Data Mining

CS 5140 / CS 6140

Jeff M. Phillips

January 9, 2017

www.cs.utah.edu/~jeffp/teaching/cs5140.html

Data Mining

Instructor : Jeff Phillips (email) | Office hours: TBA @ MEB 3442 (and directly after class in WEB L104)
TAs: WaiMing Tai (email) | Office hours: 11am-noon Friday, location TBA
+ Deb Paul (email) | Office Hours: 3-4pm Thursday, location TBA
+ Yang Gao (email) | Office Hours: 9-11am Tuesdays, location TBA
+ Shweta Singhal (email) | Office Hours: TBA
others TBA
Spring 2017 | Mondays, Wednesdays 3:00 pm - 4:20 pm
WEB L104
Catalog number: CS 5140 01 or CS 6140 01

Syllabus

Description:
Data mining is the study of efficiently finding structures and patterns in large data sets. We will focus on several aspects of this: (1) converting from a messy and noisy raw data set to a structured and abstract one, (2) applying scalable and probabilistic algorithms to these well-structured abstract data sets, and (3) formally modeling and understanding the error and other consequences of parts (1) and (2), including choice of data representation and trade-offs between accuracy and scalability. These steps are essential for training as a data scientist.
Algorithms, probability, and linear algebra are required mathematical tools for understanding these approaches.
Topics will include: similarity search, clustering, regression/dimensionality reduction, graph analysis, PageRank, and small space summaries. We will also cover several recent developments, and the application of these topics to modern applications, often relating to large internet-based companies.
Upon completion, students should be able to read, understand, and implement ideas from many data mining research papers.

Books:
We will in general not follow any book. My own course notes (linked below) serve as the defacto book. However, the following two free online books may serve as useful references that have good overlap with the course.
MMDS(v1.3): Mining Massive Data Sets by Anand Rajaraman, Jure Leskovec, and Jeff Ullman. The digital version of the book is free, but you may wish to purchase a hard copy.
FoDS: Foundations of Data Science by Avrim Blum, John Hopcroft and Ravindran Kannan. This provide some proofs and formalisms not explicitly covered in lecture.

Videos: We plan to videotape all lectures, and make them available online. They will appear on this playlist on our YouTube Channel.
Videos will also **livestream here**.
Lectures will also be live-streamed and available through Luum. More information to come.

Prerequisits: A student who is comfortable with basic probability, basic linear algebra, basic big-O analysis, and basic programming and data structures should be qualified for the class. There is no specific language we will use. However, programming assignments will often (intentionally) not be as specific as in lower-level classes. This will partially simulate real-world settings where one is given a data set and asked to analyze it; in such settings even less direction is provided.
For undergrads, the prerequisits are CS 3500 and CS 3130 and MATH 2270 (or equivalent), and CS 4150 is a corequisite. I will grant exceptions for those with (a reasonable grade in) CS 4964 (Fall 2016).
In the past, this class has had undergraduates, masters, and PhD students, including many from outside of Computer Science. Most (but not all) have kept up fine, and still most have been challenged. If you are unsure if the class is right for you, contact the instructor.

Schedule: (subject to change - some linked material is from the previous iteration of the class)

Date	Topic (+ Notes)	Video	Link	Assignment (latex)	Project

www.cs.utah.edu/~jeffp/teaching/cs5140.html

Data Mining

Instructor : Jeff Phillips (email) | Office hours: TBA @ MEB 3442 (and directly after class in WEB L1
TAs: WaiMing Tai (email) | Office hours: 11am-noon Friday, location TBA
+ Deb Paul (email) | Office Hours: 3-4pm Thursday, location TBA
+ Yang Gao (email) | Office Hours: 9-11am Tuesdays, location TBA
+ Shweta Singhal (email) | Office Hours: TBA
others TBA
Spring 2017 | Mondays, Wednesdays 3:00 pm - 4:20 pm
WEB L104
Catalog number: CS 5140 01 or CS 6140 01

Syllabus

Description:
Data mining is the study of efficiently finding structures and patterns in large data sets. We will focus on se
and abstract one, (2) applying scalable and probabilistic algorithms to these well-structured abstract data s
parts (1) and (2), including choice of data representation and trade-offs between accuracy and scalability. T
Algorithms, probability, and linear algebra are required mathematical tools for understanding these approa
Topics will include: similarity search, clustering, regression/dimensionality reduction, graph analysis, PageR
the application of these topics to modern applications, often relating to large internet-based companies.
Upon completion, students should be able to read, understand, and implement ideas from many data mini

Books:
We will in general not follow any book. My own course notes (linked below) serve as the defacto book. How
good overlap with the course.
MMDS(v1.3): Mining Massive Data Sets by Anand Rajaraman, Jure Leskovec, and Jeff Ullman. The digita
FoDS: Foundations of Data Science by Avrim Blum, John Hopcroft and Ravindran Kannan. This provide s

Videos: We plan to videotape all lectures, and make them available online. They will appear on this playlis

Syllabus

Syllabus

Instructor: Jeff M. Phillips. | 3442 MEB | http://www.cs.utah.edu/~jeffp

Class Meetings: Mondays and Wednesdays, 3:00pm – 4:20pm, WEB L104.

