Multiclass, Ranking, and Complex Prediction Problems

Peng Lingwei

August 15, 2019

Contents

17 Multiclass, Ranking and Complex prediction Problems

17.1 ONE-VERSUS-ALL AND ALL-PAIRS

$\mathcal{Y} = \{1, 2, 3, \ldots, k\}$

Algorithm 1 One-Versus-All

Require:
training set $S = ((\vec{x}_1, y_1), (\vec{x}_2, y_2), \ldots, (\vec{x}_m, y_m))$, algorithm for binary classification A
for $i \in \mathcal{Y}$ **do**
$S_i = ((\vec{x}_1, (-1)^{1_{y_1 \neq i}}), \ldots, (\vec{x}_m, (-1)^{1_{y_m \neq i}}))$
$h_i = A(S_i)$
end for.
return $h(\vec{x}) \in \arg\max_{i \in \mathcal{Y}} h_i(\vec{x})$

Algorithm 2 All-Pairs

Require:
training set $S = ((\vec{x}_1, y_1), (\vec{x}_2, y_2), \ldots, (\vec{x}_m, y_m))$, algorithm for binary classification A
for $i, j \in \mathcal{Y}$ s.t. $i < j$ **do**
$S_{i,j} = ()$
for $t = 1, \ldots, m$ **do**
if $y_t = i$ or $y_t = j$ **then** $S_{i,j} = S_{i,j} \cup (\vec{x}_t, (-1)^{1_{y_t \neq i}})$
end if.
end for. $h_{i,j} = A(S_{i,j})$.
end for.
return $h(\vec{x}) \in \arg\max_{i \in \mathcal{Y}} \left(\sum_{j \in \mathcal{Y}} sign(j - i) h_{i,j}(\vec{x}) \right)$

The binary learner might lead to **suboptimal results**.

17.2 LINEAR MULTICLASS PERDICTORS

1. Let $\Psi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ be a class-sensitive feature mapping.We can think of the elements of $\Psi(\vec{x}, y)$ as score functions that access how well the label y fits the instance $\vec{x}$.
2. $h(\vec{x}) = \arg\max_{y \in \mathcal{Y}} \langle \vec{w}, \Psi(\vec{x}, y) \rangle$.
3. $W = \{\vec{w} \in \mathbb{R}^d : \|\vec{w}\| \leq B\}$
4. $\mathcal{H}_{\Psi,W} = \{\vec{x} \mapsto \arg\max_{y \in \mathcal{Y}} \langle \vec{w}, \Psi(\vec{x}, y) \rangle : \vec{w} \in W\}$

17.2.1 How to Construct Ψ

1. The multivector construction: Let $\mathcal{Y} = \{1, \ldots, k\}$ and let $\mathcal{X} = \mathbb{R}^n$. Then $\Psi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$, where $d = nk$,

$$\Psi(\vec{x}, y) = [\underbrace{0, \ldots, 0}_{\in \mathbb{R}^{(y-1)n}}, \underbrace{x_1, \ldots, x_n}_{\in \mathbb{R}^n}, \underbrace{0, \ldots, 0}_{\in \mathbb{R}^{(k-y)n}}]$$

   This is sometimes equal to one-versus-all with hyperplane binary classification algorithms.

2. TF-IDF:

   - Term-frequency: $TF(j, \vec{x})$ is the number of times the word corresponding index is j appears in the document $\vec{x}$.
   - Document-frequency: $DF(j, y)$ is the number of times the word corresponding index is j appears in the documents that are not on topic y.

$$\Psi_j(\vec{x}, y) = TF(j, \vec{x}) \log\left(\frac{m}{DF(j, y)}\right)$$

17.2.2 Cost-Sensitive Classification

1. Loss function, $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$;
2. Zero-one loss, $\Delta^{0-1} = 1_{[y' \neq y]}$;
3. We also can penalize different levels of loss for different mistakes, such as: mistaking tiger for cat and mistaking tiger for durk, which is called generalized zero-one loss, or zero-nozero loss.

17.2.3 ERM

Empirical risk with respect to Δ: $L_S(h) = \frac{1}{m} \sum_{i=1}^{m} \Delta(h(\vec{x}_i), y_i)$.

In realizable case, we can use multiclass batch perception to get a good hyperplane.

