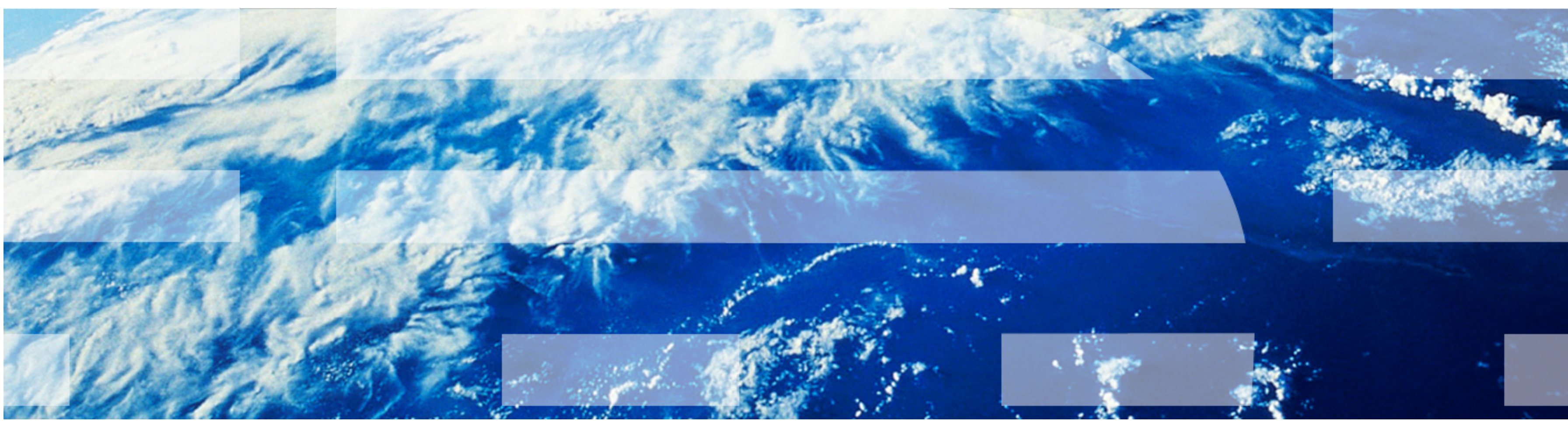
E6893 Big Data Analytics Lecture 5:

End-to-End System Workflow

Ching-Yung Lin, Ph.D.

Adjunct Professor, Dept. of Electrical Engineering and Computer Science



October 4th, 2024

A Big Data Analytics Use Case:

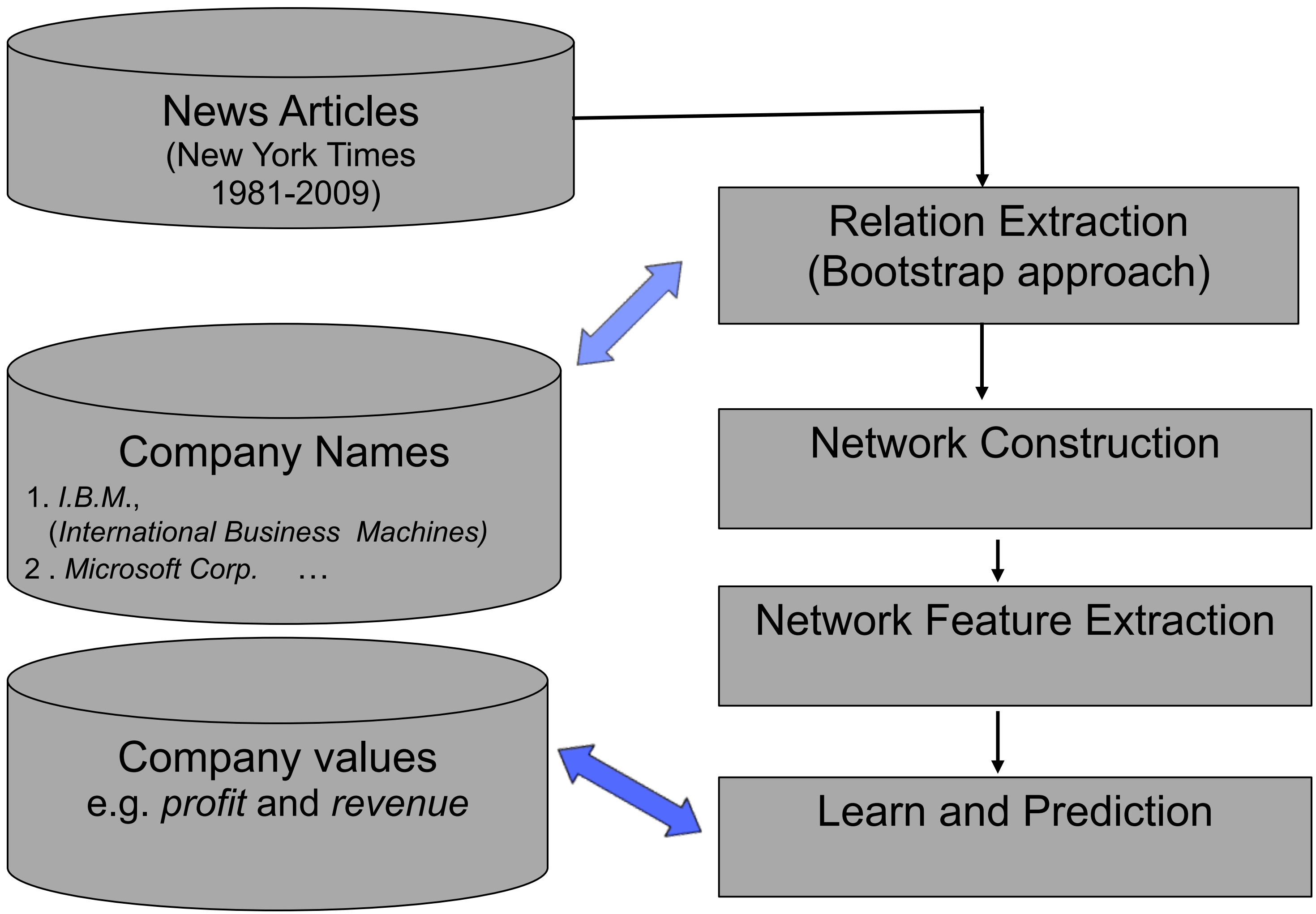
Company Network and Finance Analysis

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?

System Outline



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]

$$\hat{I}_t = f(I_{t-w} + \ldots + I_{t-1})$$

Simple Predictions

- Example 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

$$value = a + \sum_i \beta_i feature_i + \varepsilon .$$

  - SVM Regression (using RBF kernel)

$$\min_{\mathbf{w},b,\xi,\xi^*} \frac{1}{2}\mathbf{W}^T\mathbf{W} + C\sum_{i=1}^{l}\xi_i + C\sum_{i=1}^{l}\xi_i^*$$

$$\begin{aligned} \text{subject to} \quad & \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_i) + b - z_i \leq \varepsilon + \xi_i, \\ & z_i - \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_i) - b \leq \varepsilon + \xi_i^*, \\ & \xi_i \xi_i^* \geq 0, \ i = 1, \ldots, l. \end{aligned}$$

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \, \| \mathbf{x}_i - \mathbf{x}_j \|^2), \gamma > 0.$$

Are Social Networks of Companies related to Companies' Value?

Social Network Analysis

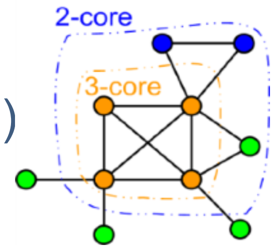
- **An Analytics research field since 1920s.**
- **Social Networks (SNs)**

  Nodes : Actors (persons, companies, organizations etc.)
  Ties : Relations (friendship, collaboration, alliance etc.)

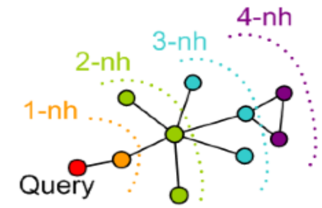
- **Network properties**
  – Degree, distance, centrality, and various kinds of positional and equivalence
- **Application of SNs**
  – Social psychology: analyzing social phenomena
  – Economics: consulting business strategy
  – Information science: Information sharing and recommendation, trust calculation, ontology construction

Example of Analytical Tools

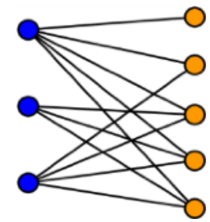
- ***Network topological analysis*** tools
  - •Centralities (degree, closeness, betweenness)
  - •PageRank
  - •Communities (connected component, K-core, triangle count, clustering coefficient)
  - •Neighborhood (egonet, K-neighborhood)
- ***Graph matching and search*** tools
  - •Graph search/filter 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

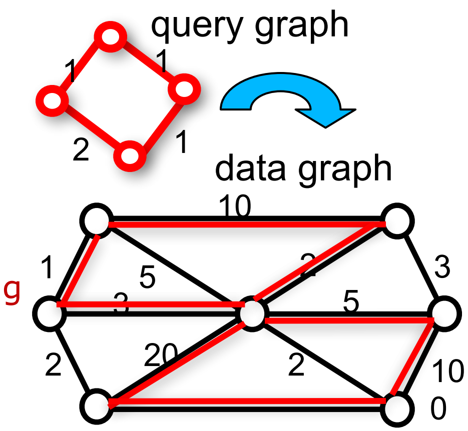


K-core

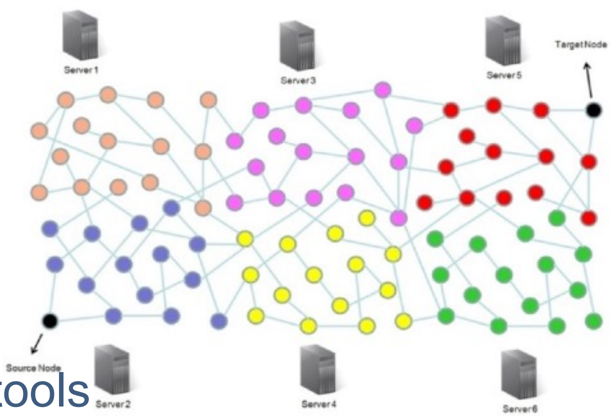


K-neighborhood


Collaborative filtering
Bipartite weighted graph matching



Graph matching



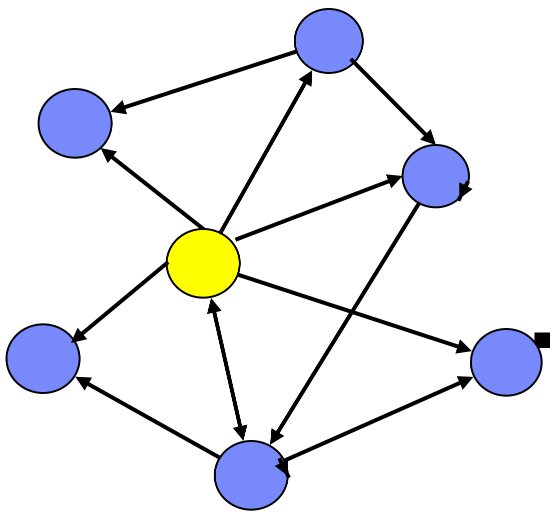
Top k-shortest paths


Bayesian network inference

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 WORKs!

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.

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.

- 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)

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)
  http://www.nytimes.com/2009/03/06/business/06layoffs.html
  Two days after I.B.M.'s report, Microsoft said that its quarterly profits were disappointing.
  http://www.nytimes.com/2009/05/07/technology/07iht-telecoms.html
  ... the world's largest software makers, including Microsoft, SAP and I.B.M., which...
  http://www.nytimes.com/2009/01/31/business/31nocera.html
  Caterpillar, Kodak, Home Depot, I.B.M., even mighty Microsoft they are all cutting jobs.
  http://www.nytimes.com/2009/03/23/technology/companies/23mainframe.html
  More recently, Sun Microsystems, Hewlett-Packard and Microsoft have made mostly unsuccessful attempts to have made mostly unsuccessful attempts to pull mainframe customers away from I.B.M. by ...
- **I.B.M. -- SPSS** (1 articles, 9 sentences, weight=13.675)
  http://www.nytimes.com/2009/07/29/technology/companies/29ibm.html
  I.B.M.to Buy SPSS, a Maker of Business Software
  I.B.M.'s $50-a-share cash offer is a premium of more than 40 percent over SPSS's closing stock price on Monday.
  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)
  http://www.nytimes.com/2009/01/22/business/22pepsi.html
  … companies that have taken steps to reduce carbon emissions includes I.B.M., Nike, Coca-Cola and BP, the oil giant.
  http://www.nytimes.com/2009/11/01/business/01proto.html
  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".

Document Weight

Title: x……
…x.. …y1. .. ..
…….y3………
……………y4..

- |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?

Sentence Weight

S1: x ….. y1 …
S2: x… y3…..y5
S3: y3..x…, y4…y1…

- $|Y_1|$: the number of company names appeared in the same sentence.
- $|Y_2|$: the number of company names appeared between "x" and "y"

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

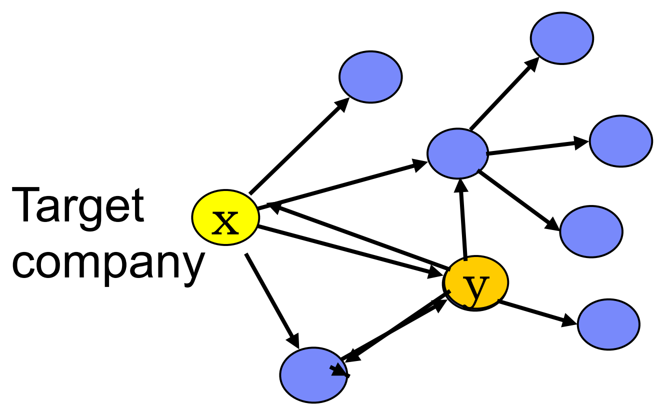
$$\mathrm{w_s} = \log\left(1+\frac{1}{|Y_1|}+\frac{1}{|Y_2|}\right)$$

$$\mathrm{w} = a\cdot\sum df\times \mathrm{w_d} + b\cdot\sum sf\times \mathrm{w_s}$$

Data and Network

- Data Source:
  - Relationships among companies from public articles
    - New York Times (NYT) articles: 1981 ~ 2009
      http://www.nytimes.com/
    - 7594 companies http://topics.nytimes.com/topics/news/business/companies/index.html
  - Company Values: profit, revenue, etc.
    - Fortune 500: 1955-2009
      http://money.cnn.com/magazines/fortune/fortune500/2009/full_list/

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



Bootstrap approach

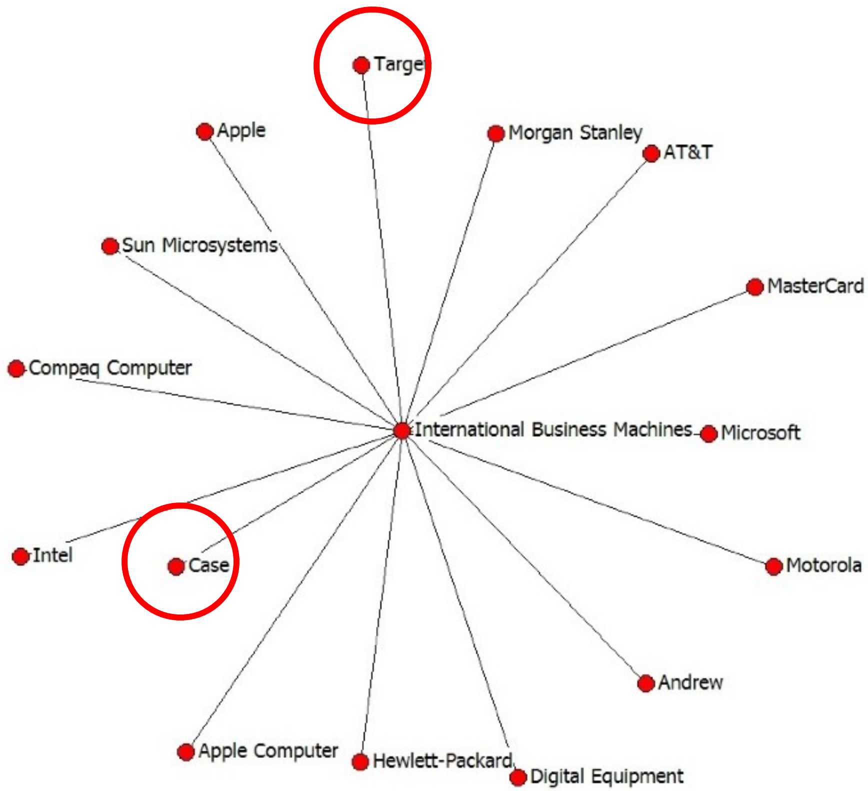
Network size (all)

