From Binary to Multiclass Predictions

CMSC 422
MARINE CARPUAT
marine@cs.umd.edu
Given an arbitrary method for binary classification, how can we learn to make multiclass predictions?

Fundamental ML concept: reductions
Multiclass classification

• Real world problems often have multiple classes (text, speech, image, biological sequences...)

• How can we perform multiclass classification?
  – Straightforward with decision trees or KNN
  – Can we use the perceptron algorithm?
Reductions

• Idea is to re-use simple and efficient algorithms for binary classification to perform more complex tasks

• Works great in practice:
  – E.g., Vowpal Wabbit
One Example of Reduction: Learning with Imbalanced Data

**Task:** \( \alpha \)-Weighted Binary Classification

*Given:*

1. An input space \( \mathcal{X} \)
2. An unknown distribution \( \mathcal{D} \) over \( \mathcal{X} \times \{-1, +1\} \)

*Compute:* A function \( f \) minimizing: \( \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[ \alpha^y 1[f(x) \neq y] \right] \)

**Subsampling Optimality Theorem:**

If the binary classifier achieves a binary error rate of \( \varepsilon \), then the error rate of the \( \alpha \)-weighted classifier is \( \alpha \varepsilon \)
Today: Reductions for Multiclass Classification

**Task: Multiclass Classification**

*Given:*

1. An input space $\mathcal{X}$ and number of classes $K$
2. An unknown distribution $\mathcal{D}$ over $\mathcal{X} \times [K]$

*Compute:* A function $f$ minimizing: $\mathbb{E}_{(x,y) \sim \mathcal{D}} [f(x) \neq y]$
**Task: Binary Classification**

*Given:*

1. An input space $\mathcal{X}$
2. An unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \{-1, +1\}$

*Compute:* A function $f$ minimizing: $\mathbb{E}_{(x,y) \sim \mathcal{D}}[f(x) \neq y]$
How many classes can we handle in practice?

• In most tasks, number of classes $K < 100$

• For much larger $K$
  – we need to frame the problem differently
  – e.g, machine translation or automatic speech recognition
Reduction 1: OVA

• “One versus all” (aka “one versus rest”)
  – Train K-many binary classifiers
  – classifier k predicts whether an example belong to class k or not

  – At test time,
    • If only one classifier predicts positive, predict that class
    • Break ties randomly
Algorithm 12 OneVersusAllTrain($D_{\text{multiclass}}$, BinaryTrain)

1: for $i = 1$ to $K$ do
2: \hspace{1em} $D_{\text{bin}} \leftarrow$ relabel $D_{\text{multiclass}}$ so class $i$ is positive and $\neg i$ is negative
3: \hspace{1em} $f_i \leftarrow$ BinaryTrain($D_{\text{bin}}$)
4: end for
5: return $f_1, \ldots, f_K$

Algorithm 13 OneVersusAllTest($f_1, \ldots, f_K, \hat{x}$)

1: score $\leftarrow \langle 0, 0, \ldots, 0 \rangle$ \hspace{1em} // initialize $K$-many scores to zero
2: for $i = 1$ to $K$ do
3: \hspace{1em} $y \leftarrow f_i(\hat{x})$
4: \hspace{1em} score$_i \leftarrow$ score$_i + y$
5: end for
6: return $\text{argmax}_k$ score$_k$
Time complexity

• Suppose you have N training examples, in K classes. How long does it take to train an OVA classifier
  – if the base binary classifier takes $O(N)$ time to learn?
  – if the base binary classifier takes $O(N^2)$ time to learn?
Error bound

• **Theorem:** Suppose that the average error of the K binary classifiers is \( \varepsilon \), then the error rate of the OVA multiclass classifier is at most \((K-1) \varepsilon\)

• To prove this: how do different errors affect the maximum ratio of the probability of a multiclass error to the number of binary errors (“efficiency“)?
Error bound proof

