Online Learning by Ellipsoid Method

Liu Yang LIUY@CS.CMU.EDU
Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA 15213-3891, USA

Rong Jin RONGJIN@CSE.MSU.EDU
Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824, USA

Jieping Ye JIEPING.YE@ASU.EDU
Department of Computer Science and Engineering, Arizona State University, Tempe, AZ 85287-8809, USA

Abstract

In this work, we extend the ellipsoid method, which was originally designed for convex optimization, for online learning. The key idea is to approximate by an ellipsoid the classification hypotheses that are consistent with all the training examples received so far. This is in contrast to most online learning algorithms where only a single classifier is maintained at each iteration. Efficient algorithms are presented for updating both the centroid and the positive definite matrix of ellipsoid given a misclassified example. In addition to the classical ellipsoid method, an improved version for online learning is also presented. Mistake bounds for both ellipsoid methods are derived. Evaluation with the USPS dataset and three UCI data-sets shows encouraging results when comparing the proposed online learning algorithm to two state-of-the-art online learners.

1. Introduction

Online learning aims to learn statistical models from sequentially received training examples. Compared to batch model learing, one of the key requirement for online learning is that the statistical model has to be updated efficiently given a new training example. In the past decades, a large number of online learning algorithms have been proposed and studied (Li & Long, 2002; Gentile, 2002; Crammer & Singer, 2003; Crammer et al., 2006; Rosenblatt, 1958; Kivinen & M.K.Warmuth, 1997; Littlestone, 1989; Gentile & M.Warmuth, 1998; Kivinen et al., 2002). Most of them are additive, i.e., given a misclassified example (x_i, y_i), the classification model, denoted by a weight vector w, is usually updated by shifting along the direction of $y_i x_i$, i.e., $w + \alpha_i y_i x_i \to w$ where α_i weights the misclassified example. (Grove et al., 2001) generalized the additive approaches by an quasi-additive framework which unifies a number of seemingly different online learning algorithms (e.g., Perception and Winnow). Several strategies were proposed to extend online learning algorithms, which were originally proposed for binary classification, to multilabel learning (Fink et al., 2006; Crammer & Singer, 2003; Crammer et al., 2006). (Herbster et al., 2005) extended graph-based approaches for online learning, and (Shalev-Shwartz & Singer, 2006; Amit et al., 2007) exploited the dual formation of optimization for online learning.

One common feature shared by most online learning algorithms is that they only maintain a single solution for the classification model at any trials. We discuss the shortcoming of this feature from two different respective: (I) *Bayesian viewpoint*. By only maintaining a single solution, these online learning approaches are similar to the point estimation in statistics. This is insufficient from the Bayesian viewpoint, which requires computing not only the most likely solution but also the distribution of all possible solutions. (II) *Information viewpoint*. These online learning approaches essentially summarize all the information of training data into a single solution, and therefore is inefficient in exploiting the training data.

To address the above problems, we propose ellipsoid methods for online learning. Instead of only maintaining a single solution, we follow the Bayesian spirit and approximate by an ellipsoid all the classification models that are consistent with the training examples received so far. Since each ellipsoid is described by two quantities, i.e., the centroid of ellipsoid and the positive definite matrix that decides the shape of ellipsoid, the ellipsoid methods are able to maintain more information of training data than most existing

Appearing in *Proceedings of the 26^{th} International Conference on Machine Learning*, Montreal, Canada, 2009. Copyright 2009 by the author(s)/owner(s).

online learning algorithms.

2. Online Learning by Ellipsoid Methods

We first introduce the classical ellipsoid method for convex programming, followed by two variants of ellipsoid method for online learning.

2.1. Introduction to Ellipsoid Method for Convex Programming

Ellipsoid method (Shor, 1977) is a first order method for convex programming. Given an optimization problem $x^* = \arg\min\{f(x) : x \in G\}$ where $f(x)$ is a convex objective function and $G \subset \mathbb{R}^d$ is a convex solid, the ellipsoid method starts with a large ellipsoid $\mathcal{E}_1 \supseteq G$. Let $\mathcal{E}_k = \{x|(x-x_k)^\top P_k^{-1}(x-x_k) \le 1\}$ be the ellipsoid available at the k iteration that includes the optimal solution x^*. Here $x_k \in \mathbb{R}^d$ is the center of $\mathcal{E}_k$, and $P_k \in S_{++}^{d\times d}$ is a positive definite matrix that defines the shape of $\mathcal{E}^k$. The key question of the ellipsoid method is how to update the ellipsoid efficiently. To this end, it computes the gradient of $f(x)$ at x^k, denoted by h_k, and constructs a half-plane $\mathcal{P}_k = \{x|h_k^\top(x-x_k) \le 0\}$. Using the convexity of $f(x)$, it is easy to show $x^* \in \mathcal{P}_k \cap \mathcal{E}_k$. Hence, the new ellipsoid $\mathcal{E}_{k+1} = \{x|(x-x_{k+1})P_{k+1}^{-1}(x-x_{k+1}) \le 1\}$ is constructed to cover the interaction $\mathcal{P}_k \cap \mathcal{E}_k$, where x_{k+1} and P_{k+1} are computed as follows

$$x_{k+1} = x_k - \frac{P_k h_k}{(d+1)\sqrt{h_k^\top P_k h_k}},$$
$$P_{k+1} = \frac{d^2}{d^2-1}\left(P_k - \frac{2P_k h_k h_k^\top P_k}{(d+1)h_k^\top P_k h_k}\right) \qquad (1)$$

2.2. The Classical Ellipsoid Method for Online Learning (CELLIP)

In this section, we focus on binary classification problems, and assume that there exists an γ-margin classifier $u \in \mathbb{R}^d$ that classifies any instance (x, y) with a margin γ, i.e., $yu^\top x \ge \gamma$, where $x \in \mathbb{R}^d$ and $y \in \{-1, +1\}$. For the convenience of discussion, we assume $|u|_2 = 1$ and $|x|_2 \le 1$ for any instance. The extension of the ellipsoid method to the inseparable case and multiple-label learning will be discussed later.