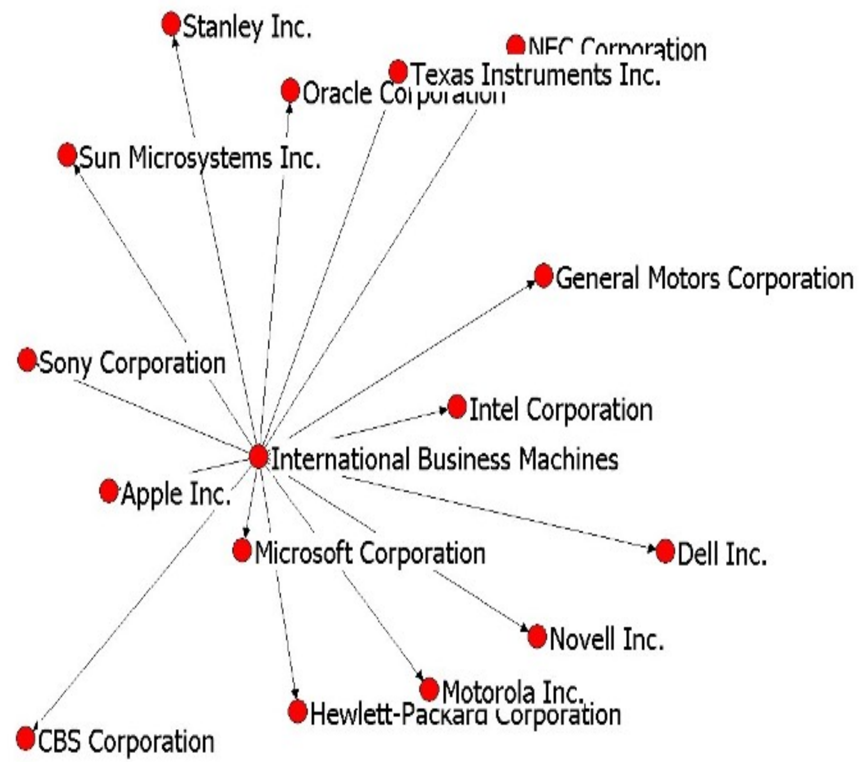
year	#nodes	#edges	year	#nodes	#edges
1981	463	4030	1996	1202	46265
1982	478	4457	1997	1266	45650
1983	477	4546	1998	1312	51362
1984	484	4606	1999	1379	53653
1985	546	6606	2000	1534	59079
1986	565	6680	2001	1496	55801
1987	**941**	**124326**	2002	1487	54713
1988	1015	108075	2003	1504	54173
1989	1066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

Financial Crisis 1987

Comparison of Naïve co-occurrence and the proposed method

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



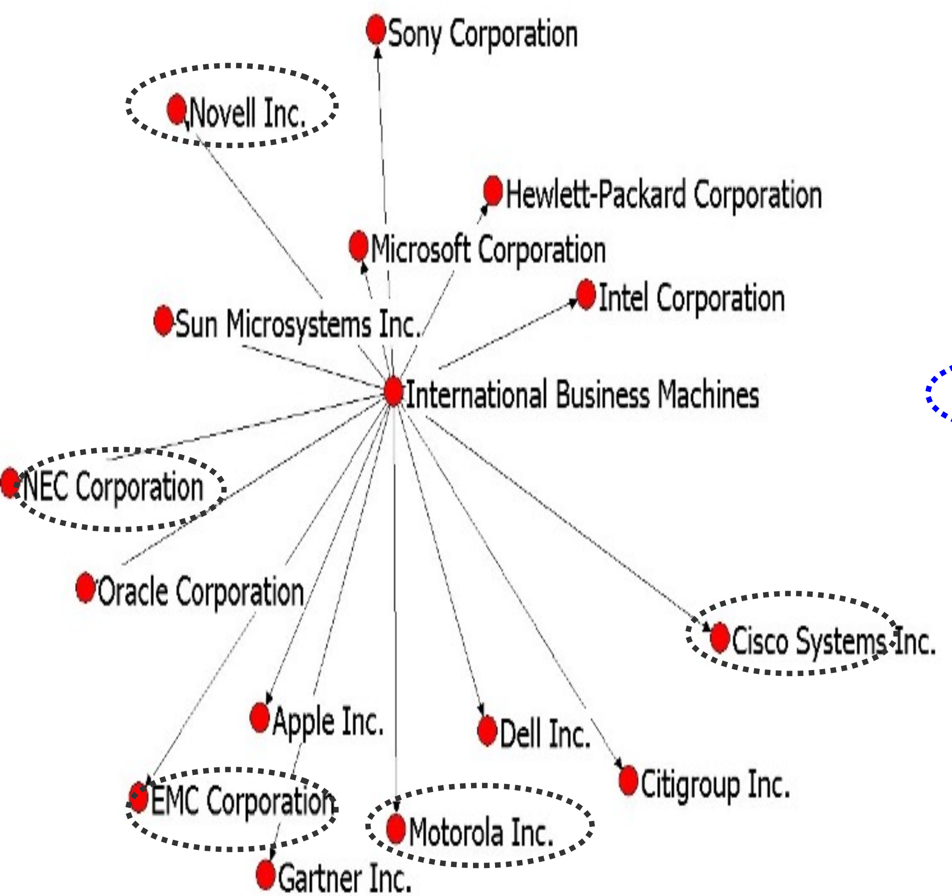


Dominated by big/general companies

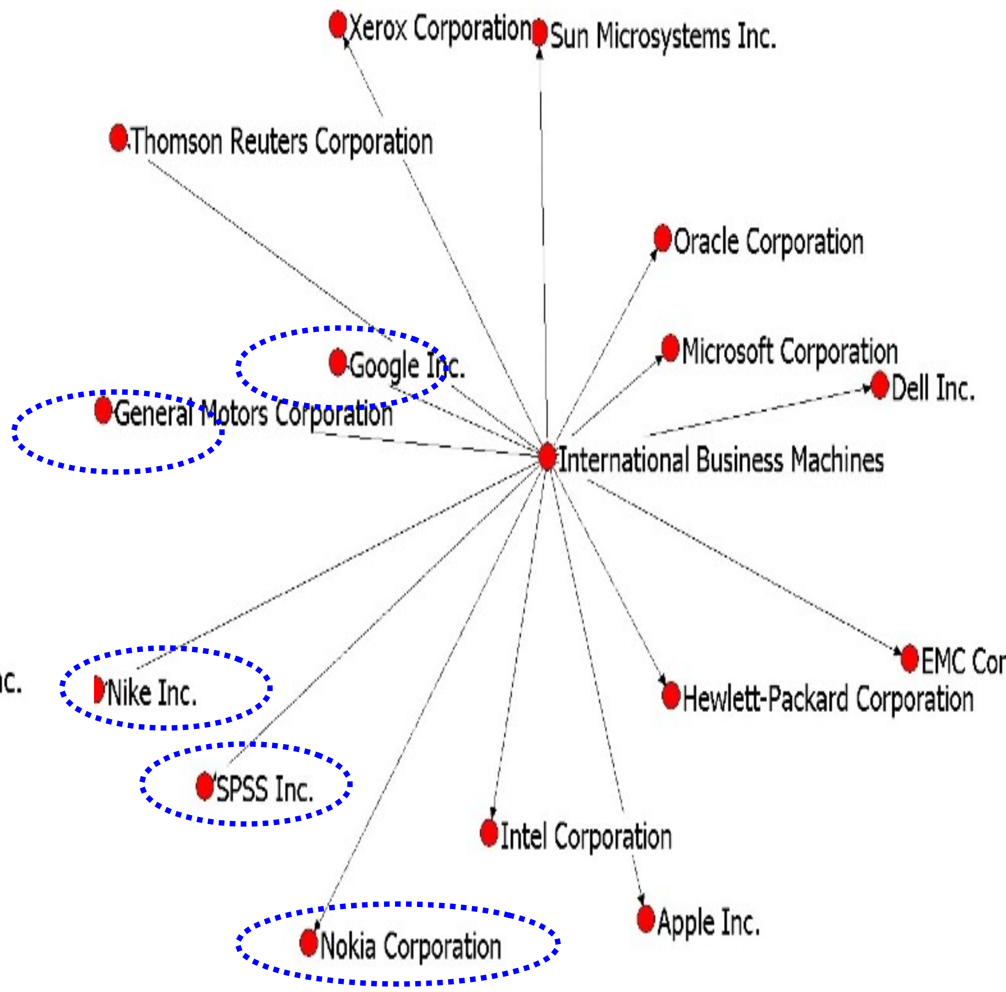
Better balance between different company sizes

Example of Network Evolution (IBM)

- IBM 2003

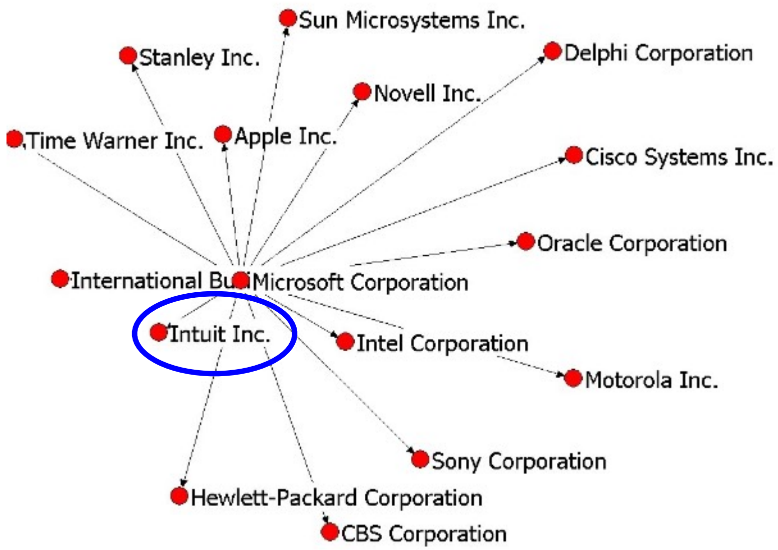


- IBM 2009



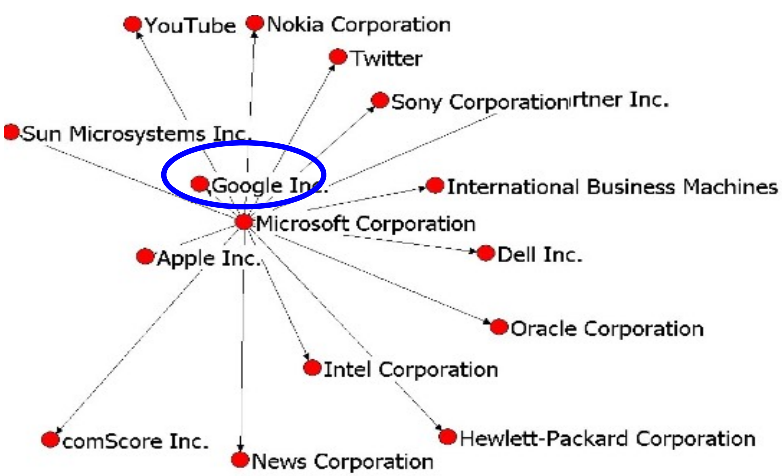
Example of Network Evolution (Microsoft)

- Microsoft 1995



- Microsoft 2003

- Microsoft 2009

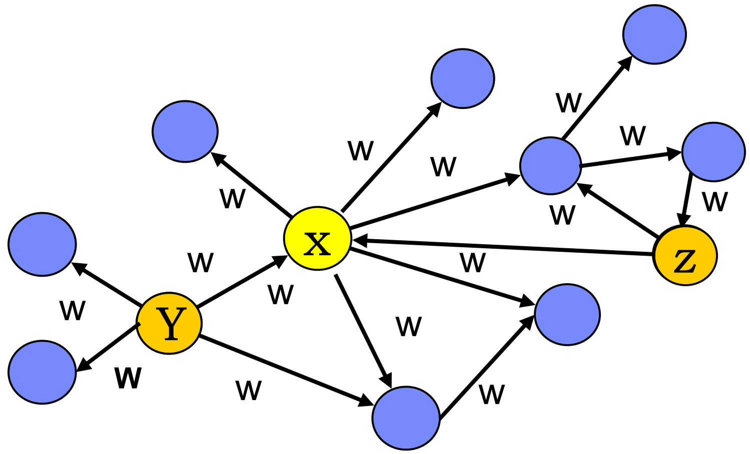


	1995	2003	2009
1	Intuit	I.B.M.	Google
2	I.B.M.	Apple	Apple
3	Intel	Intel	Intel.
4	Apple	Time Warner	Sony
5	Novell	Sony	I.B.M.

Outline

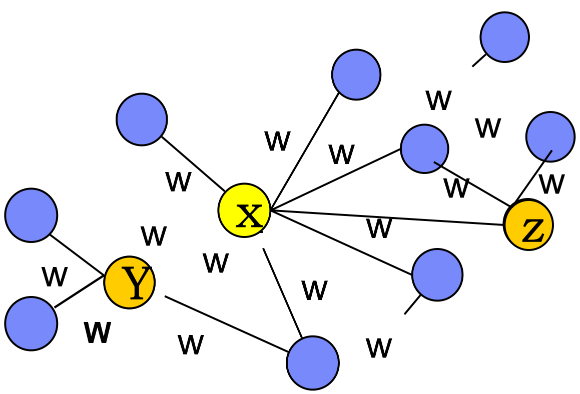
- Background and Study goal
- Infer Company Networks from Public News
- Network Feature Generation & Selection
- Predict Company Value
- Conclusion and Future work

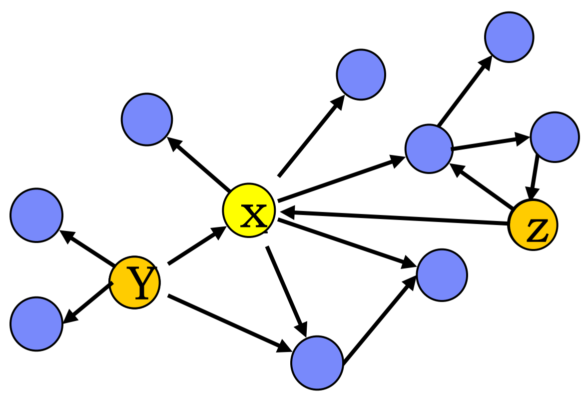
Network Type

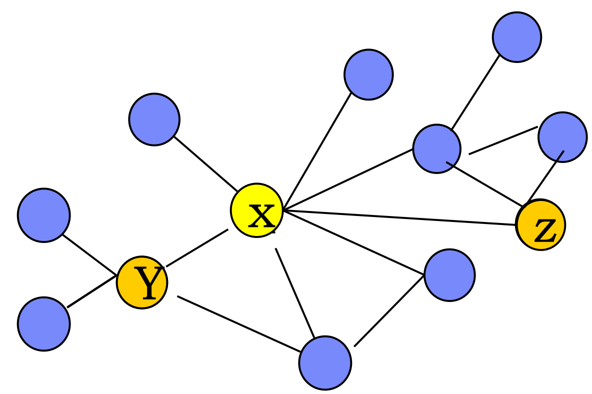


Weighted-Directed Network

$W_{i\text{-}j} = W_{i\text{->}j} + W_{j\text{->}i}$







Weighted-Undirected Network

Binary -Directed Network

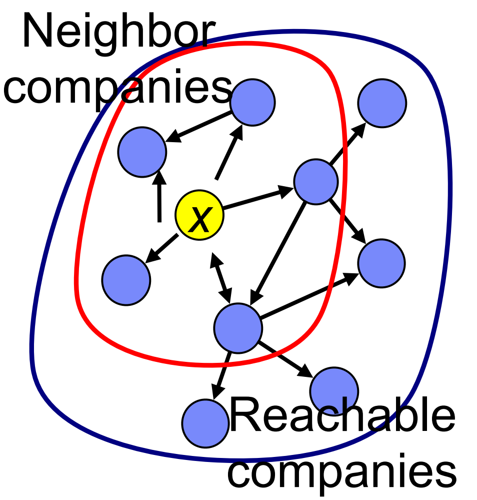
Binary -Undirected Network

Network Feature Generation (1/3)

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

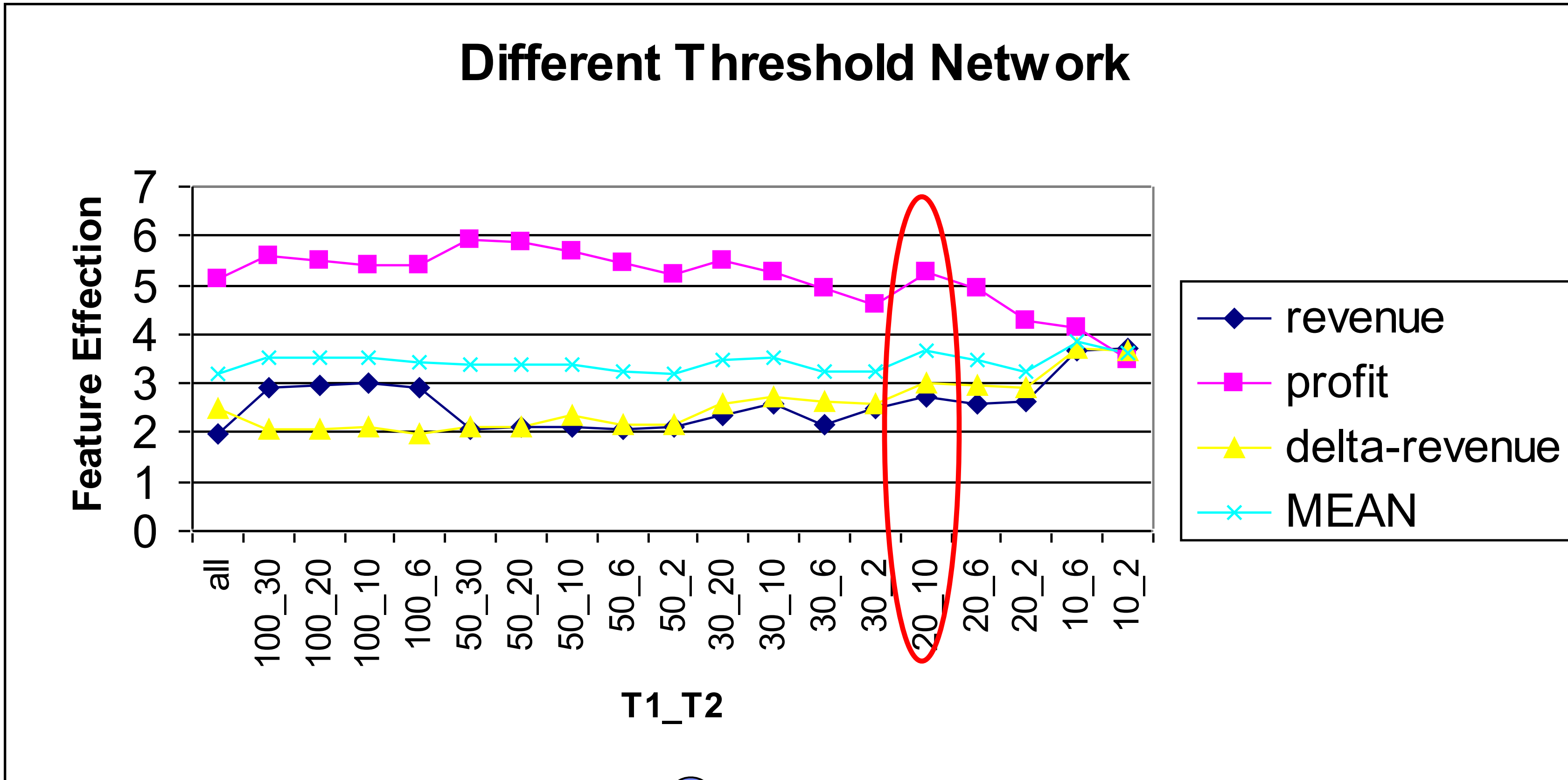
Network Features:
- number of neighbors (In-degree, Out-degree)
- number of reachable nodes
- number of connections among neighbors
- number of connections among reachable nodes
- neighbors' degree (In-degree, Out-degree)
- distance of x to all reachable nodes
- distances among neighbors
- ratio of above values between neighbors and reachable nodes …
- etc.