Algorithm 3 Multiclass Batch Perception

Require: A training set $S = ((\vec{x}_1, y_1), \ldots, (\vec{x}_m, y_m))$; A class-sensitive feature mapping $\Psi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$.
Ensure: $\vec{w}^{(1)} = (0, \ldots, 0) \in \mathbb{R}^d$
  for $t = 1, 2, \ldots$ **do**
    if $\exists i$ and $y \neq y_i$ s.t. $\langle \vec{w}^{(t)}, \Psi(\vec{x}_i, y_i) \rangle \leq \langle \vec{w}^{(t)}, \Psi(\vec{x}_i, y) \rangle$ **then**
      $\vec{w}^{(t+1)} = \vec{w}^{(t)} + \Psi(\vec{x}_i, y_i) - \Psi(\vec{x}_i, y)$
    else
      break.
    end if.
  end for.

17.2.4 Generalized Hinge Loss

Definition 1. ***(Generalized hinge loss).***

$$l^{g-hinge}(\vec{w},(\vec{x},y)) = \max_{y'\in\mathcal{Y}}(\Delta(y',y) + \langle \vec{w}, \Psi(\vec{x},y') - \Psi(\vec{x},y)\rangle)$$

The definition of $h_{\vec{w}}(\vec{x})$ implies that $\forall y \in \mathcal{Y}, \langle \vec{w}, \Psi(\vec{x},y)\rangle \leq \langle \vec{w}, \Psi(\vec{x}, h_{\vec{w}}(\vec{x}))\rangle$. Therefore,

$$\Delta(h_{\vec{w}}(\vec{x}),y) \leq \Delta(h_{\vec{w}}(\vec{x}),y) + \langle \vec{w}, \Psi(\vec{x}, h_{\vec{w}}(\vec{x})) - \Psi(\vec{x},y)\rangle$$

Because $h_{\vec{w}}(\vec{x}) \in \mathcal{Y}$, we can easily get $\Delta(h_{\vec{w}}(\vec{x}),y) \leq l(\vec{w},(\vec{x},y))$.

If $\mathcal{Y} \in \{\pm 1\}$ and $\Psi(\vec{x},y) = \frac{y\vec{x}}{2}$, then the generalized hinge loss becomes the vanilla hinge loss for binary classification,

$$l^{hinge}(\vec{w},(\vec{x},y)) = \max\{0, 1 - y\langle \vec{w}, \vec{x}\rangle\}$$

In generalized hinge-loss, the correct condition ($l(\vec{w},(\vec{x},y)) = 0$) is: If $y' = y$ then $l(\vec{w},(\vec{x},y)) = 0$, else if $y' \neq y$, we require $\langle \vec{w}, \Psi(\vec{x},y)\rangle \geq \langle \vec{w}, \Psi(\vec{x},y')\rangle + \Delta(y',y)$.

In preceeding vanilla hinge loss, we require that if $y \neq y'$,$\langle \vec{w}, y\vec{x}\rangle \geq 1$.

$l(\vec{w},(\vec{x},y))$ is $\rho - Lipschitz$ with $\rho = \sup_{\vec{x}\in\mathcal{X}} \max_{y'\in\mathcal{Y}} \|\Psi(\vec{x},y') - \Psi(\vec{x},y)\|$.

17.2.5 Multiclass SVM and SGD

Algorithm 4 Multiclass SVM

Require: S, λ, Δ, Ψ

$\min_{\vec{w}\in\mathbb{R}^d} \left(\lambda\|\vec{w}\|^2 + \frac{1}{m}\sum_{i=1}^m \max_{y'\in\mathcal{Y}} \left(\Delta(y',y_i) + \langle \vec{w}, \Psi(\vec{x_i},y') - \Psi(\vec{x}_i,y_i)\rangle\right)\right)$

return $h_{\vec{w}}(\vec{x}) = \arg\max_{y\in\mathcal{Y}}\langle \vec{w}, \Psi(\vec{x},y)\rangle$

Corollary 1. *If $\|\Psi(\vec{x},y)\| \leq \rho/2$, $\|\vec{w}\| \leq B$, and the regularization $\lambda = \sqrt{\frac{2\rho^2}{B^2 m}}$, then we have:*

$$\mathbb{E}_{S\sim\mathcal{D}^m}\left[L^{\Delta}_{\mathcal{D}}(h_{\vec{w}})\right] \leq \mathbb{E}_{S\sim\mathcal{D}^m}\left[L^{g-hinge}\mathcal{D}(h_{\vec{w}})\right] \leq \min_{\vec{w}:\|\vec{w}\|\leq B} L^{g-hinge}_{\mathcal{D}}(\vec{w}) + \sqrt{\frac{8\rho^2B^2}{m}}$$

