E6893 Big Data Analytics Lecture 10:

*End-to-End System Workflow*

Ching-Yung Lin, Ph.D.
Adjunct Professor, Dept. of Electrical Engineering and Computer Science

November 12th, 2021
Workflow

- A **scheduler**, which handles both triggering scheduled workflows, and submitting **Tasks** to the executor to run.
- An **executor**, which handles running tasks. In the default Airflow installation, this runs everything *inside* the scheduler, but most production-suited executors actually push task execution out to **workers**.
- A **webserver**, which presents a handy user interface to inspect, trigger and debug the behaviour of DAGs and tasks.
- A folder of **DAG files**, read by the scheduler and executor (and any workers the executor has)
- A **metadata database**, used by the scheduler, executor and webserver to store state.
Workflow Components
Apache Airflow

Apache Airflow

Airflow is a platform created by the community to programatically author, schedule and monitor workflows.
Workloads

There are three types of Tasks:

- **Operators**, predefined tasks that you can string together quickly to build most parts of your DAGs.
- **Sensors**, a special subclass of Operators which are entirely about waiting for an external event to happen.
- A **TaskFlow-decorated @task**, which is a custom Python function packaged up as a Task.
Operators

An Operator is conceptually a template for a predefined Task, that you can just define declaratively inside your DAG:

```
with dag("my-dag") as dag:
    ping = SimpleHttpOperator(endpoint="http://example.com/update/")
    email = EmailOperator(to="admin@example.com", subject="Update complete")

    ping >> email
```
Example of Popular Operators

- **BashOperator** - executes a bash command
- **PythonOperator** - calls an arbitrary Python function
- **EmailOperator** - sends an email

- **SimpleHttpOperator**
- **MySqlOperator**
- **PostgresOperator**
- **MsSqlOperator**
- **OracleOperator**
- **JdbcOperator**
- **DockerOperator**
- **HiveOperator**
- **S3FileTransformOperator**
- **PrestoToMySqlOperator**
- **SlackAPIOperator**
Sensors

Sensors are a special type of Operator that are designed to do exactly one thing - wait for something to occur. It can be time-based, or waiting for a file, or an external event, but all they do is wait until something happens, and then succeed so their downstream tasks can run.

Because they are primarily idle, Sensors have three different modes of running so you can be a bit more efficient about using them:

- **poke** (default): The Sensor takes up a worker slot for its entire runtime
- **reschedule**: The Sensor takes up a worker slot only when it is checking, and sleeps for a set duration between checks
- **smart sensor**: There is a single centralized version of this Sensor that batches all executions of it
TaskFlow

askFlow takes care of moving inputs and outputs between your Tasks as well as automatically calculating dependencies - when you call a TaskFlow function in your DAG file, rather than executing it, that you can then use as inputs to downstream tasks or operators.

```python
from airflow.decorators import task
from airflow.operators.email import EmailOperator

@task
def get_ip():
    return my_ip_service.get_main_ip()

@task
def compose_email(external_ip):
    return {
        'subject': f'Server connected from {external_ip}',
        'body': f'Your server executing Airflow is connected from the external IP {external_ip}'
    }

email_info = compose_email(get_ip())

EmailOperator(
    task_id='send_email',
    to='example@example.com',
    subject=email_info['subject'],
    html_content=email_info['body']
)
```
Control Flow

DAGs are designed to be run many times, and multiple runs of them can happen in parallel.

DAGs are parameterized, always including an interval they are "running for" (the data interval), but with other optional parameters as well.

Tasks have dependencies declared on each other. You'll see this in a DAG either using the >> and << operators:

```
first_task >> [second_task, third_task]
third_task << fourth_task
```

Or, with the set_upstream and set_downstream methods:

```
first_task.set_downstream([second_task, third_task])
third_task.set_upstream(fourth_task)
```
Triggering

These dependencies are what make up the "edges" of the graph, and how Airflow works out which order to run your tasks in. By default, a task will wait for all of its upstream tasks to succeed before it runs, but this can be customized using features like Branching, LatestOnly, and Trigger Rules.
Branching

**Note**

When a Task is downstream of both the branching operator *and* downstream of one of more of the selected tasks, it will not be skipped:

The paths of the branching task are `branch_a`, `join` and `branch_b`. Since `join` is a downstream task of `branch_a`, it will be still be run, even though it was not returned as part of the branch decision.
## Airflow User Interface

### DAGs

<table>
<thead>
<tr>
<th>DAG Name</th>
<th>Owner</th>
<th>Runs</th>
<th>Schedule</th>
<th>Last Run</th>
<th>Recent Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>example_bash_operator</td>
<td>airflow</td>
<td>2</td>
<td>00:00</td>
<td>2020-10-26, 21:08:11</td>
<td>6</td>
</tr>
<tr>
<td>example_branch_dop_operator_v3</td>
<td>airflow</td>
<td>1</td>
<td>00:00</td>
<td>2020-10-23, 14:09:17</td>
<td>11</td>
</tr>
<tr>
<td>example_branch_operator</td>
<td>airflow</td>
<td>1</td>
<td>@daily</td>
<td>2020-10-26, 21:08:04</td>
<td>37</td>
</tr>
<tr>
<td>example_complex</td>
<td>airflow</td>
<td>1</td>
<td>None</td>
<td>2020-10-26, 21:08:04</td>
<td>37</td>
</tr>
<tr>
<td>example_external_task_marker_child</td>
<td>airflow</td>
<td>1</td>
<td>None</td>
<td>2020-10-26, 21:07:33</td>
<td>2</td>
</tr>
<tr>
<td>example_external_task_marker_parent</td>
<td>airflow</td>
<td>1</td>
<td>None</td>
<td>2020-10-26, 21:08:34</td>
<td>1</td>
</tr>
<tr>
<td>example_kubernetes_executor</td>
<td>airflow</td>
<td>1</td>
<td>None</td>
<td>2020-10-26, 21:07:40</td>
<td>5</td>
</tr>
<tr>
<td>example_kubernetes_executor_config</td>
<td>airflow</td>
<td>1</td>
<td>None</td>
<td>2020-10-26, 21:07:37</td>
<td>5</td>
</tr>
<tr>
<td>example_nested_branch_dag</td>
<td>airflow</td>
<td>1</td>
<td>@daily</td>
<td>2020-10-26, 21:07:37</td>
<td>5</td>
</tr>
<tr>
<td>example_passing_params_via_test_command</td>
<td>airflow</td>
<td>1</td>
<td>00:00</td>
<td>2020-10-26, 21:08:04</td>
<td>37</td>
</tr>
</tbody>
</table>
Tree View

DAG: example_bash_operator
Graph View

DAG: example_bash_operator

- BashOperator
- DummyOperator

Tasks:
- runme_0
- also_run_this
- runme_1
- run_after_loop
- run_this_last
- this_will_skip

Status:
- queued
- running
- success
- failed
- up_for_retry
- up_for_reschedule
- upstream_failed
- skipped
- scheduled
- no_status
Calendar View

DAG: example_bash_operator

success  | failed  | running  | no_status
# Variable View

<table>
<thead>
<tr>
<th>Key</th>
<th>Val</th>
<th>Is Encrypted</th>
</tr>
</thead>
<tbody>
<tr>
<td>airtable_api_key</td>
<td>**********</td>
<td>True</td>
</tr>
<tr>
<td>airtable_base_key</td>
<td>appzasdasdasdas</td>
<td>True</td>
</tr>
<tr>
<td>environment</td>
<td>prod</td>
<td>True</td>
</tr>
<tr>
<td>pipedrive_env</td>
<td>pipedrive</td>
<td>True</td>
</tr>
<tr>
<td>postgres_env</td>
<td>prod</td>
<td>True</td>
</tr>
<tr>
<td>snowflake_password</td>
<td>**********</td>
<td>True</td>
</tr>
</tbody>
</table>
Gantt chart

DAG: example_bash_operator

runme_0
runme_1
runme_2
also_run_this
this_will_skip
run_this_last
run_after_loop

Gantt chart for example_bash_operator DAG.
Task Duration

DAG: example_kubernetes_executor

Airflow DAGs Security Browse Admin Docs

21:20 EDT (-04:00) — RH

2020-10-01 19:05:51-04:00

Runs 25

Update

Cumulative Duration

Duration (minutes)

