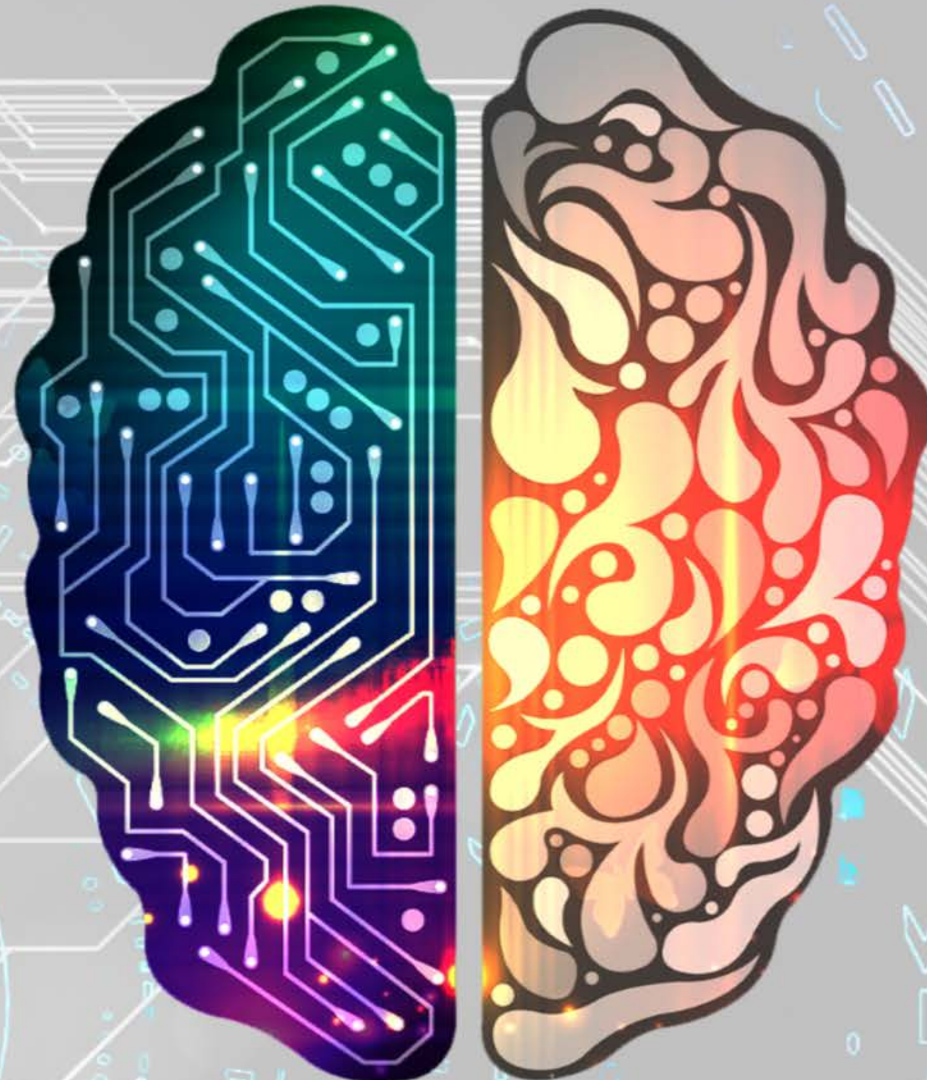
CMPS 460 – Spring 2022

MACHINE LEARNING

Tamer Elsayed



6

Beyond Binary Classification

Chapter 6

Roadmap …

- Using standard binary classifiers to solve other problems
  - Weighted classification
  - Multiclass classification

- Fundamental ML concept: reduction

Learning with Imbalanced Data

Imbalanced Data Distributions

- Sometimes training examples are drawn from an ***imbalanced distribution***.
- This results in an ***imbalanced training set***.
  - "needle in a haystack" problems
  - e.g., find fraudulent transactions in credit card histories

Why is this a big problem for the ML algorithms we know?

