# EMPLOYEE TURNOVER PREDICTION USING MACHINE LEARNING BASED METHODS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ZEHRA ÖZGE KISAOĞLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2014

Approval of the thesis:

**EMPLOYEE TURNOVER PREDICTION USING MACHINE LEARNING BASED METHODS**

submitted by **ZEHRA ÖZGE KISAOĞLU** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Adnan Yazıcı _____
Head of Department, **Computer Engineering**

Assoc. Prof. Dr. Pınar Karagöz _____
Supervisor, **Computer Engineering Department, METU**

Dr. Berkant Barla Cambazoğlu _____
Co-supervisor, **Yahoo Labs, Barcelona**

**Examining Committee Members:**

Prof. Dr. İsmail Hakkı Toroslu
Computer Engineering Department, METU _____

Assoc. Prof. Dr. Pınar Karagöz
Computer Engineering Department, METU _____

Prof. Dr. Ahmet Coşar
Computer Engineering Department, METU _____

Prof. Dr. Özgür Ulusoy
Computer Engineering Department, Bilkent University _____

Assist. Prof. Dr. İsmail Sengör Altıngövde
Computer Engineering Department, METU _____

**Date:** _____

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last Name: ZEHRA ÖZGE KISAOĞLU

Signature :

# ABSTRACT

EMPLOYEE TURNOVER PREDICTION USING MACHINE LEARNING BASED METHODS

Kısaoğlu, Zehra Özge

M.S., Department of Computer Engineering

Supervisor : Assoc. Prof. Dr. Pınar Karagöz

Co-Supervisor : Dr. Berkant Barla Cambazoğlu

September 2014, 100 pages

Employee turnover is a major problem for many companies because it brings with new issues including hiring costs, overtime costs, low productivity. Hence, preventing or reducing turnovers is a challenging task in human resource management field. At this point, employee turnover prediction plays an important role in providing early information for highly probable turnovers in near future that enables companies to take precautions against this situation. In this thesis, we work on this problem to predict whether an employee will leave his/her company within a certain time period. We formulate this binary classification problem as a supervised machine learning problem. Our study exploits publicly available employee profiles taken from the Web and job transition graphs extracted from these profiles. Main contribution of our study on predicting turnovers is the forming and use of job transitions of employees as well as the publicly available information about employees and institutions. So far, most of the turnover prediction models are built with the statistical methods or data analysis techniques and they make use of detailed employee information like age, race, job performance in company or job satisfaction survey results. To the best of our knowledge, this is the first study on predicting turnovers using job transitions and machine learning methods. With the help of job transition graph analysis and relevant features extracted from the graphs, many machine learning models under the change of year and time period parameters are composed. Several experiments with several

models on different years' employee profiles indicate that our proposed models have considerably predictive capabilities compared to different baselines.

Keywords: Employee Turnover, Employee Turnover Prediction, Job Transition, Machine Learning, Binary Classification

# ÖZ

MAKİNE ÖĞRENİMİ TABANLI YÖNTEMLERLE ÇALIŞANLARIN İŞTEN AYRILMA TAHMİNİ

Kısaoğlu, Zehra Özge
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Doç. Dr. Pınar Karagöz
Ortak Tez Yöneticisi : Dr. Berkant Barla Cambazoğlu

Eylül 2014 , 100 sayfa

Personellerin işten ayrılması birçok şirket için önemli bir problemdir. Çünkü bu durum işe alım maliyetleri, fazla mesai ücretleri, düşük verimlilik gibi yeni sorunları da beraberinde getirir. Bu nedenle işten ayrılmaları önlemek ya da en aza indirmek insan kaynakları yönetimi alanında büyük önem taşıyan bir meseledir. Bu noktada iş bırakmaların önceden tahmini yakın gelecekteki olası personel kayıplariyla ilgili erkenden bilgi vermesi açısından önemli bir role sahiptir, bu sayede şirketler bu durum karşısında önlemlerini alabilirler. Yaptığımız tez çalışmasında bu problem üzerinde yoğunlaşılarak çalışanların mevcut işlerini belli bir zaman diliminde bırakıp bırakmayacağı tahmin edilmeye çalışılmıştır. İkili sınıflandırma problemi olan bu problem gözetimli makine öğrenimi problemi olarak ifade edilip formülleştirilmiştir. Bu çalışmada Web üzerinden erişilebilir mevcut çalışan profillerinden ve bu profillerden oluşturulan iş geçiş/değiştirme ağlarından faydalanılmıştır. Çalışmamızın bahsi geçen probleme olan en büyük katkısı Web üzerinden erişilebilir çalışan ve kurum temel bilgileriyle beraber çalışanların iş geçişlerinin oluşturulup problemin çözümü için kullanılıyor olmasıdır. Şimdiye kadarki işten ayrılma tahmini yapan modellerin çoğu istatistiksel yöntemlerle ya da veri analizi teknikleriyle oluşturulmuşlardır. Aynı zamanda bu modeller çalışanların yaşı, etnik kökeni, çalıştığı şirkette işindeki performansı ya da çalışan memnuniyeti anketindeki sonuçları gibi daha detaylı, Web üzerinden doğrudan erişilemeyen özellikleri kullanarak oluşturulmuşlardır. Bu anlamda

iş geçişlerini ve makine öğrenimi yöntemlerini kullanarak çalışanların işten ayrılma tahminlerini yapan başka bir çalışma bulunamamıştır. İş geçişleri ağı analizi ve bu ağlardan çıkarılan ilgili özellikler yardımıyla yıl ve zaman dilimi parametreleri değiştirilerek birçok makine öğrenimi modeli oluşturulmuştur. Oluşturulan birçok model ile değişik yıllardaki çalışan profilleri üzerinde yapılan deneyler bu tez çalışmasında önerilen modellerin referans olarak kabul ettiğimiz, baz aldığımız degişik temellere göre önemli ölçüde tahmin etme yeteneği olduğunu göstermektedir.

Anahtar Kelimeler: Çalışanların İşten Ayrılması, Çalışanların İşten Ayrılma Tahmini, İş Geçişi, Makine Öğrenimi, İkili Sınıflandırma

*To my family, especially my mother and my spouse*

# ACKNOWLEDGMENTS

I express my sincere appreciation to my supervisor, Assoc. Prof. Pınar Karagöz, for her guidance, support, encouragement and positive attitudes during my master studies.

I also would like to express my gratitude to Dr. Berkant Barla Cambazoğlu, who is the idea owner of this thesis topic, for his support, guidance, invaluable ideas and critical thinking for the approach to the problem.

I am very grateful and would like to thank my family, for their invaluable patience, encouragement, endless support and unconditional love.

Finally, I would like to thank TÜBİTAK (Scientific and Technological Research Council of Turkey) for providing financial support during my thesis.

# TABLE OF CONTENTS

# LIST OF TABLES

TABLES

# LIST OF FIGURES

FIGURES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| Inst | Institution |
| fw | Future Window Parameter in years |
| SVM | Support Vector Machine |
| RBF | Radial Basis Function |
| DTNB | Decision Table/Naive Bayes Hybrid Classifier |
| DT | Decision Table |
| NB | Naive Bayes |
| ANN | Artificial Neural Network |

# CHAPTER 1

# INTRODUCTION

## 1.1 Motivation

Willing to keep and protect our health, our wealth or all the other things what we have own is in human nature. It is the same situation for the companies as well. Companies' products & technologies, customers, employees are all considered as their assets and companies want to protect them. However, products might be deprecated one day, customers might decide to get service from another company and employees might begin to seek for other opportunities and decide to leave current companies. Being able to predict what is going to happen to especially customer base and employees is so important that companies can take precautions even before the "churn" occurs.

The term "churn rate" is defined as the measure of the number of the items moving out in a collection over a specific period of time. When applied to customer base, it refers to the proportion of contractual customers or subscribers who discontinue receiving service or switch their service provider. Churn problem of customers are investigated in many different studies, especially in wireless telecommunication industry where customer and service provider possibly have a long term relationship (see Section 2.1).

Besides the churn problem of customers, employee turnover is another major problem that companies face. "Turnover" is commonly used term and it is more suitable than "churn" for the employees who leave a workforce and are replaced. Employee turnover problem brings with new issues for companies including hiring costs, training costs, low productivity, not being able to meet deadlines. Hence, to predict the

employee turnovers is as important as the prediction of customers' churn. There are vast amount of studies made in different fields like psychology, business management or economics to determine the factors that influence employees to retain or leave (see Section 2.2). These studies show some of the influencing factors of turnovers; demographics of the employee like age and gender, work environment factors like salary, position and work hours etc. Most of these studies generally make an analysis of the reasons and factors behind the turnovers and they try to predict the employee turnovers using statistical methods and data analysis techniques.

In this thesis, we study on the employee turnover problem. Our problem is to predict whether the employees will leave their current companies within the specific time interval. It is formulated as a binary classification problem which classifies employees as who "will leave" (turnover) and who "will not leave" (no-movement). We investigate several attributes including the traditional employee features as well as employees' job experiences. We believe that our system can successfully exploit the job transitions performed by all employees. Hence, job transitions according to past job histories of the employees are generated for each year from *2000* to *2010* and job transition networks' properties are also used for prediction.

Many models are proposed for different years based on supervised machine learning. Given a currently working employee, the common objective of the learning models is to accurately predict the employee turnover within certain time period by considering the past features and job transitions of the employee. Using publicly available employee profiles for each year, evaluations of different year models on different test years are performed. The results of our experiments demonstrate that our proposed models using job transitions have considerably predictive capabilities compared to different baselines. Moreover, our results indicate that current company features are more important in predicting turnover of the employee than the features of the past companies.

## 1.2 Contributions and Organization of Thesis

Our contributions in this thesis can be explained as follows:

- We develop a framework to predict employee turnovers using machine learning methods.
- We use a new feature set that is different from traditional employee features. We make use of job transition graphs and construct each employee's job transitions for our problem. In addition to job transitions, we use only available public information of the employees and companies.
- We compare the experimental results with different baseline models.

The rest of the thesis is organized as follows. In Chapter 2, a survey of related studies on customer churn problem and employee turnover is given. Chapter 3 gives the detailed background information about the data structures, algorithms and methods used in this thesis. In Chapter 4, all phases of our methodology in this study are explained in detail. Results of the experiments are given and discussed in Chapter 5. Finally Chapter 6 summarizes and concludes the thesis with final remarks and provides pointers to future work.

As a note, in the following chapters, "institution" refers to both company and educational institution, "experience" refers to both professional and educational experience unless specified.

# CHAPTER 2

# RELATED WORK

In this chapter, related studies are given in two sections with respect to two types of "churns"; customer churn and employee turnover. The studies on customer churn problem and employee turnover problem are summarized in Section 2.1 and Section 2.2, respectively.