Course Web Page: http://www.cs.utah.edu/~jeffp/teaching/cs5140.html

Data mining is the study of efficiently finding structures and patterns in large data sets. We will focus on several aspects of this: (1) converting from a messy and noisy raw data set to a structured and abstract one, (2) applying scalable and probabilistic algorithms to these well-structured abstract data sets, and (3) formally modeling and understanding the error and other consequences of parts (1) and (2), including choice of data representation and trade-offs between accuracy and scalability. These steps are essential for training as a data scientist.

Algorithms, probability, and linear algebra are required mathematical tools for understanding these approaches.

Topics will include: similarity search, clustering, regression/dimensionality reduction, graph analysis, PageRank, and small space summaries. We will also cover several recent developments, and the application of these topics to modern applications, often relating to large internet-based companies.

Upon completion, students should be able to read, understand, and implement many data mining research papers.

Getting Help

Take advantage of the instructor and TA office hours (posted on course web page). We will work hard to be accessible to students. Please send us email if you need to meet outside of office hours. Don't be shy if you don't understand something: come to office hours, send email, or speak up in class!

Students are encouraged to use a discussion group for additional questions outside of class and office hours. The class will rely on the Canvas discussion group. Feel free to post questions regarding any questions related to class: homeworks, schedule, material covered in class. Also feel free to answer questions, the instructors and TAs will also actively be answering questions. But, **do not post potential homework**

www.cs.utah.edu/~jeffp/teaching/cs5140.html

Data Mining

Instructor : Jeff Phillips (email) | Office hours: TBA @ MEB 3442 (and directly after class in WEB L1
TAs: WaiMing Tai (email) | Office hours: 11am-noon Friday, location TBA
+ Deb Paul (email) | Office Hours: 3-4pm Thursday, location TBA
+ Yang Gao (email) | Office Hours: 9-11am Tuesdays, location TBA
+ Shweta Singhal (email) | Office Hours: TBA
others TBA
Spring 2017 | Mondays, Wednesdays 3:00 pm - 4:20 pm
WEB L104
Catalog number: CS 5140 01 or CS 6140 01

Syllabus

Description:
Data mining is the study of efficiently finding structures and patterns in large data sets. We will focus on se
and abstract one, (2) applying scalable and probabilistic algorithms to these well-structured abstract data s
parts (1) and (2), including choice of data representation and trade-offs between accuracy and scalability. T
Algorithms, probability, and linear algebra are required mathematical tools for understanding these approa
Topics will include: similarity search, clustering, regression/dimensionality reduction, graph analysis, PageI
the application of these topics to modern applications, often relating to large internet-based companies.
Upon completion, students should be able to read, understand, and implement ideas from many data mini

Books:
We will in general not follow any book. My own course notes (linked below) serve as the defacto book. Hov
good overlap with the course.
MMDS(v1.3): Mining Massive Data Sets by Anand Rajaraman, Jure Leskovec, and Jeff Ullman. The digita
FoDS: Foundations of Data Science by Avrim Blum, John Hopcroft and Ravindran Kannan. This provide s

Videos: We plan to videotape all lectures, and make them available online. They will appear on this playlis

www.cs.utah.edu/~jeffp/teaching/cs5140.html

Date	Topic (+ Notes)	Video	Link	Assignment (latex)	Project
Mon 1.09	Class Overview	Vid	MMDS 1.1		
Wed 1.11	Statistics Principles + Chernoff Bounds	Vid	MMDS 1.2		
Mon 1.16	(MLK Day - No Class)				
Wed 1.18	Similarity : Jaccard + k-Grams	Vid	MMDS 3.1 + 3.2 | FoDS 7.3		
Mon 1.23	Similarity : Min Hashing	Vid	MMDS 3.3		
Wed 1.25	Similarity : LSH	Vid	MMDS 3.4	Statistical Principles	
Mon 1.30	Similarity : Distances	Vid	MMDS 3.5 + 7.1 | FoDS 8.1		Proposal
Wed 2.01	Similarity : SIFT and ANN vs. LSH	Vid	MMDS 3.7 + 7.1.3		
Mon 2.06	Clustering : Hierarchical	Vid	MMDS 7.2 | FoDS 8.7		
Wed 2.08	Clustering : K-Means	Vid	MMDS 7.3 | FoDS 8.3		
Mon 2.13	Clustering : Spectral (S)	Vid	MMDS 10.4 | FoDS 8.4 | Speilman | Gleich	Document Hash	
Wed 2.15	Streaming : Misra-Greis and Frugal	Vid	MMDS 4.1 | FoDS 7.1.3 | Min-Count Sketch | Misra-Gries		
Mon 2.20	(Presidents Day - No Class)				
Wed 2.22	Streaming : Count-Min + Apriori Algorithm	Vid	MMDS 6+4.3 | Careful Bloom Filter Analysis		Data Collection Report
Mon 2.27	Regression : Basics in 2-dimensions	Vid	ESL 3.2 and 3.4		
Wed 3.01	Regression : SVD + PCA	Vid	Geometry of SVD - Chap 3 | FoDS 4	Clustering	

