Logistic Regression

CMSC 422

SOHEIL FEIZI

sfeizi@cs.umd.edu

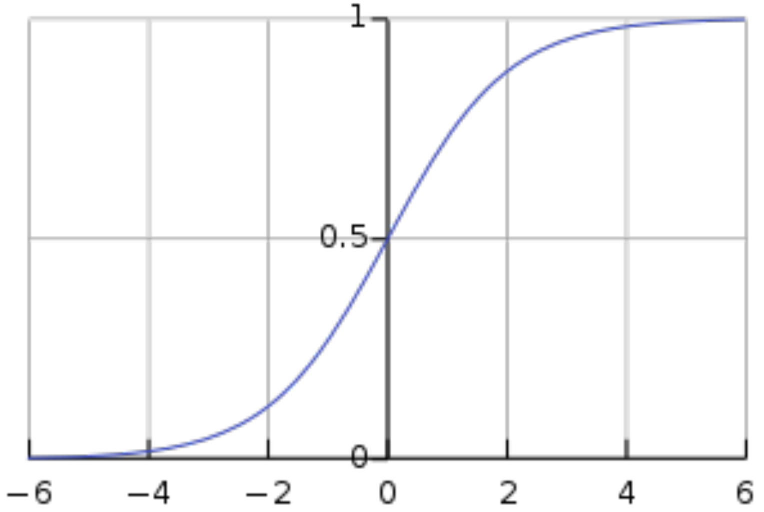
Slides partially adapted from MARINE CARPUAT

Logistic Regression

- Binary classification

$$P(Y^{(i)} = 1 | X^{(i)}, \theta) = g(< \theta, X^{(i)} >)$$

$$P(Y^{(i)} = 0 | X^{(i)}, \theta) = 1 - g(< \theta, X^{(i)} >)$$



Sigmoid function

$$g(z) = \frac{1}{1 + \exp(-z)}$$

Logistic Regression

- Maximum Likelihood

$$\max_{\theta} \quad \prod_{i=1}^{N} P(Y^{(i)} | X^{(i)}, \theta)$$

$$\Downarrow$$

$$\max_{\theta} \quad \prod_{i=1}^{N} g(<\theta, X^{(i)}>)^{Y^{(i)}} (1 - g(<\theta, X^{(i)}>))^{1-Y^{(i)}}$$

$$\Downarrow$$

$$\max_{\theta} \quad \underbrace{\sum_{i=1}^{N} Y^{(i)} \log g(<\theta, X^{(i)}>) + (1 - Y^{(i)}) \log(1 - g(<\theta, X^{(i)}>))}_{\text{Cross-entropy loss function}}$$

Cross-entropy loss function

How to solve it?

- Gradient Descent

- A good property of sigmoid:

$$\nabla_z g(z) = g(z)(1 - g(z))$$

- SGD: $\theta_{k+1} = \theta_k + \eta(Y^i - g(< \theta, X^i >))X^{(i)}$

- Why? Intuition behind the updates

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

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($\mathbf{D}^{multiclass}$, 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 $f_1, \dots, f_K$

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

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

Reduction 2: AVA

- All versus all (aka all pairs)

- How many binary classifiers does this require?

Algorithm 14 AllVersusAllTrain($\mathbf{D}^{multiclass}$, BinaryTrain)