## 2.1 Churn of a Customer Base

Churn of a customer is one of the most common problems studied in literature. Wireless telecommunication is one of the top industries that takes churn problem into consideration. The churn rate of the top US carriers is $2.2\%$ [75] and their cost to sign a new contract is ranging around $\$300$ to $\$600$ [54, 75, 76]. Considering each customer pays $\$50$ per month, companies should at least keep their customers $6$ months to compensate their costs. These numbers indicate the importance of customer churn problem for companies.

Some of the studies [25,75,76] use Support Vector Machine over traditional methods like Decision Tree, Naive Bayes and Artificial Neural Network (ANN) for customer churn prediction problem. Use of Support Vector Machine shows good accuracy and generalization compared the other methods stated. The study [76] experiments the one-class SVM with 3 different kernel functions; Linear, Polynomial and Gaussian for the solution of the same problem. It shows that Gaussian Kernel has the best accuracy among the other ones with an accuracy of $87.15\%$. It also compares the results of ANN, Decision Tree, Naive Bayes with one-class SVM over the same dataset.

Unlike the previous studies mentioned, [54] chooses to use Logit Regression, Decision Tree, Neural Networks and [61] proposes two different Genetic Algorithm based Neural Network models to predict customer churn. In [54], researchers not only compare the behavior of different models but also propose a decision making policy. The decision making policy they propose is to offer incentive promotions to a subscriber whose churn probability is above a certain threshold. By applying their findings on real life subscribers, it is found that companies can save $\$417$ per customer with high churn probability [54].

Most of the studies in the literature about churn of a customer focus on the problem in context of a single subscriber. However, some researches investigate the relation between subscribers' social network and their churn probability [57]. They propose a new churn prediction approach named Collective Classification (CC) and they consider both subscribers' demographics and the social network that the subscriber is in. They seek to answer whether the decision of a subscriber to churn depends on their social network or not. To answer this question, they compute a churn probability from a social network database with respect to the proportion of the friends who previously churned. While doing experiments over the database, they include features for a single subscriber like age, gender, race etc. as well as features from the social network graph like number of neighbors for a node, average in/out weight of a node, average Jaccard Similarity [5] of a node and its neighbors. They conclude that Collective Classification approach provides better accuracy compared to traditional classification [57].

## 2.2 Employee Turnover

Most turnover studies in literature view employee turnover in four different types under two different categories; voluntary and involuntary turnover [8, 34].

- Involuntary turnover
  - Discharge Turnover: This type of turnover is initiated by the organization and aims at an individual employee. The reasons may be related to discipline and/or performance problems of the employee.

  - Downsizing Turnover: This type of turnover is initiated by the organization and occurs as a part of organizational restructuring (lost funding, change of work requirements, reorganization).

- Voluntary turnover
  - Avoidable Turnover: This type of turnover is initiated by the employee and could be possibly prevented by the organization.
  - Unavoidable Turnover: This type of turnover is initiated by the employee and occurs in unavoidable circumstances like retirement, death, spousal relocation.

It is important to identify the causes of the turnover for organizations so that they can take actions to prevent it. Otherwise, it costs organizations up to $100\%$ - $300\%$ of the base salary of the replaced employee, including pre-turnover costs, training costs, new-hire costs, recruiting costs etc. [53]. Considering all these costs and negative effects, there are vast amount of studies on this topic to analyze the turnover factors and prevent employee turnovers. These studies can be categorized into three buckets: Studies focused on demographics like age, gender, education, ethnicity etc., studies focused on work environment like salary, work hours, recognition, position etc., and studies focused on human resource development interventions like trainings, career development etc. [41].

In a meta-analysis and review of voluntary turnover study conducted in 1987, it is clearly stated that some of the demographic attributes of an employee like age, gender combined with some of the work environment attributes like salary, tenure, job satisfaction play the strongest role in predicting a turnover [24]. Like [24], there are some other studies indicate that variables including age, economic activity, tenure, working time in current position and education are the strongest predictors of turnover [69]. Later on, another study conducted on $46$ samples with a total of $42625$ individuals shows that the relationship between age and voluntary turnover is too small (near zero) and it concludes that age is not one of the strongest indicators of turnover [33].

There are also many other studies conducted by researchers over the years to validate the relation between work environment and turnover results or intentions. These studies consider demographic attributes of employee along with work environment

attributes like salary, job satisfaction, benefits and recognition [19, 27, 39, 44, 51, 65, 67, 69]. Each of these studies validates that job satisfaction plays the key role in employee turnover.

# CHAPTER 3

# BACKGROUND

In this chapter, background information is given for the topics relevant to our thesis study. We divide the major topics into four parts in accordance with the phases explained in Chapter 4. Job Transitions section contains information about the job transition concepts and related studies, transition graph notation and graph construction algorithm from [60]. Feature Vector Construction section includes the concepts and formulas inspired during feature calculations. In thesis study, WEKA (Waikato Environment for Knowledge Analysis) [32] and LibSVM (A Library for Support Vector Machines) [21] tools are used in feature selection and classification phases. The general concepts, algorithms and methods along with the information about these tools are provided in the last two sections.

## 3.1 Job Transitions

A job transition can be described as the movement of an employee from one job to another. While moving to another company is a job transition (external), the job change can also occur within the same company (internal) by changing the department or position. Job transition term can be interchangeable with the terms *career transition* or *career change* mostly for the voluntary cases. Involuntary job loss or transition may also be regarded as a career transition/opportunity [45].

The causes, forms and outcomes of career transitions are analyzed in different studies on business management, economics and psychology fields [9, 55, 56, 66, 68]. Expecially in career management and development, this analysis is so important

to achieve the success and fulfillment desired in an individual's career [28, 49, 63]. While some studies investigate the effects of social context/network on individuals' career changes [18, 29, 35, 38], some of them are conducted on different groups of people to determine other effects including gender, education level, age and profession [10, 64, 71, 74].

For our problem, we believe that job transitions can be exploited in order to predict employee turnovers, because each transition also corresponds to a turnover at the origin point of the transition. In this sense, a relation can be established between our problem and the problem of prediction of next institution (target point of the transition). If an employee quits his/her job voluntarily or involuntarily, we focus on only the turnover occurred for the related company, regardless of the next institution. On the other hand, the next move is the main focus of the prediction problem in addition to turnover. Predicting the next institution of an employee, in other words, recommending suitable jobs to employees are the studied topics in the literature [40, 50, 60]. Among these studies, [60] makes use of previous job transitions of the employees. We take this study as a reference for the job transition graphs construction phase.

In the following sections, graph notation and graph construction algorithm studied in [60] are presented.

### 3.1.1 Job Transition Graph Notation

Job transitions can be formed using the start and end dates of employment in principle. However, in practice, it might be more complicated construction process due to mainly two issues. First one is about the multiple employment of an individual at the same time. Second one is the unemployment of a person during some periods. Therefore, in the study [60], specific notation is accepted and some rules are defined to extract job transitions.

Job transitions of the employees are described as a directed graph $G = (I, T)$, where $I$ is the set of nodes representing institutions and $T$ is the set of edges representing employee job transitions between institutions. A directed edge $(u \rightarrow v) \in T$ is expressed as a transition from institution $u \in I$ to institution $v \in I$ for only one

employee. Every edge is associated with a quintuple $\langle a, u, v, e, s \rangle$ where $e$ and $s$ denote the end time employee $a$ finished his/her job at $u$ and the start time employee $a$ started his/her new job at $v$, respectively.

The transitions graph $G$ is directed and unweighted. It may contain self-loops because same employee can work on different positions at the same time or can change position or department within the same institution. Additionally, considering millions of employee profiles and several transitions for only one employee, there may be many parallel edges between two institutions/nodes in the graph.

### 3.1.2 Graph Construction Algorithm

Job transition algorithm explained in [60] depends on two basic conditions that should be both met in order to define a job transition $(u \rightarrow v)$ of an employee:

- The experience end time $e(u)$ of an employee at institution $u$ should be before the experience start time $s(v)$ of the employee at institution $v$, i.e., $e(u) \leq s(v)$.
- There cannot be an institution $w$ such that $s(w) > e(u)$ and $e(w) < s(v)$.

These two conditions imply that any transition to a new institution can be constructed only from the most recent previous institution(s).

The two-phase algorithm from [60] are used respecting two conditions above to create a job transition graph. For each employee profile, the algorithm phases explained in Phase 1 and Phase 2 are followed.

An example constructed job transition graph with the proposed algorithm in [60] is shown in Figure 3.1. Employment history of an employee with the timeline is shown in Figure 3.1(a). In Figure 3.1(b), there are two transitions, $(C \rightarrow E)$ and $(D \rightarrow E)$, to institution $E$. Since $e(F) > s(E)$, the transition $(F \rightarrow E)$ is omitted from the graph ($1^{st}$ condition). $(A \rightarrow E)$ and $(B \rightarrow E)$ are omitted also due to the $2^{nd}$ condition ($(D \rightarrow E)$ exists and $s(D) > e(A)$ and $s(D) > e(B)$).

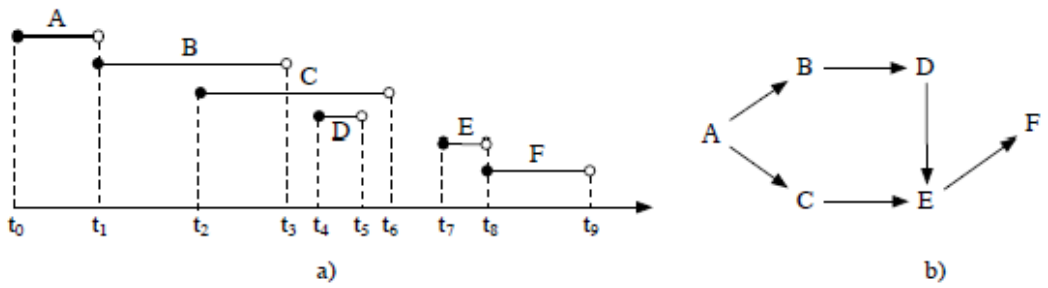



Figure 3.1: a) Past job transitions of an employee and b) Corresponding job graph from [60]

**Algorithm 1** Phase 1: Creation of acceptable transitions

**Require:** Employee profile of an individual

1: Sort the experiences of the employee by start date in **decreasing** order,
2: Set $S = \{institution\ nodes\ with\ sorted\ start\ dates\}$,
3: **for all** $maxNode \in S$ **do**
4: Put $maxNode$ to personal subgraph
5: $maxStartDate \leftarrow s(maxNode)$
6: **for all** $node \in S$ **do**
7: $nodeEndDate \leftarrow e(node)$
8: **if** $nodeEndDate \leq maxStartDate$ **then**
9: Put $(node \rightarrow maxNode)$ to $T$ {transition from $node$ to $maxNode$}
10: **else if** $nodeEndDate = maxStartDate$ **then**
11: Put $(maxNode \rightarrow node)$ to $T$ {transition from $maxNode$ to $node$}
12: **else**
13: $nodeEndDate$ cannot be greater than $maxStartDate$, $S$ is sorted
14: **end if**
15: **end for**
16: **end for**