Lecture Notes

4 Min Hashing

Last time we saw how to convert documents into sets. Then we discussed how to compare sets, specifically using the Jaccard similarity. Specifically, for two sets $A = \{0,1,2,5,6\}$ and $B = \{0,2,3,5,7,9\}$. The *Jaccard similarity* is defined

$$\begin{aligned}\mathsf{JS}(A,B) &= \frac{|A \cap B|}{|A \cup B|} \\ &= \frac{|\{0,2,5\}|}{|\{0,1,2,3,5,6,7,9\}|} = \frac{3}{8} = 0.375.\end{aligned}$$

Although this gives us a single numeric score to compare similarity (or distance) it is not easy to compute, and will be especially cumbersome if the sets are quite large.

This leads us to a technique called *min hashing* that uses a randomized algorithm to quickly estimate the Jaccard similarity. Furthermore, we can show how accurate it is through the Chernoff-Hoeffding bound.

To achieve these results we consider a new *abstract data type*, a matrix. This format is incredible useful conceptually, but often extremely wasteful if full written out.

4.1 Matrix Representation

Here we see how to convert a series of sets (e.g. a set of sets) to be represented as a single matrix. Consider sets:

$$\begin{aligned}S_1 &= \{1,2,5\} \\ S_2 &= \{3\} \\ S_3 &= \{2,3,4,6\} \\ S_4 &= \{1,4,6\}\end{aligned}$$

For instance $\mathsf{JS}(S_1, S_3) = |\{2\}|/|\{1,2,3,4,5,6\}| = 1/6$.

We can represent these four sets as a single matrix

Element	S_1	S_2	S_3	S_4
1	1	0	0	1
2	1	0	1	0
3	0	1	1	0
4	0	0	1	1
5	1	0	0	0
6	0	0	1	1

represents matrix $M = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$.

That element in the ith row and the jth column determine if element i is in set S_j. It is 1 if the element is in the set, and 0 otherwise. This captures *exactly* the same data set as the set representation, but may take much more space. If the matrix is *sparse*, meaning that most entries (e.g. $> 90\%$ or maybe $> 99\%$ ... or more conceptually, as the matrix becomes $r \times c$ the non-zero entries grows as roughly $r + c$, but the space grows as $r \cdot c$) then it wastes a lot of space. But still it is very useful *to think about*. There are also sparse matrix representations built into many languages such as Matlab which do not store all of the 0s, they just store the locations of the non-zeros.

4.2 Min Hashing

The next approach, called *min hashing*, initially seems even simpler than the clustering approach. It will need to evolve through several steps to become a useful trick.

Step 1: Randomly permute the items (by permuting the rows of the matrix).

Element	S_1	S_2	S_3	S_4
2	1	0	1	0
5	1	0	0	0
6	0	0	1	1
1	1	0	0	1
4	0	0	1	1
3	0	1	1	0

Step 2: Record the first 1 in each column, using a map function m. That is, given a permutation, applied to a set S, the function $m(S)$ records the element from S which appears earliest in this permutation.

$$\begin{aligned}m(S_1) &= 2 \\ m(S_2) &= 3 \\ m(S_3) &= 2 \\ m(S_4) &= 6\end{aligned}$$

Step 3: Estimate the Jaccard similarity $\mathsf{JS}(S_i, S_j)$ as

$$\hat{\mathsf{JS}}(S_i, S_j) = \begin{cases} 1 & m(S_i) = m(S_j) \\ 0 & \text{otherwise.} \end{cases}$$

Lemma 4.2.1. $\mathbf{Pr}[m(S_i) = m(S_j)] = \mathbf{E}[\hat{\mathsf{JS}}(S_i, S_j)] = \mathsf{JS}(S_i, S_j)$.

Proof. There are three types of rows.

(Tx) There are x rows with 1 in both column

(Ty) There are y rows with 1 in one column and 0 in the other

(Tz) There are z rows with 0 in both column

The total number of rows is $x + y + z$. The Jaccard similarity is precisely $\mathsf{JS}(S_i, S_j) = x/(x+y)$. (Note that usually $z \gg x, y$ (mostly empty) and we can ignore these.)

Let row r be the $\min\{m(S_i), m(S_j)\}$. It is either type (Tx) or (Ty), and it is (Tx) with probability exactly $x/(x+y)$, since the permutation is random. This is the only case that $m(S_i) = m(S_j)$, otherwise S_i or S_j has 1, but not both. □

Thus this approach only gives 0 or 1, but has the right expectation. To get a better estimate, we need to repeat this several (k) times. Consider k random permutations $\{m_1, m_2, \ldots, m_k\}$ and also k random variables $\{X_1, X_2, \ldots, X_k\}$ where

$$X_\ell = \begin{cases} 1 & \text{if } m_\ell(S_i) = m_\ell(S_j) \\ 0 & \text{otherwise.} \end{cases}$$

Now we can estimate $\mathsf{JS}(S_i, S_j)$ as $\hat{\mathsf{JS}}_k(S_i, S_j) = \frac{1}{k}\sum_{\ell=1}^{k} X_\ell$, the average of the k simple random estimates.