0.12

0.10

0.00

01 Oct 2020

01 Oct 2020

01 Oct 2020

01 Oct 2020

start_task

con_task

two_task

three_task

four_task
Airflow  DAGs  Security  Browse  Admin  Docs

DAG: example_bash_operator

# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements. See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership. The ASF licenses this file
# to you under the Apache License, Version 2.0 (the
# "License"); you may not use this file except in compliance
# with the License. You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing,
# software distributed under the License is distributed on an
# "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
# KIND, either express or implied. See the License for the
# specific language governing permissions and limitations
# under the License.

"""Example DAG demonstrating the usage of the BashOperator.""""
Task Instance Context Menu

DAG: example_bash

Task Instance: run_after_loop
at: 2020-10-01T00:00:00+00:00

Task Actions
- Ignore All Deps
- Ignore Task State
- Ignore Task Deps
- Run
- Clear
- Mark Failed
- Mark Success
End-to-End Workflow Management is Important

"Apache Airflow is highly extensible and its plugin interface can be used to meet a variety of use cases. It supports ..."

"Apache Airflow is highly extensible and its plugin interface can be used to meet a variety of use cases. It supports ..."

"Apache Airflow is a great open-source workflow orchestration tool supported by an active community. It provides all the ..."

"Airflow is Batteries-Included. A great ecosystem and community that comes together to address about any (batch) data ...

"Airflow can be an enterprise scheduling tool if used properly. Its ability to run "any command, on any node" is amazing. ..."

"Airflow is extensible enough for any business to define the custom operators they need. Airflow can help you in your ...

"Apache Airflow helps us efficiently tackle crucial game dev tasks, such as working with churn or sorting bank offers."

"Airflow helped us increase the visibility of our batch processes, decouple our batch jobs, and improve our development ...

https://airflow.apache.org/use-cases/

More details to follow in today’s TA tutorial session of Homework #4
A Big Data Analytics Use Case:
Company Network and Value Analysis
Are we able to find out answers for these questions?

Finding answers of,

- Is it possible to **predict** a company’s profit or revenue changes based on dynamic company networks?
- How can we **infer** evolutionary company networks from public news?
- How **accurate** can network characteristics help predicting profit/revenue changes?
- What are the most important – positive or negative – **feature** measures of networks to predict profit/revenue?
Social Network Analysis

- An Analytics research field since 1920s.
- Social Networks (SNs)
  - **Nodes**: Actors (persons, companies, organizations etc.)
  - **Ties**: Relations (friendship, collaboration, alliance etc.)

- **Network properties**
  - Degree, distance, centrality, and various kinds of positional and equivalence

- **Application of SNs**
  - **Social psychology**: analyzing social phenomena
  - **Economics**: consulting business strategy
  - **Information science**: Information sharing and recommendation, trust calculation, ontology construction
Example of Company Value Analysis

Accounting-based financial statement information

Fundamental values:
ROE(Return On Equity), ROA(Return On Asset), PER(Price Earnings Ratio), PBR(Price Book-value Ratio), Employee Number, Dividend Yield, Capital Ratio, Capital, etc.
E.g. “Fundamental Analysis, Future Earnings and Stock Prices”, [Abarbane&Bushee97]

Applying historical trends to predict stock market index (Heng Seng Index in Hong Kong)

E.g. “Support Vector Machine Regression for Volatile Stock Market Prediction” [H.Yang02]

\[ \hat{I}_t = f(I_{t-w} + \ldots + I_{t-1}) \]
Example of Analytical Tools

- **Network topological analysis** tools
  - Centralities (degree, closeness, betweenness)
  - PageRank
  - Communities (connected component, K-core, triangle count, clustering coefficient)
  - Neighborhood (egonet, K-neighborhood)

- **Graph matching and search** tools
  - Graph search/filter by label, vertex/edge properties (including geo locations)
  - Graph matching
  - Collaborative filtering

- **Graph path and flow** tools
  - Shortest paths
  - Top K-shortest paths

- **Probabilistic graphical model** tools
  - Bayesian network inference
  - Deep learning
Are Social Networks of Companies related to Companies’ Value?
Outline