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

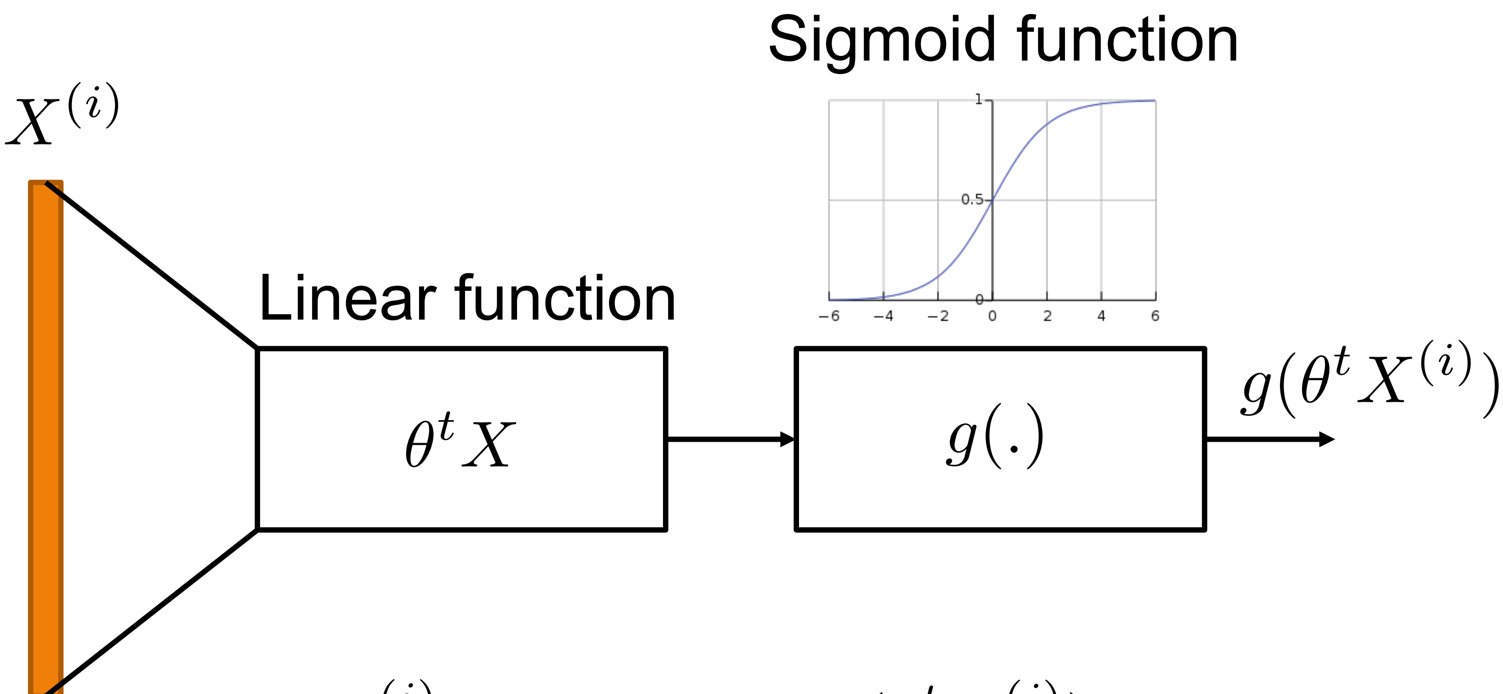
Algorithm 15 AllVersusAllTest(all f_{ij}, $\hat{\boldsymbol{x}}$)

$score \leftarrow \langle 0, 0, \dots, 0 \rangle$ // 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?

A High-Level View



$$Y^{(i)} = 1 \longrightarrow g(\theta^t X^{(i)}) \approx 1$$

$$Y^{(i)} = 0 \longrightarrow g(\theta^t X^{(i)}) \approx 0$$

Does cross entropy optimization encourage this?

Multi-Label Classification

- Suppose we have labels {0,1,…,k}

- How can we extend logistic regression's formulation for the general case?