So how large should we set k so that this gives us an accurate measure? Since it is a randomized algorithm, we will have an error tolerance $\varepsilon \in (0,1)$ (e.g. we want $|\mathsf{JS}(S_i,S_j) - \hat{\mathsf{JS}}_k(S_i,S_j)| \leq \varepsilon$), and a probability of failure δ (e.g. the probability we have more than ε error). We will now use Theorem 2.4.2 where $M = \sum_{\ell=1}^{k} X_\ell$ and hence $\mathbf{E}[M] = k \cdot \mathsf{JS}(S_i, S_j)$. We have $0 \leq X_i \leq 1$ so each $\Delta_i = 1$. Now we can write for some value α:

$$\begin{aligned}\mathbf{Pr}[|\hat{\mathsf{JS}}_k(S_i,S_j) - \mathsf{JS}(S_i,S_j)| \geq \alpha/k] &= \mathbf{Pr}[|k \cdot \hat{\mathsf{JS}}_k(S_i,S_j) - k \cdot \mathsf{JS}(S_i,S_j)| \geq \alpha] \\ &= \mathbf{Pr}[|M - \mathbf{E}[M]| \geq \alpha] \leq 2\exp\left(\frac{-2\alpha^2}{\sum_{i=1}^{k}\Delta_i^2}\right) = 2\exp(-2\alpha^2/k).\end{aligned}$$

Setting $\alpha = \varepsilon k$ and $k = (1/(2\varepsilon^2))\ln(2/\delta)$ we obtain

$$\mathbf{Pr}[|\hat{\mathsf{JS}}_k(S_i,S_j) - \mathsf{JS}(S_i,S_j)| \geq \varepsilon] \leq 2\exp(-2(\varepsilon^2 k^2)/k) = 2\exp(-2\varepsilon^2 \frac{1}{2\varepsilon^2}\ln(2/\delta)) = \delta.$$

Or in other words, if we set $k = (1/2\varepsilon^2)\ln(2/\delta)$, then the probability that our estimate $\hat{\mathsf{JS}}_k(S_i,S_j)$ is within ε of $\mathsf{JS}(S_i,S_j)$ is at least $1 - \delta$.

Say for instance we want error *at most* $\varepsilon = 0.05$ and can tolerate a failure 1% of the time ($\delta = 0.01$), then we need $k = (1/(2 \cdot 0.05^2))\ln(2/0.01) = 200\ln(200) \approx 1060$. Note that the modeling error of converting a structure into a set may be more than $\varepsilon = 0.05$, so this should be an acceptable loss in accuracy.

Top k. It is sometimes more efficient to use the top-k (for some small number $k > 1$) hash values for each hash function, than just the top one. For instance, see Cohen and Kaplan (*Summarizing Data using Bottom-k Sketches*, PODC 2007). This approach requires a bit more intricate analysis, as well as a bit more careful implementation.

4.2.1 Fast Min Hashing Algorithm

This is still too slow. We need to construct the full matrix, and we need to permute it k times. A faster way is the *min hash algorithm*.

Make one pass over the data. Let $n = |\mathcal{E}|$. Maintain k random hash functions $\{h_1, h_2, \ldots, h_k\}$ chosen from a hash family at random so $h_i : \mathcal{E} \to [n]$ (one can use a larger range $n' > n$ where $n' = 2^t$ is a power of two). An initialize k values at $\{v_1, v_2, \ldots, v_k\}$ so $v_i = \infty$.

Algorithm 4.2.1 Min Hash on set S

```
for i ∈ S do
  for j = 1 to k do
    if (h_j(i) < v_j) then
      v_j ← h_j(i)
```

On output $m_j(S) = v_j$. The algorithm runs in $|S|k$ steps, for a set S of size $|S|$. Note this is independent of the size n of all possible elements $\mathcal{E}$. And the output space of a single set is only $k = (1/2\varepsilon^2)\ln(2/\delta)$ which is independent of the size of the original set. The space for N sets is only $O(Nk)$.

