

CS 744: BIG DATA SYSTEMS

Shivaram Venkataraman

Fall 2018

ADMINISTRIVIA

- Assignment 2
- Guest Lecture

MACHINE LEARNING

MACHINE LEARNING

Classification



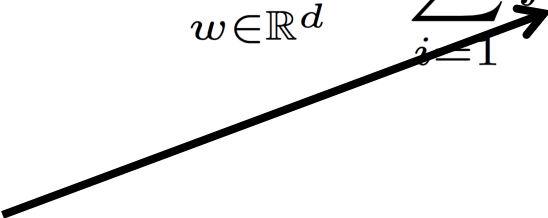
Recommendation



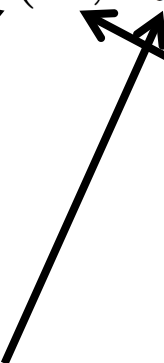
OPTIMIZATION

$$\min_{w \in \mathbb{R}^d} \sum_{i=1}^N f(w, z_i) + P(w)$$

Function



Data (Examples)



Regularization



Model



CONVEX OPTIMIZATION

$$\min_{w \in \mathbb{R}^d} \sum_{i=1}^N f(w, z_i) + P(w)$$

What is convex ?

Linear Regression, Linear SVM

Kernel SVMs, Logistic Regression,

What is not convex ?

Graph mining, Deep Learning

GRADIENT DESCENT

$$w^{(k+1)} = w^{(k)} - \alpha_k \nabla f(w^{(k)})$$

Initialize w

For many iterations:

 Compute Gradient

 Update model

End

INCREMENTAL GRADIENT DESCENT

$$w^{(k+1)} = w^{(k)} - \alpha_k \nabla f_{\eta(k)}(w^{(k)})$$

Initialize w

For many iterations:

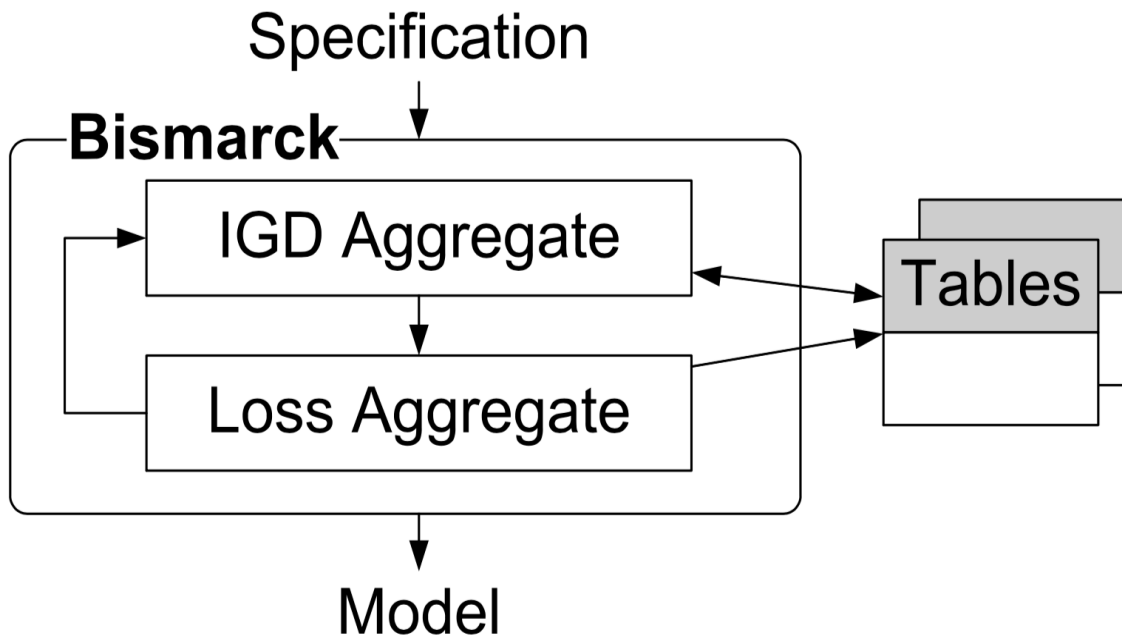
Pick one point

Compute Gradient

Update model

End

BISMARCK ARCHITECTURE



BISMARCK: USER DEFINED AGGREGATE

Three steps:

1. initialize(state)

2. transition(state, data)

3. terminate(state)

BISMARCK: LOGISTIC REGRESSION

```
LR_Transition(ModelCoef *w, Example e) { ...  
    wx = Dot_Product(w, e.x);  
    sig = Sigmoid(-wx * e.y);  
    c = stepsize * e.y * sig;  
    Scale_And_Add(w, e.x, c); ... }
```

DATA ORDERING

Random sampling

- Sample **without replacement**
- Shuffle the data after each **epoch**

Shuffle once

- Avoids pathological ordering
- Much cheaper

RESERVOIR SAMPLING

Select first m items

On the k^{th} additional item

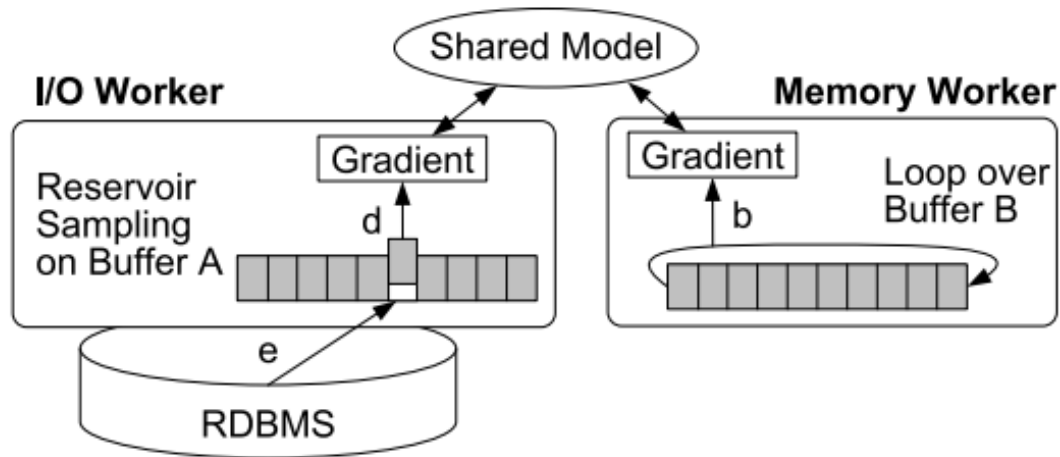
$s = \text{random in } [0, m + k)$

if $s < m$

Put in slot s

else

Drop the item



PARALLEL GRADIENTS

Shared Memory:

- Compute gradients in parallel
- Average their updates
- Or update in parallel (locks ?)

More in next lecture !

QUESTIONS / DISCUSSION ?