• If we have a **false negative** on one of the binary classifiers (assuming all other classifiers correctly output negative)
• What is the probability that we will make an incorrect multiclass prediction?

\[
\frac{K - 1}{K}
\]

Efficiency: \[
\frac{K - 1}{K} / 1 = \frac{K - 1}{K}
\]
Error bound proof

• If we have k **false positives** with the binary classifiers

• What is the probability that we will make an incorrect multiclass prediction?
  – If there is also a false negative: 1
    • Efficiency = $1 / k + 1$
  – Otherwise $k / (k + 1)$
    • Efficiency = $k / (k + 1) / k = 1 / (k + 1)$
Error bound proof

• What is the worst case scenario?

  – False negative case: efficiency is \(\frac{K-1}{K}\)
    • Larger than false positive efficiencies

  – There are \(K\)-many opportunities to get false negative, \textbf{overall error bound is} \(\frac{K-1}{\varepsilon}\)
Reduction 2: AVA

• All versus all (aka all pairs)

• How many binary classifiers does this require?
Algorithm 14 AllVersusAllTrain($D_{\text{multiclass}}$, BinaryTrain)

1: $f_{ij} \leftarrow \emptyset, \forall 1 \leq i < j \leq K$
2: for $i = 1$ to $K-1$ do
3:   $D^{pos} \leftarrow \text{all } x \in D_{\text{multiclass}} \text{ labeled } i$
4:   for $j = i+1$ to $K$ do
5:     $D^{neg} \leftarrow \text{all } x \in D_{\text{multiclass}} \text{ labeled } j$
6:     $D^{bin} \leftarrow \{(x,+1) : x \in D^{pos}\} \cup \{(x,-1) : x \in D^{neg}\}$
7:     $f_{ij} \leftarrow \text{BinaryTrain}(D^{bin})$
8:   end for
9: end for
10: return all $f_{ij}$s

Algorithm 15 AllVersusAllTest(all $f_{ij}$, $\hat{x}$)

1: $score \leftarrow \langle 0,0,\ldots,0 \rangle$ \hfill // initialize $K$-many scores to zero
2: for $i = 1$ to $K-1$ do
3:   for $j = i+1$ to $K$ do
4:     $y \leftarrow f_{ij}(\hat{x})$
5:     $score_i \leftarrow score_i + y$
6:     $score_j \leftarrow score_j - y$
7:   end for
8: end for
9: return $\text{argmax}_k score_k$
Time complexity

- Suppose you have $N$ training examples, in $K$ classes. How long does it take to train an AVA classifier
  - if the base binary classifier takes $O(N)$ time to learn?
  - if the base binary classifier takes $O(N^2)$ time to learn?
Error bound

• **Theorem:** Suppose that the average error of the $K$ binary classifiers is $\epsilon$, then the error rate of the AVA multiclass classifier is at most $2(K-1)\epsilon$

• Question: Does this mean that AVA is always worse than OVA?
Extensions

• Divide and conquer
  – Organize classes into binary tree structures

• Use confidence to weight predictions of binary classifiers
  – Instead of using majority vote
Given an arbitrary method for binary classification, how can we learn to make multiclass predictions?

OVA, AVA

Fundamental ML concept: reductions
A taste of more complex problems: Collective Classification

• Examples:
  – object detection in an image
  – finding part of speech of words in a sentence
**Task: Collective Classification**

Given:

1. An input space $\mathcal{X}$ and number of classes $K$

2. An unknown distribution $\mathcal{D}$ over $\mathcal{G}(\mathcal{X} \times [K])$

Compute: A function $f : \mathcal{G}(\mathcal{X}) \rightarrow \mathcal{G}([K])$ minimizing:

$$\mathbb{E}_{(V,E) \sim \mathcal{D}} \left[ \sum_{v \in V} [\hat{y}_v \neq y_v] \right],$$

where $y_v$ is the label associated with vertex $v$ in $G$ and $\hat{y}_v$ is the label predicted by $f(G)$. 
How would you address collective classification?