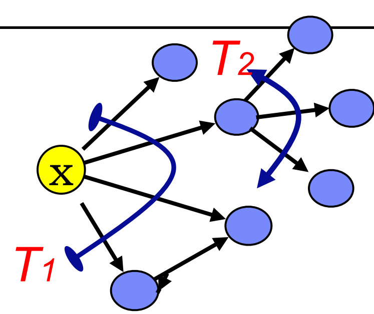
*(Normalize by network size)



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

Thresholding of Networks



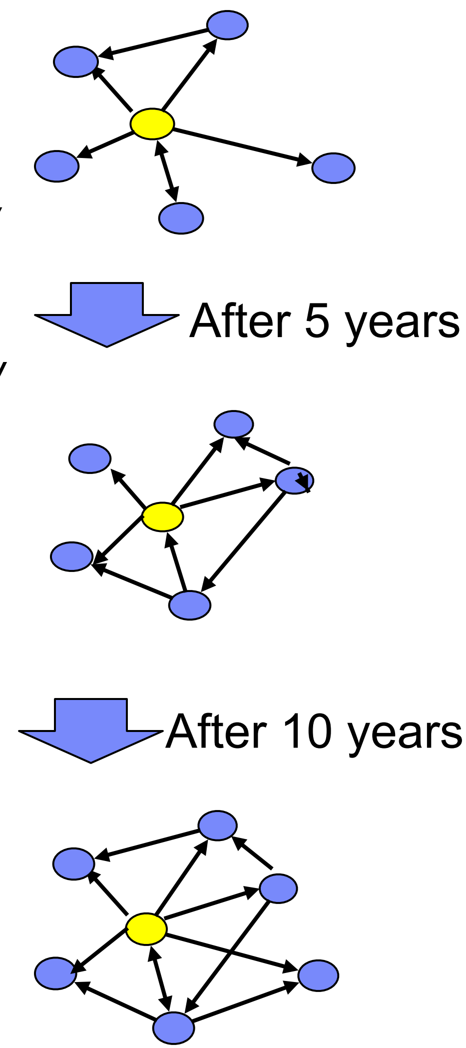


T1=20
T2=10

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×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×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

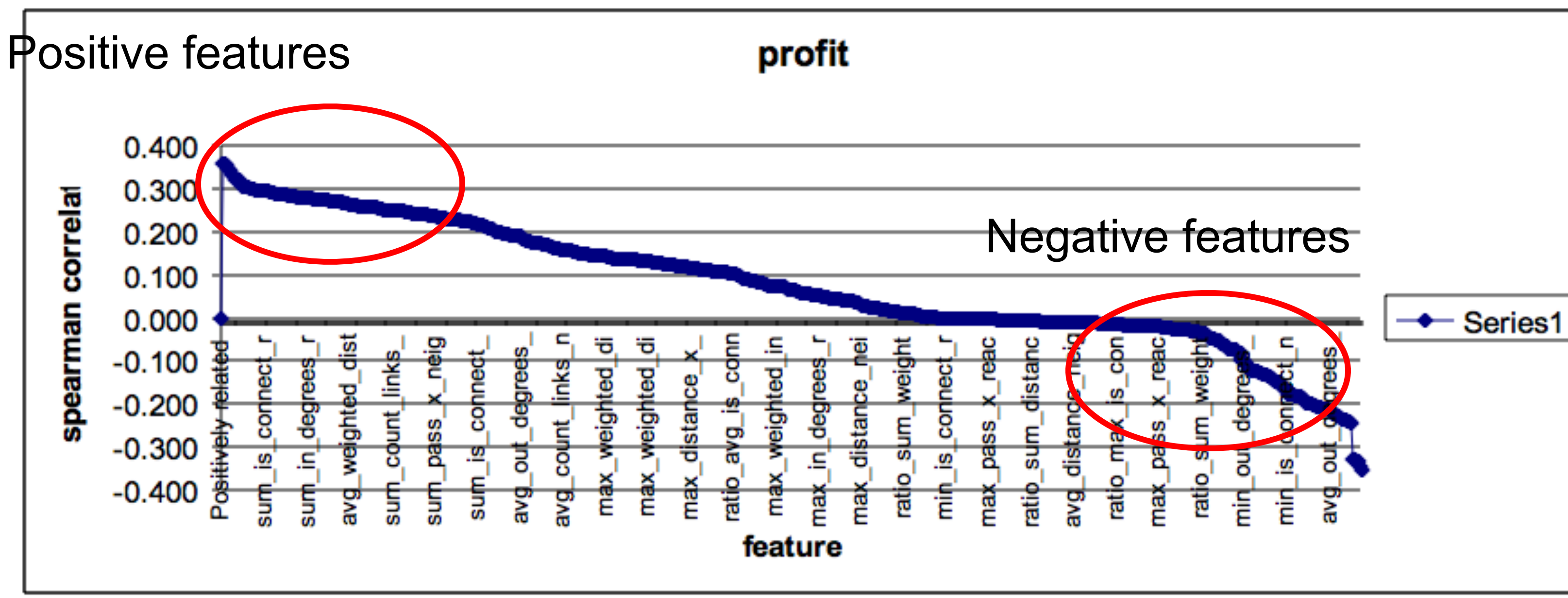
  + Financial statements of companies
    - previous year's profit/ revenue
      - delta-change of profit /revenue
      - … etc.

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



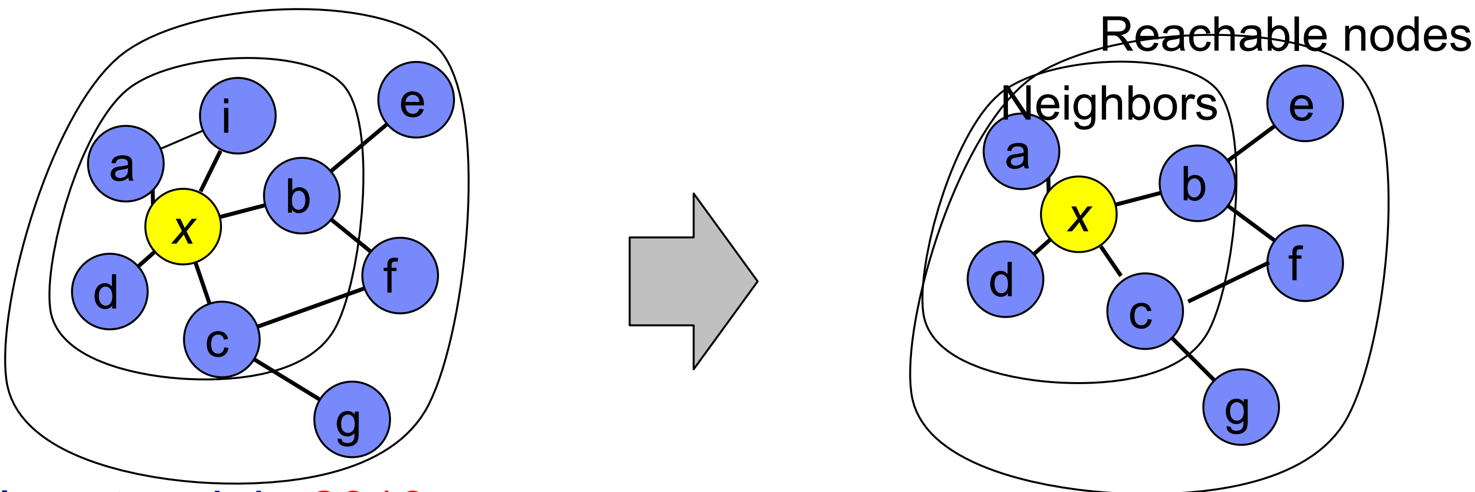
Positive and Negative Features **(example)**

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

Correlation	Negatively related features
-0.487	previous year's connections among neighbors in binary-undirected network
-0.477	delta value with 2 year's ago of sum of degrees among neighbors in binary-undirected network
-0.462	previous year's connection among neighbors in weighted-undirected network
-0.462	previous year's connection among neighbors in binary-undirected network
-0.381	ratio of connection among neighbors and reachable nodes in weighted-undirected network
-0.379	previous year's ratio of connection among neighbors and reachable nodes in weighted-undirected network

Positive Feature Example

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



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 2007

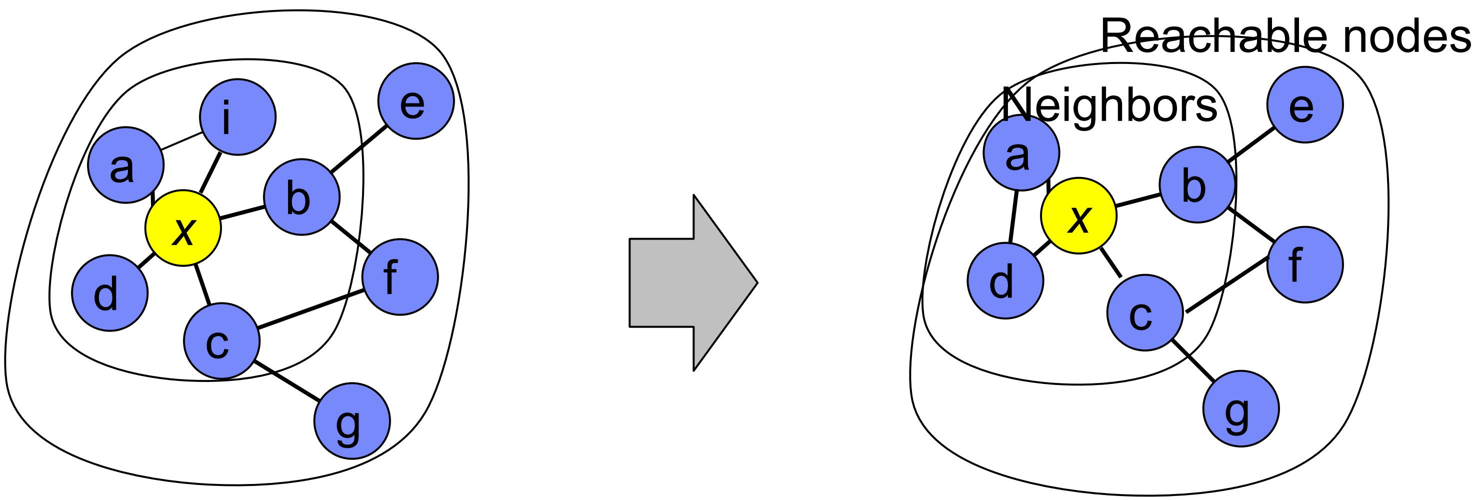
N1= {a,b,c,d}
N2={a,b,c,d,e,f,g}

2010: |N1| = 5, |N2|= 8, ratio(|N1|,|N2|) = 5/8 = 0.625
2007: |N1| = 4, |N2|= 7, ratio’(|N1|,|N2|) = 4/7= 0.57

➔ Delta (ratio – ratio’) = 5/8 – 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

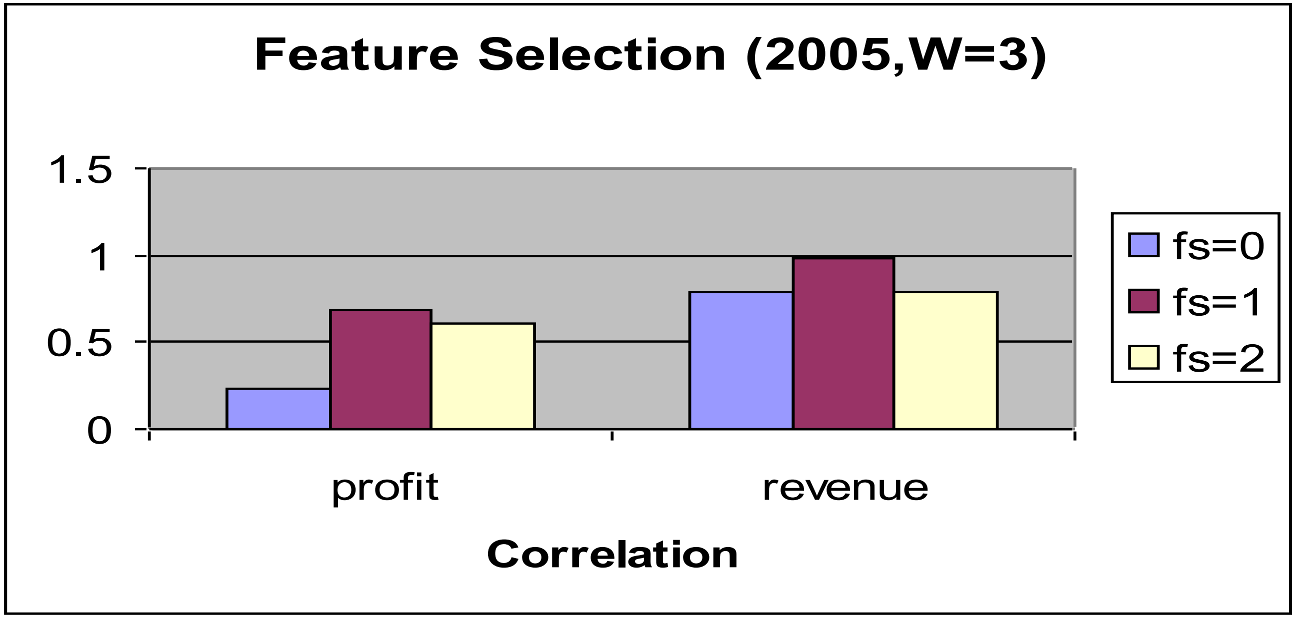
N1= {a,b,c,d}

N2={a,b,c,d,e,f,g}

Connection_N1= {b-c, a-d}

➔Connection_t-1= 2

Feature Set Selection



From Leaning samples, move out features which |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)

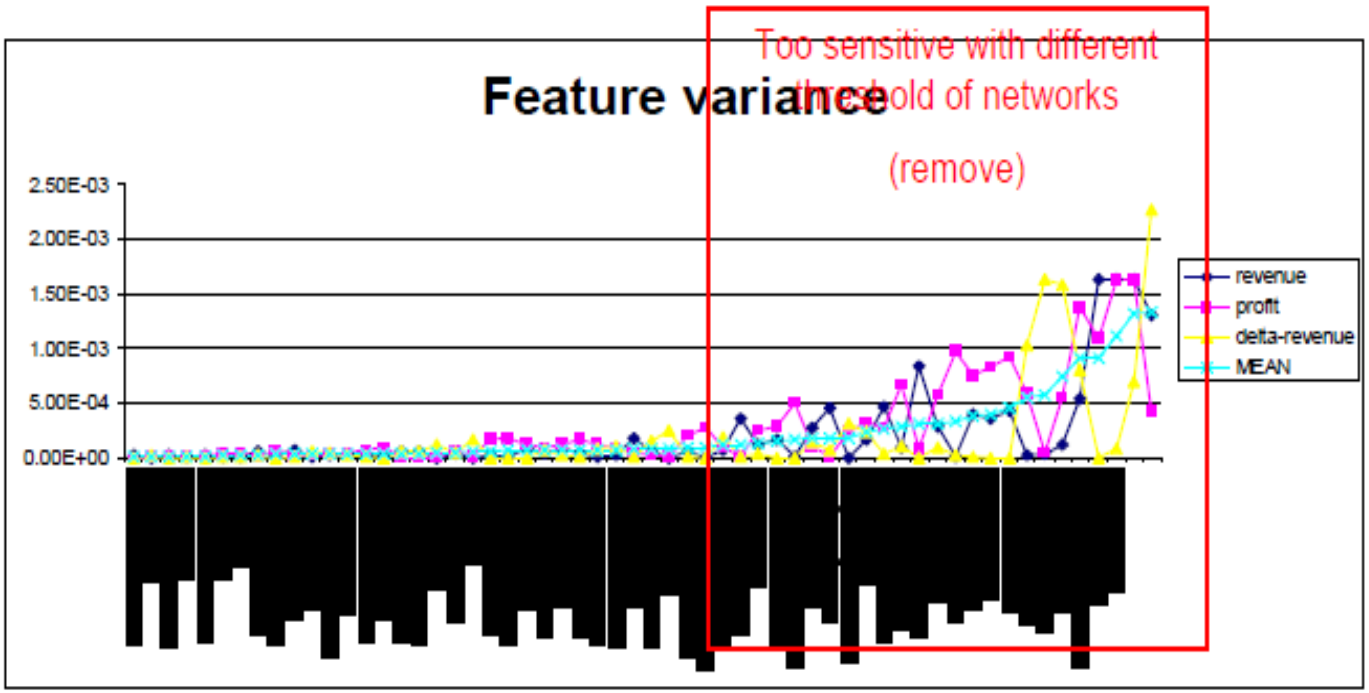
Using positive features are enough for our prediction model.