To use SGD, we need the subgradient of $l^{g-hinge}(\vec{w},(\vec{x},y))$:

$$\partial_{\vec{w}} l^{g-hinge}(\vec{w},(\vec{x},y)) = \Psi(\vec{x},\hat{y}) - \Psi(\vec{x},y), where\ \hat{y} \in \arg\max_{y'\in\mathcal{Y}}(\Delta(y',y) + \langle \vec{w}^{(t)}, \Psi(\vec{x},y') - \Psi(\vec{x},y)\rangle)$$

Corollary 2. *If $\|\Psi(\vec{x},y)\| \leq \rho/2$, $\|\vec{w}\| \leq B$, then $T \geq \frac{B^2\rho^2}{\epsilon^2}$ guarantees*

$$\mathbb{E}_{S\sim\mathcal{D}^m}\left[L^{\Delta}_{\mathcal{D}}(h_{\vec{w}})\right] \leq \mathbb{E}_{S\sim\mathcal{D}^m}\left[L^{g-hinge}\mathcal{D}(h_{\vec{w}})\right] \leq \min_{\vec{w}:\|\vec{w}\|\leq B} L^{g-hinge}_{\mathcal{D}}(\vec{w}) + \epsilon$$

17.3 STRUCTURED OUTPUT PREDICTION

In multiclass problems, if $\mathcal{Y}$ is very large but endowed with a predefined structure, we can use structured output prediction.

In optical character recognition, the words in $\mathcal{Y}$ are of length r, the size of alphabet is q. We defined $\Delta(\vec{y}',\vec{y}) = \frac{1}{r}\sum_{i=1}^r 1_{[y_i\neq y'_i]}$.

We also assume sample space is $x \in \mathcal{X} \subset \mathbb{R}^{n\times r}$, where n is the number of pixels in each image, and r is the number of images in the sequence.

Then, we construct class-sensitive feature mapping $\Psi(\vec{x},\vec{y}) \in \mathbb{R}^{nq+q^2}$:

1. $\Psi_{i,j,1}(\vec{x},\vec{y}) = \frac{1}{r}\sum_{t=1}^{r} x_{i,t}1_{[y_t=j]}$, which shows the gray level value of certain letter;

2. $\Psi_{i,j,2}(\vec{x},\vec{y}) = \frac{1}{r}1_{[y_t=i]}1_{[y_{t-1}=j]}$, which shows the likelihood probability of "qa" (or "rz") in the words;

$$h_{\vec{w}}(\vec{x}) = \arg\max_{\vec{y}\in\mathcal{Y}}\langle\vec{w},\Psi(\vec{x},\vec{y})\rangle$$

We can rewrite $\Psi(\vec{x},\vec{y}) = \sum_{t=1}^{r}\phi(\vec{x},y_t,y_{t-1})$, where $\phi_{i,j,1}(\vec{x},y_t,y_{t-1}) = x_{i,t}1_{[y_t=j]}$ and $\phi_{i,j,2}(\vec{x},y_t,y_{t-1}) = 1_{[y_t=i]}1_{[y_{t-1}=j]}$. Then

$$h_{\vec{w}}(\vec{x}) = \arg\max_{\vec{y}\in\mathcal{Y}}\sum_{t=1}^{r}\langle\vec{w},\phi(\vec{x},y_t,y_{t-1})\rangle$$

by which we can use dynamic programming procedure to predict a new x. Such as: for y_{-1} is fixed, we can easily minimized y_0:

$$\arg\max_{y_0}\langle\vec{w},\phi(\vec{x},y_0,y_{-1})\rangle$$

17.4 RANKING

1. $x = (\vec{x}_1,\ldots,\vec{x}_r)\in\mathcal{X}$;

2. $\mathcal{Y}\in\mathbb{R}^r$, such as $\vec{y} = (2,1,6,-1,0.5)$ induces the permutaion $\pi(\vec{y}) = (4,3,5,1,2)$ (ranking);

3. $Z = \cup_{r=1}^{\infty}(\mathcal{X}^r\times\mathbb{R}^r)$;

4. $\mathcal{H}$ is some set of ranking hypothesis.

5. Loss functions $l(h,(\vec{x},\vec{y})) = \Delta(h(\vec{x}),\vec{y})$, where $\Delta:\cup_{r=1}^{\infty}(\mathbb{R}^r\times\mathbb{R}^r)\to\mathbb{R}_+$.