From Binary Classification …

TASK: BINARY CLASSIFICATION

Given:

1. An input space $\mathcal{X}$
2. An unknown distribution $\mathcal{D}$ over $\mathcal{X}\times\{-1,+1\}$
3. A training set D sampled from $\mathcal{D}$

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

to α-Weighted Binary Classification

TASK: α-WEIGHTED BINARY CLASSIFICATION

Given:

1. An input space $\mathcal{X}$
2. An unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \{-1, +1\}$
3. A training set D sampled from $\mathcal{D}$

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

We define cost of misprediction as:
α > 1 for y=+1 and 1 if y=-1

Given a good binary classifier, how can we solve the α-weighted binary classification?

Solution: Train a binary classifier on an "*induced*" distribution

Subsampling

Undersample the negative class.

- Positive examples: retain all
- Negative examples: retain only $1/\alpha$ fraction of them.

- Pass the induced distribution to binary classification.

Pros/Cons?

Oversampling

Oversample the positive class.

- Positive example: include α copies of it in the induced distribution.
- Negative example: include a single copy.

- Pass the induced distribution to binary classification.

Pros/Cons?

- Efficient implementations incorporate weight in learning algorithm, instead of explicitly duplicating data!

kNN?

Reduction (in this case …)

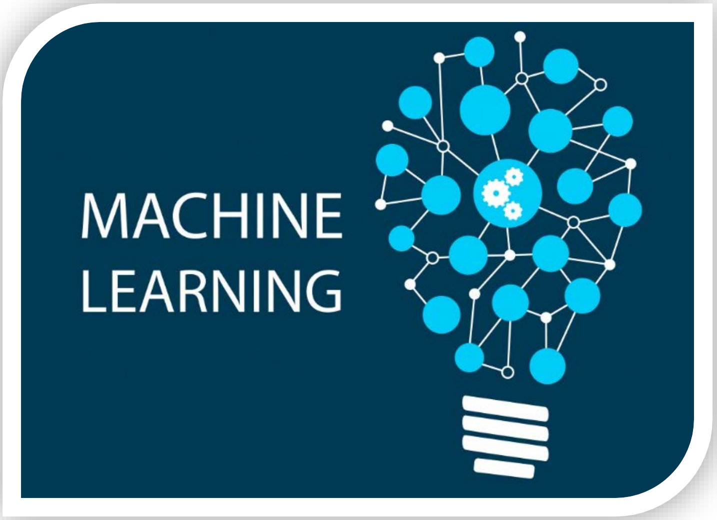
Re-using simple and efficient algorithms for binary classification to perform more complex tasks

Subsampling Optimality

- Theorem: If the binary classifier used in subsampling (*on the induced distribution*) achieves a binary error rate of ε, then the error rate of the α-weighted classifier (*on the original distribution*) is α ε.

- Same for oversampling!

Both methods have same error rate?!



Multiclass Classification

Multiclass Classification

- Real world problems often have multiple classes.

- How can we perform multiclass classification?
  - Decision trees?
  - kNN?
  - Perceptron?

Reduction to binary classification …

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

How many classes in practice?

- In most tasks, number of classes $K < 100$
- For much larger K
  - we need to frame the problem differently

Reduction 1: One Versus All (OVA)

aka "one versus rest"

- Train K 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 13 $\textsc{OneVersusAllTrain}(\mathbf{D}^{multiclass}, \textsc{BinaryTrain})$

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

Algorithm 14 $\textsc{OneVersusAllTest}(f_1, \dots, f_K, \hat{x})$

$score \leftarrow \langle 0, 0, \dots, 0 \rangle$ // initialize K-many scores to zero
for $i = 1$ **to** K **do**
  $y \leftarrow f_i(\hat{x})$
  $score_i \leftarrow score_i + y$
end for
return $\text{argmax}_k \; score_k$

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 $(K-1)\,\varepsilon$.