31

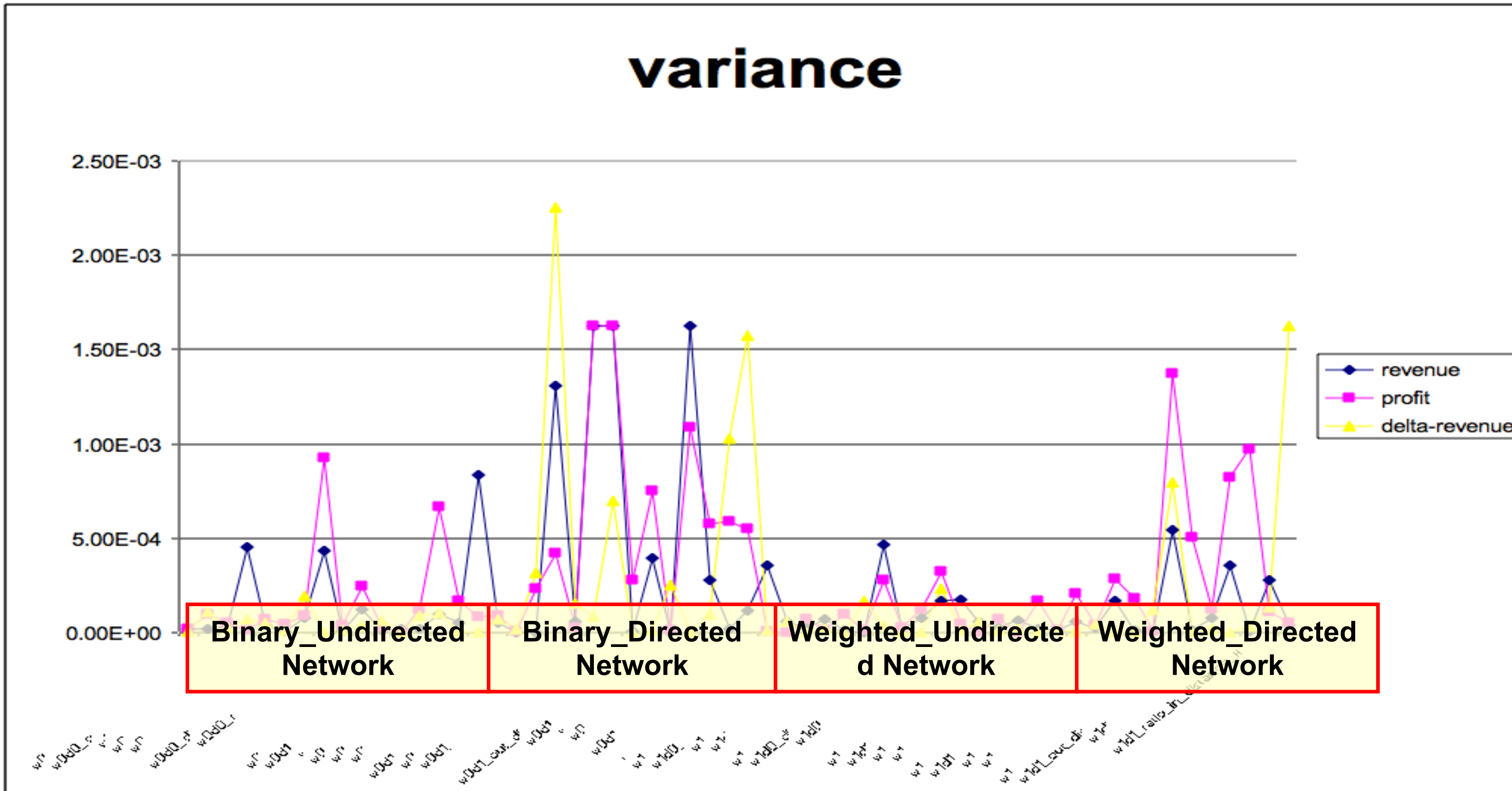
Feature Variances

$$\mathrm{var}_{\mathrm{F}_i} = \frac{\sum_{k \in K} \left(corr_k(\mathrm{F}_i, \mathrm{T}_i) - \overline{corr} \right)^2}{|K|}$$

k: various networks in different threshold
i: different features



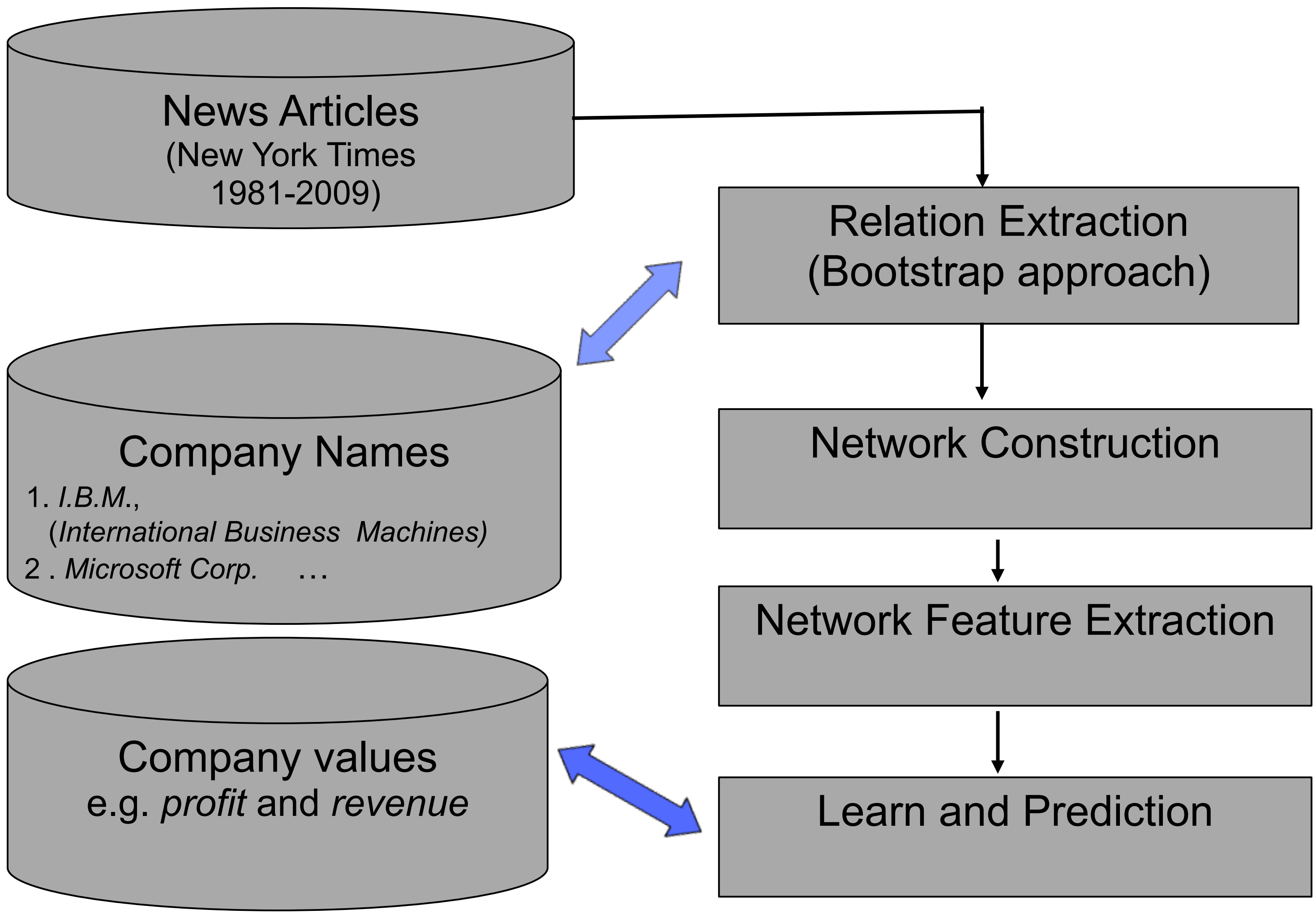
Feature Selection based on Stability of values with different network thresholding



Outline

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

System Outline



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

$$value = a + \sum_i \beta_i feature_i + \varepsilon .$$

  - SVM Regression (using RBF kernel)

$$\min_{\mathbf{w},b,\xi,\xi^*} \frac{1}{2}\mathbf{W}^T\mathbf{W} + C\sum_{i=1}^{l}\xi_i + C\sum_{i=1}^{l}\xi_i^*$$

$$\text{subject to} \quad \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_i) + b - z_i \le \varepsilon + \xi_i,$$

$$z_i - \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_i) - b \le \varepsilon + \xi_i^*,$$

$$\xi_i \xi_i^* \ge 0,\ i = 1, \ldots, l.$$

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \,\|\, \mathbf{x}_i - \mathbf{x}_j \,\|^2), \gamma > 0.$$

Performance Measures

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

$$R^2 = \frac{(\bar{l}\sum_{i=1}^{\bar{l}} f(\mathbf{x}_i)y_i - \sum_{i=1}^{\bar{l}} f(\mathbf{x}_i)\sum_{i=1}^{\bar{l}} y_i)^2}{(\bar{l}\sum_{i=1}^{\bar{l}} f(\mathbf{x}_i)^2 - (\sum_{i=1}^{\bar{l}} f(\mathbf{x}_i))^2)(\bar{l}\sum_{i=1}^{\bar{l}} y_i^2 - (\sum_{i=1}^{\bar{l}} y_i)^2)}$$

- **MSE** (Mean Squared Error)

$$\text{MSE} = \frac{1}{l}\sum_{i=1}^{\bar{l}}(f(\mathbf{x}_i) - y_i)^2$$

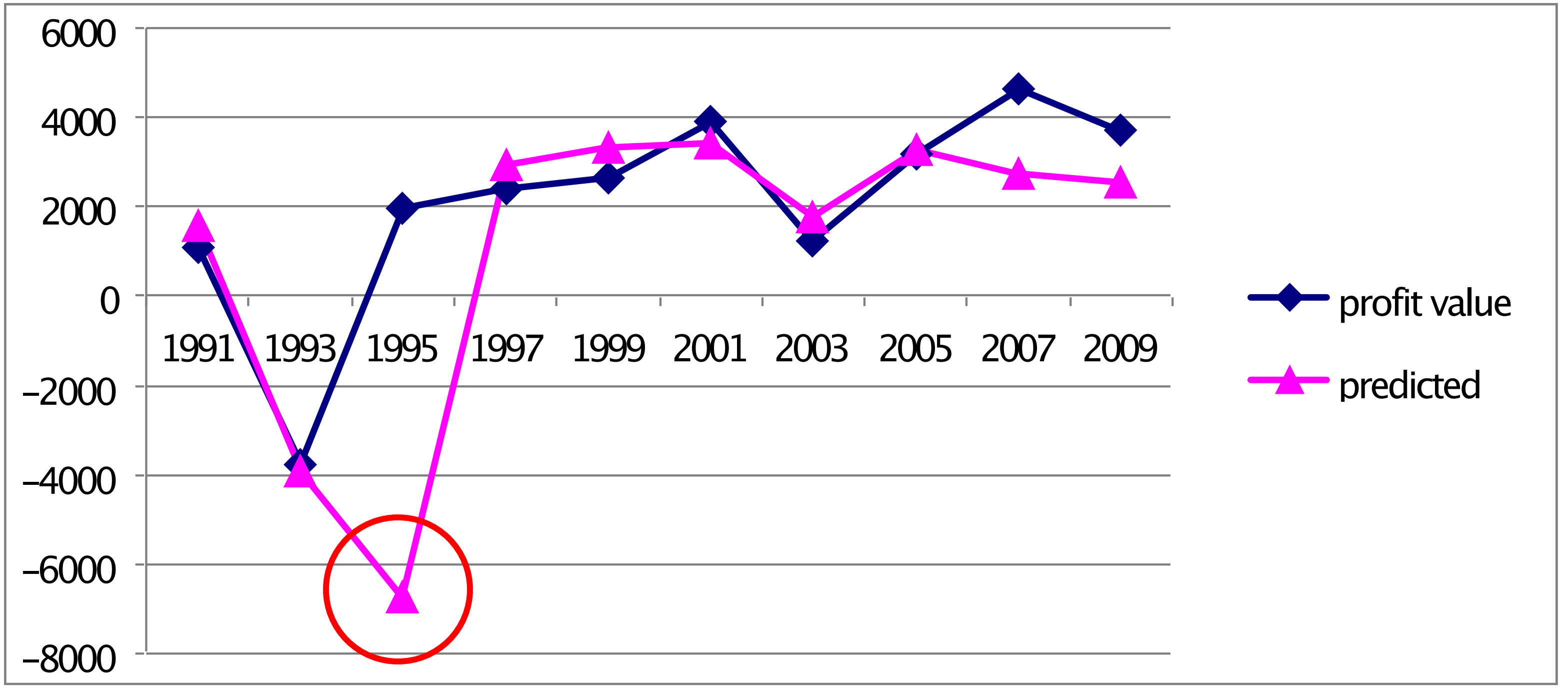
$$\text{Testing data}: \mathbf{x}_1, \cdots, \mathbf{x}_{\bar{l}}$$

$$\text{Target values}: y_i, \cdots, y_{\bar{l}}$$

$$\text{Predicted values}: f(\mathbf{x}_1), \ldots, f(\mathbf{x}_{\bar{l}})$$

Profit Prediction for Fortune Companies

- Predict 20 companies' mean value of profits
  "*I.B.M, Intel, Microsoft, GM, HP, Honda, Nissan, AT&T, Wal-Mart, Yahoo!, Nike, Dell, Starbucks, Chase, PepsiCo, Cisco, FedEx, Gap, AEP, Sun*"

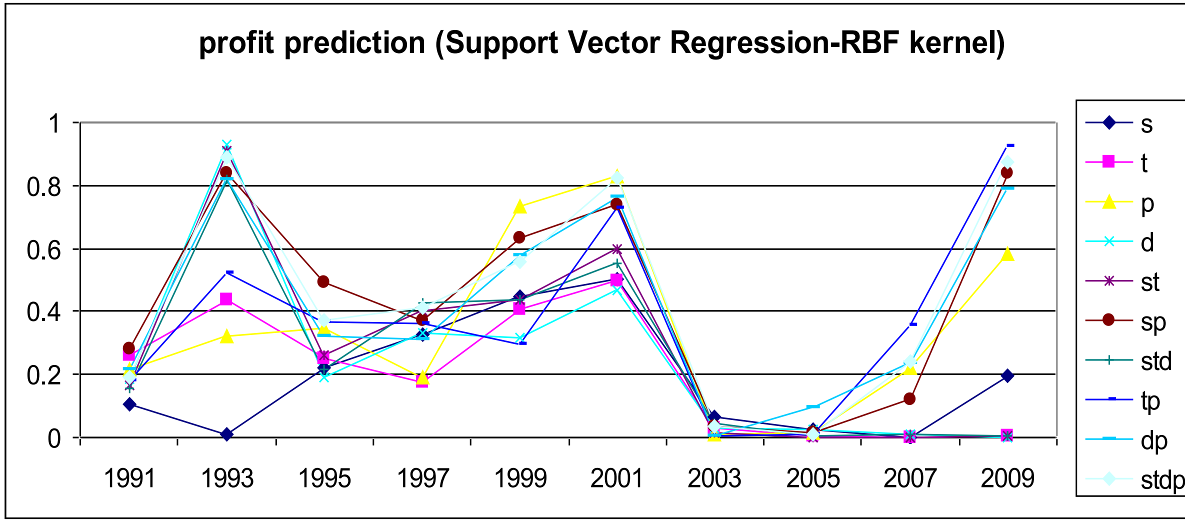


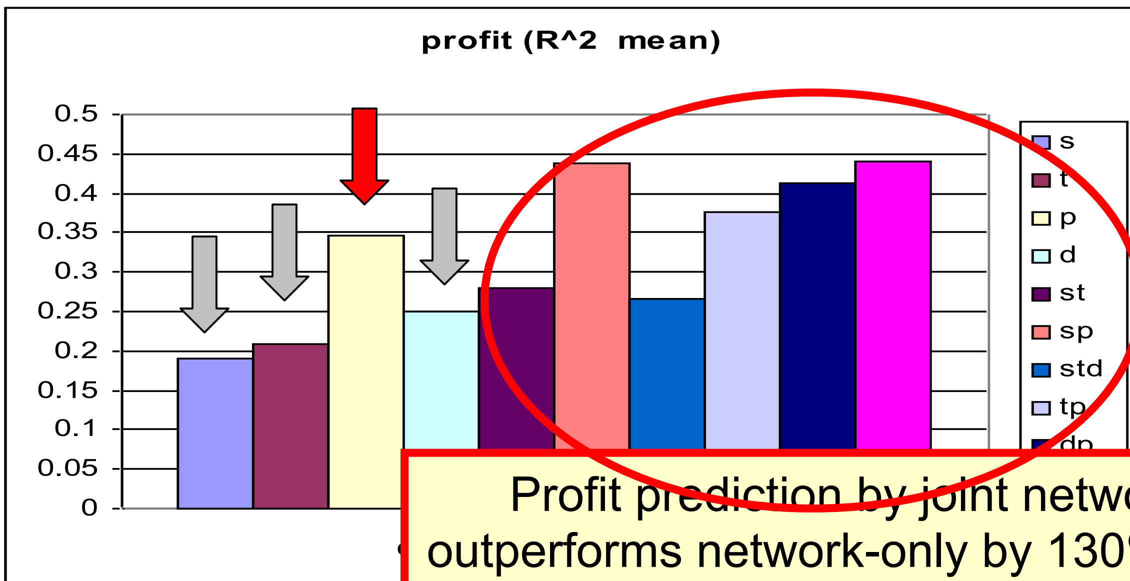
Profit Prediction using different feature sets (SVR)

