CSCI567 Machine Learning (Fall 2014)

Drs. Sha & Liu

{feisha,yanliu.cs}@usc.edu

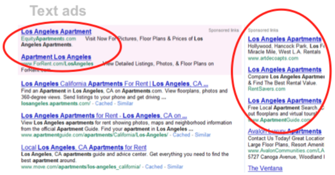
November 5, 2014

Outline

Admin Stuff

Outline

Motivating Examples



Text ads

Los Angeles Apartment
EquityApartments.com Visit Now For Pictures, Floor Plans & Prices of **Los Angeles Apartments**.

Apartment Los Angeles
www.ForRent.com/LosAngeles View Detailed Listings, Photos, & Floor Plans on ForRent.com

Sponsored links

Los Angeles California Apartments For Rent | Los Angeles, CA ...
Find an **Apartment** in **Los Angeles**, CA on **Apartments**.com. View floorplans, photos and 360-degree views. Send listings to your phone and get driving ...
losangeles.apartments.com/ - Cached - Similar

Los Angeles Apartments for Rent - Los Angeles, CA on ...
View **Los Angeles apartments** for rent showing photos, maps and neighborhood information from the official **Apartment** Guide. Find your **apartment** in **Los Angeles** ...
www.apartmentguide.com/apartments/California/Los-Angeles/ - Cached - Similar

Local Los Angeles, CA Apartments for Rent
Los Angeles, CA **apartments** guide and advice center. Get everything you need to find the best **apartment** around.
www.move.com/apartments/los-angeles_california/ - Cached - Similar

Sponsored links

Los Angeles Apartments
Hollywood, Hancock Park, **Los** F
Miracle Mile, West L.A. Rentals
www.artdecoapts.com

Los Angeles Apartments
Compare **Los Angeles Apartme**
& Find The Best Rental Value.
RentSavers.com

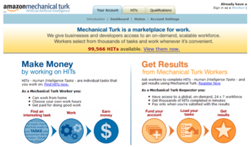
Los Angeles Apartments
Free Local **Apartment** Search.
out floorplans and virtual tours
www.ApartmentGuide.com

Avalon Luxury Apartments
Contact Us Today! Great Locatio
Large Floor Plans, Resort Ameni
www.AvalonCommunities.com/LA
5727 Canoga Avenue, Woodland I

The Ventana

(a) Computational Advertising

(b) Recommendation System



amazonmechanical turk
Artificial Artificial Intelligence

Your Account | HITs | Qualifications

Already have a
Sign in as a Worker |

Introduction | Dashboard | Status | Account Settings

Mechanical Turk is a marketplace for work.
We give businesses and developers access to an on-demand, scalable workforce.
Workers select from thousands of tasks and work whenever it's convenient.
99,566 HITs available. View them now.

Make Money
by working on HITs

HITs - *Human Intelligence Tasks* - are individual tasks that you work on. Find HITs now.

As a Mechanical Turk Worker you:

- Can work from home
- Choose your own work hours
- Get paid for doing good work

Find an interesting task → Work → Earn money

Get Results
from Mechanical Turk Workers

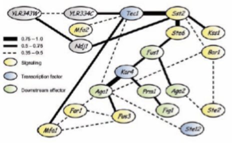
Ask workers to complete HITs - *Human Intelligence Tasks* - and get results using Mechanical Turk. Register Now

As a Mechanical Turk Requester you:

- Have access to a global, on-demand, 24 x 7 workforce
- Get thousands of HITs completed in minutes
- Pay only when you're satisfied with the results

Fund your account → Load your tasks → Get results

(c) Crowd sourcing



(d) Scientific discovery

Summary of Techniques

A variety of techniques have been developed: some have been discussed and others to be discussed in this class

- Parallel algorithms: will be discussed
- Online algorithms: e.g., perception
- Stochastic algorithms: e.g., stochastic gradient descent
- Locality sensitive hashing: will be discussed
- Distributed optimization: will be discussed
- Greedy algorithms: e.g. greedy search algorithm for solving Lasso
- First-order algorithms: e.g., NESTA

Outline

Parallel Algorithms

We are at the beginning of the multicore era (Moore's law)

[Sutter and Larus, 2005] points out that multicore mostly benefits concurrent applications

- Best match if the data can be subdivided and stay local to the cores:

$$\sum f(x_i) = f(x_1) + f(x_2) + \ldots + f(x_n).$$

- Permit the learning algorithm to access the learning problem only through a statistical query oracle. E.g. given a function of $f(x, y)$, it returns an estimate of the expectation of $f(x, y)$.

