**Page 1:**

Introduce not just my work, but myself.

Projects I’ve done and their impacts.

My career with data science and user behavior modeling.

**Page 2:**

What is behavior? Interactions, environment.

Value. Imagine when you have…

We did not… but now we have…

**Page 3:**

What content you generate and disseminate on the Web.

Different decisions with/without friends’ influence.

Different behaviors in different location at different time.

Different patterns and effects with different intentions.

Driven by preference, curiosity or monetary incentives.

**Page 4:**

Many applications need behavioral analysis.

Typically, prediction for recommendation and fraud detection.

Engineers who work on the systems share the same methodology.

Previously, we have experience-driven approaches. We observe by…

Example: governor of a world of plants, zombies; residents, bitter melon.

Feedback from a few customers, your family, friends: biased; decision=mistake.

Observe from data, represent the data and develop models/algorithms with the data.

**Page 5:**

Tencent company, Weibo department (Twitter).

5.78%

**Page 6:**

Since then, I start the journey of studying Behavioral Analytics with big data.

Prepare my toolbox. I collect tools from every step I made.

**Page 7:**

The first research problem I studied was given by the industry.

How to predict information sharing/retweeting behavior?

**Page 8:**

CF: user-based and item-based:

two users similar ratings to the same book, other books.

two items from the same person, other users;

book/movie recommenders.

Koren proposed matrix factorization-based CF,

Netflix $1M competition.

Content: filtering the content with previously adopted contents.

SoRec: Book/movie recommenders enable social features.

Use social relational links between users to recommend movies.

Trust-based: assume trust can be propagated in the network.

Random walk methods.

We have retweeting behavior data, the tweets’ content,

the social relational network, the users’ interaction frequency.

How to integrate? Given the environment of social recommenders,

what are the key factors of users’ decision?

**Page 9:**

The answer is Yes.

When she received the message, what did she think before making the decision?

Reason 1: It’s about “happy birthday”, not foreign-policy debate (preference).

Reason 2: It’s from “my husband”, not an ordinary fan of the First Lady (influence).

Social contextual factors.

**Page 10:**

Red circle: receive and retweet, item adopted.

Blue star: receive but ignore, item refused/rejected.

Facebook and Twitter, Renren and Weibo.

X-axis preference: topic distribution of receiver and item.

Y-axis influence: interaction frequency between receiver and sender.

See red circles are mostly on the top-right corner.

Not easy to adopt a message: both high preference scores and high influence scores.

We observe the factors in the real data.

**Page 11:**

Learning latent features from “content” and “behavior” data.

Preference: the similarity between the user and item’s latent features.

Influence: Learning influence score from the sender.

with “social relation”, “interaction frequency” and “behavior” data.

**Page 12:**

Propose a matrix factorization based model, ContextMF.

Left hand is the influence score given by the item’s sender.

Right hand is the preference score by comparing the receiver and item’s latent features.

**Page 13:**

S multiplied by transposed G, influence.

Transposed U multiplied by V, preference.

Hadamard product.

Optimization function, to minimize the sum of the Frobenius norms:

1st term: behavior; 2nd term: interaction frequency;

3rd term: content; 4th term: social relation;

other terms: regularization to avoid over-fitting.

**Page 14:**

Gradient: vector field whose components are the partial derivative of optimization function J.

Gradient descent: first-order iterative optimization, local optimum of the function.

**Page 15:**

Comparing error-based and ranking-based metrics: ContextMF vs Influence, vs Preference.

Comparing with the best state-of-the-art method [TIST 2011].

Prediction @ top K.

**Page 16:**

Cross-domain: all the Weibo users can post images, videos, join groups and attach themselves with social tags.

Sparsity, knowledge transfer across domains through the social domain.

I published a paper in CIKM’12 as the 1st author…

Cross-platform: can we use Facebook data to improve behavior prediction on Uber?

For our data, can we use Weibo data to predict movie ratings on Douban?

E-mail to register both platforms. Overlapping accounts we have take only 1% of the population.

Can we transfer knowledge across platforms through the overlapping part?

I published a paper in AAAI’16 as the 1st author…

**Page 17:**

Tools I collected from my experience on behavior prediction.

**Page 18:**

Weibo was very satisfied with our recommender design and deployment.

They told us a more serious problem on their platform: fake accounts.

Zombie followers/botnets.