To exploit the ellipsoid method, we convert an online learning problem into a feasibility problem, namely how to efficiently find a solution that is close to the γ-margin classifier u given the sequentially received training examples. In particular, at each trial t, we consider constructing the set $\mathcal{A}_t$ that is defined as follows:

$$\mathcal{A}_t = \{z \in \mathbb{R}^d|y_i x_i^\top z \ge a\gamma, i = 1, \dots, t\} \qquad (2)$$

According to the above definition, $\mathcal{A}_t$ includes all the classifiers z that are able to classify with margin $a\gamma$ the training examples received in the first t iterations. Here $0 \le a \le 1$ is predefined constant. The following lemma shows an important property of $\mathcal{A}_t$.

Lemma 1. *Let $\mathcal{B} = \{z||z-u|_2 \le (1-a)\gamma\}$ denote a ball centering at u with radius $r = (1-a)\gamma$, where $u \in \mathbb{R}^d$ is a γ-margin classifier for all labeled instances. We have $\mathcal{A}_t \supseteq \mathcal{B}$.*

Proof. First, we have $u \in \mathcal{A}_t$ because $y_i x_i^\top u \ge \gamma \ge a\gamma, i = 1, \dots, t$. Hence, to show $\mathcal{A}_t \supseteq \mathcal{B}$, it is sufficient to show the distance between u and hyper-plane $y_t x_t^\top z = a\gamma$ is upper bounded by $(1-a)\gamma$, which can be verified easily. □

Lemma 1 indicates that if there exists an γ-margin classifier u, the volume of $\mathcal{A}_t$, denoted by $vol(\mathcal{A}_t)$, is lower bounded by $vol(\mathcal{B})$, which becomes the key to the proof of mistake bound. To efficiently represent $\mathcal{A}_t$, we construct an ellipsoid

$$\mathcal{E}_t = \{z \in \mathbb{R}^d|(z-w_t)^\top P_t^{-1}(z-w_t) \le 1\} \qquad (3)$$

such that $\mathcal{E}_t \supseteq \mathcal{A}_t$. Since $\mathcal{E}_t \supseteq \mathcal{A}_t \supseteq \mathcal{B}$, our goal is to efficiently reduce $vol(\mathcal{E}_t)$. Below we describe how to efficiently update the ellipsoid $\mathcal{E}_t$ given a misclassified example.

Let $x_t \in \mathbb{R}^d$ be an example that is misclassified by w_t, i.e., $y_t w_t^\top x_i \le 0$ where $y_t \in \{-1, +1\}$ is the binary class label assigned to x_t. Let $\mathcal{C}_t = \{z \in \mathbb{R}^d|y_t x_t^\top z \ge a\gamma\}$ denote the half plane generated by the misclassified example. Evidently, we have $u \in \mathcal{C}_t \cap \mathcal{E}_t$ since $y_t u^\top x_t \ge \gamma$. For the convenience of discussion, we rewrite the set $\mathcal{C}_t$ as follows

$$\mathcal{C}_t = \{z \in \mathbb{R}^d|\alpha_t - g_t^\top(z-w_t) \le 0\} \qquad (4)$$

where α_t and g_t are defined as

$$\alpha_t = \frac{a\gamma - y_t w_t^\top x_t}{\sqrt{x_t^\top P_t x_t}}, \quad g_t = \frac{y_t x_t}{\sqrt{x_t^\top P_t x_t}} \qquad (5)$$

Note that $\alpha_t \ge 0$ since $y_t w_t^\top x_t \le 0$ and $g_t^\top P_t g_t = 1$. The following theorem shows a family of updating equations for w_t and P_t that ensures $\mathcal{E}_{t+1} \supseteq \mathcal{E}_t \cap \mathcal{C}_t$.

Theorem 1. *Given a misclassified instance (x_t, y_t), the following updating equations for w_{t+1} and P_{t+1} will guarantee that the resulting new ellipsoid $\mathcal{E}_{t+1}$ covers the intersection $\mathcal{E}_t \cap \mathcal{C}_t$:*

$$w_{t+1} = w_t + (\alpha_t + \rho)P_t g_t \qquad (6)$$
$$P_{t+1} = \mu^2 P_t + ([1-\alpha_t-\rho]^2 - \mu^2)P_t g_t g_t^\top P_t \qquad (7)$$

Algorithm 1 The classical ellipsoid method (CELLIP) for online learning

```
INPUT:
    • $\gamma \geq 0$: the desired classification margin
    • $a \in [0,1]$: a tradeoff parameter
INITIALIZE: $w_1 = \mathbf{0}$ and $P_1 = (1+(1-a)\gamma)I_d$
for $t = 1, 2, \dots, T$ do
    receive an instance $x_t$
    predict its class label: $\hat{y}_t = \text{sign}(w_t^\top x_t)$
    receive correct class label $y_t$
    if $y_t \neq \hat{y}_t$ then
        compute $w_{t+1}$ and $P_{t+1}$ using (10) and (11)
    else
        $w_{t+1} \leftarrow w_t$ and $P_{t+1} \leftarrow P_t$
    end if
end for
```

if parameter $\rho > 0$ and $\mu > 0$ satisfy the following constraint

$$\frac{1-\alpha_t^2}{\mu^2} + \frac{\rho^2}{(1-\alpha_t-\rho)^2} \leq 1 \tag{8}$$

The proof can be found in the Appendix A of the supplementary materials. The following corollary shows the volume reduction after the update.