   - $0-1$ **Ranking Loss**: $\Delta(\vec{y}',\vec{y}) = 1_{[\pi(\vec{y}')\neq\pi(\vec{y})]}$, neverused;
   - **Kendall-Tau Loss**:

$$\Delta(\vec{y}',\vec{y}) = \frac{2}{r(r-1)}\sum_{i=1}^{r-1}\sum_{j=i+1}^{r}1_{[sign(y'_i-y'_j\neq sign(y_i-y_j))]}$$

   - **Normalized Discounted Dumulative Gain (NDCG)**: A monotonically nondecreasing discount function $D:\mathbb{N}\to\mathbb{R}_+$, a discounted cumulative gain measure:$G(\vec{y}',\vec{y}) = \sum_{i=1}^{r}D(\pi(\vec{y}')_i)y_i$.

$$\Delta(\vec{y}',\vec{y}) = 1-\frac{G(\vec{y}',\vec{y})}{G(\vec{y},\vec{y})} = \frac{1}{G(\vec{y},\vec{y})} = \frac{1}{G(\vec{y},\vec{y})}\sum_{i=1}^{r}(D(\pi(\vec{y})_i)-D(\pi(\vec{y}')_i))y_i$$

   A typical way to define the discount function is by

$$D(i) = \begin{cases}\frac{1}{\log_2(r-i+2)} & i\in\{r-k+1,\ldots,r\}\\ 0, & otherwise\end{cases}$$

17.4.1 Linear Predictors for Ranking

$\mathcal{H}_W = \{h_{\vec{w}} : (\vec{x}_1, \ldots, \vec{x}_r) \to (\langle \vec{w}, \vec{x}_1 \rangle, \ldots, \langle \vec{w}, \vec{x}_r \rangle); \vec{w} \in W\}, \quad W = \left\{w \in \mathbb{R}^d, \|\vec{w}\| \leq B\right\}$

1. A Hinge Loss for the Kendall Tau Loss Function:

$$\Delta(\vec{y}', \vec{y}) \leq \frac{2}{r(r-1)} \sum_{i=1}^{r-1} \sum_{j=i+1}^{r} \max\{0, 1 - sign(y_i - y_j)\langle \vec{w}, \vec{x}_i - \vec{x}_j \rangle\}$$

2. A Hinge Loss for the NDCG Loss Function: Let V be the set of all permutations of $[r]$ encoded as vectors, then $\pi(\vec{y}') = \arg\max_{\vec{v} \in V} \sum_{i=1}^{r} v_i y'_i$. Denote $\Psi(x, \vec{v}) = \sum_{i=1}^{r} v_i \vec{x}_i$, and

$$\pi(h_{\vec{w}}(x)) = \arg\max_{\vec{v} \in V} \sum_{i=1}^{r} v_i \langle \vec{w}, \vec{x}_i \rangle = \arg\max_{\vec{v} \in V} \langle \vec{w}, \sum_{i=1}^{r} v_i \vec{x}_i \rangle = \arg\max_{\vec{v} \in V} \langle \vec{w}, \Psi(x, \vec{v}) \rangle$$

$$\begin{aligned}
\Delta(h_{\vec{w}}(\vec{x}), \vec{y}) \leq & \Delta(h_{\vec{w}}(x), \vec{y}) + \langle \vec{w}, \Psi(x, \pi(h_{\vec{w}}(x))) \rangle - \langle \vec{w}, \Psi(x, \pi(\vec{y})) \rangle \\
\leq & \max_{\vec{v} \in V} \left[\Delta(\vec{v}, \vec{y}) + \langle \vec{w}, \Psi(x, \vec{v}) \rangle - \langle \vec{w}, \Psi(x, \pi(\vec{y})) \rangle\right] \\
= & \max_{\vec{v} \in V} \left[\Delta(\vec{v}, \vec{y}) + \sum_{i=1}^{r} (v_i - \pi(\vec{y})_i) \langle \vec{w}, \vec{x}_i \rangle\right]
\end{aligned}$$

To calculate the subgradient of the loss function, we need to find $\vec{v}$ that minimize the hinge loss, which is equal to solve

$$\arg\min_{\vec{v} \in V} \sum_{i=1}^{r} \left(-\langle \vec{w}, \vec{x}_i \rangle v_i + \frac{y_i D(v_i)}{G(\vec{y}, \vec{y})}\right) = \arg\min_{\vec{v} \in V} \sum_{i=1}^{r} (\alpha_i v_i + \beta_i D(v_i))$$

If we construct matrix $A \in \mathbb{R}^{r,r}$, and $A_{i,j} = j\alpha_i + D(j)\beta_i$, then we can think about each j as a "worker", each i as a "task", and $A_{i,j}$ as the cost of assigning task i to worker j. The discuss of doubly stochastic matrix for solving preceeding problem is interesting, please read the book.