- Background and Study goal
- Infer Company Networks from Public News
- Network Feature Generation & Selection
- Predict Company Value
- Conclusion and Future work
Company Relationship Detection

- **Specific Relation**
  Cooperation, competition, acquisition, incorporation, supply chain, stock share…
  “Extracting Inter-business Relationship from World Wide Web” [Jin08]
  “Temporal Company Relation Extraction” [Changjian09]
  - Focus on details, deeper NLP
  - Rare events, sparse, ad-hoc

- **Generic Relation**
  - Who give me more impact [in a period]? (maybe positive or negative)
  - Comprehensive, dynamic relations (like Google rank)
  - Shallow NLP, easy to get weighted and directed networks, much more events.
  ➔ THIS WORK!
Generic Relation Extraction

Article (document)

I.B.M. Will Buy a Maker of Data Analysis Software

By STEVE LOHR
Published: July 28, 2009

I.B.M. took a big step to expand its fast-growing stable of data analysis offerings by agreeing on Tuesday to pay $1.2 billion to buy SPSS Inc., a maker of software used in statistical analysis and predictive modeling.

Basic Idea:

- For each company x, we extract companies who
  - Frequently co-appear with x in x’s important financial articles
  - Frequently mentioned together with x in important sentences
  
  In a period of time (e.g. one year)

Sentence

software. In the last couple of years, I.B.M., Oracle, SAP and Microsoft have collectively spent more than $15 billion buying makers of such software.
Example (from NYT 2009 articles about I.B.M.)

**About 300 articles mentioned I.B.M.**

*International Business Machines* (84 articles), *I.B.M.* (277 articles)

- **I.B.M. -- Microsoft** (55 articles, 264 sentences, weight=85.85455)
  
  Two days after I.B.M.'s report, Microsoft said that its quarterly profits were disappointing.

- **I.B.M. -- SPSS** (1 articles, 9 sentences, weight=13.675)
  
  I.B.M. took a big step to expand its fast-growing stable of data analysis offerings by agreeing on Tuesday to pay $1.2 billion to buy SPSS Inc.,...

- **I.B.M. -- Nike**. (4 articles, 9 sentences, weight=8.212)
  
  Others are water-based shoe adhesives from Nike and a packing insert from I.B.M.
Generic Relation Extraction

For target company “x”, first download NYT articles for a year, and select candidate companies Y={y1,y2,…} appeared on the articles, then calculate each candidate company’s relation strength with “x”.

![Download articles](image)

<table>
<thead>
<tr>
<th>Document Weight</th>
<th>Sentence Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Title</strong>: x…...</td>
<td><strong>S1</strong>: x ...... y1 ......</td>
</tr>
<tr>
<td>…x... y1... ...</td>
<td><strong>S2</strong>: x... y3......y5</td>
</tr>
<tr>
<td>.................</td>
<td><strong>S3</strong>: y3...x..., y4...y1...</td>
</tr>
</tbody>
</table>

- |Y|: How many companies on the article?
- Σtf\_{XY}: How many times companies appeared?
- tf\_x: How many times "x" company appear?
- w: Does names appeared on the title?
- |Y\_1|: the number of company names appeared in the same sentence.
- |Y\_2|: the number of company names appeared between “x” and “y”

$$w_d = \log(1 + \frac{1}{|Y|}) \times \sum_{y_i \in Y} \frac{w \cdot tf_x}{w \cdot tf_{x,y_i}}$$

$$w_s = \log(1 + \frac{1}{|Y_1|} + \frac{1}{|Y_2|})$$

$$w = a \cdot \sum df \times w_d + b \cdot \sum sf \times w_s$$
Data and Network

- Data Source:
  - Relationships among companies from public articles
      http://www.nytimes.com/
  - Company Values: profit, revenue, etc.
    - Fortune 500: 1955-2009

- Target companies:
  - 308 companies (from NYT & Fortune500)
  - 656,115 articles about target companies:

![Bootstrap approach](image)
### Network size (all)

