Tackling The Challenges of Big Data

Big Data Analytics

Machine Learning Tools

Tommi Jaakkola

Professor

Massachusetts Institute of Technology

Tackling The Challenges of Big Data

Big Data Analytics

Machine Learning Tools

Introduction

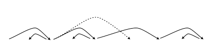
Tommi Jaakkola

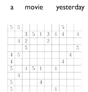
Professor

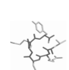
Massachusetts Institute of Technology

Our Research Group

- **Our research focuses on machine learning, from theory, algorithms, to applications**
- **There are several problems that drive our machine learning research**
  - natural language processing (e.g., parsing)
  - recommender systems (e.g., sparsity, scaling, privacy)
  - predictive user modeling (e.g., mobile)
  - uncovering how biological systems work (e.g., reconstructing transcriptional control)
  - large scale inverse problems (e.g., reservoir modeling)







Machine Learning

- **Machine learning is about forecasting**
- **Machine learning methods are computer programs that learn to predict based on data**

Machine Learning

- **Machine learning is about forecasting**
- **Machine learning methods are computer programs that learn to predict based on data**
  - modern engineering problems are hard to specify, solve directly (e.g., detecting fraudulent transactions)

Machine Learning

- **Machine learning is about forecasting**
- **Machine learning methods are computer programs that learn to predict based on data**
  - modern engineering problems are hard to specify, solve directly (e.g., detecting fraudulent transactions)
  - but it is often easy to provide examples of how the system should work (e.g., examples of fraudulent/normal transactions)

CC transaction	Fraudulent?
description 1	yes
description 2	no
. . .	. . .

Machine Learning

- **Machine learning is about forecasting**
- **Machine learning methods are computer programs that learn to predict based on data**
  - modern engineering problems are hard to specify, solve directly (e.g., detecting fraudulent transactions)
  - but it is often easy to provide examples of how the system should work (e.g., examples of fraudulent/normal transactions)

CC transaction	Fraudulent?
description 1	yes
description 2	no
. . .	. . .

  - machine learning methods learn to map descriptions to predictions based on such example data

Machine Learning

- **The mapping from examples (e.g., descriptions of transactions) to labels (e.g., fraudulent or not) is known as a classifier**

$$\text{transaction as features} \quad \begin{bmatrix} \text{time} \\ \text{location} \\ \text{customer} \\ \text{company} \\ \text{history} \\ \cdots \end{bmatrix} \xrightarrow[\text{parameterized classifier}]{} \text{prediction } (+1/-1)$$

Machine Learning

- **The mapping from examples (e.g., descriptions of transactions) to labels (e.g., fraudulent or not) is known as a classifier**

$$\text{transaction as features} \quad \begin{bmatrix} \text{time} \\ \text{location} \\ \text{customer} \\ \text{company} \\ \text{history} \\ \cdots \end{bmatrix} \xrightarrow[\text{parameterized classifier}]{} \text{prediction } (+1/-1)$$

- **Simple classification problems are everywhere**
  - classifying news articles, images, reviews, etc.
  - classifying biomedical samples, measurements, etc.
  - mapping genotype (SNP) signatures to phenotypes
  - predicting the success of financial strategies, etc.

Machine Learning

- **The mapping from examples (e.g., descriptions of transactions) to labels (e.g., fraudulent or not) is known as a classifier**

- **Lots of scalable on-line algorithms are available for learning classifiers from data**

Beyond (simple) Classification

- **We can extend classifiers to predict more complex objects, not just labels**
  - e.g., annotate genomes (genes, their control)
  - e.g., transcribe speech
  - e.g., map natural language sentences to their syntactic parses
- **How do such methods scale?**

Beyond (simple) Classification

- **We may have lots of data, much of it incomplete, fragmented, potentially erroneous**
- **How do we share, distill such data to obtain more accurate predictions?**

Beyond (simple) Classification

- **We may have lots of data, much of it incomplete, fragmented, potentially erroneous**
- **How do we share, distill such data to obtain more accurate predictions?**
  - e.g., in recommender problems, little data may be available about any particular user (e.g., amazon.com visitor) but there are lots of such users
  - the question is how we can leverage other users' experiences to better predict the behavior of any particular user?