Finally, we can now estimate $\mathsf{JS}(S, S')$ for two sets S and S' as

$$\mathsf{JS}_k(S, S') = \frac{1}{k}\sum_{j=1}^{k} \mathbf{1}(m_j(S) = m_j(S'))$$

where $\mathbf{1}(\gamma) = 1$ if $\gamma = \textsc{True}$ and 0 otherwise. This only takes $O(k)$ time, again independent of n or $|S|$ and $|S'|$.

www.cs.utah.edu/~jeffp/teaching/cs5140.html

Mon 3.06	Regression : Matrix Sketching	Vid	MMDS 9.4 | FoDS 2.7 + 7.2.2 | arXiv		
Wed 3.08	**MIDTERM TEST**				
Mon 3.13	(Spring Break - No Class)				
Wed 3.15	(Spring Break - No Class)				
Mon 3.20	Regression : Random Projections	Vid	FoDS 2.9		Intermediate Report
Wed 3.22	Regression : Compressed Sensing and OMP	Vid	FoDS 10.3 | Tropp + Gilbert	Frequent	
Mon 3.27	Regression : L1 Regression and Lasso	Vid	Davenport | ESL 3.8 | bias-variance example		
Wed 3.29	Noise : Noise in Data	Vid	MMDS 9.1 | Tutorial		
Mon 4.03	Noise : Privacy	Vid	McSherry | Dwork		
Wed 4.05	Graph Analysis : Markov Chains (S)	Vid	MMDS 10.1 + 5.1 | FoDS 5 | Weckesser notes		
Mon 4.10	Graph Analysis : PageRank	Vid	MMDS 5.1 + 5.4	Regression	
Wed 4.12	Graph Analysis : MapReduce	Vid	MMDS 2 |		
Mon 4.17	Graph Analysis : Communities	Vid	MMDS 10.2 + 5.5 | FoDS 8.8 + 3.4		Final Report
Wed 4.19	Graph Analysis : Graph Sparsification	Vid1,2	MMDS 4.1		Poster Outline
Mon 4.24	**ENDTERM TEST**				
Mon 5.01				Graphs	
Tue 5.02	Poster Day !!! (3:30-5:30pm)				Poster Presentation

Project

Project*

Final Report Due: Monday, April 17
Turn in report by 2:45pm (through Canvas).

1 Overview

Your project will consist of five elements.

- Project Proposal : Due January 30
- Data Collection Report : Due February 22
- Intermediate Report : Due March 20
- Final Report : Due April 17
- Poster Presentation : May 2 | (3:30pm - 5:30pm or 6:00pm)

As in any research in order to get people to pay attention, you will need to be able to present your work efficiently in written and oral form.

You may work in teams of 2 or 3, but the amount of work you perform will need to scale accordingly. Teams of size 1 might be allowed under unusual circumstances with special permission from the instructor. All students will need to have clearly defined roles as demonstrated in the final report and presentation. I highly recommend groups of size 3. Although the project work will scale with students, the administrative parts will remain constant, so having a large group will make it easier for you.

Note that some topics will not be covered before many elements of the project are due. I realize this is not ideal. However, typically, most work on a project is crammed in the last week or two of the semester, which is also not ideal. In the past this has lead to much stronger projects without considerably more work required.

Example Posters

Station Evaluation and Time-Series Curve Matching for Meteorological Observation

Yan Zheng

Introduction

A meteorological observation at a given place can be inaccurate for a variety of reasons. Quality control can help spot which meteorological observations are inaccurate.

The project data is mainly from MesoWest group of Atmosphere Science Department, which are the results of UU2DVAR analysis(bias, Impact) for clustering and weather observations from 100 stations of six-year data for curve matching .

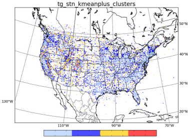
Key Idea

Based on long-term statistical information with widely neighbor stations and the pattern of a specific day of a station, QC methods are explored to distinguish high impact stations using clustering algorithm and to find a weather pattern by time-series curve matching using nearest neighbor search based LSH algorithm. Euclidean distance is used to measure the distance of two curves.

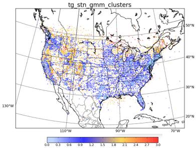
Clustering

K-Mean++ and Gaussian mixture modeling clustering algorithm have been applied and the cluster index is used as the score to evaluate the quality of a station.

Result of k-mean++ clustering



Result of Gaussian Mixture Modeling

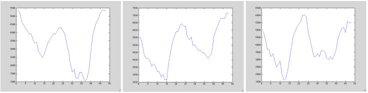


Curve Matching

LSH family: Pick a random projection of R^d onto a 1-dimensional line and chop the line into segments of length w, shifted by a random value b∈[0,w).

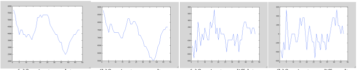
Choose L functions g_j, $j=1 \ldots L$, by setting $g_j=(h_{1,j}, h_{2,j}, \ldots h_{k,j})$, where $h_{1,j}, h_{2,j} \ldots h_{k,j}$ are chosen at random from the LSH family, H. Then construct L hash tables.

Tempreture difference data quering



(a) Quering data (b)Quering result 1 (c) Quering result 2

Tempreture difference data quering



(a)Quering temp data (b)Quering temp result (c)Quering temp diff data (b)Quering temp diff result

Conclusion

- Understanding the data, key to data mining
- Finding the right algorithm, need to explore many options.
- Correctly use the data, do experiments and compare the results.

CS 6955 Data Mining; Spring 2012

Instructor: Jeff M. Phillips, University of Utah

Data Mining

What is Data Mining?

Data Mining

What is Data Mining?

- Finding structure in data?
- Machine learning on large data?
- Unsupervised learning?
- Large scale computational statistics?