<table>
<thead>
<tr>
<th>year</th>
<th>#nodes</th>
<th>#edges</th>
<th>year</th>
<th>#nodes</th>
<th>#edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>1981</td>
<td>463</td>
<td>4030</td>
<td>1996</td>
<td>1202</td>
<td>46265</td>
</tr>
<tr>
<td>1982</td>
<td>478</td>
<td>4457</td>
<td>1997</td>
<td>1266</td>
<td>45650</td>
</tr>
<tr>
<td>1983</td>
<td>477</td>
<td>4546</td>
<td>1998</td>
<td>1312</td>
<td>51362</td>
</tr>
<tr>
<td>1984</td>
<td>484</td>
<td>4606</td>
<td>1999</td>
<td>1379</td>
<td>53653</td>
</tr>
<tr>
<td>1985</td>
<td>546</td>
<td>6606</td>
<td>2000</td>
<td>1534</td>
<td>59079</td>
</tr>
<tr>
<td>1986</td>
<td>565</td>
<td>6680</td>
<td>2001</td>
<td>1496</td>
<td>55801</td>
</tr>
<tr>
<td>1987</td>
<td>941</td>
<td>124326</td>
<td>2002</td>
<td>1487</td>
<td>54713</td>
</tr>
<tr>
<td>1988</td>
<td>1015</td>
<td>108075</td>
<td>2003</td>
<td>1504</td>
<td>54173</td>
</tr>
<tr>
<td>1989</td>
<td>1066</td>
<td>132906</td>
<td>2004</td>
<td>1461</td>
<td>51801</td>
</tr>
<tr>
<td>1990</td>
<td>1070</td>
<td>177022</td>
<td>2005</td>
<td>1193</td>
<td>43944</td>
</tr>
<tr>
<td>1991</td>
<td>1080</td>
<td>107973</td>
<td>2006</td>
<td>1355</td>
<td>51896</td>
</tr>
<tr>
<td>1992</td>
<td>1125</td>
<td>53625</td>
<td>2007</td>
<td>1280</td>
<td>44501</td>
</tr>
<tr>
<td>1993</td>
<td>1133</td>
<td>136807</td>
<td>2008</td>
<td>1260</td>
<td>43340</td>
</tr>
<tr>
<td>1994</td>
<td>1147</td>
<td>130975</td>
<td>2009</td>
<td>1203</td>
<td>37921</td>
</tr>
<tr>
<td>1995</td>
<td>1134</td>
<td>52855</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Financial Crisis 1987**
Comparison of Naïve co-occurrence and the proposed method

- IBM 1995 (doc cooccurrence)
- IBM 1995 (new algorithm – doc weights + sentence weights)

Dominated by big/general companies  Better balance between different company sizes
Example of Network Evolution (IBM)

- IBM 2003
  - Sony Corporation
  - Novell Inc.
  - Hewlett-Packard Corporation
  - Microsoft Corporation
  - Intel Corporation
  - Sun Microsystems Inc.
  - International Business Machines
  - NEC Corporation
  - Oracle Corporation
  - Apple Inc.
  - Dell Inc.
  - EMC Corporation
  - Gartner Inc.
  - Motorola Inc.
  - Citigroup Inc.

- IBM 2009
  - Xerox Corporation
  - Sun Microsystems Inc.
  - Thomson Reuters Corporation
  - Oracle Corporation
  - General Motors Corporation
  - International Business Machines
  - Microsoft Corporation
  - Dell Inc.
  - EMC Corporation
  - Hewlett-Packard Corporation
  - SPSS Inc.
  - Intel Corporation
  - Nokia Corporation
  - Apple Inc.
Example of Network Evolution (Microsoft)

- **Microsoft 1995**
  - Intuit Inc.
  - Microsoft Corporation
  - Stanley Inc.
  - Sun Microsystems Inc.
  - Delphi Corporation
  - Novell Inc.
  - Cisco Systems Inc.
  - Oracle Corporation
  - International Business Machines
  - Time Warner Inc.
  - Apple Inc.
  - Hewlett-Packard Corporation
  - CBS Corporation

- **Microsoft 2003**
  - Intel Corporation
  - Motorola Inc.
  - Oracle Corporation
  - Sun Microsystems Inc.
  - Format Inc.
  - Time Warner Inc.
  - Sony Corporation
  - RealNetworks Inc.
  - Microsoft Corporation
  - International Business Machines
  - Electoni
  - Apple Inc.
  - Hewlett-Packard
  - Google Inc.
  - Dell Inc.
  - Novell Inc.