Tackling The Challenges of Big Data

Big Data Analytics

Machine Learning Tools

Introduction

THANK YOU

PROFESSIONAL EDUCATION

CSAIL

Tackling The Challenges of Big Data

Big Data Analytics

Machine Learning Tools

Tommi Jaakkola

Professor

Massachusetts Institute of Technology

PROFESSIONAL EDUCATION

CSAIL

Tackling The Challenges of Big Data

Big Data Analytics

Machine Learning Tools

Example: Scaling Structured Prediction

Tommi Jaakkola

Professor

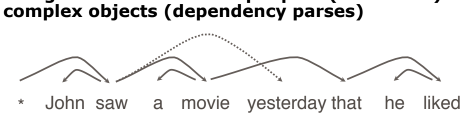
Massachusetts Institute of Technology

Structured Prediction

- **Natural language processing**
  - e.g., tagging, morphological segmentation, parsing
- **Computer vision**
  - e.g., segmentation, stereo reconstruction, object recognition
- **Computational biology**
  - e.g., annotation, molecular structures, pathway reconstruction
- **Robotics**
  - e.g., imitation learning, inverse kinematics
- **Human computer interaction**
  - e.g., interface alignment, example based design

Structured Prediction: Example

- **The goal is to learn to map inputs (sentences) to complex objects (dependency parses)**



  - in dependency parsing, we draw an arc from the head word of each phrase to words that modify it
  - the parse is a directed tree over the words. In many languages, the tree is non-projective (crossing arcs)
  - each sentence is mapped to arc scores; the parse is obtained as the highest scoring directed tree

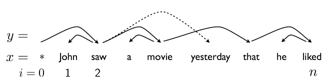
Tackling the Challenges of Big Data
© 2014 Massachusetts Institute of Technology

Scaling Structured Prediction

- **Accurate predictions of complex structures (e.g., dependency trees) require rich scoring functions**
- **Refined predictions, on the other hand, consume resources**
- **Three scaling problems we must address:**
  - Prediction (inference): finding a single highest scoring tree (a single prediction) can be already provably hard
  - Estimation: learning requires inference making it challenging to estimate models from large corpuses
  - Uncertainty: modeling uncertainty is substantially harder than solving for the highest scoring tree

The Prediction Problem

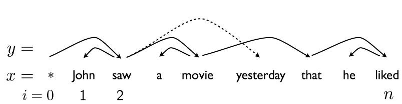
- **The goal is to learn to map sentences (x) to dependency parses (y)**



- **The mapping from x to y is typically decomposed into two parts:**
  - modeling: using sentence x to specify scores for candidate trees and/or their parts
  - computation: the predicted parse is obtained as the highest scoring tree
- **A simple way to score trees is by summing scores for individual arcs ("arc factored scoring")**

Prediction, Challenges

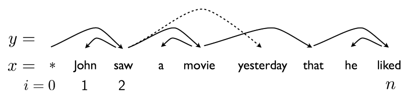
- **The goal is to learn to map sentences (x) to dependency parses (y)**



- **From the point of view of modeling, we would like to include scores for bundles of outgoing arcs ("siblings") instead of just individual arcs**

Prediction, Challenges

- The goal is to learn to map sentences (x) to dependency parses (y)



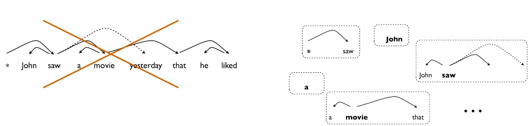
- From the point of view of modeling, we would like to include scores for bundles of outgoing arcs ("siblings") instead of just individual arcs
- But now the prediction problem -- parsing a single sentence -- is computationally hard!

Scaling Prediction (inference)

- We have to find an appropriate balance between accuracy (modeling power) and computation
- Using only the simplest models that scale without effort is limiting... can we do more?

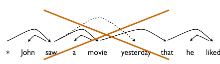
Scaling Prediction (inference)

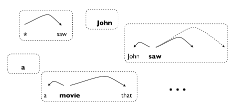
- We have to find an appropriate balance between accuracy (modeling power) and computation
- Using only the simplest models that scale without effort is limiting... can we do more?
- We can adaptively break (decompose) refined models into smaller loosely coupled pieces



Scaling Prediction (inference)

- **We have to find an appropriate balance between accuracy (modeling power) and computation**
- **Using only the simplest models that scale without effort is limiting… can we do more?**
- **We can be adaptively break (decompose) refined models into smaller loosely coupled pieces**





  - easy to solve individually
  - pieces encouraged to agree (adaptively)
  - typically results in the same prediction as the original model

Scaling Prediction (inference)

- **We have to find an appropriate balance between accuracy (modeling power) and computation**
- **Using only the simplest models that scale without effort is limiting… can we do more?**
- **We can be adaptively break (decompose) refined models into smaller loosely coupled pieces**
  - typically results in the same prediction as the original model