Reduction 2: All Versus All (AVA)

aka all pairs

- Train a classifier for each pair of classes.
- How many binary classifiers does this require?

- At test time?
  - The class with the most votes wins.

Algorithm 15 $\textsc{AllVersusAllTrain}(\mathbf{D}^{multiclass}, \textsc{BinaryTrain})$

$f_{ij} \leftarrow \emptyset, \forall 1 \le i < j \le K$
for $i = 1$ **to** $K\text{-}1$ **do**
  $\mathbf{D}^{pos} \leftarrow$ all $x \in \mathbf{D}^{multiclass}$ labeled i
  for $j = i{+}1$ **to** K **do**
    $\mathbf{D}^{neg} \leftarrow$ all $x \in \mathbf{D}^{multiclass}$ labeled j
    $\mathbf{D}^{bin} \leftarrow \{(x, +1) : x \in \mathbf{D}^{pos}\} \cup \{(x, -1) : x \in \mathbf{D}^{neg}\}$
    $f_{ij} \leftarrow \textsc{BinaryTrain}(\mathbf{D}^{bin})$
  end for
end for
return all f_{ij}s

Algorithm 16 $\textsc{AllVersusAllTest}(\text{all } f_{ij}, \hat{x})$

$score \leftarrow \langle 0, 0, \dots, 0 \rangle$ // initialize K-many scores to zero
for $i = 1$ **to** $K\text{-}1$ **do**
  for $j = i{+}1$ **to** K **do**
    $y \leftarrow f_{ij}(\hat{x})$
    $score_i \leftarrow score_i + y$
    $score_j \leftarrow score_j - y$
  end for
end for
return $\text{argmax}_k\ score_k$

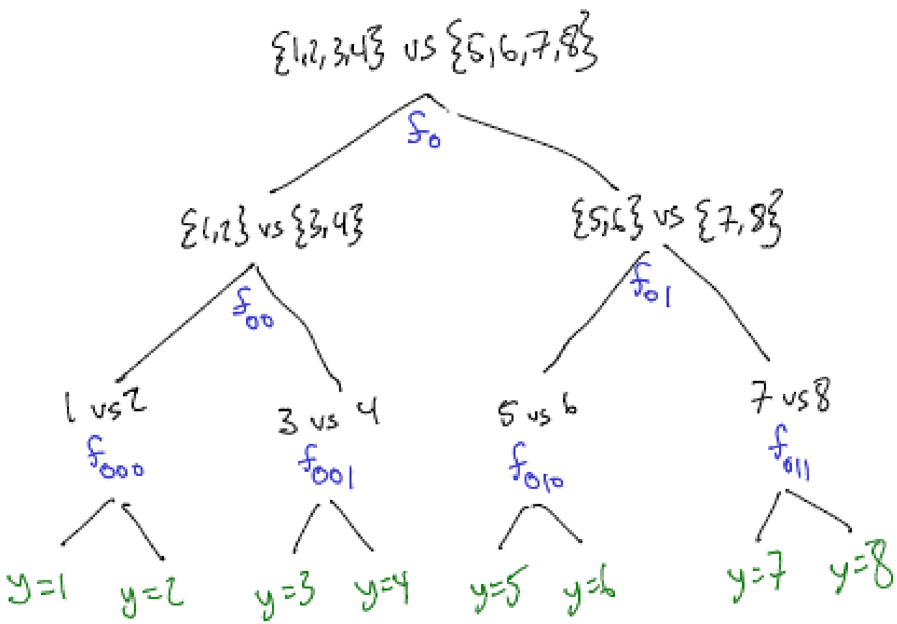
Error Bound

- 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 $2(K-1)\,\varepsilon$.

AVA is always worse than OVA?

Extensions

- **Divide and conquer**
  - Organize classes into binary tree structures
    ➔ **binary tree of classifiers**



- Use **confidence** to weight predictions of binary classifiers
  - Instead of using majority vote