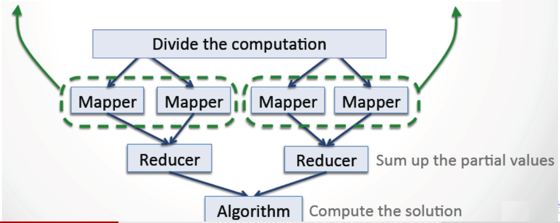
Parallel Algorithm Examples: Linear Regression

Remember in linear regression, we have the following solution of $y = \theta^T x$ as

$$\theta^\star = (X^T X)^{-1} X^T \vec{y}$$

$$\boxed{\begin{aligned} A &= X^T X \\ b &= X^T \vec{y} \\ \theta^* &= (A)^{-1} b \end{aligned}} \Rightarrow \boxed{\begin{aligned} A &= \textstyle\sum_{i=1}^{m} (x_i x_i^T) \\ b &= \textstyle\sum_{i=1}^{m} (x_i y_i) \\ \theta^* &= (A)^{-1} b \end{aligned}}$$

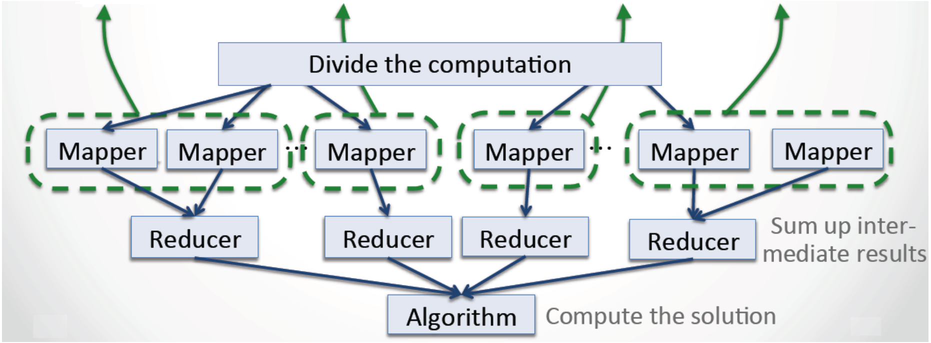
$$A = \textstyle\sum_{i=1}^{m} (x_i x_i^T) \quad \text{and } b = \textstyle\sum_{i=1}^{m} (x_i y_i)$$



Parallel Algorithm Examples: Naive Bayes

Remember in Naive Bayes, we need to estimate $P(x_j = k|y = 1)$ and $P(y)$ from the training data. In other words, we need to calculate the following terms:

$$\sum_{subgroup} 1\{x_j = k|y = 1\}, \sum_{subgroup} 1\{x_j = k|y = 0\}, \sum_{subgroup} 1\{y = 1\}, \sum_{subgroup} 1\{y = 0\}$$



Outline

Locality-sensitive hashing (LSH): Introduction

Difficulties faced by machine learning applications, where exist

- High dimensional data feature (e.g. vocabulary token histogram)
- Large size of training data (e.g. personal emails of thousands of users)

Reduce the feature dimension effectively:

- From the original features $x \rightarrow R^d$
  To the transformed features $\phi(x) \rightarrow R^m$,
  where $m << d$ through function $\phi(x)$
- With little information loss:
  preserve norm $\|x\|_\phi = \|x\|_2$, and
  preserve inner product $< \phi(x), \phi(y) >=< x, y >$.

Locality-sensitive hashing (LSH)

Given a collection of objects $\mathcal{C}$ and similarity function $sim(x, y)$ that maps pairs of objects $\mathbf{x}, \mathbf{y} \in \mathcal{C}$, a locality sensitive hash function family $\mathcal{F}$ operates on C, such that for any $x, y \in \mathcal{C}$,

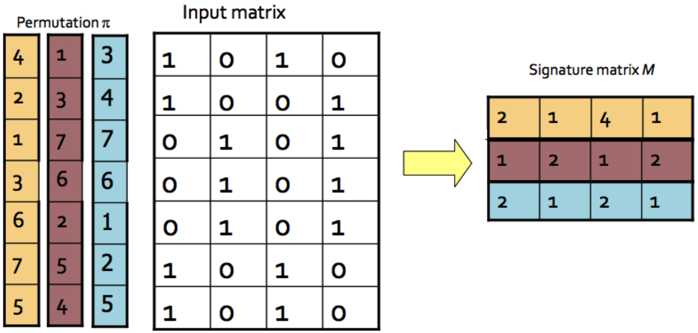
$$Prob_{h\in\mathcal{F}}[h(\mathbf{x}) = h(\mathbf{y})] = sim(\mathbf{x}, \mathbf{y}).$$