Targets: 20 Fortune companies' normalized Profits

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

Model: Support Vector Regression (RBF kernel)





Network feature:
s (current year network feature),
t (temporal network feature),
d (delta value of network feature)
Financial feature:
p (historical profits and revenues)

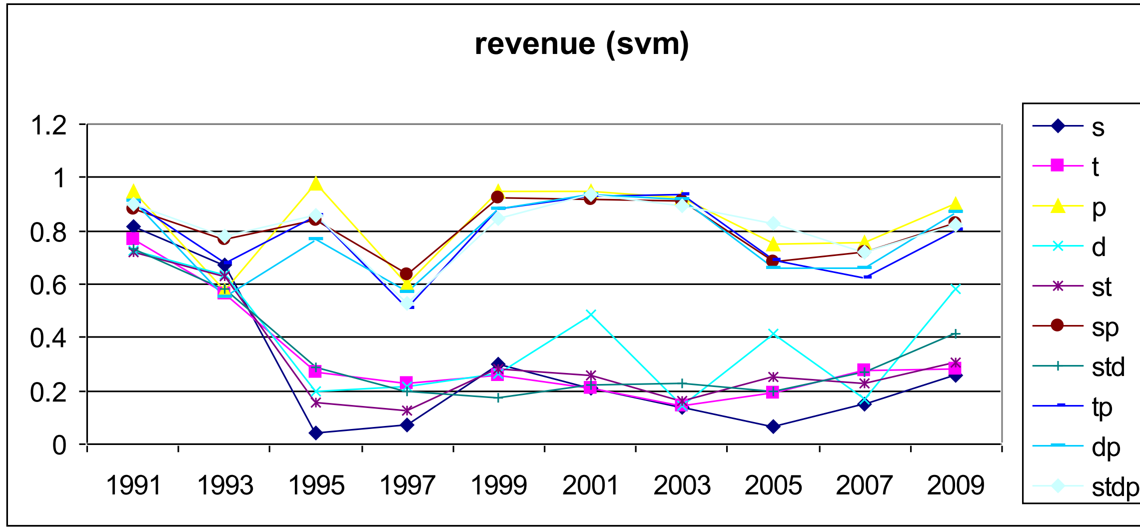
Profit prediction by joint network and financial analysis outperforms network-only by 130% and financial-only by 33%.

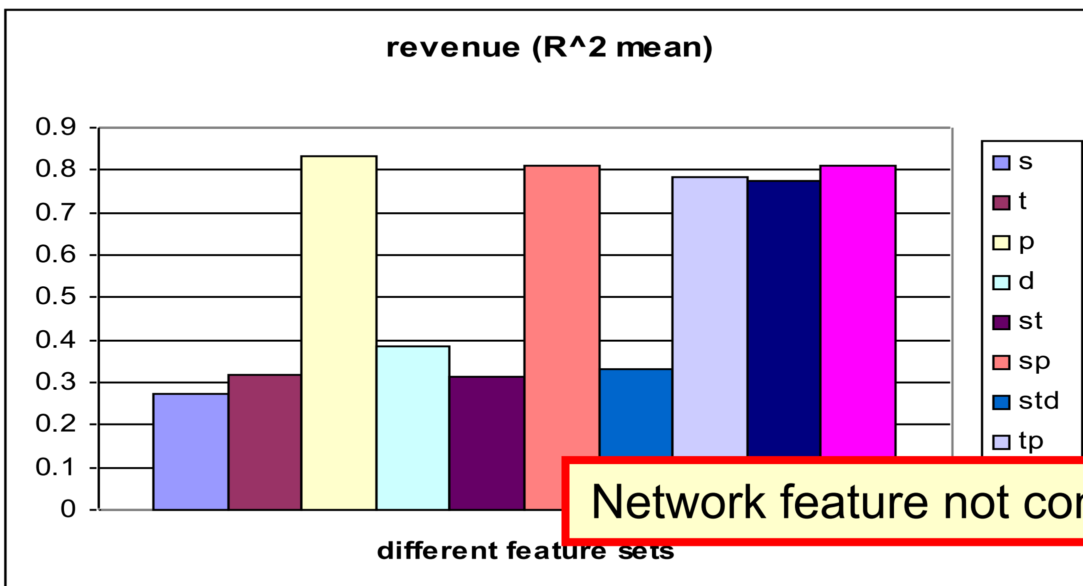
Revenue Prediction using different feature sets (SVR)

<u>Targets</u>: 20 Fortune companies' normalized Profits

<u>Goal:</u> Learn from previous 5 years, and predict next year

<u>Model</u>: 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)

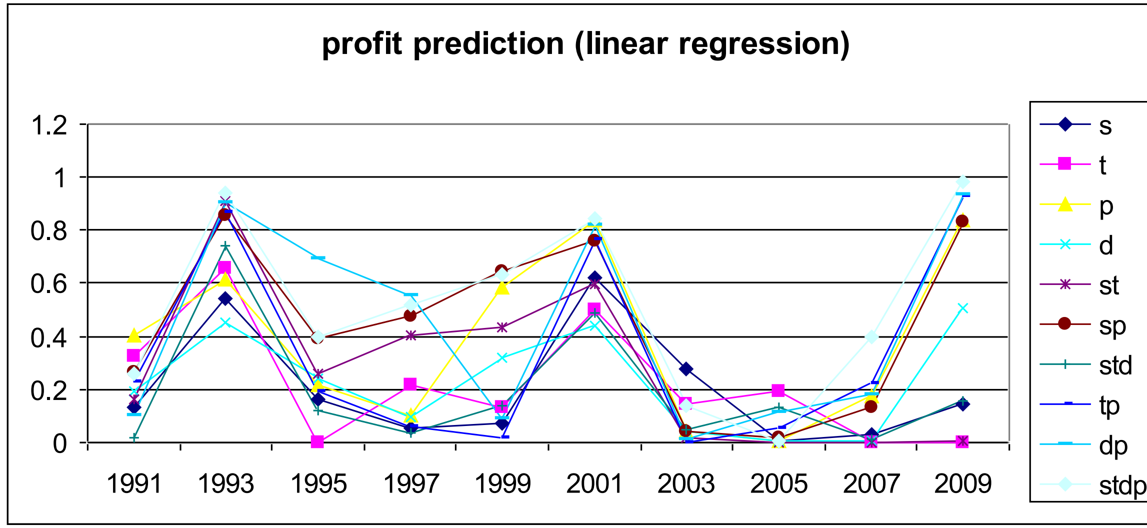
Network feature not contribute to revenue prediction.

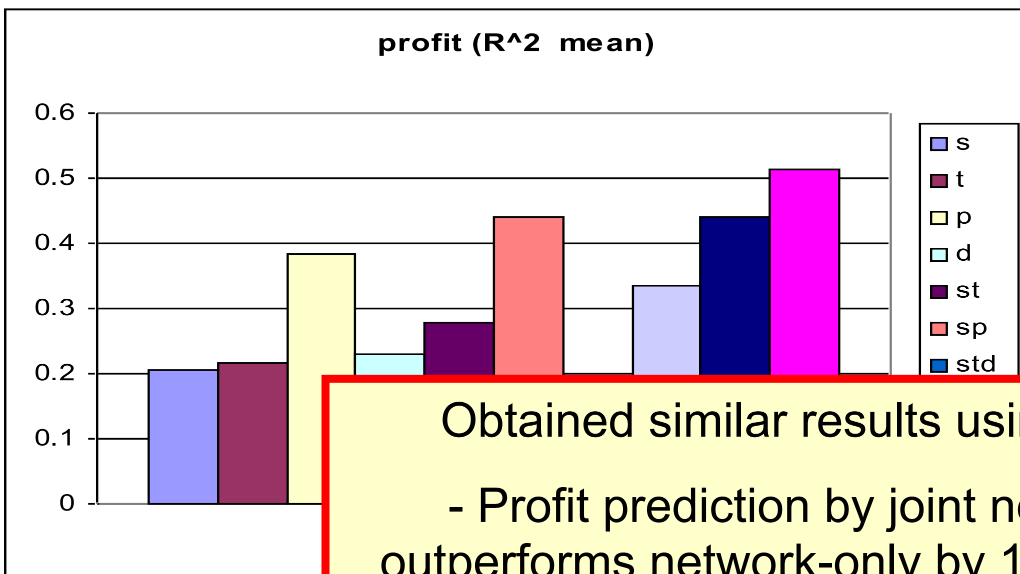
Profit Prediction (Linear Regression)

<u>Targets</u>: 20 Fortune companies' normalized Profits

<u>Goal:</u> Learn from previous 5 years, and predict next year

<u>Model</u>: 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.

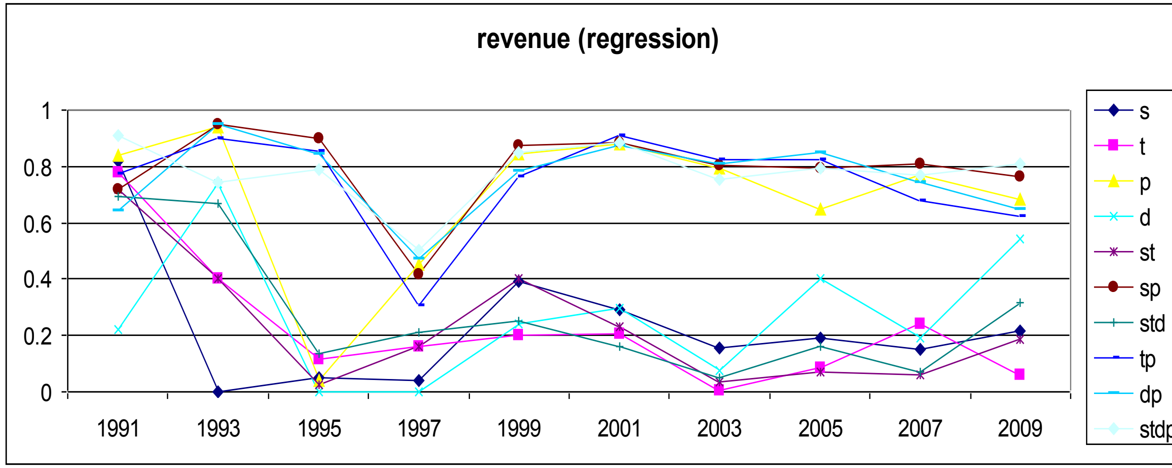
- Profit prediction by joint network and financial analysis outperforms network-only by 150% and financial-only by 37%.

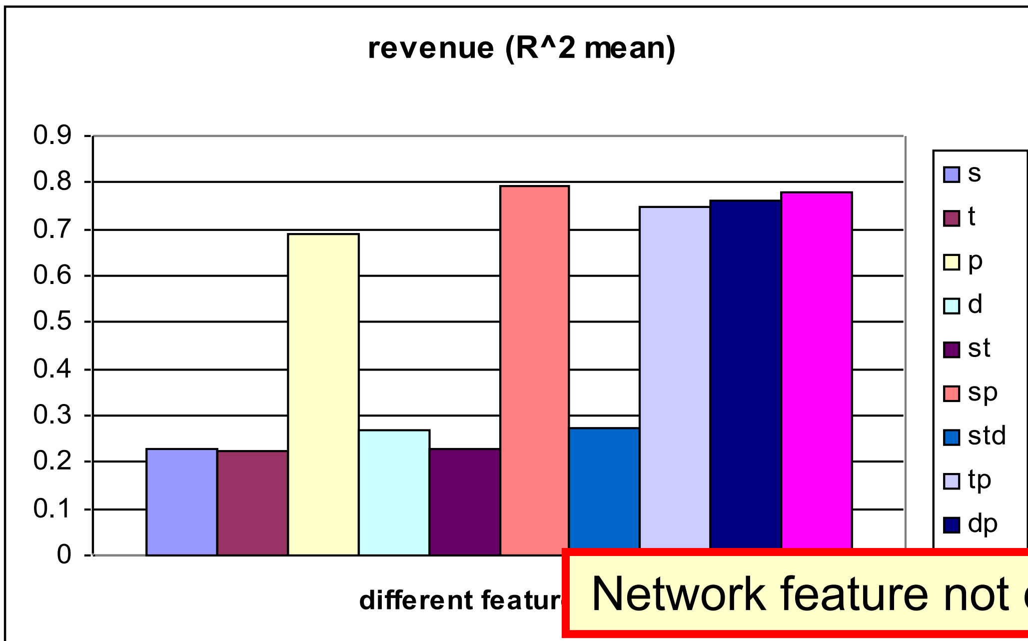
Revenue Prediction (Linear Regression)

<u>Targets</u>: 20 Fortune companies' normalized Profits

<u>Goal:</u> Learn from previous 5 years, and predict next year

<u>Model</u>: linear regression





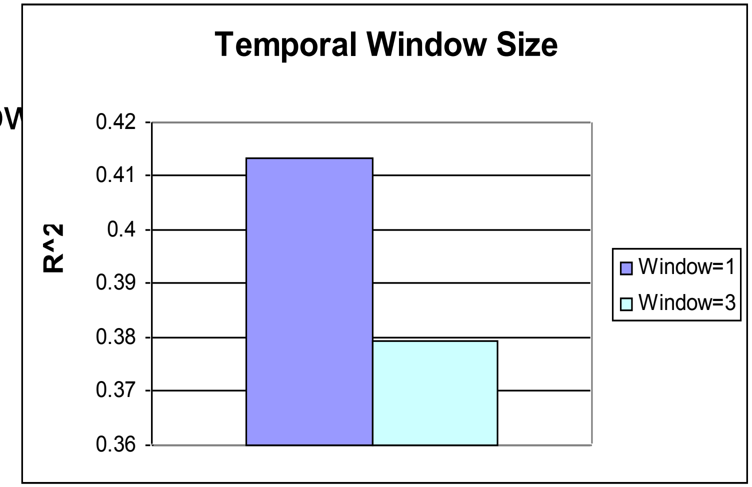
Network feature:
- s (current year network feature),
- t (temporal network feature),
- d (delta value of network feature)

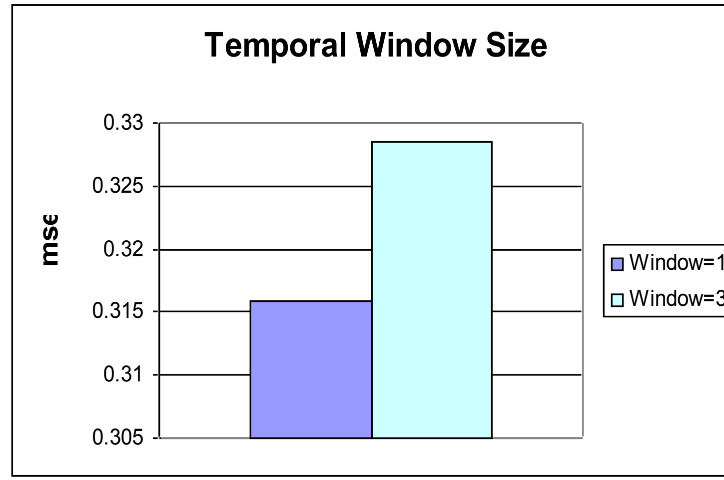
Financial feature:
- p (historical profits and revenues)

Network feature not contribute to revenue prediction.