Recall the probabilistic model

- In binary classification, we have

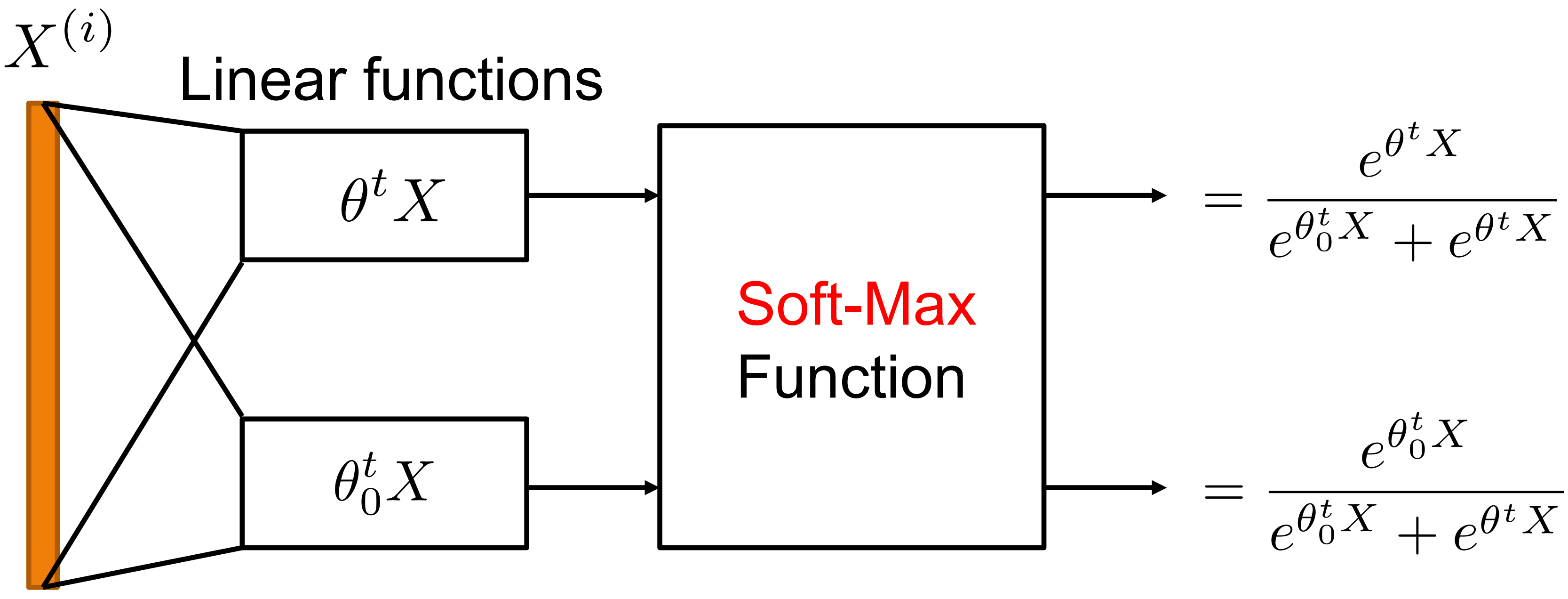
$$P(Y=1|X,\theta)=g(\theta^t X) = \frac{1}{1+e^{-\theta^t X}} = \frac{e^{\theta^t X}}{1+e^{\theta^t X}}$$

$$= \boxed{\frac{e^{\theta^t X}}{e^{\theta_0^t X}+e^{\theta^t X}}}$$

$$P(Y=0|X,\theta)=1-g(\theta^t X) = \frac{1}{1+e^{\theta^t X}} = \boxed{\frac{e^{\theta_0^t X}}{e^{\theta_0^t X}+e^{\theta^t X}}}$$

