Efficient Bandit Algorithms for Online Multiclass Prediction

Sham Kakade, Shai Shalev-Shwartz and Ambuj Tewari

Presented By: 
Nakul Verma
In many learning applications, true class labels are not fully disclosed.

Consider the setting:
- user queries a system
- system makes a recommendation $r$
- user responds (either positively or negatively) to $r$

Note: the system does not have access to how the user would have responded if some other recommendation was made.

This naturally leads to an online multiclass setting with limited feedback.

Is there an efficient learner (with guarantees) in this setting? (we will only focus on linear classification)
Talk outline

- Review of the classic Perceptron algorithm
- Multiclass generalization of the Perceptron
- Introduce the Banditron algorithm
- Theoretical analysis and experimental results
Perceptron: a review

- Online algorithm for binary linear classification

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Perceptron</th>
</tr>
</thead>
<tbody>
<tr>
<td>set $w^1 := 0$</td>
<td></td>
</tr>
<tr>
<td>for $t = 1, 2, \ldots$</td>
<td></td>
</tr>
<tr>
<td>receive example $x_t$</td>
<td></td>
</tr>
<tr>
<td>predict label $\hat{y}_t := \text{sign}(w^t \cdot x_t)$</td>
<td></td>
</tr>
<tr>
<td>nature reveals the label $y_t$</td>
<td></td>
</tr>
<tr>
<td>update weight $w^{t+1} := w^t + u^t$, where $u^t := x_t(1[y_t = 1] - 1[\hat{y}_t = 1])$</td>
<td></td>
</tr>
</tbody>
</table>

If data is linearly separable, then the number of mistakes made by Perceptron is bounded
Perceptron: quick example

- given a current weight vector $w^t$
Perceptron: quick example

- receive a new example $x_t$ such that $w^t$ makes a mistake
Perceptron: quick example

- update weight vector to $w^{t+1} := w^t + x_t$
• updated vector $w^{t+1}$ orients the hyperplane to get the example $x_t$ correct (as much as possible)
Recall: Perceptron is an online algorithm for **binary** linear classification

How can we generalize the Perceptron to **multi-class** classification?
For a $k$-class problem, we can use $k$ different weight vectors, and predict the class with largest correct margin.

---

**Algorithm** $k$-class Perceptron

- set $W^1 := 0$ ($W^t = [w^t_1, \ldots, w^t_k]^T$)
- for $t = 1, 2, \ldots$
  - receive example $x_t$
  - predict label $\hat{y}_t := \arg\max_j (W^t x_t)_j$
  - nature reveals the label $y_t$
  - update weight $W^{t+1} := W^t + U^t$
    - where $U^t_{r,j} := x_{t,j} (1[y_t = r] - 1[\hat{y}_t = r])$
In comparison with the binary case, note that the update rule for multi-class Perceptron is

\[ U_{r,j}^t := x_{t,j} (\mathbb{1}[y_t = r] - \mathbb{1}[\hat{y}_t = r]) \]

In other words, upon mistake:
- **add** \( x_t \) to \( w_i^t \) corresponding to correct label
- **subtract** \( x_t \) from \( w_i^t \) corresponding to incorrect predictor
Guarantees for multiclass Perceptron

- Define the quantities (assume $\|x_t\| \leq 1$):

  mistakes
  \[ M = \sum_{t=1}^{T} 1[\hat{y}_t \neq y_t] \]

  hinge loss
  \[ L(W) = \sum_{t=1}^{T} \max_{r \in [k] \setminus \{y_t\}} [1 - (W x_t)_y + (W x_t)_r] \]

  complexity
  \[ D(W) = 2 \|W\|_F^2 = 2 \sum_{i,j} (W_{i,j})^2 \]

For any $W$, we have (Fink et al., 2006)

\[ M \leq L(W) + D(W) + \sqrt{L(W)D(W)} \]
Challenges in this setting:

- Cannot use Perceptron update (don't know $y_t$)
- Cannot directly use bandit algorithms for online convex optimization (eg. Flaxman et al., 2005) since the only feedback we get $\mathbb{1}[y_t = \hat{y}_t]$
Banditron algorithm

Algorithm: The Banditron

Parameters: \( \gamma \in (0, 0.5) \)

Initialize \( W^1 = 0 \in \mathbb{R}^{k \times d} \)

for \( t = 1, 2, \ldots, T \) do

Receive \( x_t \in \mathbb{R}^d \)

Set \( \hat{y}_t = \arg \max_{r \in [k]} (W^t x_t)_r \)

\( \forall r \in [k] \) define \( P(r) = (1 - \gamma) 1[r = \hat{y}_t] + \frac{\gamma}{k} \)

Randomly sample \( \tilde{y}_t \) according to \( P \)

Predict \( \tilde{y}_t \) and receive feedback \( 1[\tilde{y}_t = y_t] \)

Define \( \tilde{U}^t \in \mathbb{R}^{k \times d} \) such that:

\[
\tilde{U}^t_{r,j} = x_{t,j} \left( \frac{1[y_t = \tilde{y}_t] 1[\tilde{y}_t = r]}{P(r)} - 1[\tilde{y}_t = r] \right)
\]

