Multiclass Classification

Yufei Tao

Department of Computer Science and Engineering
Chinese University of Hong Kong

Classification – Redefined

Let $A_1, ..., A_d$ be the **attributes** of a d-dimensional universe U, i.e.:

$$U = dom(A_1) \times dom(A_2) \times ... \times dom(A_d)$$

where $dom(A_i)$ represents the set of possible values on A_i.

Each **object** is an element e of $U \times \{1, 2, ..., k\}$ — for some integer k — i.e., it takes value $e[A_i]$ on every attribute A_i ($1 \leq i \leq d$), and a **class label** $e[C]$ that is an integer between 1 and k.

Denote by D a probabilistic distribution on $U \times \{1, ..., k\}$.

Classification – Redefined

Goal: Given an object e drawn from D, we want to predict its label $e[C]$ from its attribute values $e[A_1], ..., e[A_d]$.

We do so by constructing a function

$$M : dom(A_1) \times dom(A_2) \times ... \times dom(A_d) \to \{1, 2, ..., k\}$$

which we refer to as a **classifier**. Given any object e, we predict its class label as $M(e[A_1], ..., e[A_d])$.

We define the error of M on D—denoted as $err_D(M)$—as:

$$err_D(M) \quad = \quad \boldsymbol{Pr}_{e \sim D}[M(e[A_1], ..., e[A_d]) \neq e[C]].$$

Classification – Redefined

In training, we are given a sample set R of D, where each object in R is drawn independently according to D. We refer to R as the **training set**.

We would like to learn our classifier M from R.

> The key difference from what we have discussed before is that the number k of classes can be anything (in binary classifications, $k = 2$). We will refer to this version of classification as **multiclass classification**.

Think: How would you adapt the decision tree method and Bayes' method to multiclass classification?

Next we will consider that every $dom(A_i)$ $(1 \leq i \leq d)$ is the real domain $\mathbb{R}$, and discuss how linear classifiers and Perceptron can be extended to multiclass classification.

Accordingly, P is a set of points in $\mathbb{R}^d$. We consider that these points do not have the same label (otherwise, the classification task is trivial).

Linear Classification – Generalized

We will consider a class of classifiers which we will refer to as **generalized linear classifiers**.

Every such classifier is specified by k d-dimensional vectors $\vec{w}_1, \vec{w}_2, ..., \vec{w}_k$.

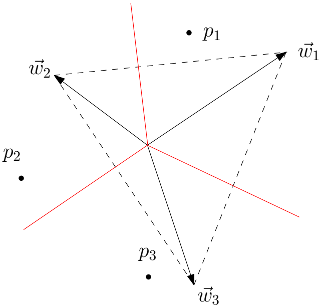
Given a point p, the classifier predicts its class label as

$$\operatorname*{argmax}_{i \in [1,k]} \vec{w}_i \cdot \vec{p}.$$

Namely, it returns the label $i \in [1, k]$ that gives the largest $\vec{w}_i \cdot \vec{p}$.

Tie breaking: In the special case where the maximum is achieved by two distinct $i, j \in [1, d]$ (i.e., $\vec{w}_i \cdot \vec{p} = \vec{w}_j \cdot \vec{p}$), we can break the tie using some consistent policy, e.g., predicting the label as the smaller between i and j.

Example



Points p_1, p_2, and p_3 will be classified as label 1, 2, and 3, respectively.

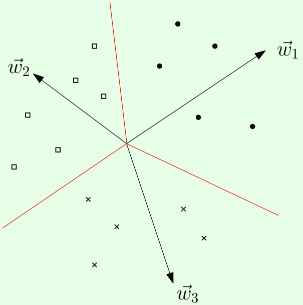
Think: What do the three red rays stand for?

Given a training set P, we say that it is **linearly separable** if there exist $\vec{w}_1, ..., \vec{w}_d$ that

- correctly classify all the points in P;
- for every point $p \in P$ with label ℓ, $\vec{w}_\ell \cdot \vec{p} > \vec{w}_z \cdot \vec{p}$ for every $z \neq \ell$.