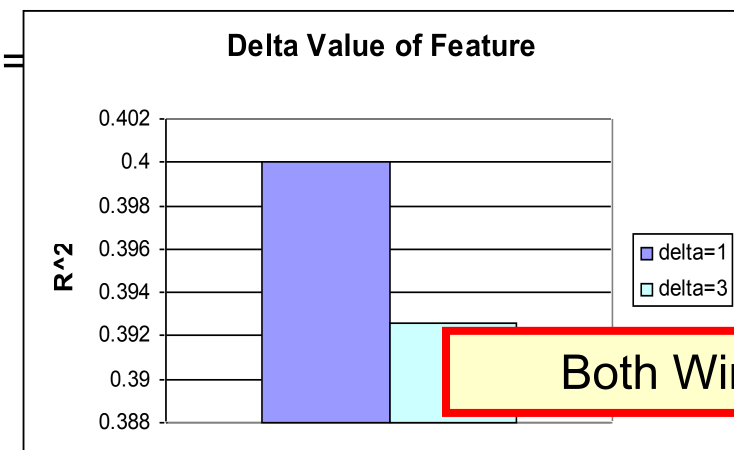
Temporal Window and Delta for Profit Prediction

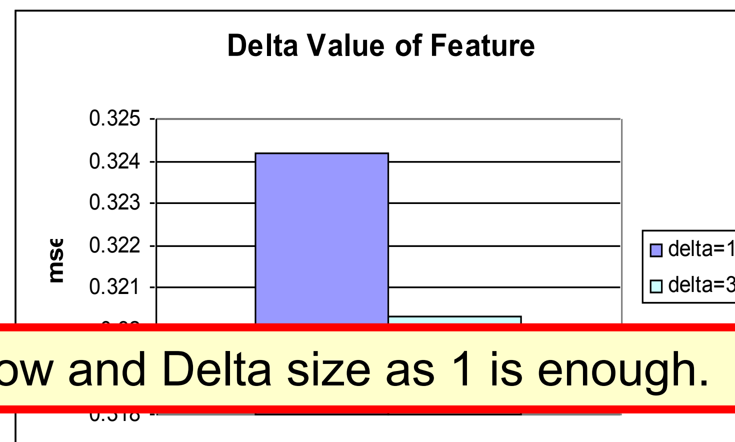
- Window





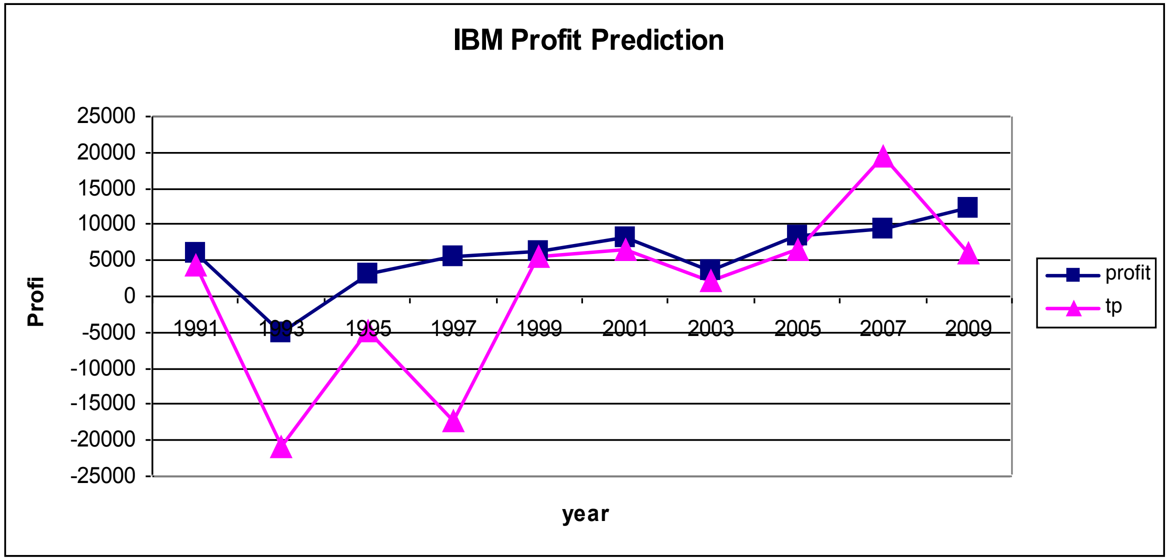
- Delta=



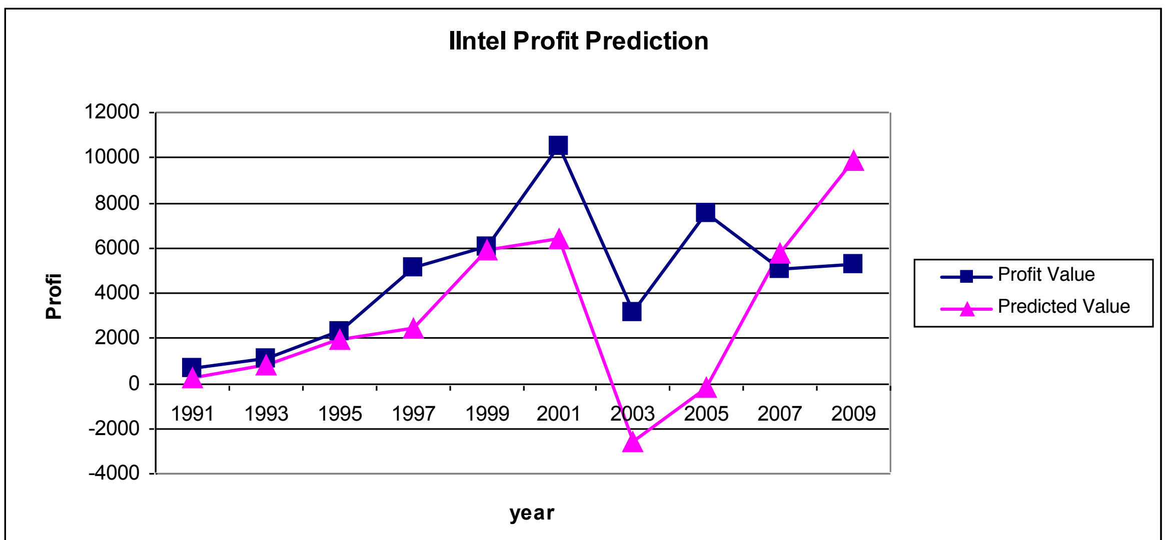


Both Window and Delta size as 1 is enough.

Profit Prediction for IBM and Intel



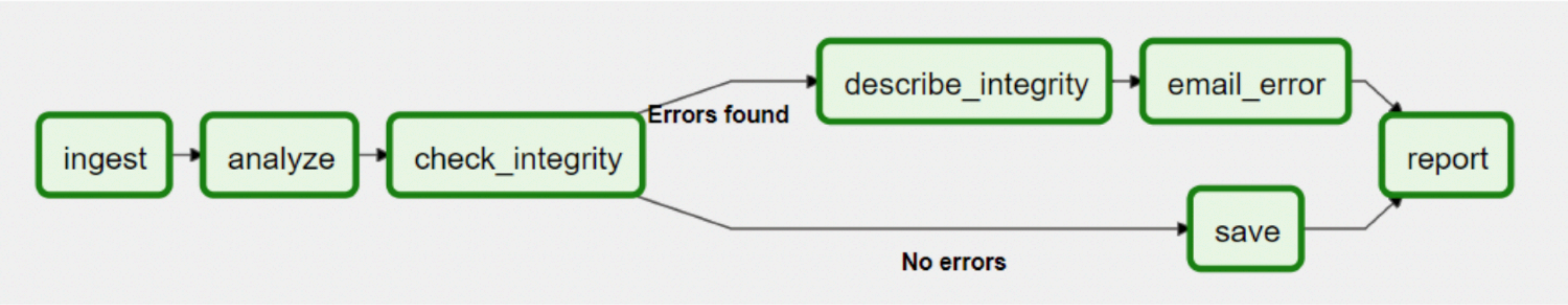
$R^2 = 0.583$



$R^2 = 0.237$

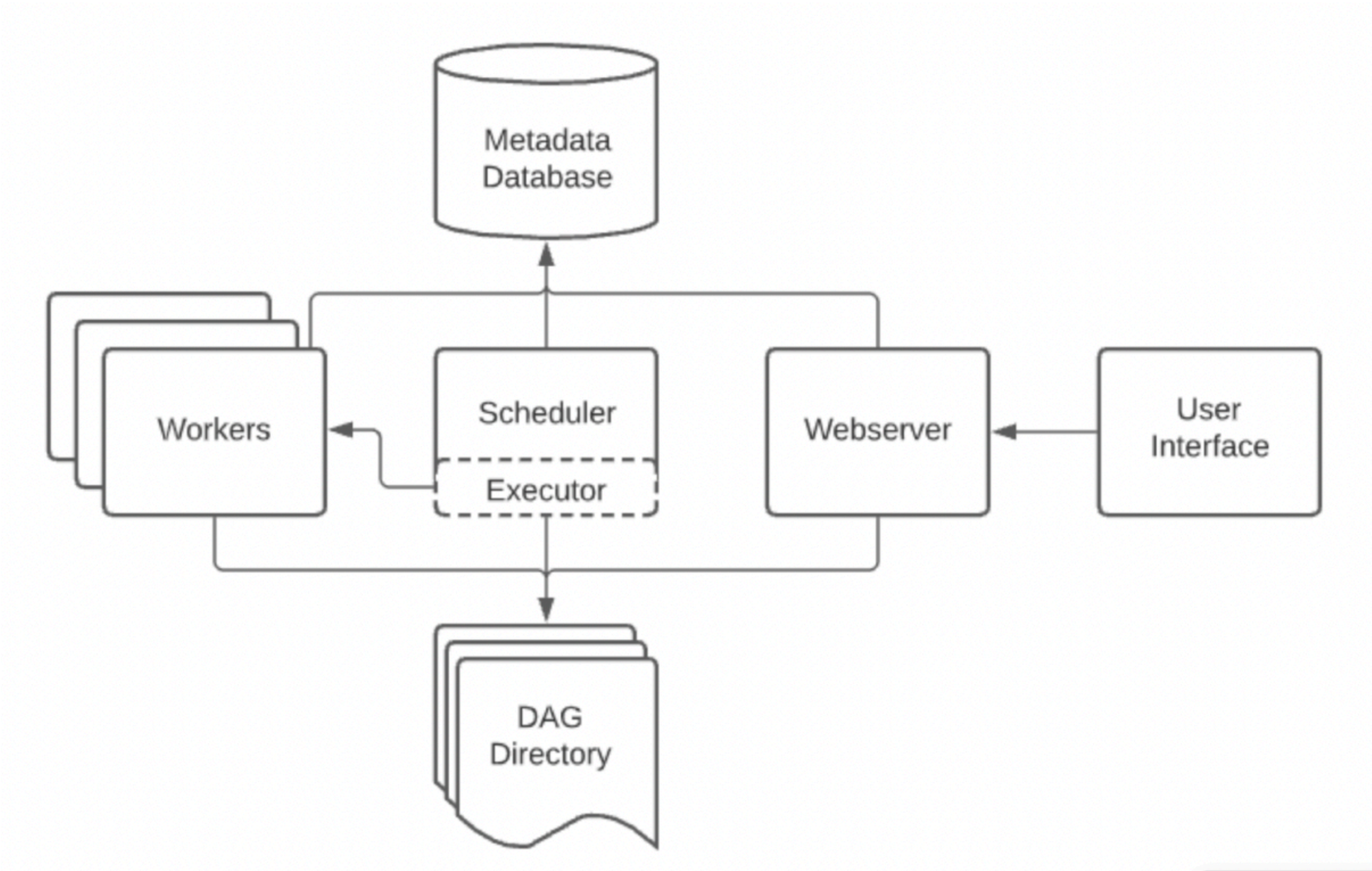
System Workflow

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-suitable 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

Apache Airflow

Airflow is a platform created by the community to programmatically 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

TaskFlow 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.

```
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 {externa
    }

email_info = compose_email(get_ip())

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

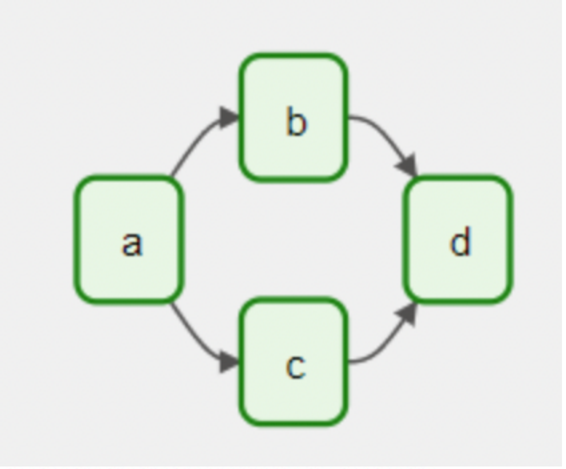
Control Flow

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
```