**Algorithm 2** Phase 2: Eliminate redundant transitions

**Require:** Personal subgraph draft of the employee constructed in Phase 1

Sort the experiences of the employee by start date in **increasing** order,
Set $S = \{institution\ nodes\ with\ sorted\ start\ dates\}$,
**for all** $minNode \in S$ **do**
  **for all** $node \in S$ **do**
    **if** $(minNode \rightarrow node) \in T$ **then**
      Remove transition from the personal subgraph
      Check if there is any other shortest path from $minNode$ to $node$
      **if** $existsShortestPath(minNode, node)$ **then**
        No need for this transition, already removed
      **else**
        Put the transition back to personal subgraph
      **end if**
    **else**
      No transition exists between $minNode$ and $node$, do nothing
    **end if**
  **end for**
**end for**

Finally, merge personal subgraph to global graph

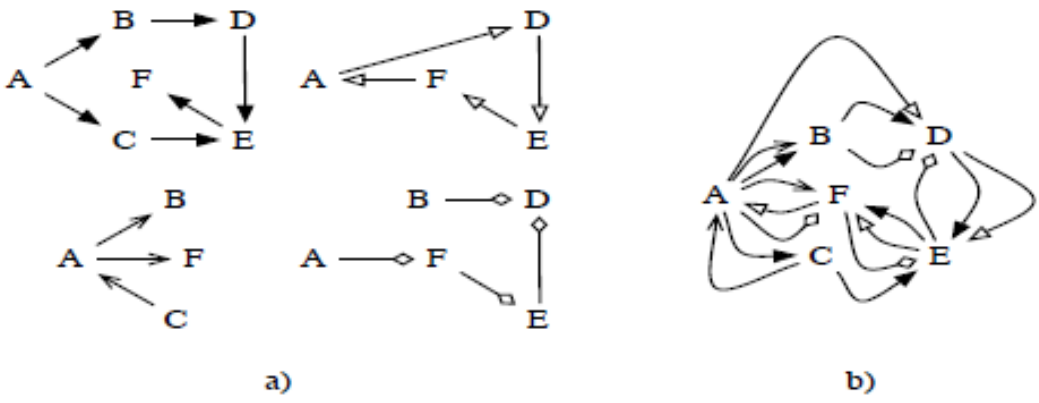



Figure 3.2: a) Job transition graphs for four individuals and b) Corresponding global transition graph

Figure 3.2 illustrates the construction of the global job transition graph by taking union of four individual transitions. In the global graph, multiple edges between same pair of nodes (parallel edges) can be seen.

## 3.2 Feature Vector Construction

To extract related features to our problem, we make use of job transition graphs in addition to relational data. During the calculation of the features from the graphs, we make a little search on the literature about that what kind of information can be extracted from the general "graph" concepts. In this section, concepts and formulas inspired during feature vector construction phase of our study are explained.

### 3.2.1 Social Network Analysis

Social network analysis (SNA) is the use of the network theory to analyze social networks by mapping and measuring the relationships and flows between people, groups, organizations and other connected information entities. The nodes in the network represents the people and groups while the links show relationships or flows between the nodes. Social networks and their analysis are in increasing interest with the ongoing growth of web-based services like facebook.com and deeply studied in many materials [20, 72]. Since social network analysis is an idea, it can be applied to many fields. Hence, in literature many network analysis studies exist in different fields (e.g., social sciences [15], biology [17], ecology [73], criminology [52], management [37], information science [58]). These studies simply cast the original problem into social network problem and use the methods in social network analysis to solve the original problem [26].

For our problem, generated job transition graphs can be considered as a kind of social network. Nodes represent institutions/organizations and links between nodes represent the transitions (flow) of the employees [70]. Therefore, we may make use of the social network methods in our approach to our problem.

A key challenge for the social network is to identify the most important nodes within

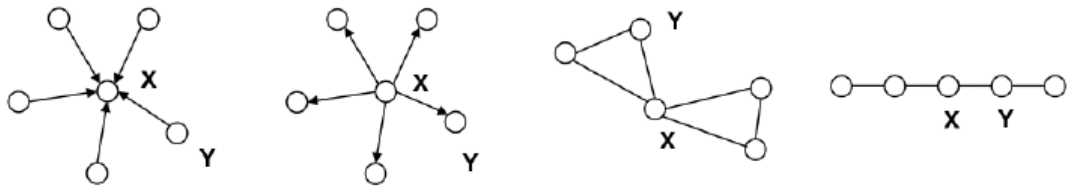



Figure 3.3: Networks in different shapes

the social network. For this purpose, network analysts describe a term "centrality" for the indicator of the node importance. Since the word "importance" has a wide number of meanings, many different centrality indices are described in the literature [12–14, 30]. Each one characterizes different aspects of structural positions/locations of nodes within networks. For example, in Figure 3.3 there are many networks in different shapes. $X$ node's importance over $Y$ node can change with the definition of influential position or advantageous position in the network. But, in a common sense among all centrality indices, high centrality value for a node (a central node) shows the prominence of a node within the network in terms of the selected centrality measure.

As mentioned above, there are many centrality definitions. But in the following section, we present only the centrality definitions used in the feature vector construction phase of our study.

### 3.2.2 Centrality Definitions

Some general considerations are taken into account when writing the centrality definitions in this section:

- Most centrality measures proposed in the literature are for undirected and connected graphs/networks. However, our job transition graphs are directed and may contain disconnected components. Therefore, the centrality definitions shown in this section may be modified versions of the centrality definitions for directed and "weakly" connected graphs.
- Links from a node to itself (self edge), or multiple incoming/outgoing links from one single node to another single node (parallel edge), are ignored usually

during centrality calculations. In other words, neighbors are more important than link counts in centrality calculations.

- $n_{in}(v)$ and $n_{out}(v)$ represent the number of neighbors considering incoming and outgoing links, respectively. $N$ is the number of nodes in the network. The distance $d(u, v)$ from $u$ to $v$ is the length of a shortest path from $u$ to $v$, or $\infty$ if no such path exists.

- In order to compare different networks, the centrality values should be normalized within the network. For normalization, the centrality value is divided by the maximum possible centrality value for the given centrality definition.

#### 3.2.2.1 Degree Centrality

In a classical definition, degree centrality is defined as the number of links the node has. But this can be modified in the form that degree centrality is the number of neighbors the node has, because no self or parallel edge is considered.

In case of a directed network where links have a direction, two separate measures of degree centrality should be calculated namely indegree and outdegree centrality. Incoming and outgoing degree centralities are shown in Equation 3.1 and Equation 3.2, respectively.

$$C_{D_{in}}(v) = n_{in}(v) \tag{3.1}$$

$$C_{D_{out}}(v) = n_{out}(v) \tag{3.2}$$

The maximum possible degree centrality of a node occurs when all other nodes in the network are connected to this node, this is $N - 1$. So, for normalization of degree centralities, calculated degree centralities (incoming and outgoing) are divided by $N - 1$. Normalized degree centrality equations are illustrated in Equation 3.3 and Equation 3.4.

$$C^N{}_{D_{in}}(v) = \frac{C_{D_{in}}(v)}{N-1} \tag{3.3}$$

$$C^N{}_{D_{out}}(v) = \frac{C_{D_{out}}(v)}{N-1} \tag{3.4}$$

#### 3.2.2.2 Closeness Centrality

In connected graphs, there is a natural distance metric between all pairs of nodes, defined by the length of their shortest paths (geodesic distance). The farness of a node $v$ is defined as the sum of its geodesic distances to all other nodes, and its closeness is calculated as the inverse of the farness. Closeness can be regarded as a measure of how long it will take to spread information from a node to all other nodes sequentially. The classical closeness centrality equation is in Equation 3.5.

$$C_C(v) = \frac{1}{\sum_{u \neq v} d(u,v)} \tag{3.5}$$

Equation 3.5 is only for connected and undirected graphs because distance is infinite (undefined) if no path exists between pair of the nodes. For the graphs with disconnected components, *reachable nodes* should be considered only.

Since distance between two nodes in directed networks is non-symmetric, "proximity" keyword is more suitable for directed graphs. Because closeness in terms of proximity is non-symmetric while closeness only is symmetric. From this perspective, the analog to closeness centrality in directed networks considers the *proximity* of a node $v$ to other nodes in its *influence domain I*. Influence domain is the set of nodes that can reach the node $v$ or can be reached by $v$ directly or undirectly (non-zero entries in the distance matrix for the node $v$) [72]. In terms of proximity (direction) and influence domain *I*, *proximity centrality* for incoming and outgoing links are shown in Equation 3.6 and Equation 3.7.

$$C_{P_{in}}(v) = \frac{1}{\sum_{u \neq v} \frac{d(u,v)}{I_{in}}} \tag{3.6}$$

$$C_{P_{out}}(v) = \frac{1}{\sum_{u \neq v} \frac{d(v,u)}{I_{out}}} \tag{3.7}$$

where $I_{in}$ is the influence domain of $v$, reaching node $v$ with incoming links, and $I_{out}$ is the influence domain of $v$, reached by $v$ to others with outgoing links.

According to the suggestion by [48], normalized proximity centrality equations are in Equation 3.8 and Equation 3.9

$$C^N{}_{P_{in}}(v) = \frac{I_{in}/(N-1)}{\sum_{u \neq v} \frac{d(u,v)}{I_{in}}} \tag{3.8}$$

$$C^N{}_{P_{out}}(v) = \frac{I_{out}/(N-1)}{\sum_{u \neq v} \frac{d(v,u)}{I_{out}}} \tag{3.9}$$

Due to the problem of the presence of the unreachable nodes and infinity distances, in study [11] *harmonic mean of all distances* are taken to calculate the closeness centrality and so the closeness equation is rewritten and renamed as *harmonic centrality* in Equation 3.10 and Equation 3.11. Since the maximum possible value for the harmonic centralities is $N-1$ (star network), in normalized equations 3.12 and 3.13, centralities are divided by $N-1$.

$$C_{H_{in}}(v) = \sum_{u \neq v} \frac{1}{d(u,v)} \tag{3.10}$$

$$C_{H_{out}}(v) = \sum_{u \neq v} \frac{1}{d(v,u)} \tag{3.11}$$

$$C^N{}_{H_{in}}(v) = \left( \sum_{u \neq v} \frac{1}{d(u,v)} \right) / N - 1 \tag{3.12}$$

$$C^N{}_{H_{out}}(v) = \left( \sum_{u \neq v} \frac{1}{d(v,u)} \right) / N - 1 \tag{3.13}$$

#### 3.2.2.3 Eigenvector Centrality

Eigenvector centrality is a natural extension of degree centrality. But differently from degree centrality, not all neighbors are equal. Eigenvector centrality is based on the concept that links to high-scoring nodes (more important nodes) contribute more to the score of the node. This centrality is also known as the Bonacich's Approach to Centrality [12].