Dan	99.07
Dut	98.19
Por	99.65
Slo	90.55
Swe	98.71
Tur	98.72
Eng^1	98.65
Eng^2	98.96
Dan	98.50
Dut	98.00

Scaling Structured Prediction

- **Accurate predictions of complex structures (e.g., dependency trees) require rich scoring functions**
- **Refined predictions, on the other hand, consume resources**
- **Three scaling problems we must address:**
  - ✓ Prediction (inference): finding a single highest scoring tree (a single prediction) can be already provably hard
  - Estimation: learning requires inference making it challenging to estimate models from large corpuses
  - Uncertainty: modeling uncertainty is substantially harder than solving for the highest scoring tree

Tackling The Challenges of Big Data
Big Data Analytics
Machine Learning Tools
Example: Scaling Structured Prediction

THANK YOU

Tackling The Challenges of Big Data
Big Data Analytics
Machine Learning Tools

Tommi Jaakkola
Professor
Massachusetts Institute of Technology

Tackling The Challenges of Big Data
Big Data Analytics
Machine Learning Tools
Example: Collaborative Filtering

Tommi Jaakkola
Professor
Massachusetts Institute of Technology

Learning to Recommend

- **Many prediction problems can be viewed as matrix completion problems**
- **Co-occurrence data**
  - e.g., ratings, viewing, purchasing, actions
  - arranged into a large incomplete matrix

Movies, books (items)

Users

5	5						5		
		3	5	1	3	4	4		4
	4	2			2				
		5							5
4	5							4	
4							4		
5		4	5	1		4			
	4								
5				4					
5						4			
		5				5		3	

Learning to Recommend

- **Many prediction problems can be viewed as matrix completion problems**
- **Co-occurrence data**
  - e.g., ratings, viewing, purchasing, actions
  - arranged into a large incomplete matrix
- **Goal**
  - fill-in missing values in the data matrix

Movies, books (items)

Users

5	5						5		
		3	5	1	3	4	4		4
	4	2			2				
		5							5
4	5							4	
4							4		
5		4	5	1		4			
	4								
5				4					
5						4			
		5				5		3	

Learning to Recommend

- **Many prediction problems can be viewed as matrix completion problems**
- **Co-occurrence data**
  - e.g., ratings, viewing, purchasing, actions
  - arranged into a large incomplete matrix
- **Goal**
  - fill-in missing values in the data matrix
- **Collaborative prediction**
  - intuition: borrow experience from other similar users

Movies, books (items)

Users

5	5						5		
		3	5	1	3	4	4		4
	4	2			2				
		5							5
4	5							4	
4							4		
5		4	5	1		4			
	4								
5				4					
5						4			
		5				5		3	

Learning to Recommend

- **Many prediction problems can be viewed as matrix completion problems**
- **Co-occurrence data**
  - e.g., ratings, viewing, purchasing, actions
  - arranged into a large incomplete matrix
- **Goal**
  - fill-in missing values in the data matrix
- **Collaborative prediction**
  - intuition: borrow experience from other similar users

Movies, books (items)

Users

5	5						5		
		3	5	1	3	4	4		4
	4	2			2				
		5							5
4	5							4	
4							4		
5		4	5	1		4			
	4								
5				4					
5						4			
		5				5		3	

Learning to Recommend

- **Many prediction problems can be viewed as matrix completion problems**
- **Co-occurrence data**
  - e.g., ratings, viewing, purchasing, actions
  - arranged into a large incomplete matrix
- **Goal**
  - fill-in missing values in the data matrix
- **Collaborative prediction**
  - intuition: borrow experience from other similar users

Movies, books (items)

Users

5	5						5		
		3	5	1	3	4	4		4
	4	2			2				
		5							5
4	5							4	
4							4		
5		4	5	1		4			
	4								
5				4					
5			5			4			
		5				5		3	

Learning to Recommend

- **Many prediction problems can be viewed as matrix completion problems**
- **Scaling issues**
  - huge matrices (millions x tens of thousands)
- **Statistical issues**
  - largely unobserved (e.g., 1%), diverse
- **Modeling issues**
  - many interpretations for missing entries
- **Privacy issues**
  - how much info to release?