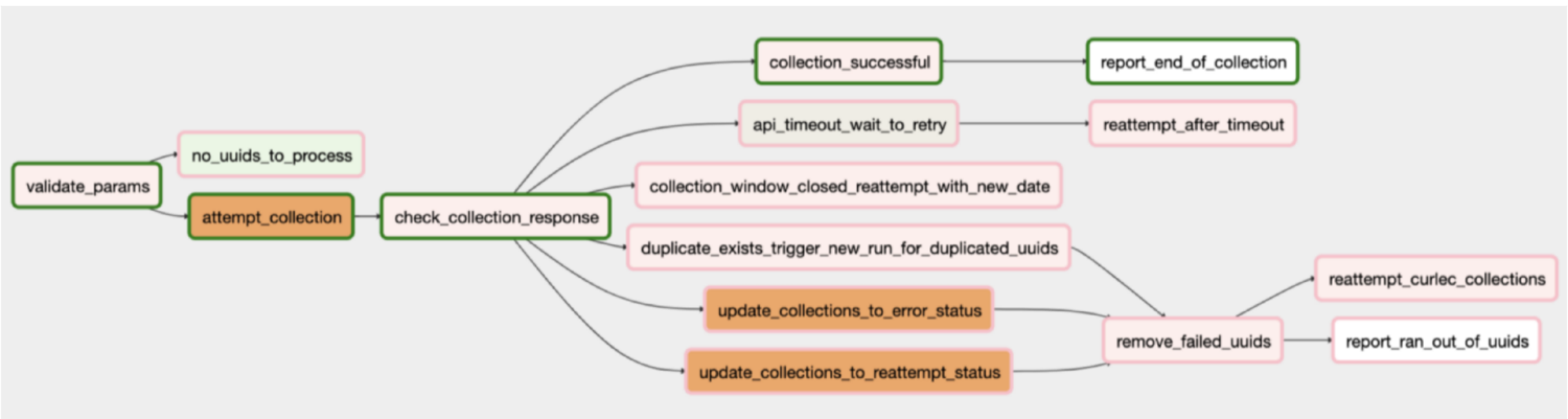


Or, with the `set_upstream` and `set_downstream` methods:

```
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.

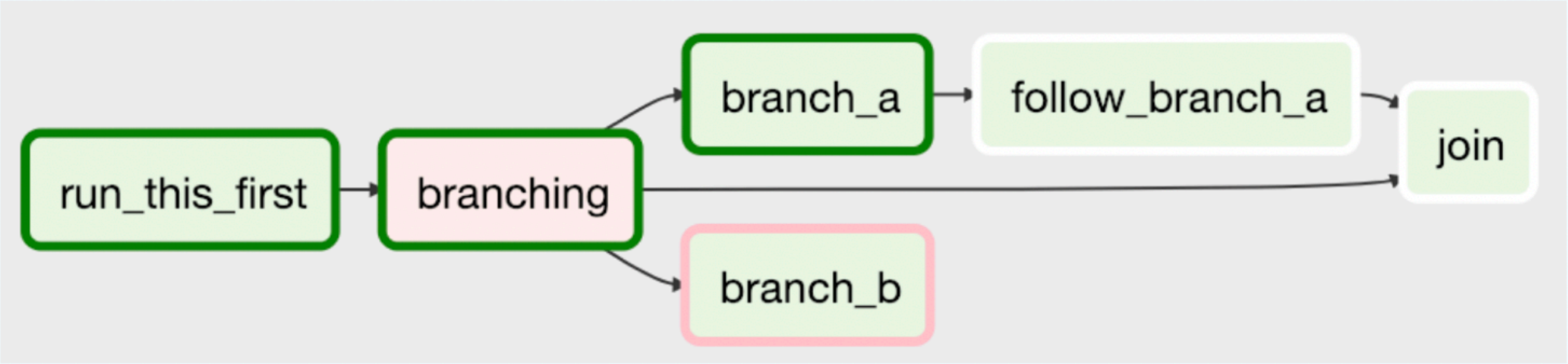


Full DAG for handling money collection

Branching

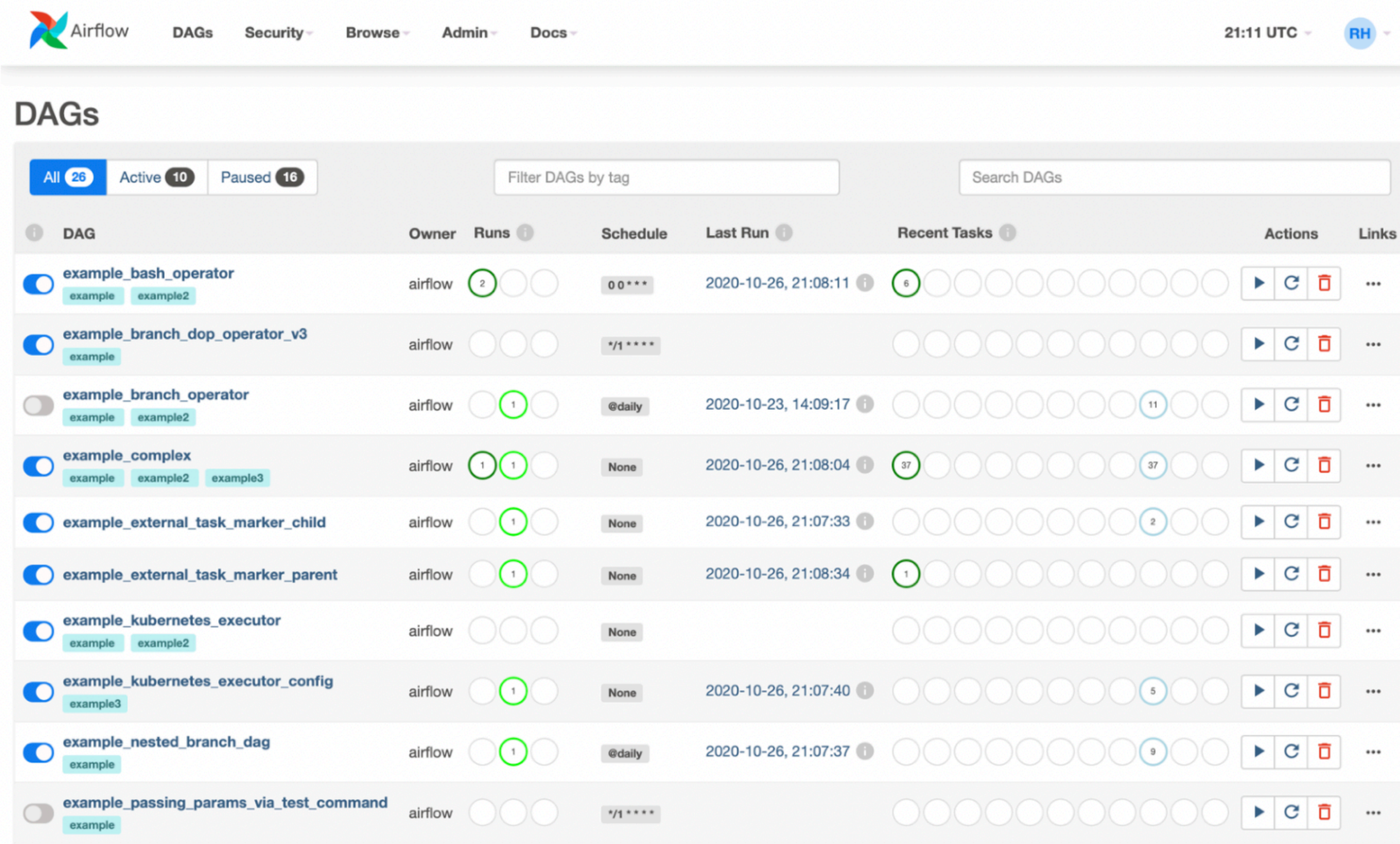
Note

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:

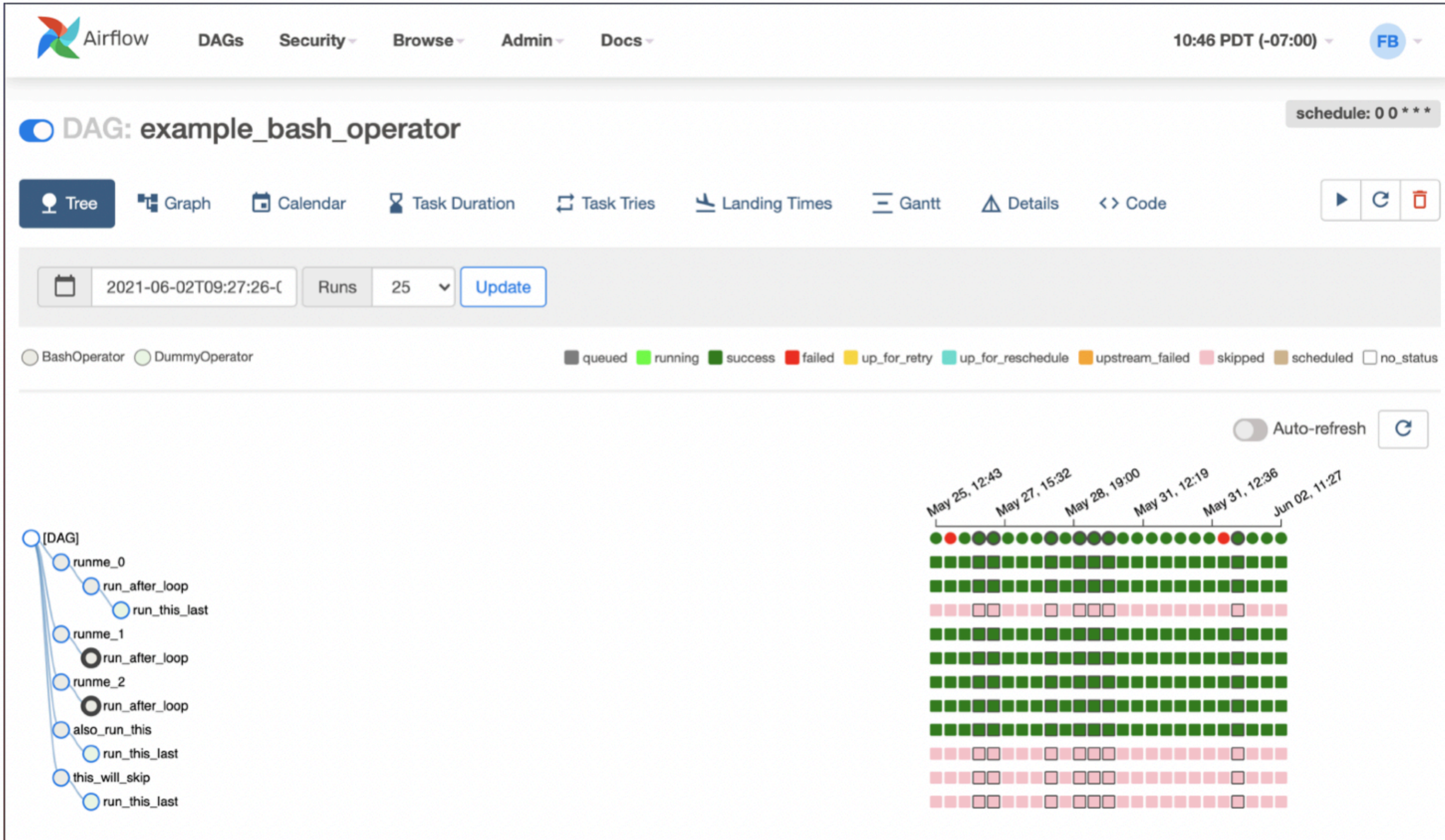


The paths of the branching task are `branch_a`, `join` and `branch_b`. Since `join` is a downstream task of `branch_a`, it will be still be run, even though it was not returned as part of the branch decision.

Airflow User Interface

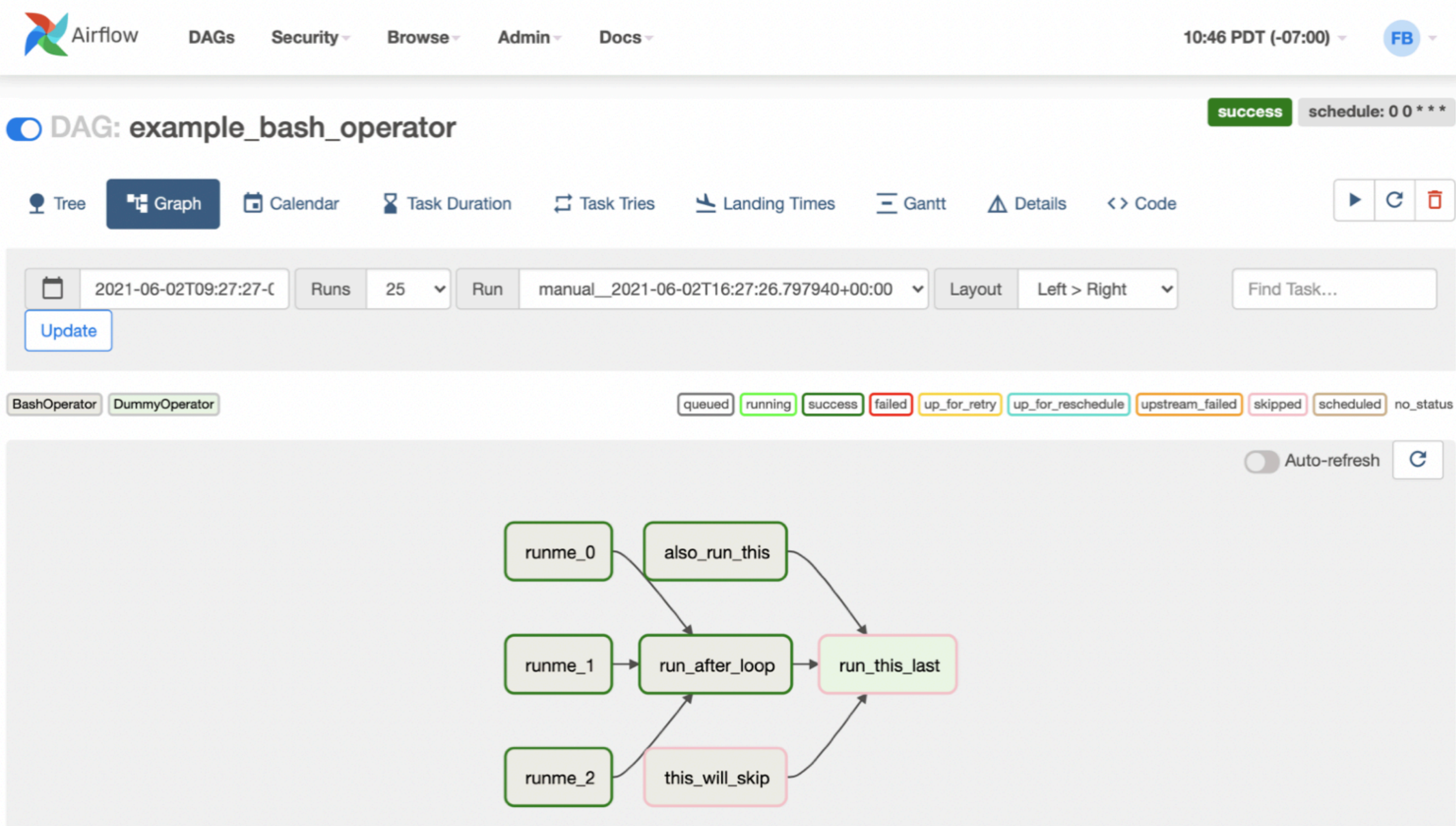


Tree View

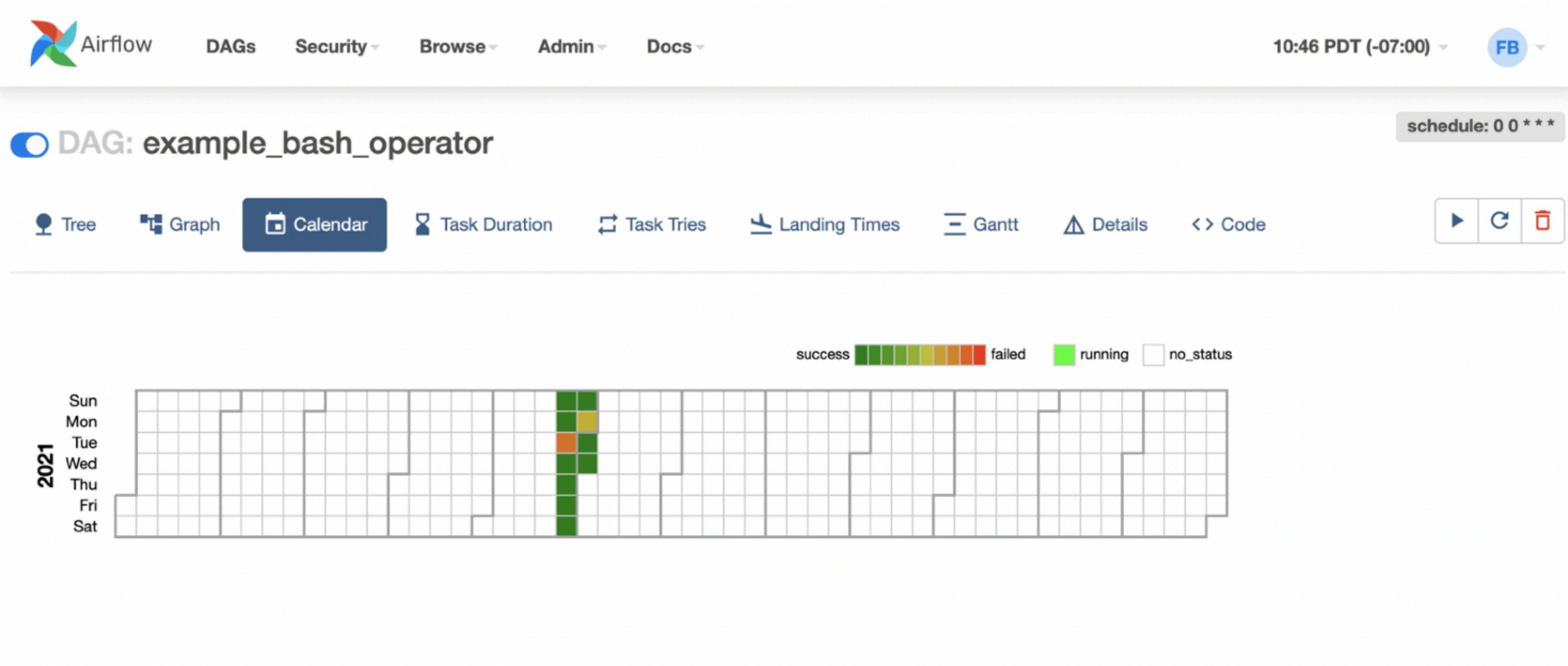


Airflow
DAGs
Security
Browse
Admin
Docs
10:46 PDT (-07:00)
FB
DAG: example_bash_operator
schedule: 0 0 * * *
Tree
Graph
Calendar
Task Duration
Task Tries
Landing Times
Gantt
Details
Code
2021-06-02T09:27:26-(
Runs
25
Update
BashOperator
DummyOperator
queued
running
success
failed
up_for_retry
up_for_reschedule
upstream_failed
skipped
scheduled
no_status
Auto-refresh
May 25, 12:43
May 27, 15:32
May 28, 19:00
May 31, 12:19
May 31, 12:36
Jun 02, 11:27
[DAG]
runme_0
run_after_loop
run_this_last
runme_1
run_after_loop
runme_2
run_after_loop
also_run_this
run_this_last
this_will_skip
run_this_last

Graph View


Airflow
DAGs
Security
Browse
Admin
Docs
10:46 PDT (-07:00)
FB
DAG: example_bash_operator
success
schedule: 0 0 * * *
Tree
Graph
Calendar
Task Duration
Task Tries
Landing Times
Gantt
Details
Code
2021-06-02T09:27:27-(
Runs
25
Run
manual__2021-06-02T16:27:26.797940+00:00
Layout
Left > Right
Find Task...
Update
BashOperator
DummyOperator