Corollary 2. *Using the updating equations in (6) and (7), we have*

$$\frac{vol(\mathcal{E}_{t+1})}{vol(\mathcal{E}_t)} = \mu^{d-1}(1-\alpha_t-\rho) \tag{9}$$

For the convenience of discussion, we choose $\rho = 0$ and $\mu = \sqrt{1-\alpha_t^2}$. The corresponding updating equations for w_t and P_t become

$$\begin{aligned} w_{t+1} &= w_t + \alpha_t P_t g_t & (10) \\ P_{t+1} &= (1-\alpha_t^2)P_t - 2\alpha_t(1-\alpha)P_t g_t g_t^\top P_t & (11) \end{aligned}$$

The volume reduction under the above updating equation is

$$\frac{\text{vol}(\mathcal{E}_{t+1})}{\text{vol}(\mathcal{E}_t)} = (1-\alpha_t^2)^{(d-1)/2}(1-\alpha_t) \tag{12}$$

Algorithm 1 summarizes the classical ellipsoid method for online learning. Note that in Algorithm 1, we initialize $P_1 = (1+(1-a)\gamma)I$ to ensure $\mathcal{B} = \{z||z-u|_2 \leq (1-a)\gamma\} \subseteq \mathcal{E}_1$. We refer to it as *Classical Ellipsoid Method for Online Learning*, or **CELLIP** for short. The following theorem shows the mistake bound for CELLIP.

Theorem 3. *Let $\mathcal{D} = \{(x_i, y_i), i = 1, 2, \dots, T\}$ be the set of training examples. Assume all the examples are normalized, i.e., $\|x_i\|_2 \leq 1$. We assume that there exists an classifier $u \in \mathbb{R}^d$ with $\|u\|_2^2 = 1$ that is able to classified all the training examples in $\mathcal{D}$ with a margin $0 \leq \gamma \leq 1$, i.e., $y_i u^\top x_i \geq \gamma$ for any (x_i, y_i) in $\mathcal{D}$. We then have the mistake made by the classical ellipsoid method when learning from $\mathcal{D}$ (Algorithm 1), denoted by M, upper bounded by*

$$M \leq \frac{2\log(1-a) + 2\log\gamma - \log(1+(1-a)\gamma)}{\log\left(1-a^2\gamma^2/(1+(1-a)\gamma)^2\right)} \tag{13}$$

The proof of the above theorem can be found in Appendix B of the supplementary materials.

2.3. Improved Ellipsoid Method for Online Learning

One major problem with the above classical ellipsoid method for online learning is that it is theoretically incapable of handling the inseparable case. In this subsection, we present an improved ellipsoid method for online learning that is able to address the inseparable case.

Clearly, for the inseparable cases, we have to drop the idea of casting online learning as a feasibility problem since no classifier can classify all the instances correctly. Instead, we treat w_t and P_t, i.e., the center and the positive definite matrix of ellipsoid, as a summarization of information from the received training examples. Since w_{t+1} is a linear combination of the training examples received in the first t trials, it can be viewed as a kind of first order statistics for training examples. To understand the relationship between P_t and received training examples, we derive the updating equation for P_t^{-1} using (11)

$$P_{t+1}^{-1} = \frac{1}{1-\alpha_t^2}P_t^{-1} + \frac{2\alpha_t}{(1-\alpha_t)^2(1-\alpha_t)}g_t g_t^\top$$

The above expression follows directly from the matrix inverse lemma. Using the above expression, it is not difficult to show

$$P_{t+1}^{-1} = \theta_0 P_1^{-1} + \sum_{i=1}^t \theta_i g_i g_i^\top \propto \theta_0 P_1 + \sum_{i=1}^t \xi_i x_i x_i^\top \tag{14}$$

where θ_i and ξ_i are functions of $\{\alpha_j\}_{j=i}^t$. The expression in (14) indicates that P_t^{-1} can be viewed as a weighted covariance matrix that stores the second order information of training examples. The above observation motivates the development of an improved ellipsoid method for online learning.