Data Mining

What is Data Mining?

- Finding structure in data?
- Machine learning on large data?
- Unsupervised learning?
- Large scale computational statistics?

- How to think about data analytics.

Data Mining

What is Data Mining?

- Finding structure in data?
- Machine learning on large data?
- Unsupervised learning?
- Large scale computational statistics?

- How to think about data analytics.

- *Principals* of converting from messy raw data to abstract representations.
- Algorithms of how to analyze data in abstract representations.
- Addressing challenges in scalability, error, and modeling.

Methods for Data Analytics

Machine Learning (CS 5350/6350)

- Classification: Given labeled data $\ell(x) \in \{\textsc{True} \text{ or } \textsc{False}\}$, build model so given new data, you can guess a label.
- More continuous optimization (DM more discrete)

Methods for Data Analytics

Machine Learning (CS 5350/6350)

- Classification: Given labeled data $\ell(x) \in \{\textsc{True} \text{ or } \textsc{False}\}$, build model so given new data, you can guess a label.
- More continuous optimization (DM more discrete)

Artificial Intelligence (CS 4300 / CS 6300)

- Interaction with World/Data: Observe, Learn, Act; repeat.

Methods for Data Analytics

Machine Learning (CS 5350/6350)

- Classification: Given labeled data $\ell(x) \in \{\textsc{True} \text{ or } \textsc{False}\}$, build model so given new data, you can guess a label.
- More continuous optimization (DM more discrete)

Artificial Intelligence (CS 4300 / CS 6300)

- Interaction with World/Data: Observe, Learn, Act; repeat.

More advanced Topics:

- Probabilistic Learning
- Structured Prediction
- Natural Language Processing
- Clustering

Methods for Data Analytics

Machine Learning (CS 5350/6350)

- Classification: Given labeled data $\ell(x) \in \{\textsc{True} \text{ or } \textsc{False}\}$, build model so given new data, you can guess a label.
- More continuous optimization (DM more discrete)

Artificial Intelligence (CS 4300 / CS 6300)

- Interaction with World/Data: Observe, Learn, Act; repeat.

More advanced Topics:

- Probabilistic Learning
- Structured Prediction
- Natural Language Processing
- Clustering

Data Mining has some ($< 10\%$) overlap with each of these.

Modeling versus Efficiency

Two Intertwined (and often competing) Objectives:

- Model Data Correctly
- Process Data Efficiently

Other Data Mining Courses

Every university teaches data mining differently!

Other Data Mining Courses

Every university teaches data mining differently!

What flavor is offered in this class:

- Focus on techniques for *very* large scale data
- Broad coverage ... with recent developments
- Formally and generally presented (proof sketches)
- ... but useful in practice (e.g. internet companies)
- Probabilistic algorithms: connections to CS and Stat

Other Data Mining Courses

Every university teaches data mining differently!

What flavor is offered in this class:

- Focus on techniques for *very* large scale data
- Broad coverage ... with recent developments
- Formally and generally presented (proof sketches)
- ... but useful in practice (e.g. internet companies)
- Probabilistic algorithms: connections to CS and Stat
- *no specific software tools / programming languages*

Other Data Mining Courses

Every university teaches data mining differently!

What flavor is offered in this class:

- Focus on techniques for *very* large scale data
- Broad coverage ... with recent developments
- Formally and generally presented (proof sketches)
- ... but useful in practice (e.g. internet companies)
- Probabilistic algorithms: connections to CS and Stat
- *no specific software tools / programming languages*

Maths: Linear Algebra, Probability, High-dimensional geometry

Outline

Statistical Principals:

- 1. **Hashing, Concentration of Measure**

Data and Distances:

- 2. **Similarity** (find duplicates and similar items)
- 3. **Clustering** (aggregate close items)

Structure in Data:

- 3. **Clustering** (aggregate close items)
- 4. **Regression** (linearity of (high-d) data)
- 5. **Noisy Data** (anomalies in data)

Controlling for Noise and Uncertainty:

- 5. **Noisy Data** (anomalies in data)
- 6. **Link Analysis** (prominent structure in large graphs)

Statistical Principals

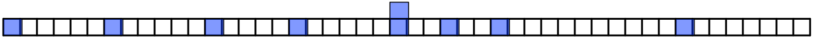
What happens as data is generated with replacement
{IP addresses, words in dictionary, edges in graph, hash table}

- When do items collide?
- When do you see all items?
- When is the distribution almost uniform?

Statistical Principals

What happens as data is generated with replacement
{IP addresses, words in dictionary, edges in graph, hash table}

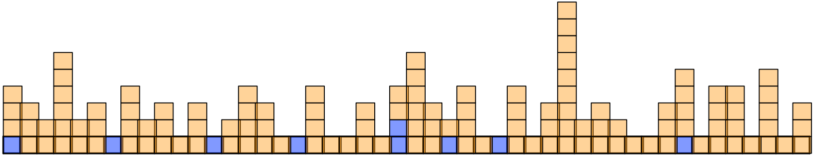
- When do items collide?
- When do you see all items?
- When is the distribution almost uniform?