queued
running
success
failed
up_for_retry
up_for_reschedule
upstream_failed
skipped
scheduled
no_status
Auto-refresh
runme_0
also_run_this
runme_1
run_after_loop
run_this_last
runme_2
this_will_skip

Calendar View



Variable View

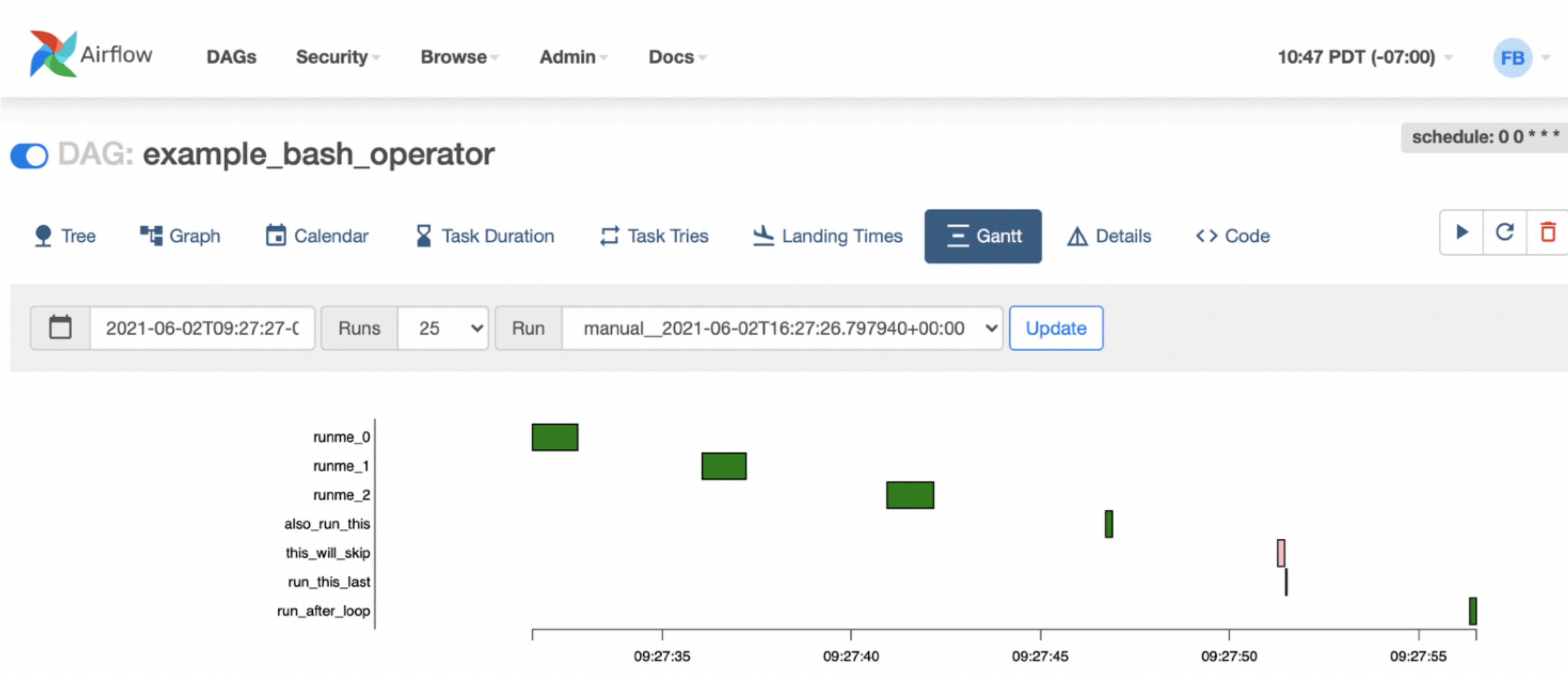
List Variable

Search

Actions | Record Count: 6

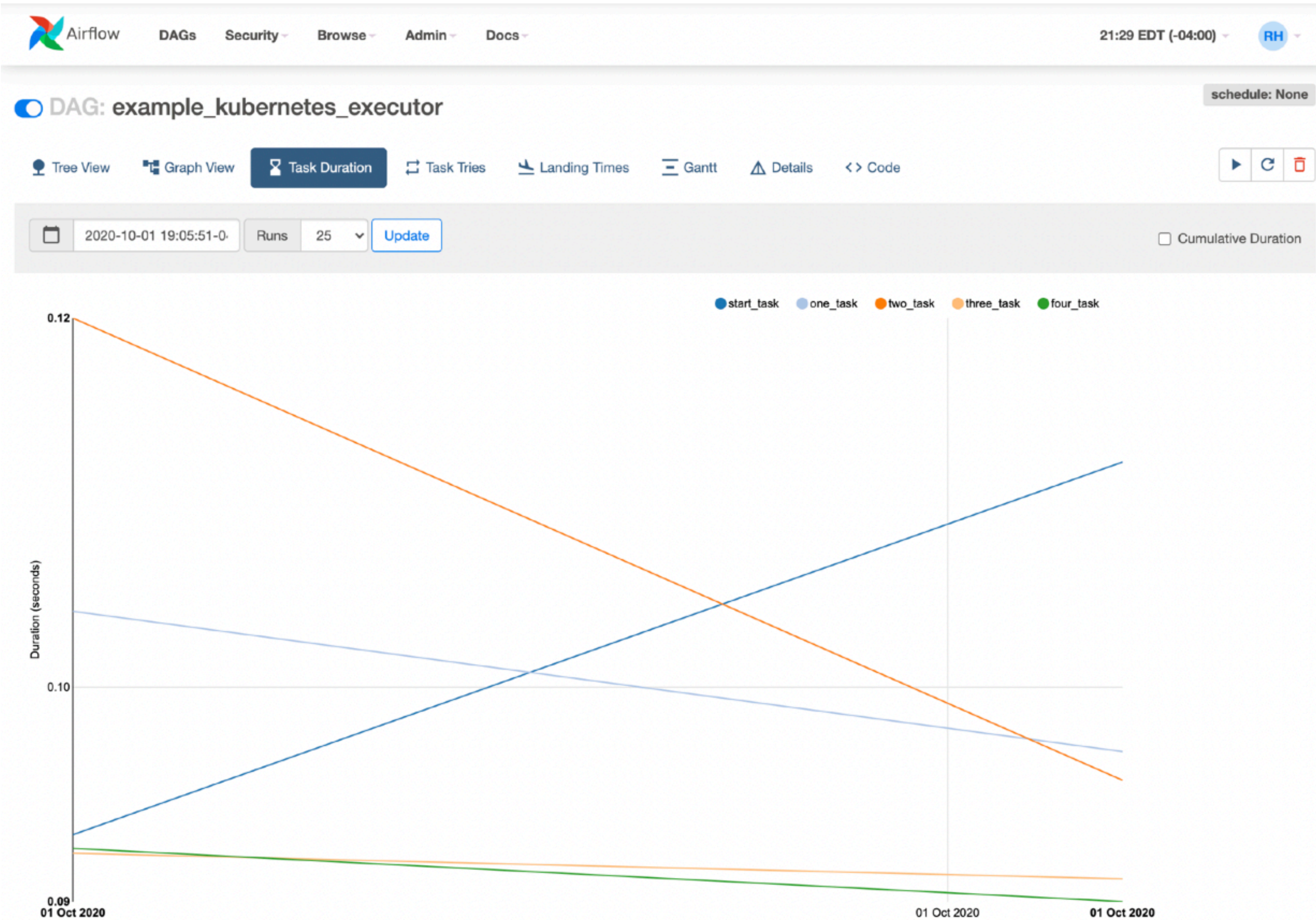
	Key	Val	Is Encrypted
	airtable_api_key	********	True
	airtable_base_key	appzasdasdasdas	True
	environment	prod	True
	pipedrive_env	pipedrive	True
	postgres_env	prod	True
	snowflake_password	********	True

Gantt chart

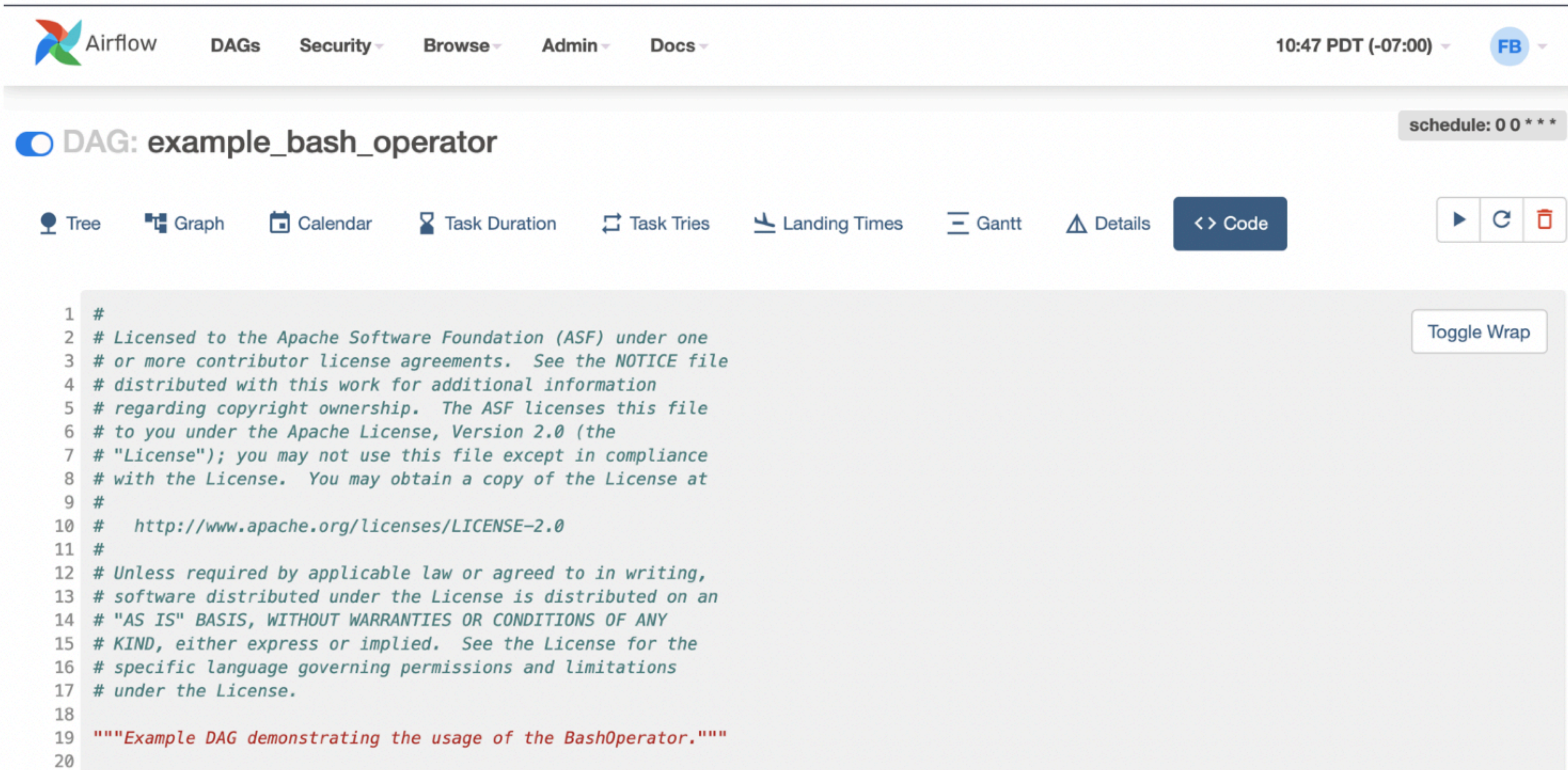


Airflow
DAGs
Security
Browse
Admin
Docs
10:47 PDT (-07:00)
FB
DAG: example_bash_operator
schedule: 0 0 * * *
Tree
Graph
Calendar
Task Duration
Task Tries
Landing Times
Gantt
Details
Code
2021-06-02T09:27:27-(
Runs
25
Run
manual__2021-06-02T16:27:26.797940+00:00
Update
runme_0
runme_1
runme_2
also_run_this
this_will_skip
run_this_last
run_after_loop
09:27:35
09:27:40
09:27:45
09:27:50
09:27:55

Task Duration

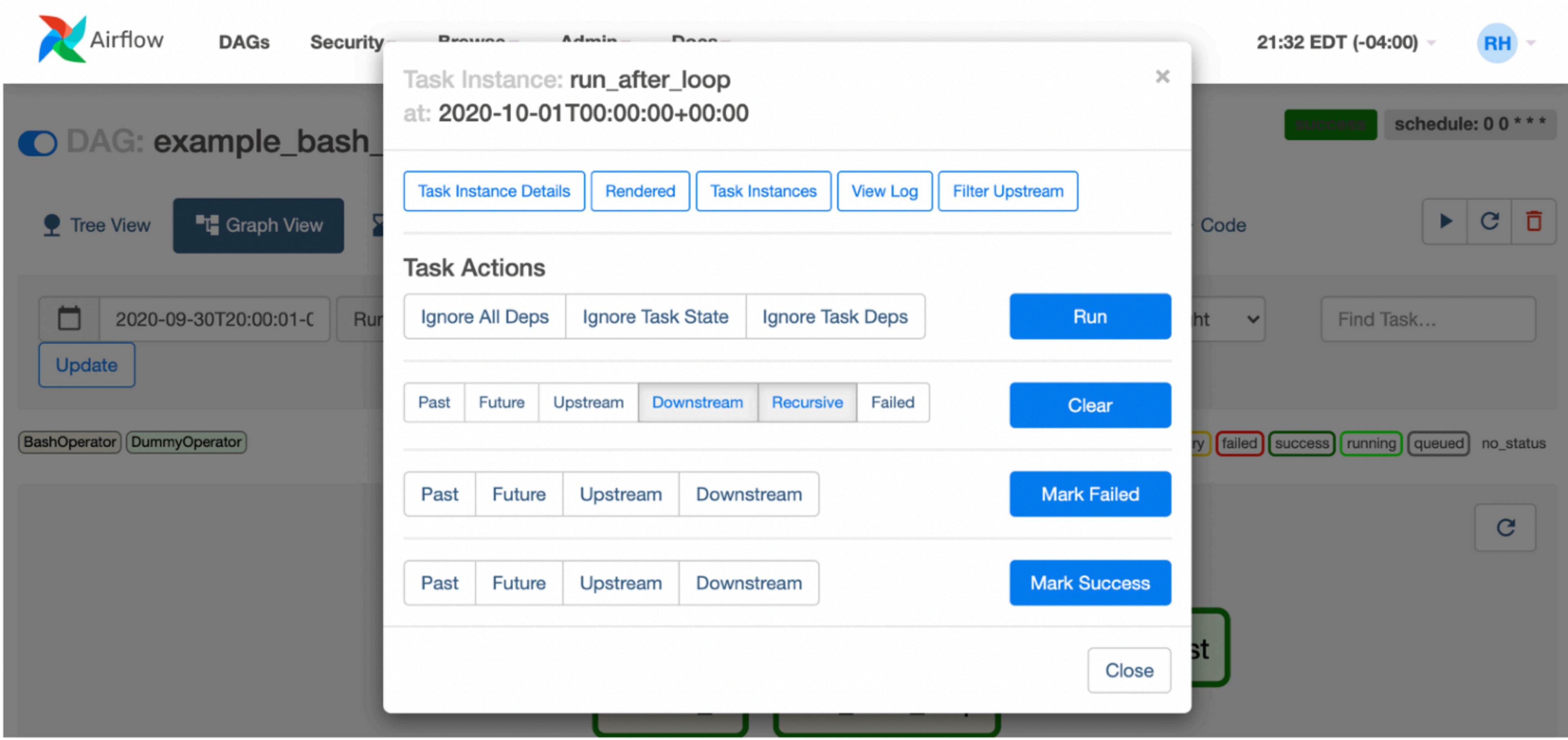


Code view

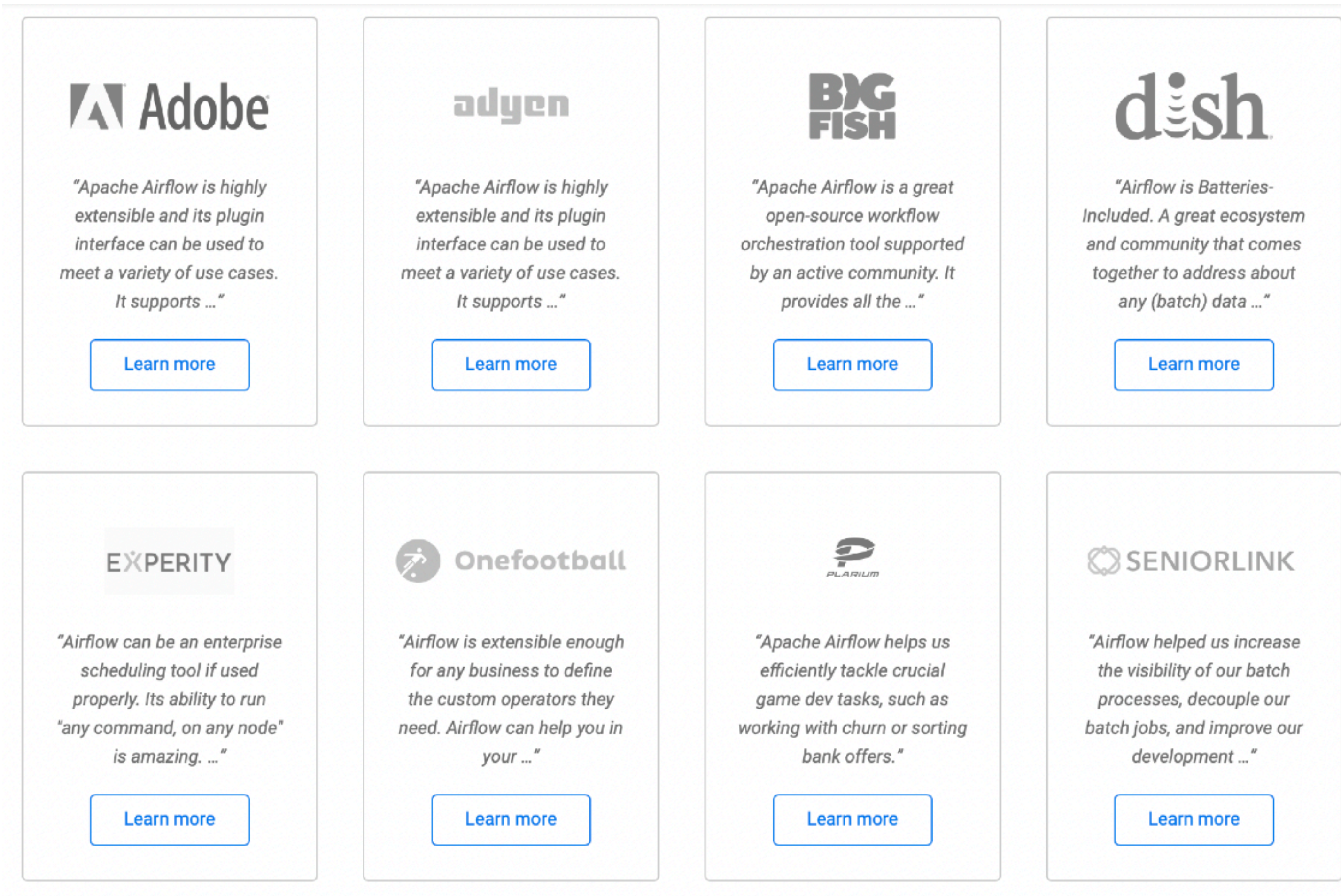


Airflow
DAGs
Security
Browse
Admin
Docs
10:47 PDT (-07:00)
FB
DAG: example_bash_operator
schedule: 0 0 * * *
Tree
Graph
Calendar
Task Duration
Task Tries
Landing Times
Gantt
Details
Code
Toggle Wrap
#
Licensed to the Apache Software Foundation (ASF) under one
or more contributor license agreements. See 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



End-to-End Workflow Management is Important



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

More details to follow in today's TA tutorial session of Homework #2

Questions?