We keep the updating equation (10) for w_t, and modify the updating equation for P_t as follows

$$P_{t+1} = \frac{1}{1-c_t}(P_t - c_t P_t g_t g_t^\top P_t) \tag{15}$$

where $c_t \in [0,1]$. We set $c_t = cb^{t-1}$ where $0 \leq c, b \leq 1$ are two constants that are set manually. The exponential

Algorithm 2 The improved ellipsoid method (IELLIP) for online learning

```
INPUT:
  • $\gamma \geq 0$: the desired classification margin
  • $0 \leq c, b \leq 1$: parameters controlling the memory of online learning
INITIALIZE: $w_1 = \mathbf{0}$ and $P_1 = I_d$
for $t = 1, 2, \ldots, T$ do
  receive an instance $x_t$
  predict its class label: $\hat{y}_t = \text{sign}(w_t^\top x_t)$
  receive correct class label $y_t$
  if $y_t \neq \hat{y}_t$ then
    compute $w_{t+1}$ and $P_{t+1}$ using (10) and (15)
  else
    $w_{t+1} \leftarrow w_t$ and $P_{t+1} \leftarrow P_t$
  end if
end for
```

form for c_t is important for the proof of mistake bound as revealed in the supplementary materials. It is not difficult to verify the inductive relationship for P_t^{-1}, i.e.,

$$P_{t+1}^{-1} \quad = \quad (1 - c_t)P_t^{-1} + c_t g_t g_t^\top$$

As indicated above, P_{t+1}^{-1} can be viewed as a mixture of matrices P_t^{-1} and $g_t g_t^\top$. Given $c_t = cb^{t-1}$, it is not difficult to see that P_{t+1} is a weighted sum of $x_i x_i$ where the weight for $x_i x_i$ decays exponentially in t. By varying constant c and b, we are able to adjust "memory" of P_t. In particular, the smaller b is, the shorter the memory is. The effect of b will be further revealed in our empirical study. Algorithm 2 summarizes the improved ellipsoid method for online learning. We refer to it as *Improved Ellipsoid Method for Online Learning*, or **IELLIP** for short.

Before we present the mistake bound for the improved ellipsoid method, like many online learning algorithms, we introduce the following quantity for measuring the progress of online learning

$$q_t \quad = \quad (u - w_t)^\top P_t^{-1}(u - w_t) \tag{16}$$

where u is some optimal classifier. Note that compared to the conventional approaches for analysis of online learning algorithms, we introduce P_t^{-1} in (20) for measuring the distance between u and w_t. The following lemma shows an important inductive property for q_t, which is key to the proof of mistake bound

Lemma 2.

$$q_{t+1} \quad \leq \quad (1 - c_t)q_t + \gamma_t^2 + c_t(u^\top g_t)^2 - 2\gamma_t u^\top g_t \tag{17}$$

where $\gamma_t = \gamma/\sqrt{x_t P_t x_t}$.

It is straightforward to verify the result in Lemma 2. We now state the mistake bound for the improved ellipsoid for online learning.

Theorem 4. *Let* $\mathcal{D} = \{(x_i, y_i), i = 1, 2, \ldots, T\}$ *be the set of training examples. Let* u *be the optimal classifier with norm* $|u|_2^2 = 1$. *Assume all the examples are normalized, i.e.,* $\|x_i\|_2 \leq 1$. *If parameter* c*, and* b *satisfy conditions* $c + b < 1$, *we have the number of mistakes made by running Algorithm 2 upper bounded by the following expression*

$$M \leq \frac{1}{\gamma^2} + \frac{2}{\gamma}\frac{1-b}{1-b-c}\sum_{i=1}^{T} l_i(u) \tag{18}$$

where $l_i(u) = \max(0, \gamma - u^\top x_i)$.

The proof of Theorem 4 can be found in Appendix C of the supplementary materials. Note that when $c = 0$, the mistake bound is reduced to $1/\gamma^2 + \sum_{i=1}^T l_i(u)/\gamma$, a common mistake bound for online learning.

2.4. Ellipsoid Methods for Multiple-Label Online Learning

We now follow the framework by Crammer et al. (Crammer & Singer, 2002) and extend the ellipsoid method to multi-label learning. Let K be the total number of classes. We denote by $w_i \in \mathbb{R}^d, i = 1, \ldots, K$ as the weight vectors for the K classes. Given an example x assigned to a subset of classes Y, we define the classification margin with respect to a classifier w as $\eta(W; x, Y) = \min_{z \in Y} w_z^\top x - \max_{z \notin Y} w_z^\top x$. We then define the loss function $l(W; x, Y)$ as $l(W; x, Y) = \max(0, \gamma - \eta(W; x, Y))$ where γ is a predefined margin.

To extend the ellipsoid method for multi-label learning, we construct vector $v = (w_1, \ldots, w_K)$. For a misclassified example (x_i, Y_i), i.e., $\eta(W; x_i, Y_i) \leq 0$, we define two class indices a_i and b_i as

$$a_i = \max_{y \notin Y_i} w_y^\top x_i, \quad b_i = \min_{y \in Y_i} w_y^\top x_i$$

We the construct a big vector $z_i \in \mathbb{R}^{K \times d}$ that includes information from x_i and Y, i.e.,

$$z_i^j = \begin{cases} x_i^k & j = (b_i - 1)d + k \\ -x_i^k & j = (a_i - 1)d + k \\ 0 & \text{otherwise} \end{cases}$$

Similar to the previous discussion, we construct a half plane $\mathcal{P}_t$ for each misclassified example z_t

$$P_t = \{v \in \mathbb{R}^{K \times d} | \alpha_t - (v - v_t)^\top g_t \leq 0\}$$

where α_t and g_t are identical the expressions in (5) except that $y_t x_t$ is replaced by z_t. Using the definition of classifier v, misclassified example z_i, α_t and g_t, we can directly extend the two ellipsoid methods described in Algorithm 1 and 2 to multi-label learning. Similar mistake bounds can be derived for multi-label learning. Since the proof is literally a word-by-word copy of the proof for binary classification, we omit them completely.

