Europhys. Lett.*, 10(7):687–692, 1989.

[2] C. Cortes and V.N. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

[3] I. Dagan, Y. Karov, and D. Roth. Mistake-driven learning in text categorization. In *Proceedings of the Second Conference on Empirical Methods in NLP*, 1997.

[4] S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of the 1998 ACM 7th*

category	Perceptron	SVM	Winnow	Regularized Winnow
earn	98.0	**98.3**	98.1	**98.3**
acq	93.3	95.3	93.2	**95.5**
money-fx	69.8	**73.9**	71.3	73.7
grain	88.6	89.9	87.9	**92.6**
crude	84.1	**84.7**	80.7	84.1
trade	71.8	**75.2**	72.6	74.0
interest	64.1	77.1	68.7	**78.6**
ship	74.9	85.4	76.4	**86.5**
wheat	**85.9**	**85.9**	**85.9**	84.5
corn	78.6	**87.5**	75.0	85.7
Avg Top 10	89.4	91.7	89.5	**91.8**

Table 1: Break-even Performance for 10 Largest Categories

international conference on Information and knowledge management, pages 148–155, 1998.

[5] A.J. Grove, N. Littlestone, and D. Schuurmans. General convergence results for linear discriminant updates. In *Proc. 10th Annu. Conf. on Comput. Learning Theory*, pages 171–183, 1997.

[6] Tommi Jaakkola, Marina Meila, and Tony Jebara. Maximum entropy discrimination. In S.A. Solla, T.K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 470–476. MIT Press, 2000.

[7] T.S. Jaakkola, Mark Diekhans, and D. Haussler. A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology*, to appear.

[8] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In *European Conference on Machine Learing, ECML-98*, pages 137–142, 1998.

[9] W. Kinzel. Statistical mechanics of the perceptron with maximal stability. In *Lecture Notes in Physics*, volume 368, pages 175–188. Springer-Verlag, 1990.

[10] J. Kivinen and M.K. Warmuth. Additive versus exponentiated gradient updates for linear prediction. *Journal of Information and Computation*, 132:1–64, 1997.

[11] N. Littlestone. Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.

[12] O.L. Mangasarian and D.R. Musicant. Successive overrelaxation for support vector machines. *IEEE Transactions on Neural Networks*, 10(5):1032–1037, 1999.

[13] M. Opper. Learning times of neural networks: Exact solution for a perceptron algorithm. *Phys. Rev. A*, 38(7):3824–3826, 1988.

[14] F. Rosenblatt. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan, New York, 1962.

[15] Bernhard Scholkopf, Christopher J. C. Burges, and Alex J. Smola., editors. *Advances in Kernel Methods : Support Vector Learning*. The MIT press, 1999.

[16] J. Shawe-Taylor, P.L. Bartlett, R.C. Williamson, and M. Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Trans. Inf. Theory*, 44(5):1926–1940, 1998.

[17] V.N. Vapnik. *Statistical learning theory*. John Wiley & Sons, New York, 1998.

[18] Tong Zhang. Analysis of regularized linear functions for classification problems. Technical Report RC-21572, IBM, 1999. An early abstract in NIPS'99, pp. 370–376.