Created by scripts. Inflate the number of followers.

Only 5% of Twitter accounts are real.

Fake Facebook Likes.

Weibo used experience-driven supervised learning approaches

Using features of … to detect the zombie followers.

However, they don’t have to…

**Page 19:**

I feel fortunate to have the chance of studying the monetary intentions of the bad actors.

I also feel fortunate to have an opportunity to study overseas for one year.

I read lots of papers to find the experts on suspicious behavior detection.

Then I contacted Prof. C. Faloutsos in CMU.

He had papers about anomaly detection on fake review detection in Ebay.

The zombie follower problem is more challenging, but I believe I can learn a lot from him.

I grabbed this opportunity and visited him for 9 months.

I took Weibo’s “who-follows-whom” graph with me.

I hope to solve the problem, collect and even develop my own tools.

**Page 20:**

I started from plotting the degree distributions.

In DBLP, an author’s out-degree is the number of his/her publications.

X-axis: out-degree; y-axis is the number of authors who have the degree value.

The out-degree distribution in log-log scale is often a straight line: called power-law.

The three Faloutsos brothers, Michalis, Petros, Christos, in SIGCOMM’99.

In Twittter data, I spot spikes!

**Page 21:**

Twitter and Weibo.

Who created the spikes?

I first found a few zombie followers and then looked for their behavioral patterns in the data.

Particularly, who are their followees? How they inflate their followees’ fame?

**Page 22:**

Jon Kleinberg, Cornell, HITS algorithm, authoritativeness.

Heat map: red: millions of users/followees, purple: very few.

Outliers. Same in-degree (#fans), smaller authoritativeness.

Who are the zombie followers’ followees?

**Page 23:**

Colorful heat map into grey-scale.

My followers.

BuyAB22’s followers: Synchronized, Abnormal.

**Page 24:**

High synchronicity means more suspicious.

**Page 25:**

Low normality means more suspicious.

**Page 26:**

What is the relation between synchronicity and normality?

Given a normality, what is the minimal synchronicity?

Dividing the feature space into G grids.

Foreground points: followees.

Background points: all users.

Normality: 0-1. Fg multiplied by Bg.

Fg: Percentage of followees in grid g by all the followees.

Bg: Percentage of users in grid g by all the users.

Synchronicity: 0-1. Sum of Fg square.

**Page 27:**

Lagrange multiplier: minimize synchronicity,

subject to conditions of normalization and normality.

The synchronicity has a lower limit which is a Parabolic function of normality.

**Page 28:**

Plot a heat map of all the users. The majority locates at the lower limit.

BuyAB22 has very high synchronicity and very law normality. It is very suspicious.

It’s easy to develop a distance-based outlier detection algorithm to catch the zombie followers.

The algorithm is called CatchSync.

**Page 29:**

CatchSync found 3 million users who had high synchronicity and law normality.

When we look at their followees, they all follow around 20 from a set of 1,500 users.

Their followees are exactly the accounts who want lots of followers.

If we remove all these links from the network, we recover the out-degree distribution to

the Power-law distribution. Before CatchSync, the distribution is the black dots, they have a big spikes.

Now it is the red dots. There is no spike any more.

**Page 30:**

In July 2014, Tencent’s Weibo department was dismissed

and Tencent stopped upgrading its micro-blogging service.

Tencent announced that it integrated the Weibo team into their news team.

In 2012, Tencent Weibo had 469 million registered users.

In our data 2011, it had 117 million. It just suffers too much from the issue of fake followers.

We developed a top-tier conference award-finalist algorithm but it was too late.

Camouflage: follow some other normal users.

The Best Paper cited my both papers.

**Page 31:**

From Christos Faloutsos and his group, I got many skills of detecting fraudsters such as…

We talked about modeling behaviors with social contexts and different intentions.

**Page 32:**

Another research problem is how to modeling behaviors with spatiotemporal context.

This is very important because people make different decisions in different locations at different time.

\*For example, in late 2013, when Weibo was not yet dismissed,

I worked on a mentioned-user recommender system.

For the mobile users, when they took some pictures at a place, they want to mention their friends.

If they have hundreds of friends, how to find the one they want to mention?

Here the data has multiple dimensions.

The historical geo-tagged tweet has the user, location and the user who was mentioned.

Now it’s not just a user-item network but many hyper-edges among the dimensions.