Below are two examples of locality sensitive hash function family:

1. Min-wise independent permutations for Jaccard coefficients: $Prob_{h\in\mathcal{F}}[\min\{\pi(\mathbf{x})\} = \min\{\pi(\mathbf{y})\}] = |\mathbf{x} \cap \mathbf{y}|/|\mathbf{x} \cup \mathbf{y}|$.
2. Random projection for cosine similarity: $Prob_{h\in\mathcal{F}}[h(\mathbf{x}) = h(\mathbf{y})] = 1 - \theta(\mathbf{x}, \mathbf{y})/\pi$, where $\theta(\mathbf{x}, \mathbf{y})$ is the angle between vector $\mathbf{x}$ and $\mathbf{y}$.

LSH Example: Min-wise independent permutations

- A k-shingle (or k-gram) is a sequence of k tokens that appears in the document.
- Represent a doc by the set of hash values of its k-shingles.
- A natural document similarity measure is then the Jaccard similarity: $sim(\mathbf{x}, \mathbf{y}) = |\mathbf{x} \cap \mathbf{y}|/|\mathbf{x} \cup \mathbf{y}|$.



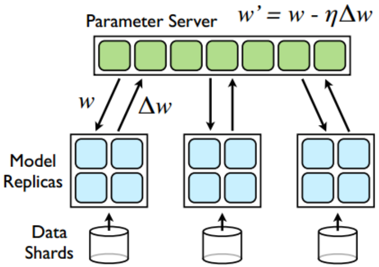
LSH Example: Random Projection

- The random projection method of LSH is designed to approximate the cosine distance between vectors.
- The basic idea of this technique is to choose a random hyperplane (defined by a normal unit vector) at the outset and use the hyperplane to hash input vectors.
- Given an input vector x and a hyperplane defined by r, we let $h(x) = sign(v \cdot r)$. That is, $h(x) = +/-1$ depending on which side of the hyperplane lies.

Outline

Distributed Optimization

1. Massive data can be stored or even collected in distributed manner.
   - Example 1: cloud computing, Hadoop/MapReduce
   - Example 2: training data of extremely large size.
2. Increasing computing resources, parallel computation.
   - e.g., features with high dimensions.



Distributed optimization for large scale problem (Dean et al. 2012)

Distributed SGD - One shot averaging

Settings:

- Divide the data into K disjoint sets such that $S = (s_1, ..., s_K)$.
- Suppose we have K machines.

Algorithm:

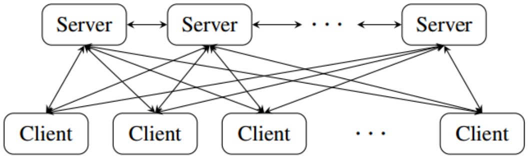
```
for each machine $i \in \{1, ..., K\}$ parallel do:
    compute $\hat{w}_i = \textbf{SGD}(s_i, T, \eta, w_0)$
return    $\frac{1}{K} \sum_{i=1}^{K} \hat{w}_i$
```

where T is the maximum number of iterations, η is the step size, and w_0 is the initial model parameter.

This method is simple, and no communication until the end!

Parameter Server (M.Li et al. 2014)

- Sometimes, it is impossible to store all the parameters in single machine. i.e. for social network with billions of edges, the size of model parameters can go up to few thousand terabyte.



Communication between clients and servers. Clients process data and while servers synchronize parameters and perform global updates.

Outline

Tips

- Start early! The data is fairly large and you will need some time to be able to handle it.
- Start with simpler algorithms like linear algorithms so that you make at least a submission.
- Make sure that you use online algorithms like stochastic gradient descent if you cannot load the data into memory.
- Always use all of the available cores in your system via the multi-threading packages. (e.g. OpenMP in C++)
- If you don't have enough time, tune your algorithms in a small amount of data, then run the selected algorithm on the entire data.

Resources

Scikit-Learn	`http://scikit-learn.org/` Easy to use, general purpose Python library for machine learning. Good for large scale learning. It has many of the important algorithms implemented.
Vowpal-Wabbit	`http://hunch.net/~vw/` Very fast C++ library for most of the fundamental algorithms.
Shogun	`http://www.shogun-toolbox.org/` Written in C++, with interface for many languages. Mainly for large-scale learning.
Graphlab	`http://graphlab.com` Very fast implementation of graphical models and graph-based algorithms.
ParameterServer	`http://parameterserver.org` Distributed machine learning algorithms

And many others. See a comparison of features in here: `http://www.shogun-toolbox.org/page/features/`.