Machine Learning and Computational Statistics
Multiclass

Nhung Le

Intro: This document consists of concepts and exercises related to Multiclass. This document discusses two main approaches One-vs-all and linear multiclass predictor.

1 Key Concepts

1. One-vs-All Our approach will assume we have a binary base classifier that returns a score, and we will predict the class that has the highest score.
   (a) pseudocode for one-vs-all (as shown in the Linear Binary Classifier Review)
   (b) Examples where one-vs-all fail: when it is too computational expensive (e.g., too many classes, a large time amount required to train one class of n classes.
2. Multiclass predictor: Our approach is to reframe multiclass learning such that rather than using a score function for each class, we use one function $h(x, y)$ that gives a compatible score between input x and output y.
   (a) Reframed multiclass hypothesis space:
      i. General [discrete] output space: Y
      ii. Base hypothesis space $\mathcal{H} = \{h : \mathcal{X} \times \mathcal{Y} \to \mathbf{R}\}$ for $h(x, y)$ gives compatible score between input x and output y
      iii. Multiclass hypothesis space $\mathcal{F} = \left\{f(x) = \arg\max_{y \in \mathcal{Y}} h(x, y) \mid h \in \mathcal{H}\right\}$.
      iv. Final prediction function is an $f \in \mathcal{F}$. Each $f \in \mathcal{F}$ has an underlying compatibility score function $h \in \mathcal{H}$.
   (b) **Compatible score**: Given class-specific score functions $h_1, h_2, \cdots, h_k$, we could define compatible score function as: $h(x, i) = h_i(x), i = 1, \cdots, k$
   (c) Reframed learning in a multiclass hypothesis space:
      i. **Margin** of the compatibility score function h on the ith example (x_i, y_i) for class y

$$m_{i,y}(h) = h(x_i, y_i) - h(x_i, y).$$

      ii. Two versions of Hinge Loss:
         A. Loss on an individual example (x_i, y_i)

$$\ell_1(h, (x_i, y_i)) = \max_{y \in \mathcal{Y} - \{y_i\}} \left(\max\left[0, \Delta(y_i, y) - m_{i,y}(h)\right]\right).$$

B. The generalized hinge loss

$$\ell_2(h,(x_i,y_i)) = \max_{y\in\mathcal{Y}} \left[\Delta\left(y_i,y\right) + h(x_i,y) - h(x_i,y_i)\right].$$

C. if $\Delta(y,y) = 0$ for all $y \in \mathcal{Y}$, then $\ell_2\left(h,(x_i,y_i)\right) = \ell_1(h,(x_i,y_i))$. [Hint: Note that $\max_{y\in\mathcal{Y}} \phi(y) = \max\left(\phi(y_i), \max_{y\in\mathcal{Y}-\{y_i\}} \phi(y_i)\right)$.]

iii. Non-negative class-sensitive loss function $\Delta : \mathcal{Y} \times \mathcal{A} \to [0,\infty)$

iv. Class-sensitive feature mapping $\Psi : \mathcal{X} \times \mathcal{Y} \to \mathbf{R}^d$

v. Linear class-sensitive score function:

$$h(x,y) = \langle w, \Psi(x,y)\rangle$$

vi. Prediction function $f : \mathcal{X} \to \mathcal{Y}$

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

vii. Training data $(x_1,y_1),\ldots,(x_n,y_n) \in \mathcal{X}\times\mathcal{Y}$

viii. $J(w)$ be the ℓ_2-regularized empirical risk function for the multiclass hinge loss for some $\lambda > 0$.

$$J(w) = \lambda\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n} \max_{y\in\mathcal{Y}} \left[\Delta\left(y_i,y\right) + \langle w, \Psi(x_i,y) - \Psi(x_i,y_i)\rangle\right],$$

3. Multiclass Concept Check

4. Map a set of linear score functions onto a single linear class-sensitive score function using a class-sensitive feature map. Give some intuition for the value of this feature map (based on features related to the target classes).

The Multivector Construction

- What if we stack w_i's together:

$$w = \left(\underbrace{-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}}_{w_1}, \underbrace{0,1}_{w_2}, \underbrace{\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}}_{w_3} \right)$$

- And then do the following: $\Psi : \mathbf{R}^2 \times \{1,2,3\} \to \mathbf{R}^6$ defined by

$$\begin{aligned} \Psi(x,1) &:= (x_1, x_2, 0, 0, 0, 0) \\ \Psi(x,2) &:= (0, 0, x_1, x_2, 0, 0) \\ \Psi(x,3) &:= (0, 0, 0, 0, x_1, x_2) \end{aligned}$$

- Then $\langle w, \Psi(x,y) \rangle = \langle w_y, x \rangle$, which is what we want.

Linear Binary Classifier Review

- Input Space: $\mathcal{X} = \mathbf{R}^d$
- Output Space: $\mathcal{Y} = \{-1, 1\}$
- Linear classifier score function:

$$f(x) \quad = \quad \langle w, x \rangle = w^T x$$

- Final classification prediction: $\text{sign}(f(x))$
- Geometrically, when are $\text{sign}(f(x)) = +1$ and $\text{sign}(f(x)) = -1$?

2 Implementation

2.1 Onv-vs-All Classifier

1. We have a BaseEstimator from sklearn (e.g., SVM Linear SVC)
2. Fit one classifier for each class, knowing that self.estimators[i] should be fit on class i vs rest. The key here is $y_{onevsall} = 1$ when y = the class.
3. Decision function returns the score of each input for each class.
4. Prediction function return the class with highest score.

2.2 Linear Multiclass SVM

1. Set-up featureMap
2. Define the Stochastic gradient descent
   (a) Objective function for some $\lambda > 0$.

$$J(w) = \lambda \|w\|^2 + \frac{1}{n} \sum_{i=1}^{n} \max_{y \in \mathcal{Y}} \left[\Delta \left(y_i, y \right) + \langle w, \Psi(x_i, y) - \Psi(x_i, y_i) \rangle \right]$$

   (b) $J(w)$ is convex so it has a subgradient at every point. For $\hat{y_i} = \max_{y \in \mathcal{Y}} [\Delta(y_i, y) + \langle w, \Psi(x_i - y) - \Psi(x_i - y_i) \rangle]$

$$g = 2\lambda w + \frac{1}{n} \sum_{i=1}^{n} (\Psi(x_i, \hat{y_i}) - \Psi(x_i, y_i))$$

   (c) Stochastic subgradient based on the point (x_i, y_i)

$$g = 2\lambda w + (\Psi(x_i, \hat{y_i}) - \Psi(x_i, y_i))$$

   (d) Minibatch subgradient based on the point $(x_i, y_i), \cdots, (x_{i+m}, y_{i+m})$

$$g = 2\lambda w + \frac{1}{m} \sum_{j=i}^{i+m-1} (\Psi(x_j, \hat{y}) - \Psi(x_j, y_j))$$

3. Fit
4. Decision function returns the score of each input for each class.
5. Predict function returns the class with the highest score.