Eigenvector centrality can be calculated with *Adjacency matrix*. Let $A = (a_{i,j})$ be the adjacency matrix of a graph, i.e. $a_{v,t} = 1$ if node $v$ is linked to node $t$, and $a_{v,t} = 0$ otherwise. Eigenvector centrality $x_v$ of a node $v$ is;

$$x_v = \frac{1}{\lambda} \sum_u a_{v,u} x_u \tag{3.14}$$

where $\lambda \neq 0$ is a constant. In a matrix form, we have an eigenvector equation;

$$\mathbf{A}\mathbf{x} = \lambda \mathbf{x} \tag{3.15}$$

There can be many different eigenvalues $\lambda$ in Equation 3.15 but only the greatest eigenvalue is accepted as the desired centrality measure. This eigenvalue problem can be solved with Power iteration method [7].

Direction notion is already included in eigenvector centrality definition (usage of adjacency matrix), so the separation into incoming and outgoing centrality is not necessary. Furthermore, there is no need to normalize eigenvector centralities additionally because the normalization parameter is automatically selected in power iteration for each iteration (the square root of the sum of squares of the node centralities).

Google's PageRank and Katz centrality are the variants of the eigenvector centrality.

#### 3.2.2.4 Katz Centrality

Katz centrality [42] is a variant of eigenvector centrality. It computes the relative influence of a node within a network by measuring the number of all nodes that can

be connected through a path. However, links with distant neighbors are penalized by an attenuation factor $\alpha$ by assigning a weight ($\alpha^d$) determined by $\alpha$ and the distance between nodes. Katz centrality $x_v$ of a node $v$ is;

$$x_v = \alpha \sum_u a_{v,u} x_u + \beta \tag{3.16}$$

where $\alpha$ is an attenuation factor in $(0, 1)$ and $\beta$ is the parameter controls the initial centrality. The principal eigenvector (the largest eigenvalue of $A$, the adjacency matrix) is the limit of Katz centrality as $\alpha < \frac{1}{\lambda_{max}}$.

In Katz centrality, large attenuation factor gives more penalty for the distant neighbors and results in more "local effect". The magnitude of $\beta$ reflects the radius of the power. Small values of $\beta$ weight local structure and large values weight global structure.

When $\alpha = \frac{1}{\lambda_{max}}$ and $\beta = 0$, Katz centrality is the same as eigenvector centrality.

#### 3.2.2.5 PageRank

PageRank is a link analysis algorithm [59] used by Google to rank websites in their search engine results. It assigns a weight to each element of the collection (World Wide Web) to measure its relative importance within set. It uses the reference links on each element (website) for this purpose. This algorithm can be applied to any entities that have directional links.

Since PageRank is the variant of eigenvector centrality, the PageRank computations require several "iterations" to adjust approximate PageRank values to more closely true value. Again, parallel and self edges are ignored during PageRank calculations.

As PageRank is initially developed for websites, the theory holds that an imaginary web-surfer who is randomly clicking on links will eventually stop clicking. At any step, the probability of the continuation of clicking is controlled by a factor called damping factor $d$. Among tests on different damping factors, generally assumed damping factor for PageRank calculations is around $0.85$. General PageRank equation is shown below:

$$PR(i) = \frac{1-d}{N} + d \sum_{j \in M(i)} \frac{PR(j)}{L(j)} \tag{3.17}$$

where $M(i)$ is the set of pages/nodes that *link to* $i$ and $L(j)$ is the number of outbound links on page $j$. Due to absence of parallel and self edges, $L(j)$ is the equivalent to the number of neighbors considering outbound links $n_{out}(j)$.

The formula 3.17 corresponds to a eigenvalue problem and can be solved by iterative methods. Initially $PR(i)$ is set to $\frac{1}{N}$. At each iteration, sum of page ranks of all nodes should be 1.

The formula above is slightly confusing in the case that the page has no links to other pages/nodes (*sink node*). If the random surfer arrives at such a sink page, it picks another URL at random and continues surfing again. Therefore, the PageRank calculation is rewritten such that it is assumed sink nodes link out to all other nodes in the collection [46].

$$\begin{aligned} PR(x) &= \frac{1-d}{N} + d \sum_{y \to x} \frac{PR(y)}{L(y)} + d \sum_{z \to \emptyset} \frac{PR(z)}{N}, \\ &= \frac{1-d+dS}{N} + d \sum_{y \to x} \frac{PR(y)}{L(y)} \end{aligned} \tag{3.18}$$

where $z$ represents the sink node and $S$ is the sum of the page ranks of the sink nodes over all collection of size $N$. $S$ is recalculated before each iteration.

Compared to eigenvector centrality and Katz centrality, PageRank has one major difference, the scaling factor $L(j)$. Moreover, PageRank vector is a left hand eigenvector. When calculating $PR(i)$, incoming links to node $i$ is taken into consideration instead of outgoing links from the node $i$.

## 3.3 Feature Selection

In machine learning and statistics, feature (attribute) selection is the process of selecting a subset of relevant features to be used in model construction. Feature selection techniques are applied to data because the data may contain many redundant or irrelevant features. Redundant features duplicate some information contained in one

or more other attributes, and irrelevant features contain no useful information for the learning task. When constructing predictive models, feature selection techniques improve model interpretability, shorten training times and enhance generalisation by reducing overfitting. Feature selection is useful part of the data analysis process showing which features are important for prediction, and how these features are related.

A feature selection algorithm is based on two points. First one is search technique for proposing feature subsets. Search approaches include exhaustive, best first, greedy forward, greedy backward searches and many others. Second point is the evaluation of subset of features as a group for suitability. Since the choice of evaluation method heavily affects the algorithm, feature selection algorithms are categorized into three with respect to evaluation methods; wrappers, filters and embedded methods.

- Wrapper methods search possible features and evaluate each feature subset by running a training model on the subset. Due to training part, wrappers can be computationally expensive.
- Filters use similar search approaches as wrappers, but instead of evaluating against a model, a simpler filter is evaluated. Filter methods are usually less computationally intensive than wrappers since they do not produce a feature set that is tuned to a specific model. Many filter methods provide a feature ranking after cross-validation rather than an explicit best feature subset.
- Embedded techniques are embedded in and specific to a model.

Feature selection techniques are a subset of the more general field, feature extraction. Feature extraction creates new reduced features from the original features, whereas feature selection returns a subset of the original features. In feature extraction, same transformation functions should be applied to both training and testing data.

In this study, we choose feature selection techniques rather than feature extraction because training and testing data is taken from different sources (years) (Section 5.1). If feature extraction technique is used for our case, it requires saving and restoring transformations several times which makes our problem so complicated.

As feature selection methods, filter methods through WEKA and a wrapper method through LibSVM are applied in this study. In the following subsections, evaluation

methods used in feature selection phase of our study are explained.

### 3.3.1 Chi-square Based Feature Selection

Chi-square ($\chi^2$) is a statistical test usually applied to categorical data. It is used to compare observed data with data that is expected to obtain according to a specific hypothesis. It evaluates how likely any observed difference between the sets arises by chance. The chi-square test is always testing the null hypothesis, which states that there is no significant difference between the expected and observed result.

The value of Chi-square statistic is calculated as the sum of the squared difference between observed and the expected data, divided by the expected data in all possible categories. The formula can be seen in Equation 3.19. The larger the $\chi^2$ value is, the more likely the attribute is related to class label.

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} \tag{3.19}$$

where $O_i$ is observed data, $E_i$ is expected data, $n$ is the number of categories for a categorical attribute.

As a feature selection method, *ChiSquaredAttributeEval* method is used within WEKA for our study. It evaluates the worth of an attribute by computing the value of the Chi-squared statistic with respect to the class. Since our data is not categorical, this WEKA method discretizes our attributes automatically within itself. After evaluating each attribute individually with 10-fold cross-validation, *Ranker* method as the "search" method ranks the attributes by their individual evaluations.

### 3.3.2 Information Gain Based Feature Selection

Information gain is an attribute evaluator metric. More information gain means more relevance of attribute with respect to class. It can be calculated as follows:

$$InfoGain(Class, Attr) = H(Class) - H(Class|Attr) \tag{3.20}$$

where $H$ specifies the entropy (information).

In our study's feature selection phase, *InfoGainAttributeEval* method is used within WEKA. It evaluates the worth of an attribute by measuring the information gain with respect to the class and discretizes numeric attributes itself. After evaluating each attribute individually with 10-fold cross-validation, *Ranker* method as the "search" method ranks the attributes by their individual evaluations.

### 3.3.3 Gain Ratio Based Feature Selection

Gain ratio is another evaluator metric used in feature selection evaluations. The formula is in Equation 3.21. A high gain ratio value shows more relevance of an attribute with respect to class label.

$$GainR(Class, Attr) = (H(Class) - H(Class|Attr))/H(Attr) \tag{3.21}$$

where $H$ specifies the entropy (information).

*GainRatioAttributeEval* method is used within WEKA as a feature selection method. It evaluates the worth of an attribute by measuring the gain ratio with respect to the class. After evaluating each attribute individually with 10-fold cross-validation, *Ranker* method as the "search" method ranks the attributes by their individual evaluations.

### 3.3.4 F-score Based Feature Selection

F-score is a simple technique measuring the discrimination of two sets of real numbers [22]. Given training vectors $\mathbf{x}_k$, and the number of positive instances $n_+$ and negative instances $n_-$, the F-score of the $i$th feature is calculated as:

$$F(i) = \frac{\left(\bar{\mathbf{x}}_i^{(+)} - \bar{\mathbf{x}}_i\right)^2 + \left(\bar{\mathbf{x}}_i^{(-)} - \bar{\mathbf{x}}_i\right)^2}{\frac{1}{n_+-1}\sum_{k=1}^{n_+}\left(x_{k,i}^{(+)} - \bar{\mathbf{x}}_i^{(+)}\right)^2 + \frac{1}{n_--1}\sum_{k=1}^{n_-}\left(x_{k,i}^{(-)} - \bar{\mathbf{x}}_i^{(-)}\right)^2} \tag{3.22}$$

where $\bar{\mathbf{x}}_i$, $\bar{\mathbf{x}}_i^{(+)}$, $\bar{\mathbf{x}}_i^{(-)}$ are the average of the $i$th feature of the whole, positive and negative data sets, respectively; $x_{k,i}^{(+)}$ is the $i$th feature of the $k$th positive instance, and $x_{k,i}^{(-)}$ is the $i$th feature of the $k$th negative instance.