3. Evaluation

We focus on the evaluation of the improved ellipsoid method for online learning. This is because the classical ellipsoid method for online learning is theoretically unable to handle the inseparable cases as pointed out before. This is further confirmed by our empirical study, which showed the classical ellipsoid method is usually outperformed by the improved version. We thus omit the discussion for the classical ellipsoid method due to the space limitation.

For IELLIP, we initialize P_1 to be an identity matrix at the scale of 0.1; the vector $\mathbf{w}$ is randomly initialized around the origin. We set $b = 0.3$ for all experiments except for the experiment that is devoted to examining the role of b in the proposed online learning algorithm. Note that since only the relatively scale between P_1 and c is useful, by setting the scale of P, we don't have to set c in the implementation of IELLIP.

3.1. Datasets

The experiments are conducted on the USPS data-set of handwritten digits and three UCI multiclass data-sets (`http://archive.ics.uci.edu/ml/data-sets.html`). The data information is summarized in Table 1. For the UCI Isolet and Letter datasets, we select 80% from each class to form the training data-set, and use the rest as testing data. For the USPS and UCI Shuttle dataset, we adopt the splitting between training and testing as provided in the original data packages.

3.2. Baseline Methods and Evaluation Metrics

To demonstrate the efficiency and efficacy of IELLIP for multi-class learning, we compare it to two baseline algorithms. The first baseline is the *Online Passive-Aggressive algorithm (PA)* (Crammer et al., 2006). We implement the PA algorithm, by using the aggressiveness parameter corresponding to the best performance evaluated in (Crammer et al., 2006). As indicated in (Crammer, 2004), PA in general performs better than the generalized Perceptron algorithms because of the aggressiveness (i.e. large margins). The second baseline algorithm is the *Margin Infused Relaxed Algorithm (MIRA)* (Crammer & Singer, 2003), an online learning algorithm for multiclass large margin classifiers with good generalization performance. We use in our experiment the implementation of MIRA downloaded from `http://www.cis.upenn.edu/~crammer/code-index.html`. For fair comparison, all methods are restricted to use linear classifiers. To this end, for MIRA, we set the polynomial degree to be 1. The margin parameter was set to be 0.1 for all algorithms and for all datasets. Test error (Crammer & Singer, 2003) is used as the main evaluation metric in our study. It is defined as the number of prediction mistakes made on a given sequence of examples normalized by the length of the sequence. The traditional concept of "epoch" is adopted as an ordering (random permutations) of all the examples in the training set. For example, in our experiments of three epoches, we cycle through all the training examples three times, with a different random permutation for each epoch, before calling the online learner. We report the results averaged over three random permutations for all the four datasets.

3.3. Results of Multiclass Classification

Fig 1 shows the classification results of the three online learning algorithms for datasets USPS, UCI Letter, UCI Isolet, and UCI Shuttle: the first row shows the test errors, and the second row shows the number of updates; the three columns, from left to right, correspond to the results of the first, second, and third epoch, respectively.

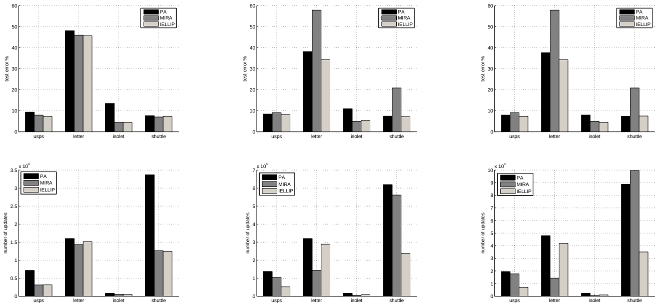
First, as indicated in the first row of Fig 1, we observe that the test error of IELLIP is either comparable to or better than the best performance between PA and MIRA. The second row of Fig 1 reveals that in general, a smaller number of updates are required by IELLIP to achieve a test error that is either comparable to or better than that of PA and MIRA. For instance, for dataset UCI Shuttle, we found that both PA and IEELIP achieve similar test errors across three epoches. But, the number of updates made by IELLIP is significantly smaller than that of PA. One exception is dataset UCI Letter, in which the number of updates made by IELLIP and PA are significantly larger than that of MIRA for the second and third epoch. However, it is also important to note that the test error of IELLIP and PA are significantly lower than that of MIRA for both epoches. When we compare IELLIP with PA on dataset UCI Letter, we still observe a noticeable reduction in the number of updates by IELLIP. Therefore, we conclude that the proposed online learning algorithm is more efficient than the two baselines. Furthermore, since the number of updates is closely related to the number of examples used to construct the classifier, the above analysis indicates that the proposed approach tends to favor a sparse solution than PA and MIRA, a desirable property to have.