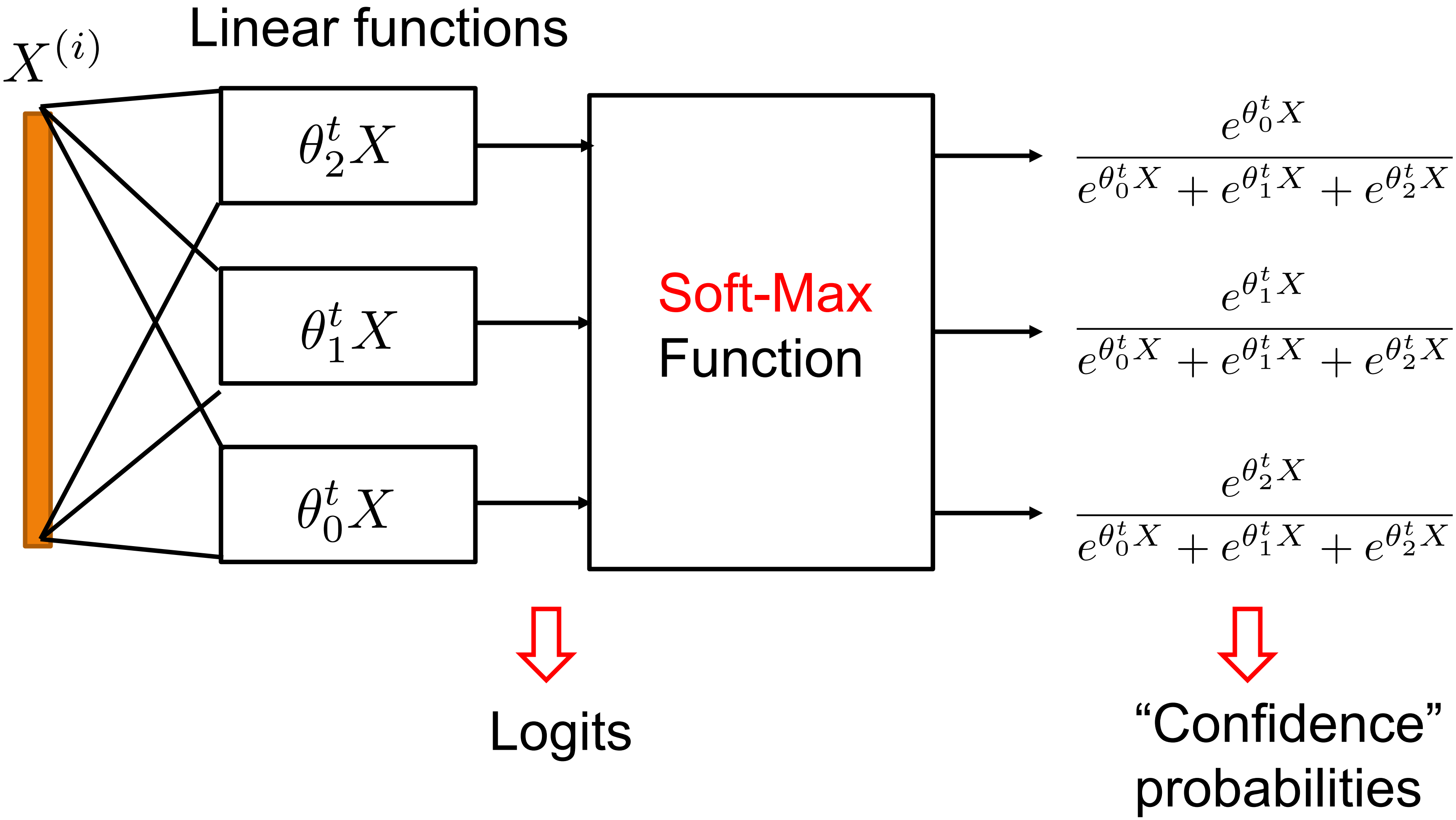
- If $\theta_0 = 0 \longrightarrow e^{\theta_0^t X} = 1$

A High-Level View:Binary Classification



How to extend this to the multi label classification?

Multi-Label Classification



Cross-Entropy Loss for Multi-Label Case

- Recall the binary case

$$\max_{\theta} \quad \sum_{i=1}^{N} Y^{(i)} \log g(< \theta, X^{(i)} >) + (1 - Y^{(i)}) \log(1 - g(< \theta, X^{(i)} >))$$

- Multi-label case

$$\sum_{\text{all samples}} \quad 1\{Y^{(i)} = \text{label}\} \log(\text{corresponding confidence prob.})$$