The larger the F-score is, the more likely this feature is more discriminative. F-score evaluation of the features is used within LibSVM as "fselect.py" tool. This tool calculates the F-score value of each feature. Then, it picks some possible thresholds to cut low and high F-scores. For each threshold, the features below threshold are dropped and SVM algorithm is applied to train and test the subset of features with $5$-fold cross validation. For each threshold and feature subset, average validation error is calculated. Finally, the tool selects a threshold and corresponding feature subset with lowest validation error.

Since F-score based feature selection evaluates each feature subset by running a training model (SVM) on the subset, it can be considered as a wrapper method.

In our work, we use F-score rankings of the features in addition to the subset results of the training part.

### 3.4 Classification Methods

Machine Learning is a subfield of computer science and artificial intelligence in addition to strong ties to statistics and optimization. It concerns the learning from data, rather than following only explicitly programming instructions. A machine learning framework begins with a preparation of information (training) by extracting knowledge from training data, then it uses trained knowledge to predict the output of new data (testing). There are various forms of machine learning such as supervised, unsupervised, semi-supervised, reinforcement learning. These types of algorithms are organized based on the desired outcome of the algorithm or the type of input available during training of the machine. In this thesis, we do binary classification using supervised learning methods. In supervised learning, training set contains labelled data and the algorithm infers a function mapping from inputs (typically vector) to outputs/labels. During the testing process, the algorithm uses this inferred function from training process to classify new data.

Support Vector Machines, Decision Table/Naive Bayes Hybrid Classifier and Artificial Neural Networks are the types of supervised algorithms applied in classification phase of our study. These algorithms are used through LibSVM (A Library for Support Vector Machines) [21] and WEKA (Waikato Environment for Knowledge Analysis) [32] tools.

### 3.4.1 Support Vector Machines

Support Vector Machines (SVMs) are supervised learning models for data classification. The goal of SVM is to produce a model based on the training data which predicts the target values (class labels) of the test data given only the test data attributes. By doing this, SVM maps input vectors into high dimensional feature space and separates different class points by a clear gap (hyperplane) that is as wide as possible. SVM can be seen as a problem of finding the optimal hyperplanes with the biggest clear gap between the classes [23]. The graphical representation of this situation can be seen in Figure 3.4. According to figure, margin of the largest separation between two classes is defined by "support vectors" marked with grey squares which lie on the margin.

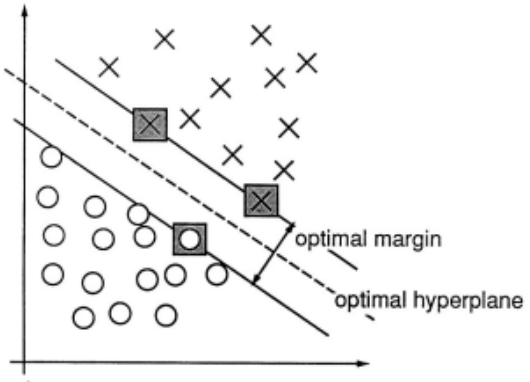



Figure 3.4: An example of separation in 2D (Figure taken from [23])

Formally, a data point in training data is represented with $p$-dimensional "attributes" vector $\mathbf{x}_i$ and a "target value" (class label) $y_i$. Optimal hyperplane can be written as the set of points $\mathbf{x}$ satisfying Equation 3.24, where $\cdot$ denotes the dot product, $\mathbf{w}$ the (not necessarily normalized) normal vector to the hyperplane and $b$ constant.

$$D = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in R^p,\ y_i \in \{-1, 1\}\}_{i=1}^n \tag{3.23}$$

$$\mathbf{w} \cdot \mathbf{x} - b = 0 \tag{3.24}$$

This problem requires the solution of the following optimization problem ( [16, 23, 36]):

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0 \end{aligned} \tag{3.25}$$

Here training vectors $\mathbf{x}_i$ are mapped into a higher dimensional space by the function $\phi$. SVM finds a linear separating maximum margin hyperplane in this higher dimensional space. Non-negative slack variable $\xi_i$ measures the degree of misclassification of the data $\mathbf{x}_i$. $C > 0$ is the penalty parameter of the error term $\xi_i$. $C$ can be viewed as a soft margin parameter which trades off between a large margin and a small error. A high $C$ aims at classifying all training examples correctly.

Furthermore, $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$ is called the *kernel* function. By applying kernel trick, nonlinear classification is also possible in addition to the linear one. Some common kernel types are linear, polynomial, radial basis function (RBF), sigmoid. In our study, we choose nonlinear classification with the Gaussian RBF kernel $K(\mathbf{x}, \mathbf{y}) = e^{-\gamma||\mathbf{x}-\mathbf{y}||^2}$ for SVM algorithm. RBF kernel nonlinearly maps samples into a higher dimensional space and unlike the linear kernel can handle the nonlinear relation between class labels and attributes. It has fewer numerical difficulties/parameters compared to other kernels. Additionally, other kernels (linear, sigmoid) can be reduced to RBF kernel with suitable parameters [43, 47].

There are two parameters for a RBF kernel: $C$ and $\gamma$. While the penalty parameter $C$ is common to all SVM kernels, $\gamma$ is RBF-specific kernel parameter which controls the shape of the separating hyperplane. Increasing gamma usually increases number of support vectors and makes the decision boundary more contorted. With the help of "grid search" on $C$ and $\gamma$ using cross-validation, various pairs of ($C$,$\gamma$) are tried and the one with the best cross-validation accuracy is picked.

In our study, SVM algorithm is applied through LibSVM tool's command line. Best parameters for used RBF kernel are also extracted by LibSVM's grid search script "grid.py".

### 3.4.2 Decision Table/Naive Bayes Hybrid Classifier (DTNB)

In study [31], a hybrid model combining naive Bayes with induction of decision tables is proposed. The model is a simple Bayesian network in which decision table (DT) represents a conditional probability table. Each entry in the table is associated with the class probability estimates.

The algorithm for learning the DTNB model proceeds as the same way as DTs. Using the forward selection search, at each point in the search, the algorithm evaluates the merit of dividing the attributes into two disjoint subsets: one for the decision table, the other for naive Bayes. Then, at each step a set of selected attributes are modeled by naive Bayes and the rest by the decision table.

The class probability estimates of the decision table (DT) and naive Bayes (NB) are combined to generate overall class probability estimates. Assuming $X^T$ and $X^\perp$ are the sets of attributes in DT and NB respectively, the overall class probability is computed as;

$$Q(y|X) = \alpha \times Q_{DT}(y|X^T) \times Q_{NB}(y|X^\perp)/Q(y), \tag{3.26}$$

where , $Q_{DT}(y|X^T)$ and $Q_{NB}(y|X^\perp)$ are the class probability estimates obtained from DT and NB respectively, $\alpha$ is normalization constant, and $Q(y)$ is the prior probability of the class.

Cross-validation according to selected evaluation metrics (e.g., accuracy, area under the curve - AUC) is used to evaluate the quality of a split based on the probability estimates generated by combined model. The algorithm also considers dropping entirely a feature (attribute selection) from the model at each step.

In our study, DTNB hybrid algorithm is applied through WEKA command line. Use of decision tables and probabilistic models may not be suitable to our all numeric data

because numeric attributes should be discretized with the intervals from the training data to apply the algorithm. We choose this algorithm since it is used in our reference study [60] for job recommendation problem using job transitions. We apply this hybrid algorithm only to the best model years obtained from SVM to make a comparison with the SVM results.

### 3.4.3 Artificial Neural Network

In machine learning and related fields, artificial neural networks (ANNs) are computational models using learning algorithms that are inspired by our understanding of how the brain learns. Neural networks are used in a variety of practical applications such as speech recognition, object recognition like handwriting recognition, image retrieval.

An artificial neural network consists of a set of processing units, "neurons", which communicate by sending signals to each other over a large number of weighted connections. The word "network" here refers to the inter–connections between the neurons in the different "layers" of each system. Each neuron is activated by neighbors or external source and activations of neurons are then passed on to other neurons (propagation). This process is repeated until finally, an output neuron is activated. A simple neural network can be seen in Figure 3.5.

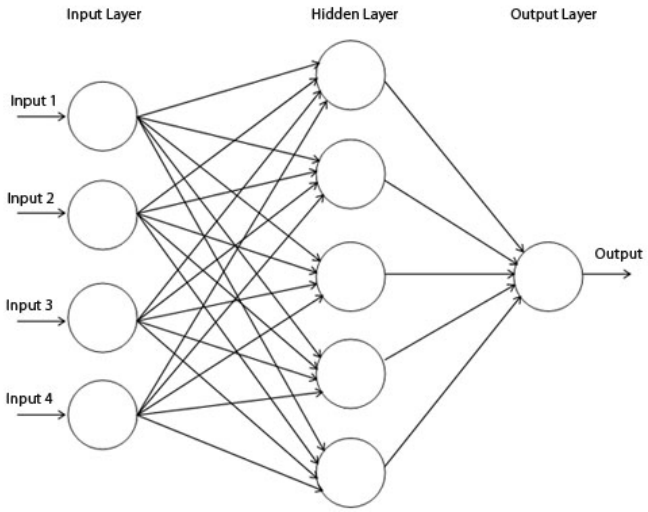



Figure 3.5: A two-layer feedforward artificial neural network

The learning process takes place in updating/adjusting weights of interconnections. The cost function $C$, that measures how far away a particular solution is from an optimal solution, is tried to be minimized. In supervised learning, backpropagation is a common method adjusting weights of ANN. Modifications are made in the backwards direction from the output layer through each hidden layer down to the first hidden layer.

Mainly two classes of artificial neural networks are feedforward neural network and recurrent neural network. For feedforward one, connections between the units/neurons do not form a directed cycle unlike the recurrent one. The data processing can extend over multiple layers of units, but no feedback connections or connections between units of the same layer are present.

In this study, we use Multilayer Perceptron through WEKA. It is a feedforward artificial neural network model that consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. It utilizes a supervised learning technique called backpropagation for training the network and can distinguish data which is not linearly separable.

Like DTNB algorithm, we apply neural network algorithm only to the best model years from SVM to make a comparison with the SVM results. We generate neural network with default parameters using the algorithm in WEKA.