Statistical Principals

What happens as data is generated with replacement
{IP addresses, words in dictionary, edges in graph, hash table}

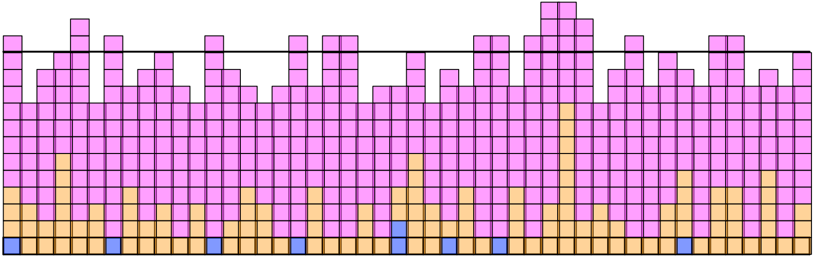
- When do items collide?
- When do you see all items?
- When is the distribution almost uniform?



Statistical Principals

What happens as data is generated with replacement
{IP addresses, words in dictionary, edges in graph, hash table}

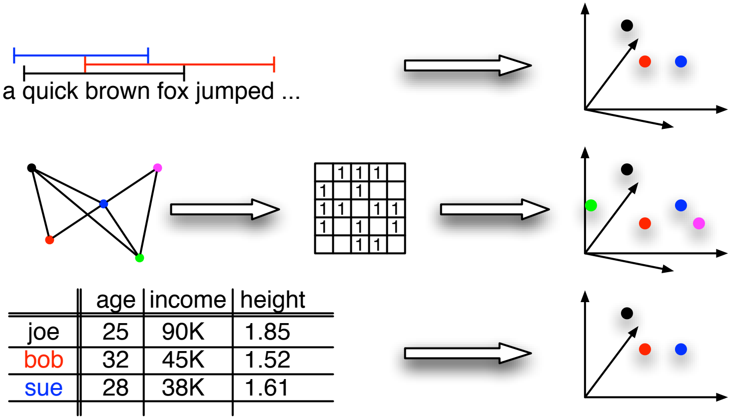
- When do items collide?
- When do you see all items?
- When is the distribution almost uniform?



Raw Data to Abstract Representations

How to measure similarity between data?

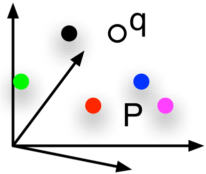
Key idea: data $\rightarrow$ point



Similarity

Given a large set of data P.
Given new point q, is q in P?

Given a large set of data P.
Given new point q, what is the *closest* point in P to q?



Clustering

How to find groups of similar data.

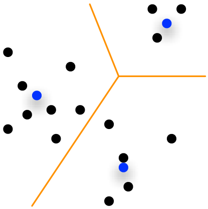
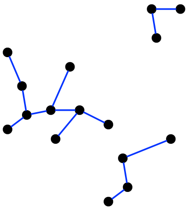
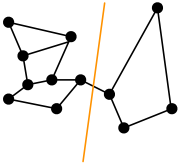
- do we need a representative?
- can groups overlap?
- what is structure of data/distance?

Clustering

How to find groups of similar data.

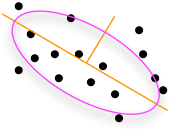
- do we need a representative?
- can groups overlap?
- what is structure of data/distance?

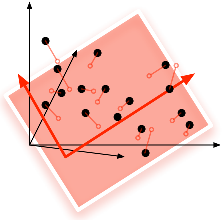
- **Hierarchical clustering** : When to combine groups?
- ***k*-means clustering** : k-median, k-center, k-means++
- **Graph clustering** : modularity, spectral

Regression

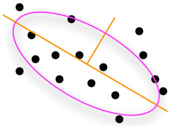
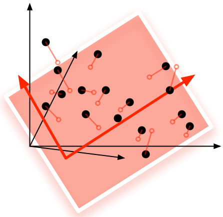
Consider a data set $P \in \mathbb{R}^d$, where d is BIG!
Want to find representation of P in some $\mathbb{R}^k$
$\mu(P) \to Q \in \mathbb{R}^k$ so $\|p_i - p_j\| \approx \|q_i - q_j\|$
$Q \in \mathbb{R}^k$ should capture most data in P





Regression

Consider a data set $P \in \mathbb{R}^d$, where d is BIG!
Want to find representation of P in some $\mathbb{R}^k$
$\mu(P) \to Q \in \mathbb{R}^k$ so $\|p_i - p_j\| \approx \|q_i - q_j\|$
$Q \in \mathbb{R}^k$ should capture most data in P

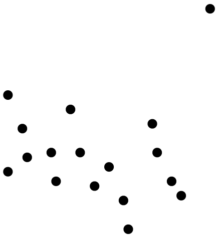



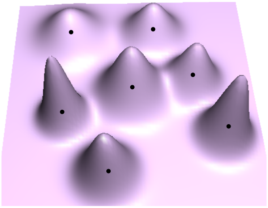
- L_2 **Regression + PCA** : Common easy approach
- **Multidimensional Scaling** : Fits in $\mathbb{R}^k$ with k small
- **Matrix Sketching**: Random Projections, Sampling, FD
- L_1 **Regression** : "Better", Orthogonal Matching Pursuit
- **Info Recovery** : Compressed Sensing

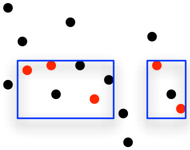
Noisy Data

What to do when data is noisy?