3.4. The Role of Parameter b: Tradeoff Between Accuracy and Sparseness

As indicated in the previous analysis, parameter b controls the "memory" of the proposed algorithm. In order to examine the role of parameter b, we follow (Crammer & Singer, 2003), in which a natural tradeoff between accuracy and sparseness of solutions was revealed for a family of additive online learners. Fig 2 shows the number of updates vs. test errors of IELLIP for all four datasets at the 3rd epoch

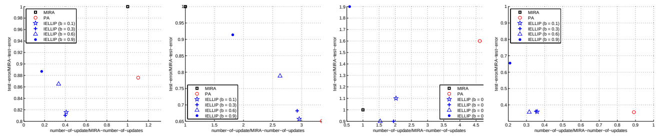
Table 1. Data-sets used in the online learning experiments

NAME	NO. OF TRAINING EG.S	NO. OF TESTING EG.S	NO. OF CLASSES	NO. OF ATTRIBUTES
USPS [1]	7,291	2,007	10	256
UCI LETTER[2]	15,998	4,002	26	16
UCI ISOLET	800	200	10	200
UCI SHUTTLE	43,500	14,500	7	9



(a) 1st epoch (b) 2nd epoch (c) 3rd epoch

Figure 1. Experimental results for datasets USPS, UCI Letter, UCI Isolet, and UCI Shuttle. The first row shows the test errors, and the second row shows the number of updates.



(a) USPS (b) UCI Letter (c) UCI Isolet (d) UCI Shuttle

Figure 2. Experimental results for IELLIP using different b. X-axis and Y-axis represent the number of updates and test errors that are normalized by the corresponding quantities of MIRA.

when we vary b from 0.1 to 0.3, 0.6, and 0.9. For the convenience of comparison, both X-axis and Y-axis, which respond to the number of updates and test errors respectively, are normalized by the quantities of MIRA. Note that since the sparsity of solutions is closely related to the number of updates, the plots in Fig 2 essentially reveal the tradeoff between accuracy and sparseness of solutions that are controlled by the parameter b. We clearly see a overall trend between accuracy and sparseness across all four datasets. In particular, a larger b usually leads to a higher sparseness (i.e., a smaller number of updates) and a lower accuracy (i.e., a higher test error). This can be understood as follows: when we keep a longer history of training examples in P matrix (i.e., a large b), the learning algorithm is less likely to be updated, and as a consequence, those important examples may not be assigned enough weights, which could lead to a lower classification accuracy. Hence, by setting b a modest value (e.g., 0.3), we able to achieve a balance between accuracy and sparseness of solutions, as revealed by the previous study.

4. Conclusion

We present novel methods for online learning method (ELLIP) by exploiting the ellipsoid method for convex programming. Unlike the conventional approaches for online learning that only maintain a single classifier, the proposed method is able to capture all the classifiers that are consistent with training examples via an ellipsoid. In addition, the shape of the ellipsoid, represented by a positive definite matrix, allows us to store more information of training examples, and provide additional controls for online updating. We also present an analysis of mistake bound and a generalization to multi-label learning for the ellipsoid method. Empirically studies demonstrates the effectiveness of the proposed method, compared with two state-of-the-art online learners. In the future, we plan to examine other variants of ellipsoid methods for online learning.

Acknowledgments

The work was partially supported by NSF (DBI-0640543 and IIS-0643494) and US Army Research Office (W911NF-08-1-0403). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF and ARO.

Appendix A: Proof of Theorem 1

We define $v = P_t^{-1/2}(z - w_t)$, and a unit ball and a half plane for v as $\widetilde{\mathcal{E}} = \{v||v|_2 \leq 1\}$ and $\widetilde{\mathcal{C}}_t = \{v|\alpha_t - g_t^\top P_t^{1/2} v \leq 0$. We then rewrite $\mathcal{E}_t$ and $\mathcal{C}_t$ as $\mathcal{E}_t = \{z = w_t + P_t^{1/2} v | v \in \widetilde{\mathcal{E}}\}$ and $\mathcal{C}_t = \{z = w_t + P_t^{1/2} v | v \in \widetilde{\mathcal{C}}_t\}$. We thus have

$$vol(\mathcal{E}_t \cap \mathcal{C}) = |P_t|^{1/2} vol(\widetilde{\mathcal{E}} \cap \widetilde{\mathcal{C}}_t)$$

Figure 3 shows an example of the intersection between the unit ball $\widetilde{\mathcal{E}}$ and the hyper-plane $\alpha_t - g_t^\top P_t^{1/2} v \leq 0$. Note that $P_t^{1/2} g_t$ is an unit vector because $[P_t^{1/2} g_t]^\top P_t^{1/2} g_t = g_t^\top P_t g_t = 1$. Using the symmetry argument, the new ellipsoid in the transformed space, denoted by $\widetilde{\mathcal{E}}_{t+1} =$

$\{v|(v-v_0)^\top Q^{-1}(v-v_0) \le 1\}$, should have its center v_0 move along the direction of $P_t^{1/2}g_t$. We denote by ρ the distance between the center of $\widetilde{\mathcal{E}}_{t+1}$ and the hyper-plane $\widetilde{\mathcal{C}}_t$. As shown in Figure 3, the center v_0 is written as

$$v_0 = (\alpha_t + \rho)P_t^{1/2}g_t \tag{19}$$

Furthermore, based on the argument of symmetry, the matrix Q of ellipsoid $\widetilde{\mathcal{E}}_{t+1}$ should be isometric along almost all the directions except for $P_t^{1/2}g_t$, and therefore can be written as

$$Q = \mu^2 I + ((1-\alpha_t-\rho)^2 - \mu^2)P_t^{1/2}g_t g_t^\top P_t^{1/2} \tag{20}$$