### 3.4.4 LibSVM

During our study, Support Vector Machine algorithm is applied through LibSVM [21] which is an integrated open source software for support vector classification and regression. It is written in C++ though with a C API. It implements the Sequential Minimal Optimization algorithm (SMO) [62] for kernelized support vector machines (SVMs), supporting classification and regression. LibSVM also requires data to be classified in LibSVM file format.

In addition to the support of different classification and regression problems by training, predicting and data scaling programs, LibSVM also provides subset selection, parameter selection and data format checking tools (Python scripts) which are all

used during our study.

In this study, proposed procedure explained in the guide of LibSVM [36] is taken into account to get robust results. Command line interface of LibSVM is used.

### 3.4.5 WEKA

WEKA [32] is the project that collects visualization tools and algorithms for data mining and machine learning tasks. It is an open source software written in Java language. It contains tools for data preprocessing, classification, regression, clustering, attribute/subset evaluation, association rules, and visualization.

WEKA has its own Attribute Relationship File Format (ARFF) for describing and storing data. An ARFF file consists of header and data sections. Header section includes the name of the relation, a list of the attributes and attribute types. Data section contains data of instances in lines storing the attribute values and class labels of instances. In addition to ARFF, WEKA can also process data in some other formats like CSV, LibSVM's format, C4.5.

The algorithms in WEKA can either be applied directly to a dataset from WEKA interfaces or called from own Java code. While WEKA provides graphical user interfaces, simple command line option is also available.

In this thesis, classification algorithms except for SVM and some feature selection methods are applied through WEKA tool. Instead of using WEKA API in our code, we choose command line interface of WEKA for applying algorithms and methods.

# CHAPTER 4

# METHODOLOGY

Our methodology is mainly composed of five phases; data preprocessing, job transition graphs construction, feature extraction, feature selection and classification. General information about these phases is as follows:

- Data preprocessing phase includes many steps such as removing noisy profiles, completing missing date information, skipping some institutions' data etc. This phase aims to eliminate imperfect information and mistakes in consequence of users' erroneous data entry and collect suitable data for our problem.

- Job transition graphs construction is the phase of forming of job transitions of employees for each year using the algorithm explained in Section 3.1.2.

- Feature extraction is another important phase of our study. From job transition graphs and relational data, employee and company features are extracted. For employees, features of the companies for which employees work in their whole professional life are aggregated separately with aggregate functions.

- Since current and past companies' features are collected separately with different aggregate functions, employee feature vector size becomes large. In order to pick most important features, feature ranking algorithms are used and ranking thresholds are found.

- The last phase, classification, is the whole process of training and testing parts. Support vector machine algorithm is used and tested as classification algorithm. Different classifiers are tested with the best model obtained.

The general overview can be seen in Figure 4.1. Each phase's details are explained in the following sections. At the beginning, dataset will be described first.

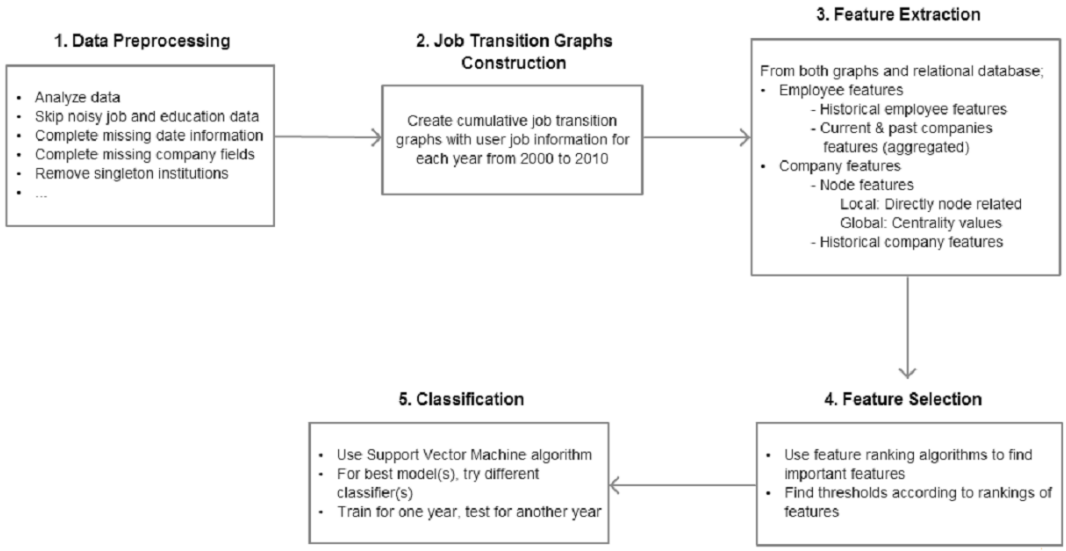



Figure 4.1: Overview of Our Methodology

To give some brief explanations about the implementation of our study, Java programming language is used for implementation and as a development environment Eclipse is chosen. While employee profiles and companies' general data are stored in a PostgreSQL relational database, job transition graphs should be stored in a graph database and Neo4j [6] is used for this purpose.

## 4.1 Dataset Description

For the purpose of our thesis work, we use publicly available employee profiles crawled from a widely used professional social network in the Web. Original profiles dataset contains $6,909,746$ employee profiles. The profiles contain information about employees' professional and educational experiences as well as companies' basic information. It can be possible to split each profile's data fields into three sections:

1. **Personal information:** This section contains personal information about the employee as follows;

- URL of user/employee homepage
- Name and surname of the employee
- Country of the employee
- Number of connections in professional social network

2. **Professional experience:** Employee's experience details in his/her professional life are described. For each employee job experience, there are many fields shown below:
   - Job title
   - Company name
   - Metadata of the company (company type, industry of the company, number of employees working, stock market ticker)
   - Job experience start and end date
3. **Educational experience:** Information about employee's each educational experience is included in data fields as follows:
   - Educational institution name
   - Degree given
   - The field of education
   - Educational experience start and end date

An example employee profile record can be seen in Table 4.1. URL and name information is anonymous.

An important detail about our dataset is the last date entry. Since dataset is crawled from the Web, the date on which the crawling process is completed is an important issue. The last job date entry in our dataset is *'January 2011'* showing that dataset is crawled about year *2011*. *'January 2011'* is set as the 'present' time in the dataset. Some given educational experiences have start or end dates after this date, however this may not tell any information because of the possible educational plans in the future.

More details of the dataset can be found while explaining data preprocessing steps in the following section.

Table 4.1: Example employee profile record

| | | |
|---|---|---|
| **Personal Info.** | URL | <User URL> |
| | Name | <Name Surname> |
| | Country | Albuquerque, New Mexico Area |
| | Connections | 34 |
| | # of Job Exper. | 3 |
| | # of Educ. Exper. | 1 |
| **Job Experience # 1** | Title | Sales Consultant - Small Business |
| | Company Name | Paychex |
| | Company Metadata | Public Company; PAYX; Human Resources industry |
| | Start Date | July 2009 |
| | End Date | Present |
| **Job Experience # 2** | Title | Associate District Sales Manager |
| | Company Name | ADP |
| | Company Metadata | Public Company; 10,001 or more employees; ADP; Information Services industry |
| | Start Date | January 2009 |
| | End Date | June 2009 |
| **Job Experience # 3** | Title | Branch Manager |
| | Company Name | Enterprise Rent-A-Car |
| | Company Metadata | Privately Held; 10,001 or more employees; Automotive industry |
| | Start Date | June 2006 |
| | End Date | December 2008 |
| **Educ. Experience # 1** | Institution Name | The University of New Mexico - Robert O. Anderson School of Management |
| | Degree | BBA |
| | Field | Human Resources Management |
| | Start Date | 1998 |
| | End Date | 2003 |

## 4.2 Data Preprocessing

Since our dataset contains publicly available employee profiles taken from a professional social network in the Web, it can be said that the data have been provided by the users themselves. This also means that their correctness is not guaranteed. There may be typos when people enter their own data to the system or people may intentionally report incorrect information. Due to absence of naming rule, some may use the full name of the company whereas others may abbreviate the name, add location of the company to the name or provide specific department. Similar situation also applies to educational data entry, people might again abbreviate the institution name, given degree or field of education. The most complex situation occurs when the experience date information is absent or missing.

For the mentioned reasons, data preprocessing step is essential for our study to obtain clean and useful data. At this point, analysis of dataset plays an important role. In this section, data preprocessing phase is explained in steps along with some analysis results.

### 4.2.1 Invalid Data Fields

This section states invalid situations due to invalid data entry.

- In this study, at first glance we prefer to work with the US profiles because professional social network usage is more common in US than other countries. But, this determination with our limited information is hard, country of the employee may not give the correct information about the employee's geolocation. Therefore, as a first step **elimination of country-specific account aliases** is applied for this purpose.

- We prefer to work with the profiles whose language is English. But in our dataset, dates in other languages except English such as *'Novembre 2009'* or *'Mayo de 2008'* exist. Hence, **employee profiles giving the date information in other languages except English** are ignored. This elimination/validation mostly helps us to eliminate non-US profiles. Because it is observed that if

an employee profile has the date information in other languages, other given information like company name, educational experience, job position title etc. is also in that language and this profile is most probably non-US profile.

- Another language concerning preprocessing is applied to industry fields. Industry fields of companies are counted in decreasing order according to users' data entries. A threshold is found such that below that threshold (decreasingly), industry fields are in other languages except from English. So, **industry fields** commonly given by users and remaining above threshold count **(only in English)** are taken only. **Employee profiles having invalid industry fields in their professional experiences data** are ignored as they give other related information again in non-English language.

  Table 4.2 clearly shows the numeric results of industry thresholding process. As it can be seen from this table, industry threshold count is selected as $41$. This means that we expect an industry field given by at least $41$ users' entries. Above this threshold number, industry counts start from $1214$ in increasing order. This threshold is the border which separates English industries from other language industries. Among $338$ given industry fields, only $148$ industries are in English.

Table 4.2: Industry field counts

| **Industry Name** | **Count** |
|---|---|
| Information Technology and Services industry | 1504559 |
| Computer Software industry | 819677 |
| Marketing and Advertising industry | 634535 |
| Telecommunications industry | 598470 |
| Financial Services industry | 548056 |
| Higher Education industry | 547339 |
| Internet industry | 423129 |
| Retail industry | 353556 |
| Banking industry | 350345 |
| ... | ... |
| Fishery industry | 1930 |
| Ranching industry | 1214 |
| Servicios y tecnología de la información industry | 40 |
| Venta al por menor industry | 26 |
| ... | ... |

- Date field become the most problematic field in out dataset. One of these problems is the empty dates in dataset like *'X'*. Since these date fields cannot be filled, **empty date fields** are considered as invalid.