- **Microsoft 2009**
  - YouTube
  - Nokia Corporation
  - Twitter
  - Sun Microsystems Inc.
  - Sony Corporation
  - Apple Inc.
  - Microsoft Corporation
  - Dell Inc.
  - International Business Machines
  - Oracle Corporation
  - Intel Corporation
  - comScore Inc.
  - News Corporation
  - Hewlett-Packard Corporation

<table>
<thead>
<tr>
<th></th>
<th>1995</th>
<th>2003</th>
<th>2009</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Intuit</td>
<td>I.B.M.</td>
<td>Google</td>
</tr>
<tr>
<td>2</td>
<td>I.B.M.</td>
<td>Apple</td>
<td>Apple</td>
</tr>
<tr>
<td>3</td>
<td>Intel</td>
<td>Intel</td>
<td>Intel</td>
</tr>
<tr>
<td>4</td>
<td>Apple</td>
<td>Time Warner</td>
<td>Sony</td>
</tr>
<tr>
<td>5</td>
<td>Novell</td>
<td>Sony</td>
<td>I.B.M.</td>
</tr>
</tbody>
</table>
Outline

- Background and Study goal
- Infer Company Networks from Public News
- Network Feature Generation & Selection
- Predict Company Value
- Conclusion and Future work
Network Type

Weighted-Directed Network

\[ W_{i-j} = W_{i->j} + W_{j->i} \]

Weighted-Undirected Network

Binary -Directed Network

Binary -Undirected Network
Network Feature Generation (1/3)

Who give company x impact?
- Neighbor companies on the network
- Reachable companies on the network

Network Features:
- number of neighbors (In-degree, Out-degree)
- number of reachable nodes
- number of connections among neighbors
- number of connections among reachable nodes
- neighbors’ degree (In-degree, Out-degree)
- distance of x to all reachable nodes
- distances among neighbors
- ratio of above values between neighbors and reachable nodes ...
- etc.
*(Normalize by network size)*

Generate 57 Network features from weighted/binary, directional/undirectional networks
Thresholding of Networks

Different Threshold Network

- Revenue
- Profit
- Delta-revenue (△)
- MEAN

Thresholds:
- $T_1 = 20$
- $T_2 = 10$
Network Feature Generation (2/3)

Temporal Network Features:
- number of neighbors (In-degree, Out-degree) \( last \) year (or \( w \) years ago)
- number of connections among neighbors \( last \) year (or \( w \) years ago)
- number of connections among reachable nodes \( last \) year (or \( w \) years ago)
- number of neighbors degree \( last \) year (or \( w \) years ago)
- distance of \( x \) to all reachable nodes \( last \) year (or \( w \) years ago)
- … etc.

\[ 57 \times \text{Window} \] temporal network features

Similar to,
- What’s last year’s (or \( w \) years ago) revenue?
- What’s last year’s (or \( w \) years ago) profit?
Network Feature Generation (3/3)

**Delta Change of Network Features:**

- *Delta change of* the number of neighbors (In-degree, Out-degree) *from last year (or d years ago)*
- *Delta change of* the number of connections among neighbors *from last year (or d years ago)*
- *Delta change of* the number of connections among reachable nodes *from last year (or d years ago)*
- *Delta change of* the number of neighbors degree *from last year (or d years ago)*
- *Delta change of* the distance of $x$ to all reachable nodes *from last year (or d years ago)*
- … etc.