It is straightforward to use multi-way tensor to represent the data.

Every cell is the number of tweets that have the specific value at each dimension.

We can decompose the tensor for patterns, however, it costs too much time

if we re-decompose the tensor every time there is new incremental data coming.

So I proposed an incremental tensor decomposition method to do evolutionary analysis.

This work is published in KDD’14.

\*On the other side,

Weibo also suffers a lot from the spammers.

The spammers consistently retweet the same set of hashtags to inflate their popularity.

Weibo offers us tweets with user accounts, the hashtags, their IP addresses and the time information.

Again, we can represent the multidimensional data with multi-way tensor.

Now the question is, how to evaluate the suspiciousness across dimensions.

For example, we spot 10 users posted 1,000 tweets of 5 hashtags.

We also spot the other 1,000 tweets by more users, say, 100 users, but they operate at only 2 IP addresses.

Which is more suspicious?

I proposed a principled scoring function of the suspiciousness.

The general philosophy is that given the data distribution,

if it has less probability to appear, it’s more suspicious.

The scoring function was proved to satisfy intuitive axioms on dimension, size, density and mass.

For example, when the sizes are the same, higher density is more suspicious.

In Weibo data, we detected something very suspicious with no requirement of parameters…

That’s millions of tweets every month.

This could be another reason that users don’t like using Weibo

and Tencent dismissed the Weibo department.

**Page 33:**

On my way of studying multidimensional data for

behavior prediction and suspicious behavior detection,

I learnt methods and tools such as …

**Page 34:**

In 2015, I graduated from Tsinghua with a Ph.D. degree.

My thesis title is “…”. I won the Dissertation Award by the University.

**Page 35:**

During my study of mining the behavior network,

I have been working with the well-structured data.

I realize one of the critical bottlenecks in understanding human behaviors

is \*modeling the rich unstructured content\*.

If we can mine the structure in the text data,

for example, we extract the entities, attributes, relationships and even sentiments

from the tweets, news, product reviews,

if we can integrate our analysis with this rich information network,

we can better understand the behavioral content.

**Page 36:**

Then I started working as a postdoc with…

The UIUC group has… (automatic text mining)

With SegPhrase, the quality phrase mining technique, I start thinking

how to bring phrases that were automatically extracted from behavioral content into behavior modeling.

This could be a simple but the first and fundamental step of combining behavioral and information networks.

**Page 37:**

Here are two typical behaviors, one is tweeting behavior, the other is paper-publishing behavior.

Suppose the task is to have event summaries and research trend summaries.

SegPhrase extracted quality phrases from the tweets as well as paper titles.

You can see phrases such as … and …

Now I realize there is a serious and fundamental problem in representing the behaviors.

**Page 38:**

The behavior has multiple dimensions and for each dimension,

it may have a set of values instead of one-guaranteed value.

The set can be empty if a tweet does not have a tagged location nor a mentioned user.

The set can have three values if a tweet has 3 phrases, a paper also has 3 phrases, 3 authors.

Many researchers have used tensors to represent multidimensional data. We also did this.

But here, a tensor requires every cell to have exactly one value on every dimension, which is impossible.

We have to change the representation to cover every behavior.

With tensor, we used dense block detection to summarize interesting patterns.

Now we also have to change the representation of summaries.

**Page 39:**

So I proposed a new representation called “Two-Level Matrix”.

Instead of adding dimensions into multi-way tensors,

I added a new level on the columns of the matrix.

The first level of the columns has the dimensions.

For tweets, we have user, phrase, URL, location and hashtag dimensions.

The second level is the values for each dimension.

On the rows, the first level has the time slices in order.

The second level has the behaviors, or tweet, in each time slice.

Now let’s see if we can represent any tweet in the two-level matrix.

See the first row. This tweet has one user, check; two different phrases, check;

one URL, check; no location, no hashtag, check, check.

This shows that the two-level matrix is more general than a tensor.

Now we need to define what is a summary.

A summary is a pattern that takes a set of dimensions, and a set of values in each selected dimensions,

and a list of consecutive time slices, and a set of behaviors in each time slice.

For example, these 3 time slices, 6 tweets created a small advertising campaign:

This single user frequently tweeted the same 2 phrases, and the same URL.

I name it “Tartan”. I use Tartan to define a behavioral summary.

Another Tartan is the purple one. It takes phrase, location and hashtag dimensions.