- We expect the existence of both start and end date of experiences. Therefore, **if either start date or end date is invalid for an experience**, this experience is skipped.

- As mentioned at the end of Section 4.1, present time in the dataset is accepted as *'January 2011'*. Hence, **(educational) experiences having start dates after accepted present date** are ignored.

- Some company and educational institution names contain only punctuation characters or numbers. So, **institutions that doesn't have any letter in their names** are considered as invalid/noisy due to user's probable erroneous data entry.

### 4.2.2 Missing Data Fields

This section describes the situations such that missing data is a problem. Proposed filling methods are explained.

- Some date fields contain only year information. We expect date information with month and year together. Therefore we have to fill the missing month information. We prefer to arrange **'only year' date fields** in a way that they can **span whole year** and so no information loss occurs. While beginning of the year is set for the start dates of experiences, end of the year is set for the end ones. For example, if start date of an experience is given as *'2006'*, this date is converted to *'January 2006'*. If end date is again given as that year *'2006'*, *'December 2006'* is set for itself. By this way, whole *'2006'* year can be covered.

- ***'Present'* date fields** are set to accepted present time *'January 2011'*.

- As explained in Section 4.1, companies have four metadata fields; company type, industry, number of employees and stock market ticker. There is no ne-

cessity to give all these metadata fields for one company. Same company can be described with company type and industry according to one user's information whereas another user can only say industry and stock ticker for the same company. Additionally, same metadata field for a company may be given different due to user's data entry. Therefore, to find the correct fields of the companies, it is decided to **count given metadata fields of each company. The most frequent given metadata field is set to that company**.

An example of this process is shown in Table 4.3 for company *Oracle*. While three industry values are given by the users to describe the *Oracle* company, "Information Technology and Services" which has the greatest value count among others is selected as the last industry value. Same process can be seen on the results of the other metadata fields. Values in the table are not real count values, it is just demonstration purpose.

Table 4.3: Example metadata filling process for one company

| Metadata Field | Given Value | Count | Last Value |
|---|---|---|---|
| Industry | Information Technology and Services | 8694 | Information Technology and Services |
| | Computer Software | 345 | |
| | Internet | 69 | |
| Company Type | Public Company | 3489 | Public Company |
| | Privately Held | 498 | |
| | <Empty> | 2958 | |
| Employee Number | 10,001+ | 3454 | 10,001+ |
| | 5001-10,000 | 436 | |
| | 1001-5000 | 29 | |
| Stock Ticker | ORCL | 4458 | ORCL |
| | IFLX | 50 | |
| | STEL | 15 | |
| | <Empty> | 3599 | |

### 4.2.3 Other Preprocessing Steps and Assumptions

- When considering URL uniqueness of a user/employee, **duplicate user profiles** are skipped.

- **Case-folding** process is performed and institution names are converted to lowercase letters.

- **After the preprocessing steps** mentioned above, if **an employee profile doesn't have any "valid" professional or educational experience**, this means this profile doesn't have useful information and should be skipped.

- Noisy institutions can be found by looking at number of institution declarations given by users. If only one user says his/her experience at one institution, the existence of that institution can be questioned. Therefore, **"singleton" institutions** are removed from our dataset.

- Some profiles seen as having **non human readable characters** are removed.

While applying data preprocessing steps, dataset is analyzed and some assumptions are made, some rules are extracted.

- URL field determines an employee profile's uniqueness.

- It is decided for institutions that uniqueness is determined by institution name field only (lowercase).

- Start date of an experience should be before or equal to end date of the same experience. Experience data is ignored if it doesn't follow this rule due to erroneous data entry.

Even if we try to get US profiles in English, we know that after these processes non-English information about employees' experiences still exists. Another situation about our dataset is caused by absence of naming rule. Same institution can be named in a different way by the users. In both mentioned cases, we prefer to leave them as they are.

After preprocessing and cleaning steps, statistics about our data is reported and shown in Table 4.4. Average number of company affiliatians in this table represents the average number of professional experiences that is reported by an employee in our dataset. According to the results shown on the table, while an employee states nearly

Table 4.4: Dataset statistics

| Description | Value |
|---|---|
| Number of profiles | $3,771,095$ |
| Number of unique companies | $1,255,705$ |
| Number of unique educational institutions | $148,813$ |
| Number of job experiences | $10,826,911$ |
| Number of educational experiences | $5,030,626$ |
| Average number of company affiliations | $2.87$ |
| Average number of educational affiliations | $1.33$ |

three professional experiences, about one educational experience is declared for an employee.

## 4.3 Job Transition Graphs Construction

As mentioned before, we believe that job transitions can be exploited in order to predict employee turnovers, because of the nature of each transition corresponding to a turnover also. To create job transitions, we take study [60] as a reference and the job transition graph construction algorithm is explained according to study [60] in Section 3.1.1 and Section 3.1.2.

Even if we use the algorithm from [60], we define our own rules to process employee profiles before given to the algorithm. These rules are below:

- Educational institutions are not included in the node set of the graph. This means that when constructing the graph, educational experiences of employees are skipped. This restriction is applied because it is already known fact that educational experience corresponds to a "temporary" duration in one's career life. Therefore, completion of the education is not a right indicator of a turnover. However, if an educational institution is given with the professional experience, this institution is added to the graph.

  Nodes in the job transition graphs are mostly composed of company institutions for the reason mentioned above.

- In this thesis, we analyze the turnovers at years. Hence, we have to split the

transitions and graphs into years. We start analysis with the year *2000*. Since the present time is accepted as *'January 2011'* and full year information is available lastly for year *2010*, end year of the analysis is selected as *2010*.

For each year graph, professional experiences having start date before the end of the year (*'December <year>'*) are taken into consideration and used for the formation of job transitions. Thus, some employee profiles' information is included in that year's graph, some of them are not considered. Additionally, some experiences belong to an employee are taken for the year graph, some later experiences of the same employee may not be put to graph.

Each year's graph is "cumulative" because of the constraint about the formation of graphs which is the comparison of the experience start date and end of the year. For example, *2003* graph contains the job transitions from *2000*, *2001* and *2002* years in addition to *2003* transitions.

In Figure 4.2, an example employment history and the corresponding *2006* and *2007* job transition graphs for an employee are illustrated. *2006* graph doesn't include the transitions from *C* to *D*, because when the jobs at the company *C* are left, no information about the next institution exists until the end of *2006*. Job transition occurs at year *2007* and so it is included in *2007* and the following year graphs.

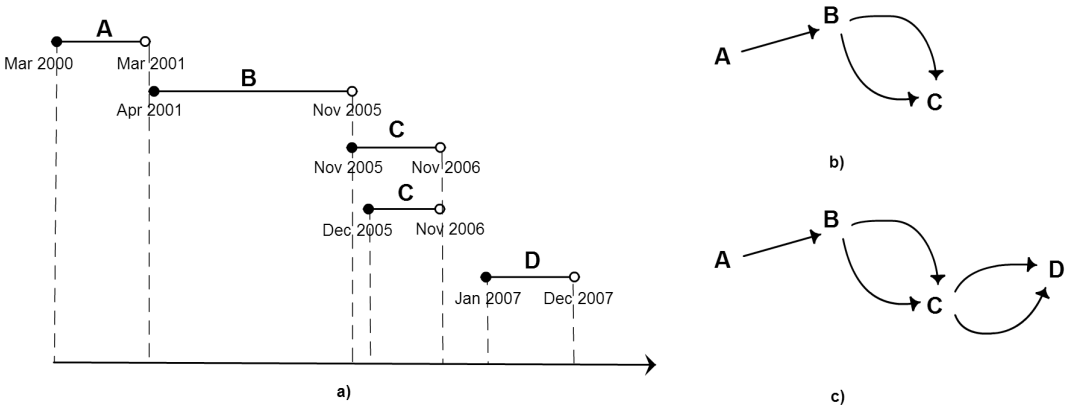



Figure 4.2: For an employee, a) Employment history and b) Job transition graph in *2006* and c) Job transition graph in *2007*

- Company institutions that no transition is available for them aren't included in graphs. In other words, there is no isolated node (with degree of zero) in graphs.

Respecting the rules explained above and applying algorithm from [60], 11 job transition graphs are constructed for the years from *2000* to *2010*. Information about the year graphs is shown in Table 4.5. According to this table, institution and job transition counts increase by the year owing to cumulative graphs. While institution count is multiplied by almost three, job transition count is multiplied by about six towards year *2010*.

Table 4.5: Counts about Job Transition Graphs

| Year | Institutions at nodes | Transitions at edges |
|---|---|---|
| 2000 | 415,784 | 1,320,768 |
| 2001 | 472,584 | 1,588,217 |
| 2002 | 528,649 | 1,861,190 |
| 2003 | 589,718 | 2,180,977 |
| 2004 | 661,121 | 2,590,205 |
| 2005 | 744,366 | 3,109,790 |
| 2006 | 838,463 | 3,768,758 |
| 2007 | 942,768 | 4,613,971 |
| 2008 | 1,045,028 | 5,618,470 |
| 2009 | 1,133,871 | 6,684,900 |
| 2010 | 1,182,836 | 7,724,282 |

Another detail is that the counts shown in Table 4.5 for the last year *2010* are less than the counts of whole dataset shown in Table 4.4 after preprocessing operations. For example, in Table 4.4, the number of unique companies is 1,255,705, whereas institution count is 1,182,836 for *2010* graph. This situation is probably caused by some employee profiles that have only one experience declaration and so no job transition can be constructed for this employee. The reason is related to last rule mentioned above.

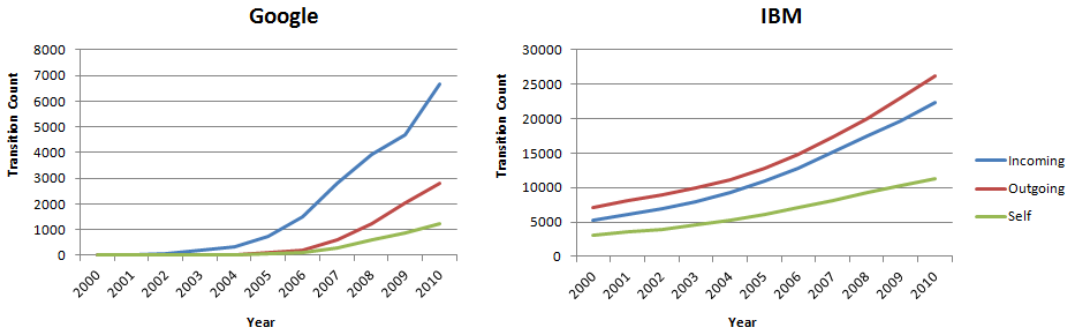



Figure 4.3: Transition counts for companies *Google* and *IBM*

In Figure 4.3, an example of transition counts for companies (nodes) *Google* and *IBM* is illustrated. *Google* has very few transitions at *2000* but reaches thousands at *2010* probably caused by the growth of the company. On the other hand, *IBM* already has several transitions at the beginning and increases its transition count in the following years due to cumulative effect of our graphs, but not as fast as *Google*. Job transition graphs enable us to do such kind of analysis.