- **Identify it** : Find and remove outliers
- **Model it** : It may be real, affect answer
- **Exploit it** : Differential privacy *(ethics in data)*



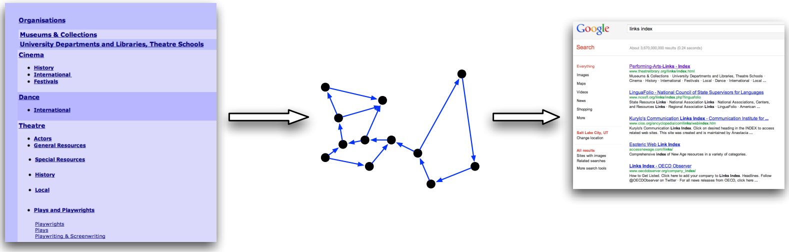




Link Analysis

How does Google Search work?
Converts webpage links into directed graph.

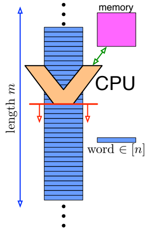
- **Markov Chains** : Models movement in a graph
- **PageRank** : How to convert graph into important nodes
- **MapReduce** : How to scale up PageRank
- **Communities** : Other important nodes in graphs

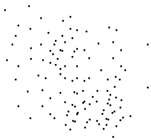


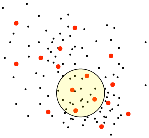
Summaries

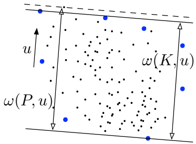
Reducing *massive* data to small space.
Want to retain as much as possible (not specific structure)
error guarantees

- **OnePass Sampling** : Reservoir Sampling
- **MinCount Hash** : Sketching data, $\rightarrow$ abstract features
- **Density Approximation** : Quantiles
- **Matrix Sketching** : Preprocessing complex data
- **Spanners** : graph approximations









Themes

What are course goals?

- Intuition for data analytics
- How to model data (convert to abstract data types)
- How to process data efficiently (balance models with algorithms)

Themes

What are course goals?

- Intuition for data analytics
- How to model data (convert to abstract data types)
- How to process data efficiently (balance models with algorithms)

Work Plan:

- 2-3 weeks each topic.
  - Overview classic techniques
  - Focus on modeling / efficiency tradeoff
  - Special topics
  - Short homework for each (analysis + with data) **(45% grade)**
- 2 Tests **(10% grade)**
- Course Project **(45% grade)**.
  - Focus on specific data set
  - Deep exploration with technique
  - Ongoing refinement of presentation + approach

On Homeworks

Managed through Canvas (should be up)

- No restriction on programming language.
- Some designed for matlab, others better in python or C++.
- Programming assignments with not too many specifications.
- Bonus Questions!

On Canvas

Class management communication through Canvas

- All homework turn ins (typically as pdfs).
- Grades assigned
- Announcements
- Discussion (emails to instructor may not be responded)
  no posting potential solutions

Videos

Class will be video-recorded and live-streamed.

- `https://www.youtube.com/channel/UCDUS8ObdunpmvWVPyFRPqFQ`
- links off of webpage to live stream and playlist
- Experiment with Luum.io

Videos

Class will be video-recorded and live-streamed.

- `https://www.youtube.com/channel/UCDUS8Obdunpmv WVPyFRPqFQ`
- links off of webpage to live stream and playlist
- Experiment with Luum.io

Come to class if you can.

- Easier to ask questions, interact
  (mechanism through video, with delay)
- Talk to me before/after class!
- Attendance required for MIDTERM, FINAL, Poster Day
- Help your grade, and understanding.

Data Group

Data Group Meeting

Thursdays @ 12:15-1:30 in MEB 3147 (LCR)

CS 7941 *Data Reading Group*

requires one presentation if taken for credit

`http://datagroup.cs.utah.edu`

Utah Data Science Day

UTAH DATA SCIENCE DAY 2017

`http://datascience.utah.edu/dataday`

Friday, January 13 : 11:30 - 6pm Union Ballroom

11:30 AM - 1:00 PM	Data Science Job Fair
1:00 PM - 1:10 PM	Welcome: Data Science at Utah
1:10 PM - 2:00 PM	Panel: Data Science in Industry
2:00 PM - 3:30 PM	Posters and Demos
3:30 PM - 4:50 PM	Data Science + X Talks
5:00 PM - 6:00 PM	Keynote
6:00 PM - 6:15 PM	Poster Awards !!