It is somehow a local event.

**Page 40:**

Perhaps you are not familiar with the word Tartan.

I knew this word since the first day I visited CMU.

All the sports team of CMU are named Tartans.

Even the logo of this university has Tartan as the background image.

However, I watched lots of their games, men’s, women’s, basketball, football, tennis…

We lost most of the games. By saying “most”, I mean over 90% of the games.

I was very depressed, so to memorize this experience, I decided to write a paper about the Tartans.

**Page 41:**

Now the problem is how to catch a dense Tartan is the sparse two-level matrix.

I used the Minimum Description Length principle to address this problem.

The general philosophy is we need many bits to encode the two-level matrix;

but if we encode the Tartan and the other entries separately,

we can save a lot of the bits, because the Tartan is dense and the other entries become even sparser.

We propose a local optimum algorithm, called CatchTartan.

The algorithm updates the shape of the Tartan to maximize the number of bits we can save.

Specifically, it iteratively updates the set of behaviors in each time slice,

update the set of values in each dimension, consider if we want a previous or next time slice in the Tartan,

and try if adding one more dimension can save more bits.

In our experiments, it takes 4-12 iterations to achieve the convergence. The algorithm is very fast.

Note that this algorithm provides a principle scoring function of the Tartan and thus

it requires no parameter. I felt great that I didn’t have to tune the parameters.

**Page 42:**

Here are the experimental results…

Only with the two-level matrix, not the tensor, we can represent all the papers,

and only with the Tartan, we can summarize the trend of different sets of dimensions.

**Page 43:**

Here is what we summarize from Super Bowl tweets.

We automatically find these Tartans.

They are the five phases of the Super Bowl event, from the score prediction,

to 1st qtr, halftime show, 2nd qtr and reflections on the result.

Note that different Tartans may have different sets of dimensions.

For example, the live video advertising of half time show has the URL dimension;

the comments about the show has the location and hashtag dimension.

**Page 44:**

This is my first trial to integrate phrases that were extracted from the behavioral content

into behavior modeling, which makes me believe that

the value of mining and integrating the content is super big.

**Page 45:**

I learnt and used summarization, MDL principle and phrase mining in this paper.

Before I wrote the paper, I had kept thinking of

integrating spatiotemporal structured data with the phrases from unstructured text data.

It will be interesting if we extend this work into a general framework

that can not only summarize the social media data but also

do event detection, fraud detection, prediction and recommendation.

Professor Jiawei Han and I wrote a proposal on this direction and submitted to NSF in Nov’15.

I wrote 8 pages of the proposal. Fortunately, … and I am now the major supported member.

**Page 46:**

After the KDD paper was accepted, Professor Han and I had a long conversation.

He was excited to see I can integrate the phrase mining with structured data mining.

He suggested that I should replace this module with in-depth unstructured data mining.

That is to construct and integrate the heterogeneous information network.

He told me that I should the person who integrates behavior network and information network.

I’ve learnt a lot from my Ph.D. experience and from Professor Christos Faloutsos.

Now I have the chance to learn how to work with information network from Professor Han and the group.

**Page 47:**

After that conversation, I had a full clear picture of my research interests,

“Data-Driven Behavioral Analytics with Networks”.

Only by working on the second theme: … I can complete the study.

I can upgrade the third theme: …

So I work with the text including tweets, news and even paper abstracts.

For example, the tweets and news can tell how old is Barack Obama,

and what is the title of Justin Trudeau in Canada.

Therefore, I was thinking how to use the behavioral content, mainly the text,

to construct a rich attributed information network.

**Page 48:**

Suppose we only have the text corpus,

we want to construct such a network: it has person’s ages, the relationships between countries and persons.

Simply say, the attributes of the entities. The task is called automatic attribute discovery. Give a class…

**Page 49:**

…

**Page 50:**

…

**Page 51:**

Based on the meta pattern definition, we propose a framework to do attribute discovery, called MetaPAD.

Meta Pattern-driven Attribute Discovery.

It has three parts. The first part is integrated data-driven text mining.

It will convert the documents into sequences of only the four elements that consist the meta pattern:

class symbol, work, phrase and punctuation marks.

In this text mining module, the first step is…

**Page 52:**

The second module is “Meta Pattern Mining”.

We first need a classifier to find high-quality meta patterns that can introduce attributes.

