From Binary to Multiclass Predictions

CMSC 422

Marine Carpuat

marine@cs.umd.edu

Topics

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: α-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: $\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\alpha^{y=1}[f(\boldsymbol{x}) \neq y]\right]$

Subsampling Optimality Theorem:
If the binary classifier achieves a binary error rate of ε, then the error rate of the α-weighted classifier is α ε

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: $\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[f(\boldsymbol{x}) \neq y\right]$

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: $\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[f(\boldsymbol{x}) \neq y\right]$

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($\mathbf{D}^{multiclass}$, BinaryTrain)

for $i = 1$ **to** K **do**
    $\mathbf{D}^{bin} \leftarrow$ relabel $\mathbf{D}^{multiclass}$ so class i is positive and $\neg i$ is negative
    $f_i \leftarrow$ BinaryTrain($\mathbf{D}^{bin}$)
end for
return $f_1, \ldots, f_K$

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

$score \leftarrow \langle 0, 0, \ldots, 0 \rangle$ // initialize K-many scores to zero
for $i = 1$ **to** K **do**
    $y \leftarrow f_i(\hat{\boldsymbol{x}})$
    $score_i \leftarrow score_i + y$
end for
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 ε, then the error rate of the OVA multiclass classifier is at most (K-1) ε

- 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?

$$(K - 1) / K$$

Efficiency: $(K - 1) / K / 1 = (K - 1) / K$

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 (K-1)/K
    - Larger than false positive efficiencies
  - There are K-many opportunities to get false negative, **overall error bound is (K-1) ε**

Reduction 2: AVA

- All versus all (aka all pairs)

- How many binary classifiers does this require?

Algorithm 14 ALLVERSUSALLTRAIN($\mathbf{D}^{multiclass}$, BINARYTRAIN)

```
f_ij ← ∅, ∀1 ≤ i < j ≤ K
for i = 1 to K-1 do
    D^pos ← all x ∈ D^multiclass labeled i
    for j = i+1 to K do
        D^neg ← all x ∈ D^multiclass labeled j
        D^bin ← {(x, +1) : x ∈ D^pos} ∪ {(x, −1) : x ∈ D^neg}
        f_ij ← BinaryTrain(D^bin)
    end for
end for
return all f_ijs
```

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

```
score ← ⟨0, 0, . . . , 0⟩                // initialize K-many scores to zero
for i = 1 to K-1 do
    for j = i+1 to K do
        y ← f_ij(x̂)
        score_i ← score_i + y
        score_j ← score_j - y
    end for
end for
return 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 ε, then the error rate of the AVA multiclass classifier is at most 2(K-1) ε

- 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

Topics

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 $f : \mathcal{G}(\mathcal{X}) \to \mathcal{G}([K])$ minimizing: $\mathbb{E}_{(V,E)\sim\mathcal{D}}\left[\sum_{v\in V}\left[\hat{y}_v \neq y_v\right]\right]$, 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?