The set of these weight vectors is said to **separate** P.

Example:



The dots have label 1, squares label 2, and crosses label 3.

Next we will discuss an algorithm that extends the Perceptron algorithm to find a set of weight vectors to separate P, **provided that** P is linearly separable. We will refer to the algorithm as **multiclass Perceptron**.

Multiclass Perceptron

```
$\vec{w}_i \leftarrow \vec{0}$ for all $i \in [1, k]$
**while** there is a **violation point** $p \in P$
/* namely, $p$ mis-classified by $\{\vec{w}_1, ..., \vec{w}_k\}$ */
    $\ell \rightarrow$ the **real label** of $p$
    $z \rightarrow$ the **predicted label** of $p$
    /* $\ell \neq z$ since $p$ is a violation point */
    $\vec{w}_\ell \leftarrow \vec{w}_\ell + \vec{p}$
    $\vec{w}_z \leftarrow \vec{w}_z - \vec{p}$
```

When $k = 2$, the above algorithm degenerates into (the conventional) Perceptron. Can you see why?

“Margin”

Let W be a set of weight vectors $\{\vec{w}_1, ..., \vec{w}_k\}$ that separates P.

Given a point $p \in P$ with label ℓ, let us define its **margin under** W as

$$margin(p \mid W) = \min_{z \neq \ell} \frac{\vec{w}_\ell \cdot \vec{p} - \vec{w}_z \cdot \vec{p}}{\sqrt{2 \sum_{i=1}^{k} |\vec{w}_i|^2}}.$$

> The margin of p under W is a way to measure how “confidently” W gives p the class label ℓ. **Think:** why?

The **margin** of W equals the **smallest** margin of all points under W:

$$margin(W) = \min_{p \in P} margin(p \mid W).$$

“Margin”

Let W^* be a set of weight vectors that (i) separates P, and (ii) has the largest margin.

Define

$$\gamma = margin(W^*).$$

As before, define the **radius** of P as

$$R = \max_{p \in P} |p|.$$

Theorem: Multiclass Perceptron stops after processing at most R^2/γ^2 violation points.

This is the general version of the theorem we have already learned on (the old) Perceptron.

Before proving the theorem, let us first familiarize ourselves with some definitions about matrices that will be useful.

Let M be a $d \times k$ matrix. We use $M[i,j]$ to denote the element at the i-th row and j-th column $(1 \le i \le d, 1 \le j \le k)$.

The **Frobenius norm** of M, denoted as $|M|_F$, is:

$$|M|_F = \sqrt{\sum_{i,j} M[i,j]^2}.$$

Here is an easy way to appreciate the above norm: think of M as a (dk)-dimensional vector by concatenating all its rows; then $|M|_F$ is simply the length of that vector.

Given two $d \times k$ matrices M_1, M_2, the (matrix) **dot product** operation gives a new $d \times k$ matrix M where

$$M[i,j] = M_1[i,j] \cdot M_2[i,j].$$

Proof of the theorem on Slide 13: The algorithm maintains a set of vectors $\{\vec{w}_1, ..., \vec{w}_k\}$. Each $\vec{w}_i$ $(1 \le i \le k)$ is a $d \times 1$ vector.

Henceforth, we will regard a set of vectors $\{\vec{w}_1, ..., \vec{w}_k\}$ as a $d \times k$ matrix W, where the i-th ($i \in [1, k]$) row of W is the **transpose** of $\vec{w}_i$ (i.e., a $1 \times d$ vector).

Define t as the number of violation points.

The algorithm performs t adjustments to W. Denote by W_j ($j \in [1, t]$) as the W after the j-th adjustment. Define specially W_0 the $d \times k$ matrix with all 0's.

Denote by W^* the $d \times k$ matrix that corresponds to an optimal set of weight vectors $\{w_1^*, ..., w_d^*\}$ whose margin is γ.

Claim 1: $W^* \cdot W_t \geq \sqrt{2} t\gamma \cdot |W^*|_F$.