Here are the features of a high-quality meta pattern: …

**Page 53:**

We input the sentences in which the entities have been replaced by their class symbols.

We hope the output meta patterns have appropriate granularity.

For example, when the pattern is about a country’s prime minister,

we want the class symbol to be $PrimeMinister;

when the pattern is about a person’s age, even the entity is a president,

we want the class symbol to be $Person.

First of all, we use the classifier to find high-quality patterns.

The entities in the patterns could be all typed at the top level, such as Location and Person;

Or all typed at the bottom level, such as Country, PrimeMinister and President.

Then we re-type the patterns for appropriate granularity.

Before the re-typing, we spot a long-tail distribution of the meta patterns,

which means a lot of patterns could be too rare for re-typing.

But the rare patterns could be the synonyms of some frequent patterns.

For example, …

Therefore, again, we first classify the frequent patterns for high-quality one;

Then detect the synonym meta patterns and group them into a cluster;

Finally, we re-type the meta patterns for appropriate granularity.

We will have these two good meta patterns…

**Page 54:**

For the re-typing, we can do either top-down or bottom-up.

Top-down means the class symbols in meta patterns were first all at the top level.

For example, we have “$Person, a $Digit -year-old”, “$Location President $Person”.

We need to decide whether we want to go down the type hierarchy to the child types.

Here are two important metrics.

The first is graininess. If the child types cover over 80% of the parent type,

for example, for this “$Location”, most of the entities either have been

typed as “$Country” or “Ethnicity”. These two types can cover 84% of the Location entities.

For the “$Person”, the total number of entities with child types

such as $Artist, $Athlete and $Victim, is too small, only 171.

The second metric is called support. The idea is if we decide to go down,

we split the parent meta patterns into child meta patterns that have the popular child types.

For example, there are only 2 counties and only one prime minister.

So finally, we split ““$Location President $Person” into two patterns,

One is “$Country President $President”, the other is “$Ethnicity President $President”.

The bottom-up re-typing uses the same two metrics.

**Page 55:**

Comparing with Biperpedia… In our setting, we cannot supply the query log data.

So we feed Biperpedia with the sentences. Suppose every sentence is a query.

The 1st column lists the attribute names by Biperpedia.

In the 2nd and 3rd columns, we automatically extract not only the attribute names but also the values…

**Page 56:**

$Location entities and Time values (events)

CEOs of organizations including companies

Persons’ ages. President of the ethnicities.

High-quality at the long-tail of the distribution.

**Page 57:**

In four datasets (3 news, 1 tweet), every component of our MetaPAD can increase the F1 score.

MetaPAD integrated all the components. So it can significantly outperform the Biperpedia.

With PubMed abstracts, we can discover attributes of the biomedical terms. BMI, square.

**Page 58:**

General philosophy of our data-driven approach, especially of Meta Pattern, is

We use distant supervision from Knowledge Bases

such as Freebase, Wiki or Bio database called MeSH to initiate entity typing.

Then we take the advantage of data redundancy to find attributes.

This work was praised by Prof. Jiawei Han as a milestone,

because it’s the key step towards constructing attributed networks with only text corpus.

**Page 59:**

I evaluated the ($Country, $President) pairs extracted by MetaPAD. The accuracy is just 0.829.

So an interesting question is can we construct a trustworthy attributed network?

Let’s look at the number of false positives by these meta patterns.

We see that…

So how to model the reliability of sources? Here the sources are the meta patterns.

This future work is quite related with the recent popular research domain, Truth Finding,

led by Professor Jing Gao and many other researchers.

**Page 60:**

By developing the MetaPAD, I learnt the literature of text mining and machine learning work

on the NLP tasks such as entity recognition, fine-grained typing and slot filling.

**Page 61:**

Now let’s summarize my research topic, Data-Driven Behavioral Analytics with Networks, as follows.

Theme 1: I introduced… (including prediction, recommendation and fraud detection.

Specifically, I introduced ContextMF and CatchSync.

Also we talked about modeling multidimensional behavioral data)

Theme 2: I introduced… (especially, the MetaPAD, for automatic attributed network construction)

Theme 3: I introduced CatchTartan, which is a preliminary study of

integrating behavior networks with phrases from the information networks.

The network integration directs us to build intelligent and trustworthy systems

for prediction, summarization and detection.

In the future, I look forward to studying along this line and collecting more powerful tools into my toolbox.