where $1-\alpha_t-\rho$ is the length for axle $P_t^{1/2}g_t$ and $\mu > 0$ is the length of other axles. Using the transform $z = w_t + P_t^{1/2}v$, we have $\mathcal{E}_{t+1}$ expressed in terms of both v_0 and Q, which further leads to the updating equations in Theorem 1. To ensure $\mathcal{E}_{t+1} \supseteq \mathcal{E}_t \cap \mathcal{C}_t$, we enforce the point e in Figure 3, i.e., an intersection point between $\widetilde{\mathcal{E}}$ and $\widetilde{\mathcal{C}}_t$, to be on the surface of the new ellipsoid $\widetilde{\mathcal{E}}_{t+1}$. Note, if we use the center of $\widetilde{\mathcal{E}}_{t+1}$ as the origin and its axles as bases, the coordinates of point e becomes $(\rho, \sqrt{(1-\alpha_t^2)/(d-1)}, \ldots, \sqrt{(1-\alpha_t^2)/(d-1)})$. Since $e \in \widetilde{\mathcal{E}}_{t+1}$, we have

$$\frac{\rho^2}{(1-\alpha_t-\rho)^2} + (d-1)\frac{(1-\alpha_t^2)/(d-1)}{\mu^2} \le 1,$$

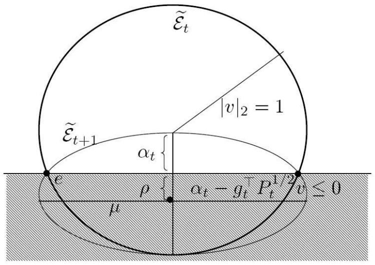

$\widetilde{\mathcal{E}}_t$
$|v|_2 = 1$
$\widetilde{\mathcal{E}}_{t+1}$
α_t
e
ρ
$\alpha_t - g_t^\top P_t^{1/2} v \le 0$
μ

Figure 3. Illustration of updating ellipsoids

Appendix B: Proof of Theorem 3

We will first show the properties of α_t and P_t that are useful for our proof of mistake bound.

Lemma 3. *We have the following properties for* P_t.

$$g_t^\top P_t g_t = 1 \tag{21}$$

$$x_t P_t x_t \le \prod_{i=1}^{t-1}(1-\alpha_t)^2 x_t^\top P_1 x_t \tag{22}$$

$$P_{t+1}^{-1} = (1-\alpha_t^2)P_t^{-1} + \frac{2\alpha_t}{(1-\alpha_t)(1-\alpha_t^2)} g_t g_t^\top \tag{23}$$

Proof. The property in (21) can be easily verified by using the expressions for g_t and P_t. The property in (22) follows from the fact

$$P_t \prec (1-\alpha_{t-1}^2)P_{t-1} \prec \prod_{i=1}^{t-1}(1-\alpha_i^2)P_1$$

The property in (23) follows from the fact

$$\begin{aligned}
&\left(P_t - \frac{2\alpha_t(1-\alpha_t)}{1-\alpha_t^2}P_t g_t g_t^\top P_t\right)^{-1} \\
&= P_t^{-1/2}\left(I + \frac{2\alpha_t}{1-\alpha_t}P_t^{1/2}g_t g_t^\top P_t^{1/2}\right)P_t^{-1/2} \\
&= P_t^{-1} + \frac{2\alpha_t}{1-\alpha_t}g_t g_t^\top
\end{aligned}$$

□

Lemma 4. *We have the following properties for* α_t

$$\frac{a\gamma}{\sqrt{\lambda_{\max}(P_1)}}\prod_{i=1}^{t-1}(1-\alpha_i^2)^{-1/2} \le \alpha_t \le 1 \tag{24}$$

Proof. Since there exists an classifier u that classifies any labeled example with margin γ, we will have $u \in \mathcal{E}_t \cap \{z|\alpha_t - g_t^\top(z-w_t) \le 0\}$. Therefore, we have

$$\begin{aligned}
\alpha_t &\le g_t^\top(u-w_t) = (P_t^{1/2}g_t)^\top(P_t^{-1/2}(u-w_t)) \\
&\le |P_t^{1/2}g_t||P_t^{-1/2}(u-w_t)| = 1,
\end{aligned}$$

which proves the the upper bound for α_t. The lower bound follows from the fact

$$\alpha_{t+1} = \frac{a\gamma - y_t w_t^\top x_t}{\sqrt{x_t^\top P_t x_t}} \ge \frac{a\gamma}{\sqrt{x_t^\top P_t x_t}},$$

the property of P_t in (22) and the fact $\max\limits_{|x|_2 \le 1} x^\top P_1 x \le \lambda_{\max}(P_1)$. □

Using the results in the above lemmas, we now show how to prove the mistake bound stated in Theorem 3. After receiving T misclassified examples, the volume of the ellipsoid, denoted by $vol(\mathcal{E}_T)$ is reduced to

$$\begin{aligned}
&\mathrm{vol}(\mathcal{E}_{T-1}) = |P_1|^{1/2}\prod_{t=1}^{T-1}(1-\alpha_t^2)^{(d-1)/2}(1-\alpha_t) \\
&\ge |P_1|^{1/2}\left(1 - \frac{a\gamma}{\sqrt{\lambda_{\max}(P_1)}}\right)^{d(T-1)/2} \\
&= \left((1+(1-a)\gamma)\left(1 - \frac{a\gamma}{\sqrt{1+(1-a)\gamma}}\right)^{T-1}\right)^{d/2}
\end{aligned}$$

Since $\mathcal{E}_T \supseteq \mathcal{B}$, we have

$$\left((1+(1-a)\gamma)\left(1 - \frac{a\gamma}{\sqrt{1+(1-a)\gamma}}\right)^{T-1}\right)^{d/2} \ge (1-a)^d\gamma^d$$