Proof: Consider any $j \in [1, t]$. Let p be the violation point that caused the j-th adjustment. Let ℓ be the real label of p, and z the label predicted by W_{j-1}.

Define Δ the $d \times k$ matrix such that

- The ℓ-th row of Δ is the transpose of $\vec{p}$.
- The z-th row of Δ is the transpose of $(-1) \cdot \vec{p}$.
- All the other rows are 0.

Hence, $W_j = W_{j-1} + \Delta$, which means:

$$W^* \cdot W_j = W^* \cdot W_{j-1} + W^* \cdot \Delta.$$

We will prove $W^* \cdot \Delta \geq \sqrt{2}\gamma \cdot |W^*|_F$, which will complete the proof of Claim 1.

$$\begin{aligned} W^* \cdot \Delta &= \vec{w_\ell^*} \cdot \vec{p} - \vec{w_z^*} \cdot \vec{p} \\ &\geq \gamma \sqrt{2 \sum_{i=1}^{k} |w_i^*|^2} \\ &= \gamma \sqrt{2 |W^*|_F^2} \\ &= \sqrt{2}\gamma \cdot |W^*|_F. \end{aligned}$$

□

Claim 2: $|W_t|_F^2 \leq 2tR^2$.

Proof: Consider any $j \in [1, t]$. Let p be the violation point that caused the j-th adjustment. Let ℓ be the real label of p, and z the label predicted by W_{j-1}. Suppose that $W_{j-1} = \{\vec{u}_1, ..., \vec{u}_k\}$.

Since p is a violation point, we must have:

$$\vec{u}_\ell \cdot \vec{p} \leq \vec{u}_z \cdot \vec{p}$$

Denote by $\vec{v}_\ell$ the new vector for class label ℓ after the update, and similarly by $\vec{v}_z$ the new vector for class label z after the update. By how the algorithm runs, we have:

$$\begin{aligned} \vec{v}_\ell &= \vec{u}_\ell + \vec{p} \\ \vec{v}_z &= \vec{u}_z - \vec{p} \end{aligned}$$

We have

$$\begin{aligned} |\vec{v}_\ell|^2 + |\vec{v}_z|^2 &= (\vec{u}_\ell + \vec{p})^2 + (\vec{u}_z - \vec{p})^2 \\ &= |\vec{u}_\ell|^2 + |\vec{u}_z|^2 + 2|\vec{p}|^2 + 2(\vec{u}_\ell \cdot \vec{p} - \vec{u}_z \cdot \vec{p}) \\ \text{(as } p \text{ is a violation point)} &\leq |\vec{u}_\ell|^2 + |\vec{u}_z|^2 + 2|\vec{p}|^2 \\ &\leq |\vec{u}_\ell|^2 + |\vec{u}_z|^2 + 2R^2. \end{aligned}$$

Observe that

$$|W_j|_F^2 - |W_{j-1}|_F^2 = (|\vec{v}_\ell|^2 + |\vec{v}_z|^2) - (|\vec{u}_\ell|^2 + |\vec{u}_z|^2)$$

We therefore have

$$|W_j|_F^2 - |W_{j-1}|_F^2 \leq 2R^2.$$

This completes the proof of the claim. □

Claim 3: $W^* \cdot W_t \leq |W^*|_F \cdot |W_t|_F$.

Proof: The claim follows immediately from the following general result:

Let $\vec{u}$ and $\vec{v}$ be two vectors of the same dimensionality; it always holds that $\vec{u} \cdot \vec{v} \leq |\vec{u}||\vec{v}|$.

The above is true because $\vec{u} \cdot \vec{v} = |\vec{u}||\vec{v}| \cos\theta$ where θ is the angle between the two vectors. □

By combining Claims 1-3, we have:

$$\begin{aligned}\sqrt{2}t\gamma|W^*|_F &\leq |W^*|_F \cdot |W_t|_F \leq |W^*|_F \cdot \sqrt{2t}R \\ \Rightarrow t &\leq R^2/\gamma^2.\end{aligned}$$

This completes the proof of the theorem.