$57 \times \text{Delta Network features}$
Network Features

- Network Features for each company
  1. Current Network features: 57
  2. Temporal Network features: 57 × Window
  3. Delta change of Network features: 57 × Delta

  + Financial statements of companies
    - previous year’s profit/revenue
      - delta-change of profit/revenue
      - … etc.
Steps to Learn for Network Feature Selection

- Correlations between ranking of each individual feature and ranking of revenue/profit
- Stability of feature values which should be consistent with different network thresholding
- Selecting Independent Features sets (orthogonal with each other)
Feature Selection

- Feature Selection
  - Filter out some un-useful features from leaning samples.
  - Positive features VS negative features
  - Company-specific selections or General selections
### Positive and Negative Features (example)

<table>
<thead>
<tr>
<th>Correlation</th>
<th>Positively related features</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.421</td>
<td>difference of the ratio of x's neighbors and reachable nodes in binary-undirected network in 3 years</td>
</tr>
<tr>
<td>0.421</td>
<td>delta value with 3 years ago of x's degree in binary-undirected network</td>
</tr>
<tr>
<td>0.420</td>
<td>2 year ago x's degree in binary-undirected network</td>
</tr>
<tr>
<td>0.413</td>
<td>x's degree in binary-undirected network</td>
</tr>
<tr>
<td>0.413</td>
<td>ratio number of x's neighbors and reachable nodes in binary-undirected network</td>
</tr>
<tr>
<td>0.353</td>
<td>2 year ago x's in-degree in weighted-undirected network</td>
</tr>
<tr>
<td>0.344</td>
<td>delta value with 3 years ago of x's out-degree in weighted-directed network</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Correlation</th>
<th>Negatively related features</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.487</td>
<td>previous year's connections among neighbors in binary-undirected network</td>
</tr>
<tr>
<td>-0.477</td>
<td>delta value with 2 year's ago of sum of degrees among neighbors in binary-undirected network</td>
</tr>
<tr>
<td>-0.462</td>
<td>previous year's connection among neighbors in weighted-undirected network</td>
</tr>
<tr>
<td>-0.462</td>
<td>previous year's connection among neighbors in binary-undirected network</td>
</tr>
<tr>
<td>-0.381</td>
<td>ratio of connection among neighbors and reachable nodes in weighted-undirected network</td>
</tr>
<tr>
<td>-0.379</td>
<td>previous year's ratio of connection among neighbors and reachable nodes in weighted-undirected network</td>
</tr>
</tbody>
</table>
Positive Feature Example

“difference of the ratio of x's neighbors and reachable nodes in binary-undirected network in 3 years”

2010: $|N_1| = 5$, $|N_2| = 8$, $\text{ratio}(|N_1|,|N_2|) = \frac{5}{8} = 0.625$
2007: $|N_1| = 4$, $|N_2| = 7$, $\text{ratio'}(|N_1|,|N_2|) = \frac{4}{7} = 0.57$

$\Rightarrow \text{Delta (ratio – ratio')} = \frac{5}{8} - \frac{4}{7} = 0.054$
Negative Feature Example

“previous year's connections among neighbors in binary-undirected network”

x’s network in 2010
N1= {a, b, c, d, i}
N2={a, b, c, d, e, f, g, h, i}

x’s network in 2009
N1= {a, b, c, d}
N2={a, b, c, d, e, f, g}
Connection_N1= {b-c, a-d}

➔ Connection_t-1= 2
Feature Set Selection

From learning samples, move out features which $|\text{correlation}|<0.2$, \#sample<50.

- $fs=0$: No feature selection
- $fs=1$: Feature selection (positive features only)
- $fs=2$: Feature selection (positive and negative features)

Using positive features are enough for our prediction model.
Feature Variances

\[
\text{var}_{F_i} = \frac{\sum_{k \in K} (\text{corr}_k (F_i, T_i) - \text{corr})^2}{|K|}
\]

- \(k\): various networks in different threshold
- \(i\): different features
Feature Selection based on Stability of values with different network thresholding

![Graph showing variance for different network types](image)
Outline

- Background and Study goal
- Infer Company Networks from Public News
- Network Feature Generation & Selection
- Predict Company Value
- Conclusion and Future work
System Outline

News Articles

Company Names
1. I.B.M., (International Business Machines)
2. Microsoft Corp. …

Company values
e.g. profit and revenue

Relation Extraction
(Bootstrap approach)

Network Construction

Network Feature Extraction

Learn and Prediction

e.g. profit and revenue

Company values

© CY Lin, Columbia University
Experiments

- **Tasks:**
  - For individual companies, learn from last 10 years, and predict next year’s company value
  - For 20 fortune companies, learn from past 5 years, and predict next year’s Companies Value.
  - Company Value: revenue, profit