Thus, T is upper bounded by

$$T \leq \frac{2\log(1-a) + 2\log\gamma - \log(1+(1-a)\gamma)}{\log\left(1 - \frac{a\gamma}{\sqrt{1+(1-a)\gamma}}\right)} + 1$$

Appendix C: Proof of Theorem 4

We first simplified the inequality in Lemma 2 of the submitted draft

$$\begin{aligned} q_{t+1} &\leq (1-c_t)q_t + \gamma_t^2 + c_t(u^\top g_t)^2 - 2\gamma_t u^\top g_t \\ &\leq (1-c_t)q_t + \gamma_t^2 + c_t(u^\top g_t)^2 - 2\frac{\gamma(\gamma - l_t(u))}{x_t^\top P_t x_t} \\ &= (1-c_t)q_t - \frac{\gamma^2 - c_t|u^\top x_t|^2}{x_t^\top P_t x_t} + 2\frac{\gamma l_t(u)}{x_t^\top P_t x_t} \\ &\leq (1-c_t)q_t - \frac{\gamma^2(1-c_t)}{x_t^\top P_t x_t} + 2\frac{\gamma l_t(u)}{x_t^\top P_t x_t} \\ &\leq (1-c_t)q_t - \gamma^2\prod_{i=1}^{t}(1-c_i) + 2\gamma l_t(u) \end{aligned}$$

The last step in the above derivation follows from the fact $P_t \succeq P_1$ and

$$x_t^\top P_t x_t \leq x_t^\top P_1 x_t \prod_{i=1}^{t-1}\frac{1}{1-c_i} \leq \prod_{i=1}^{t-1}\frac{1}{1-c_i}$$

We then put inequalities of all iterations together as

$$\begin{aligned} q_{t+1} \leq \textstyle\prod_{i=1}^{t}(1-c_i) - t\gamma^2\prod_{i=1}^{t}(1-c_i) \\ +2\gamma\textstyle\sum_{i=1}^{t} l_i(u)\prod_{j=i+1}^{t}(1-c_j) \end{aligned}$$

$$\begin{aligned} t\gamma^2 &\leq 1 + 2\gamma\sum_{i=1}^{t}\frac{l_i(u)}{\prod_{j=1}^{i}(1-c_j)} \\ &\leq 1 + 2\gamma\frac{1-b}{1-b-c}\sum_{i=1}^{t} l_i(u) \end{aligned}$$

We have the number of misclassified examples after training with T examples, denoted by M, is upper bounded

$$M \leq \frac{1}{\gamma^2} + \frac{2}{\gamma}\frac{1-b}{1-b-c}\sum_{i=1}^{T} l_i(u)$$

References

Amit, Y., Shalev-Shwartz, S., & Singer, Y. (2007). Online classification for complex problems using simultaneous projections. *In Advances in Neural Information Processing Systems* (pp. 17–24).

Crammer, K. (2004). *Online learning of complex categorial problems*. Doctoral dissertation, Hebrew Univeristy of Jerusalem.

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., & Singer, Y. (2006). Online passive-aggressive algorithms. *J. Mach. Learn. Res.*, *7*, 551–585.

Crammer, K., & Singer, Y. (2002). On the algorithmic implementation of multiclass kernel-based vector machines. *J. Mach. Learn. Res.*, *2*, 265–292.

Crammer, K., & Singer, Y. (2003). Ultraconservative online algorithms for multiclass problems. *J. Mach. Learn. Res.*, *3*, 951–991.

Fink, M., Shalev-Shwartz, S., Singer, Y., & Ullman, S. (2006). Online multiclass learning by interclass hypothesis sharing. *Proc. Int. Conf. on Mach. Learn.* (pp. 313–320).

Gentile, C. (2002). A new approximate maximal margin classification algorithm. *J. Mach. Learn. Res.*, *2*, 213–242.

Gentile, C., & M.Warmuth (1998). Linear hinge loss and average margin. *In Advances in Neural Information Processing Systems* (pp. 225–231).

Grove, A. J., Littlestone, N., & Schuurmans, D. (2001). General convergence results for linear discriminant updates. *Mach. Learn.*, *43*, 173–210.

Herbster, M., Pontil, M., & Wainer, L. (2005). Online learning over graphs. *Proc. Int. Conf. on Mach. Learn.* (pp. 305–312).

Kivinen, J., & M.K.Warmuth (1997). Additive versus exponentiated gradient updates for linear prediction. *Information and Computation*, *132*, 1–64.

Kivinen, J., Smola, A. J., & C.Williamson, R. (2002). Online learning with kernels. *IEEE Transactions on Signal Processing*, *52*, 2165–2176.

Li, Y., & Long, P. M. (2002). The relaxed online maximum margin algorithm. *Mach. Learn.*, *46*, 361–387.

Littlestone, N. (1989). *Mistake bounds and logarithmic linear-threshold learning algorithms*. Doctoral dissertation, U. C. Santa Cruz.

Rosenblatt, F. (1958). The perceptron : A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*, 386–407.

Shalev-Shwartz, S., & Singer, Y. (2006). Online learning meets optimization in the dual. *Proc. of the 19th Annual Conf. on Learning Theory* (pp. 423–437).

Shor, N. (1977). Cut-off method with space extension in convex programming problems. *Cybernetics*, *12*, 94–94.