Update: \( W^{t+1} = W^t + \tilde{U}^t \)

end for
In comparison to the full information case, note that the update rule for Banditron is

\[
\tilde{U}_{r,j}^t = x_{t,j} \left( \frac{1[y_t = \hat{y}_t]1[\hat{y}_t = r]}{P(r)} - 1[\hat{y}_t = r] \right)
\]

Two cases:

- if \( \hat{y}_t = y_t \) (full information)
  - if \( y_t = \hat{y}_t \) (correct prediction) => do tiny update
  - if \( y_t \neq \hat{y}_t \) (incorrect prediction) => do large update

- if \( \hat{y}_t \neq y_t \) (partial information)
  (incorrect prediction) => do large update
Theoretical guarantees

For any $W$, the number of mistakes $M$ made by Banditron satisfies:

$$
\mathbb{E}[M] \leq L + \gamma T + 3 \max \left\{ \frac{kD}{\gamma}, \sqrt{D \gamma T} \right\} + \sqrt{\frac{kD L}{\gamma}}
$$

- expectation is over the randomness of the algorithm
- $L := L(W), D := D(W)$

Consequence:
By setting $\gamma = \sqrt{\frac{kD}{T}}$
we have expected mistake bound: $O(\sqrt{kD/T})$
Two key observations:

1. $\mathbb{E}_t[\tilde{U}^t] = U^t$

2. $\mathbb{E}_t[\|\tilde{U}^t\|_F^2] \leq 2 \|x_t\|^2 \left( \frac{k}{\gamma} 1[y_t \neq \hat{y}_t] + \gamma 1[y_t = \hat{y}_t] \right)$

Notation, for any $W^*$:

$$\langle W^*, W^t \rangle := \sum_{i,j} W^*_{i,j} W^t_{i,j}$$

Key quantity to analyze:

$\mathbb{E}[\langle W^*, W^{T+1} \rangle]$
Proof sketch (cont.)

**Lower bound:**
\[
\mathbb{E}[\langle W^*, W^{T+1} \rangle] = \sum_{t=1}^{T} \mathbb{E}[\langle W^*, W^{t+1} \rangle] - \mathbb{E}[\langle W^*, W^{t} \rangle]
\]
(Def. of \( W^{t} \) and Obs. 1)
\[
= \sum_{t=1}^{T} \mathbb{E}[\langle W^*, \tilde{U}^{t} \rangle] = \sum_{t=1}^{T} \mathbb{E}[\langle W^*, U^{t} \rangle]
\]
(Def. of hinge loss \( L \))
\[
\geq \sum_{t=1}^{T} \mathbb{E}[\mathbf{1}[\hat{y}_t \neq y_t]] - L
\]

**Upper bound:**
\[
\mathbb{E}[\langle W^*, W^{T+1} \rangle] \leq \mathbb{E}[\|W^*\|_F \|W^{T+1}\|_F]
\]
(Def. of \( W^{T+1} \) and term \( D \))
\[
\leq \sqrt{\frac{D}{2}} \mathbb{E}[\|W^{T+1}\|_F^2] \leq \sqrt{\frac{D}{2}} \sum_{t=1}^{T} \mathbb{E}[\|U^{t}\|_F^2]
\]
(by Obs. 2)
\[
\leq \sqrt{\frac{Dk}{\gamma}} \sum_{t} \mathbb{E}[\mathbf{1}[\hat{y}_t \neq y_t]] + D\gamma T
\]
Combining the upper and lower bounds yields:

$$\sum_{t=1}^{T} \mathbb{E}[\mathbf{1}[\hat{y}_t \neq y_t]] \leq L + \sqrt{\frac{D k L}{\gamma}} + 3 \max \left\{ \frac{D k}{\gamma}, \sqrt{D \gamma T} \right\}$$

Finally noting that in expectation we explore no more than $\gamma T$ rounds, we have

$$\mathbb{E}[M] \leq \sum_{t=1}^{T} \mathbb{E}[\mathbf{1}[\hat{y}_t \neq y_t]] + \gamma T$$
Compare performance of $k$-class Perceptron with Banditron on two datasets:

- **Synthetic dataset**: 9-class, 400-dim dataset that is linearly separable. (each datapoint is sparse to simulate text data)

- **Real dataset**: subset of Reuters RCV1 collection. 4-class, 350k-dim (bag-of-words model).
Experimental results (synthetic data)

- $k$-Perceptron (full info) does **better** than Banditron (limited info)
- error rate of $k$-Perceptron: $1 / T$
- error rate of Banditron: $1 / T^{0.5}$
Experimental results (real data)

- error rates of $k$-Perceptron and Banditron are comparable
References

- S. Kakade, S. Shalev-Shwartz and A. Tewari. Efficient bandit algorithms for online multiclass prediction. ICML 2008.

- M. Fink, S. Shalev-Shwartz, Y. Singer and S. Ullman. Online multiclass learning by interclass hypothesis sharing. ICML 2006.