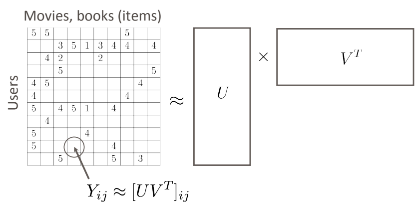
Movies, books (items)

Users

5	5						5		
		3	5	1	3	4	4		4
	4	2			2				
		5							5
4	5							4	
4							4		
5		4	5	1		4			
	4								
5				4					
5						4			
		5				5		3	

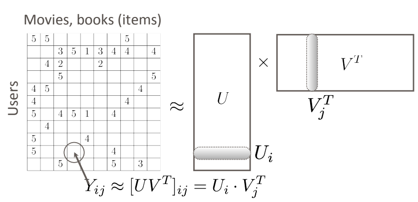
Typical Approach: Factorization

- **We reconstruct the data matrix by finding the "simplest" matrix (nearly) consistent with the limited observations**



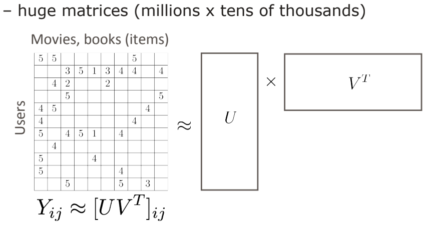
Typical Approach: Factorization

- **We reconstruct the data matrix by finding the "simplest" matrix (nearly) consistent with the limited observations**
- **In doing so, we identify "user features" and "item features", and predict by comparing the two**



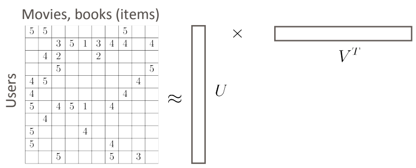
Typical Approach: Factorization

- **We identify "user features" and "item features", and predict responses by comparing the two**
- **Scaling issues**
  - huge matrices (millions x tens of thousands)



Typical Approach: Factorization

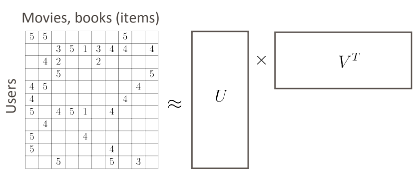
- **We identify "user features" and "item features", and predict responses by comparing the two**
- **Scaling issues**
  - huge matrices (millions x tens of thousands)
  - we can estimate each feature coordinate pair at a time



$$Y_{ij} \approx [UV^T]_{ij}$$

Typical Approach: Factorization

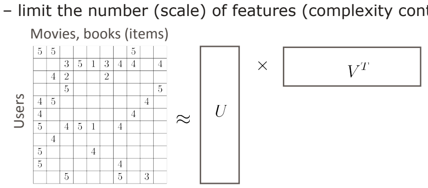
- **We identify "user features" and "item features", and predict responses by comparing the two**
- **Statistical issues**
  - largely unobserved (e.g., 1%), diverse



$$Y_{ij} \approx [UV^T]_{ij}$$

Typical Approach: Factorization

- **We identify "user features" and "item features", and predict responses by comparing the two**
- **Statistical issues**
  - largely unobserved (e.g., 1%), diverse
  - limit the number (scale) of features (complexity control)



$$Y_{ij} \approx [UV^T]_{ij}$$

Approach: factorization

- **We identify “user features” and “item features”, and predict responses by comparing the two**
- **Modeling issues**
  - many interpretations for missing entries
  - need to predict “selections” as well as responses (model selection bias)

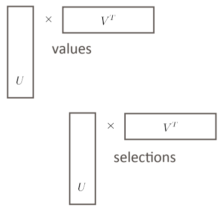
Movies, books (items)

Users

5	5						5		
		3	5	1	3	4	4		4
	4	2			2				
		5							5
4	5							4	
4							4		
5		4	5	1		4			
	4								
5				4					
5						4			
		5				5		3	

$$Y_{ij} \approx [UV^T]_{ij}$$

$\approx$



Approach: Factorization

- **We identify “user features” and “item features”, and predict responses by comparing the two**
- **Privacy issues**
  - how much info (each user needs) to release?
  - need to trade privacy for accuracy
  - users respond to queries about preferences rather than revealing ratings directly

Movies, books (items)

Users

5	5						5		
		3	5	1	3	4	4		4
	4	2			2				
		5							5
4	5							4	
4							4		
5		4	5	1		4			
	4								
5				4					
5						4			
		5				5		3	

$$Y_{ij} \approx [UV^T]_{ij}$$

query

.1	.3	−.2	.2	.6	−.3	−.5	.5	0	0

Tackling The Challenges of Big Data

Big Data Analytics

Machine Learning Tools

Example: Collaborative Filtering

THANK YOU

PROFESSIONAL EDUCATION

CSAIL

Tackling The Challenges of Big Data

Big Data Analytics
Machine Learning Tools

Tommi Jaakkola

Professor

Massachusetts Institute of Technology