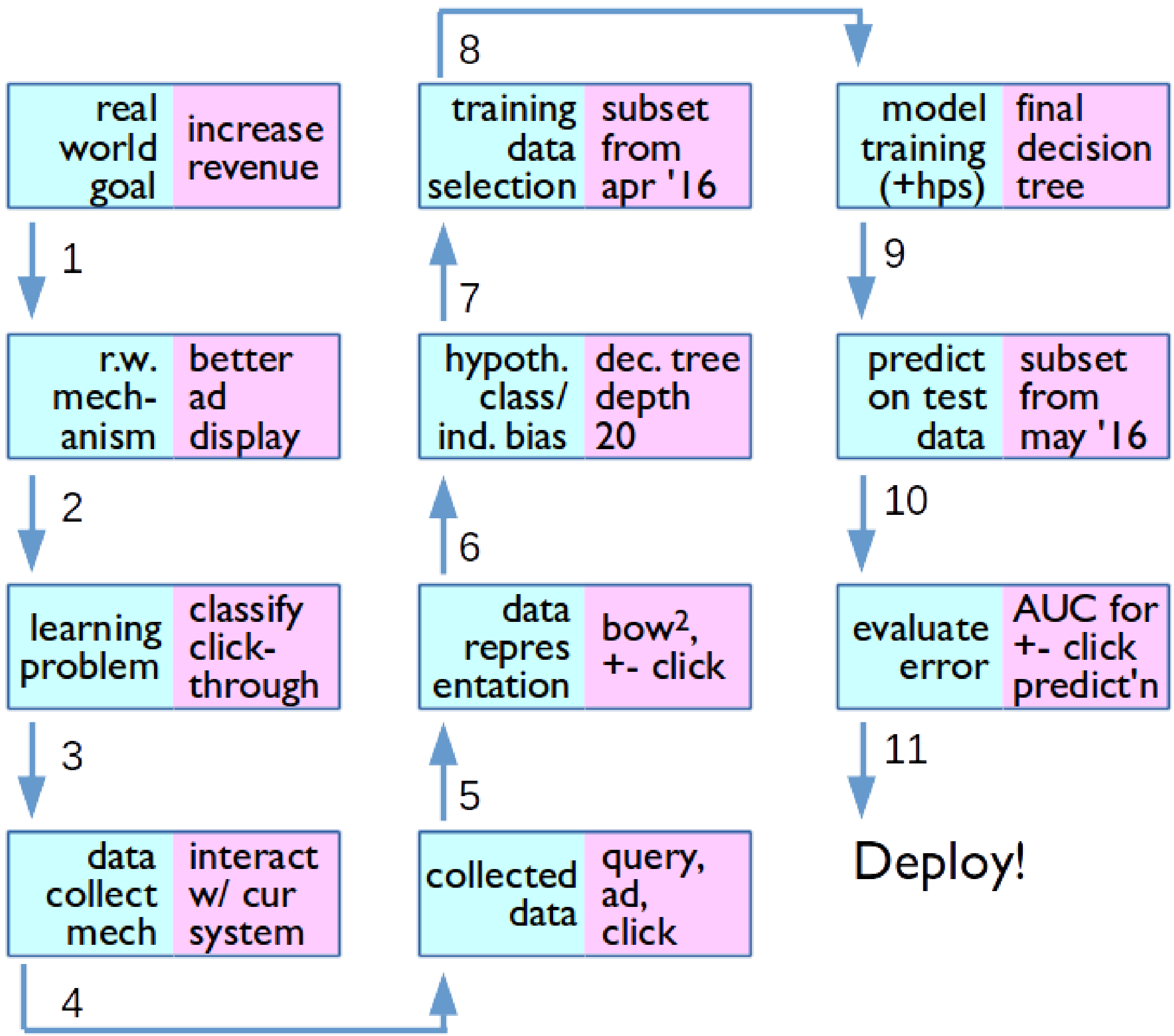
Imbalanced Data and Reductions

CMSC 422

MARINE CARPUAT

marine@cs.umd.edu

Typical Design Process for an ML Application



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?

Recall: Machine Learning as Function Approximation

Problem setting

- Set of possible instances X
- Unknown target function $f: X \rightarrow Y$
- Set of function hypotheses $H = \{h \mid h: X \rightarrow Y\}$

Input

- Training examples $\{(x^{(1)}, y^{(1)}), \ldots (x^{(N)}, y^{(N)})\}$ of unknown target function f

Output

- Hypothesis $h \in H$ that best approximates target function f

Recall: Loss Function

$l(y, f(x))$ where y is the truth and $f(x)$ is the system's prediction

$$\text{e.g. } l(y, f(x)) = \begin{cases} 0 & if\ y = f(x) \\ 1 & otherwise \end{cases}$$

Captures our notion of what is important to learn

Recall: Expected loss

- f should make good predictions
  - as measured by loss l
  - on **future** examples that are also drawn from D

- Formally
  - ε , the expected loss of f over D with respect to l should be small

$$\varepsilon \triangleq \mathbb{E}_{(x,y)\sim D}\{l(y, f(x))\} = \sum_{(x,y)} D(x,y) l(y, f(x))$$

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

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

Given a good algorithm for solving the binary classification problem, how can I solve the α-weighted binary classification problem?

We define cost of misprediction as:

α > 1 for y = +1

1 for y = -1

Solution: Train a binary classifier on an induced distribution

Algorithm 11 $\textsc{SubsampleMap}(\mathcal{D}^{weighted}, \alpha)$

```
while true do
    (x, y) ~ D^weighted        // draw an example from the weighted distribution
    u ~ uniform random variable in [0, 1]
    if y = +1 or u < 1/α then
        return (x, y)
    end if
end while
```

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

- Let's prove it.

  (see also CIML 6.1)

Strategies for inducing a new binary distribution

- Undersample the negative class
- Oversample the positive class

Strategies for inducing a new binary distribution

- Undersample the negative class
  - More computationally efficient
- Oversample the positive class
  - Base binary classifier might do better with more training examples
  - Efficient implementations incorporate weight in algorithm, instead of explicitly duplicating data!

Algorithm 1 DecisionTreeTrain(*data*, *remaining features*)

```
guess ← most frequent answer in data                       // default answer for this data
if the labels in data are unambiguous then
    return Leaf(guess)                                 // base case: no need to split further
else if remaining features is empty then
    return Leaf(guess)                                 // base case: cannot split further
else                                                   // we need to query more features
    for all f ∈ remaining features do
        NO ← the subset of data on which f=no
        YES ← the subset of data on which f=yes
        score[f] ← # of majority vote answers in NO
                   + # of majority vote answers in YES
                                   // the accuracy we would get if we only queried on f
    end for
    f ← the feature with maximal score(f)
    NO ← the subset of data on which f=no
    YES ← the subset of data on which f=yes
    left ← DecisionTreeTrain(NO, remaining features \ {f})
    right ← DecisionTreeTrain(YES, remaining features \ {f})
    return Node(f, left, right)
end if
```

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

Learning with Imbalanced Data is an Example of Reduction

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

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

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 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

What you should know

- How can we take the standard binary classifier and adapt it to handle problems with
  - Imbalanced data distributions
  - Multiclass classification problems

- Algorithms & guarantees on error rate

- Fundamental ML concept: reduction

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

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 $f_1, \ldots, f_K$

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

$score \leftarrow \langle 0, 0, \ldots, 0 \rangle$ // 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 $\operatorname{argmax}_k score_k$

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

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

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

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