- **Prediction Model**
  - **Linear Regression**
    \[
    \text{value} = a + \sum_i \beta_i \text{feature}_i + \varepsilon .
    \]
  - **SVM Regression** (using RBF kernel)
    \[
    \begin{align*}
    \min_{w,b,\xi,\xi^*} & \frac{1}{2} w^T w + C \sum_i \xi_i + C \sum_i \xi_i^* \\
    \text{subject to} & \quad w^T \phi(x_i) + b - z_i \leq \varepsilon + \xi_i, \\
    & \quad z_i - w^T \phi(x_i) - b \leq \varepsilon + \xi_i^*, \\
    & \quad \xi_i, \xi_i^* \geq 0, \quad i = 1, \ldots, l.
    \end{align*}
    \]
    \[
    K(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|^2), \gamma > 0.
    \]
Performance Measures

- **R^2** (squared Correlation Coefficient)

\[
R^2 = \frac{(\bar{l} \sum_{i=1}^{l} f(x_i) y_i - \sum_{i=1}^{l} f(x_i) \sum_{i=1}^{l} y_i)^2}{(\bar{l} \sum_{i=1}^{l} f(x_i)^2 - (\sum_{i=1}^{l} f(x_i))^2)(\bar{l} \sum_{i=1}^{l} y_i^2 - (\sum_{i=1}^{l} y_i)^2)}
\]

- **MSE** (Mean Squared Error)

\[
MSE = \frac{1}{l} \sum_{i=1}^{l} (f(x_i) - y_i)^2
\]

Testing data: \(x_1, \ldots, x_l\)

Target values: \(y_1, \ldots, y_l\)

Predicted values: \(f(x_1), \ldots, f(x_l)\)
Profit Prediction for Fortune Companies

- Predict 20 companies’ mean value of profits
Profit Prediction using different feature sets (SVR)

**Targets:** 20 Fortune companies' normalized Profits

**Goal:** Learn from previous 5 years, and predict next year

**Model:** Support Vector Regression (RBF kernel)

Profit prediction by joint network and financial analysis outperforms network-only by 130% and financial-only by 33%.

Network feature:
- s (current year network feature),
- t (temporal network feature),
- d (delta value of network feature)

Financial feature:
- p (historical profits and revenues)
Revenue Prediction using different feature sets (SVR)

**Targets:** 20 Fortune companies’ normalized Profits

**Goal:** Learn from previous 5 years, and predict next year

**Model:** Support Vector Regression (RBF kernel)

Network feature: 
s (current year network feature), 
t (temporal network feature), 
d (delta value of network feature)

Financial feature: 
p (historical profits and revenues)

Network feature not contribute to revenue prediction.
Profit Prediction (Linear Regression)

**Targets:** 20 Fortune companies' normalized Profits

**Goal:** Learn from previous 5 years, and predict next year

**Model:** linear regression

![Graph showing profit prediction (linear regression)]

Network feature:
- s (current year network feature),
- t (temporal network feature),
- d (delta value of network feature)

Financial feature:
- p (historical profits and revenues)

Obtained similar results using different prediction model.

- Profit prediction by joint network and financial analysis outperforms network-only by 150% and financial-only by 37%.
Revenue Prediction (Linear Regression)

**Targets:** 20 Fortune companies' normalized Profits

**Goal:** Learn from previous 5 years, and predict next year

**Model:** linear regression

Network feature: 
- s (current year network feature), 
- t (temporal network feature), 
- d (delta value of network feature)

Financial feature: 
- p (historical profits and revenues)

Network feature not contribute to revenue prediction.
Temporal Window and Delta for Profit Prediction

- Window = 1 VS Window = 3
- Delta = 1 VS Delta = 3
- $R^2$
- mse
- Both Window and Delta size as 1 is enough.
Profit Prediction for IBM and Intel

**IBM Profit Prediction**
- Profit values for IBM from 1991 to 2009.
- The R^2 value for IBM is 0.583.

**Intel Profit Prediction**
- Profit values for Intel from 1991 to 2009.
- The R^2 value for Intel is 0.237.
Questions?