17.5 BIPARTITE RANKING AND MULTIVARIATE PERFORMANCE MEASURES

Bipartite ranking problem: $\vec{y} \in \{\pm 1\}^r$.

The threshold transforms the vector $\vec{y}' \in \mathbb{R}^r$ in to the vector $(sign(y'_i - \theta), \ldots, sign(y'_r - \theta)) \in \{\pm 1\}^r$.

1.
   - True positives: $a = |\{i : y_i = +1 \wedge sign(y'_i - \theta) = +1\}|$
   - False positives: $b = |\{i : y_i = -1 \wedge sign(y'_i - \theta) = +1\}|$
   - False negatives: $c = |\{i : y_i = +1 \wedge sign(y'_i - \theta) = -1\}|$
   - True negatives: $d = |\{i : y_i = -1 \wedge sign(y'_i - \theta) = -1\}|$
2.
   - **recall, sensitivity**: $\frac{a}{a+c}$;
   - **precision**: $\frac{a}{a+b}$;
   - **specificity**: $\frac{d}{d+b}$

3.
- **Averaging sensitivity and specificity**: $\Delta(\vec{y}', \vec{y}) = 1 - \frac{1}{2}\left(\frac{a}{a+c} + \frac{d}{d+b}\right)$;
- F_1**−score**: $\frac{2}{1/Precision+1/Recall}$, $\Delta(\vec{y}', \vec{y}) = 1 - F_1 = 1 - \frac{2a}{2a+b+c}$;
- F_β**−score**: $\frac{1+\beta^2}{1/Precision+\beta^2/Recall}$, $\Delta(\vec{y}', \vec{y}) = 1 - F_\beta = 1 - \frac{(1+\beta^2)a}{(1+\beta^2)a+b+\beta^2 c}$;
- **Recall at k**: Set θ so that $a + b \leq k$;
- **Precision at k**: Set θ so that $a + b \geq k$.

17.5.1 Linear Predictors for Bipartite Ranking

$$h_{\vec{w}}(x) = (\langle \vec{w}, \vec{x}_1 \rangle, \ldots, \langle \vec{w}, \vec{x}_r \rangle) = \vec{y}'.$$

$$\vec{b}_\theta(\vec{y}') = (sign(y'_1 - \theta), \ldots, sign(y'_r - \theta)) \in \{\pm 1\}^r.$$

$$V \subset \{\pm 1\}^r, \vec{b}_0(\vec{y}') = \arg\max_{\vec{v} \in V} \sum_{i=1}^{r} v_i y'_i.$$

For any θ, let $\vec{b}_\theta(\vec{y})$ is recall at k, then $\vec{b}_0(\vec{y}') = \arg\max_{\vec{v} \in V_{\geq k}} \sum_{i=1}^{r} v_i y'_i$; For any θ, let $\vec{b}_\theta(\vec{y})$ is precision at k, then $\vec{b}_0(\vec{y}') = \arg\max_{\vec{v} \in V_{\leq k}} \sum_{i=1}^{r} v_i y'_i$;

Hinge-loss of preceeding loss:

$$\begin{aligned}
\Delta(h_{\vec{w}}(x), \vec{y}) =& \Delta(\vec{b}(h_{\vec{w}}(x)), \vec{y}) \\
\leq& \Delta(\vec{b}(h_{\vec{w}}(x)), \vec{y}) + \sum_{i=1}^{r} (b_i(h_{\vec{w}}(x)) - \vec{y}) \langle \vec{w}, \vec{x}_i \rangle \\
\leq& \max_{\vec{v} \in V} \left[\Delta(\vec{v}, \vec{y}) + \sum_{i=1}^{r} (v_i - y_i) \langle \vec{w}, \vec{x}_i \rangle \right]
\end{aligned}$$

If we want use SGD, we need to calculate the hinge-loss function's subgradient, the computational bottleneck is calculating the argmax $\vec{v} \in V$.

We denote:

$$\mathcal{Y}_{a,b} = \{\vec{v} : |\{i : v_i = 1 \wedge y_i = 1\}| = a \wedge |\{i : v_i = 1 \wedge y_i = -1\}| = b\}$$

if we fix some a, b, then Δ is fixed, and we only need to calculate $\max_{\vec{v} \in \mathcal{Y}_{a,b}} \sum_{i=1}^{r} v_i \langle \vec{w}, \vec{x}_i \rangle$.

The pseudocode to get maximum $v \in V$ is left in the book.