## 4.4 Feature Vector Construction

After the job transition graphs construction, features are extracted from both graphs and relational data of employees and institutions, to be given to classification phase. We categorize the calculated features into two; employee and company features. We use the term "company" because we only consider the professional experiences for our problem as explained previously in Section 4.3. Still, there can be educational institutions given with the professional experience information.

In this thesis, we analyze the employee turnovers at years. Hence, both company and employee features are calculated for different years from *2000* to *2010*. This means that each year has its own company and employee feature values calculated according to the following feature definitions. Since working employees and companies for each year can change, the sets of employees and companies, for which features are calculated, are evaluated according to the given year.

### 4.4.1 Company Features

Companies (professional institutions) are located at the nodes of the generated job transition graphs. Since the job transition graphs can be considered as a kind of social network, general social network analysis methods explained in Section 3.2.1 and 3.2.2 can be applied to our problem. Additionally, we have basic company information (metadata fields) associated with the professional experiences from our dataset. Hence, in this section we divided the features into two; node features from the graphs and company historical features from the relational data.

#### 4.4.1.1 Company Node Features

In Section 3.2.2, many centrality definitions and formulas are shown. We use these formulas to calculate the company node features.

For all job transition graphs from year *2000* to *2010*, features for graph nodes are calculated within their own networks. Moreover, as explained at the beginning of the centrality definitions (Section 3.2.2), parallel or self transitions between company nodes are ignored during node features' calculations. And also each node centrality value is normalized.

Some centrality definitions are more "global" to the network whereas some has usually "local" effect. In other words, to calculate the scores of the nodes, global definitions consider all nodes in the graph whereas local ones only look at the close links of the nodes. For example, degree related centralities are the local ones. If a node has more links or neighbors than the other node, this shows the close connectivity of the node with neighbors but in network level the situation can be different. The following node features are categorized considering this effect.

- **Local Node Features**

  - Incoming Degree Centrality: Normalized incoming degree centrality for a node is calculated according to Equation 3.3.

  - Outgoing Degree Centrality: Normalized outgoing degree centrality for a node is calculated according to Equation 3.4.

  - Incoming Link Ratio: Among all links/transitions of a company node $v$, ratio of incoming transitions to $v$ is calculated according to Equation 4.1.

  - Outgoing Link Ratio: Among all transitions of a company node $v$, ratio of outgoing transitions from $v$ is calculated according to Equation 4.2.

  - Self Link Ratio: Among all transitions of a company node $v$, ratio of self transitions for $v$ is calculated according to Equation 4.3.

$$percnt_{in}(v) = \frac{(\#\ in\ transitions) - (\#\ self\ transitions)}{\#\ all\ transitions} \quad (4.1)$$

$$percnt_{out}(v) = \frac{(\#\ out\ transitions) - (\#\ self\ transitions)}{\#\ all\ transitions} \quad (4.2)$$

$$percnt_{self}(v) = \frac{\#\ self\ transitions}{\#\ all\ transitions} \quad (4.3)$$

- **Global Node Features**

  Almost all pair of the nodes are scanned during the calculations of "global" centrality values. However, very low scores (about zero) are assigned to less important nodes. No enough information for less important company nodes is available and can be obtained. Furthermore, scanning all pair of nodes slows the calculation process. For these reasons, we bring some limitation to the nodes for which calculations are done. The aim is to get information for only more important nodes. We calculate the centrality values for only nodes having degree greater than $20$ for year graphs *2000-2006* and $25$ for *2007-2010*. This doesn't mean that we ignore other nodes during calculations. Centrality values are computed again over whole network according to formulas, but we obtain centrality values for only some nodes.

  - Proximity Incoming Centrality: Normalized proximity incoming centrality for a node as explained in Section 3.2.2.2 is calculated according to Equation 3.8 for nodes having the degree above the degree threshold.
  - Proximity Outgoing Centrality: Normalized proximity outgoing centrality for a node is calculated according to Equation 3.9.
  - Harmonic Incoming Centrality: Normalized harmonic incoming centrality for a node is calculated according to Equation 3.12.
  - Harmonic Outgoing Centrality: Normalized harmonic outgoing centrality for a node is calculated according to Equation 3.13.
  - Eigenvector Centrality: For eigenvector centrality, there is no separation as incoming/outgoing centrality, direction of notion is already included in the definition. However, we compute eigenvector centralities in two

ways by treating our graphs directed and undirected separately. Thus, two versions of eigenvector centralities are obtained. The formula used is Equation 3.14.

- Katz Centrality: There are two parameters for Katz centrality, attenuation factor $\alpha$ and initial centrality $\beta$. In our study, we calculate Katz centralities with $\alpha = 0.5$ and $\beta = 0.01$ giving better results in order to distinguish more important nodes from others. Like eigenvector centrality, two versions of Katz centrality values (directed and undirected) for nodes are computed according to Equation 3.16.
- PageRank: For PageRank, there is a controlling parameter $d$ which is damping factor. Usually $0.85$ is set for it. However, in our case employees may leave their companies with a probability of $0.5$, we don't have any information about it. Therefore, we calculate the PageRank's of the nodes with $d = 0.5$ and $d = 0.85$ separately. PageRank Equation 3.18 is used.

#### 4.4.1.2 Company Historical Features

Companies also have their own data with metadata fields. Furthermore, we can obtain the data and statistics about the companies from user professional experiences. So, independently from job transition graphs, company past features are extracted.

As mentioned before, company features are calculated for different years. If no employee experience information is available for the given company "until" the given year, no feature calculation is done for that company for the given year.

- External Turnover Rate: Employee turnover for a company can describe the number of employees moved within a certain period. This rate, also known as the "employee attrition rate", is calculated monthly, quarterly or annually [1]. In this thesis, we calculate annual employee turnover rate of each company for different years from *2000* to *2010* according to the formula;

$$turnover\ rate = \frac{\#\ of\ employees\ resigned\ during\ the\ year}{avg\ \#\ of\ employees\ during\ the\ year} \tag{4.4}$$

where average number of employees during the year is;

$$\frac{(total\ \#\ employees\ at\ January) + (total\ \#\ employees\ at\ December)}{2} \tag{4.5}$$

For external turnover rate feature, we only consider the "external" turnovers. This means that we only count the employees, in terms of resignation, who left the company during the year, internal position changes are ignored.

- Internal Turnover Rate: In our study, we believe that internal position or department changes can help us to find external turnovers for our problem. Therefore, this time Equation 4.4 is applied to internal turnovers and employees who make internal job changes at their companies are counted instead of resignations for internal turnover rate feature.

- External Turnover Rate Difference: External turnover rates are calculated for companies associated with different years from *2000* to *2010*. Additionally, we calculate the differences of external turnover rates comparing the given year's values and previous year's values. If the difference value is positive for a year (external turnover rate increases during that year), this means that employees tend to leave the company more than the previous year; if the difference value is negative, vice versa.

- Internal Turnover Rate Difference: Same approach mentioned above can be applied to internal turnovers. We calculate the difference values of internal turnover rates comparing the given year's values and previous year's values.

- Annual Stock Price Change: Stock ticker symbols are unique identifiers for publicly traded shares of particular stocks on a particular stock market. Especially, they are assigned to public companies. For example, 'AAPL' is a stock ticker for Apple Inc. and 'GOOG' is for Google Inc. In our dataset, we have stock ticker symbols for only some companies. As a metadata field, they are filled with metadata filling process as explained in Section 4.2.2.

  For stock price change feature, we get the historical stock prices from [4] for the companies which have a stock ticker symbol given in our data. We search the historical prices with ticker symbols between *2000* and *2010* years for each

month. According to monthly stock prices for companies, "annual" stock price change within the year is calculated as a feature for all years considering the difference of values between the first and last month of the year. Since not all years or months have corresponding stock prices (the company may become public later than the searched date), companies may have this feature for some years only. Also, some companies may have invalid stock ticker symbols or may not have a given stock ticker at all. Then, for these companies, this feature and the following feature cannot be calculated.

If annual stock price change is positive, this may mean that company's valuation is increased during the year; if it is negative, vice versa. This change can also affect the employees working at the company in terms of turnover.

- Annual Stock Price Change Difference: We calculate the differences of annual stock price changes of the public companies comparing the given year's values and previous year's ones. We can compute this feature only if both previous and given year have annual stock price change feature for a specified company. If difference feature value is positive, this means that at the given year, stock price increases at a faster rate compared to the previous year. If it is negative, then increase rate for the given year is not as fast as the previous year's one or stock price decreases.

### 4.4.2 Employee Features

In our study, we analyze mainly the employees and so each feature vector should be constructed with employee features. Employee features can be categorized into two; features calculated from employee's own data and features calculated from employee's current and past companies.

There can be more than one companies that employee worked for in the past. Hence, corresponding feature values of many companies should be reduced to only one feature value for each company feature definition. For this purpose, past company features are aggregated with "aggregate" functions. These aggregate functions include *minimum, maximum, average*. For example, three company feature values of *pageRank* are calculated for each employee as *past-pageRank-min, past-pageRank-max,*

*past-pageRank-avg*. They indicate the aggregated pageRank values of the past companies of the employee with three different perspectives. Throughout the employee feature calculations, directly company related features are aggregated by this way.

Another issue about employee features is the degree threshold. Not all companies have enough information and so companies having a few transitions (mostly noise) can not contribute to our problem due to lack of information. Like in the company feature calculations, degree threshold (20 for *2000-2006* graphs and 25 for *2007-2010*) also applies to employee features. In the corresponding graph, companies having a few transitions below degree threshold are considered as "invalid" and do not included/counted in the feature calculations. If an employee do not have any "valid" past/current companies, the related feature would be empty for the employee.

As mentioned before, each employee feature is calculated for different years.

#### 4.4.2.1 Employee Historical Features

Employee historical features are the ones from employee's own data and past experiences. Surely, current and past companies given in experiences are used for calculating historical features, but they are not used as "nodes".

- Current Working Time: This is the working time (in months) of the employee at the current "valid" company. For example, if an employee started to work for a company at *March 2005* and still continues for his/her company at given year *2006*, difference between *December 2006* and *March 2005* (22 months) will be current working time feature of the employee.

- Number of Past Companies: This is the number of "valid" companies (having enough transitions in the graphs) for which the employee worked. Past companies are counted until the end of given year. The current company is not included in this number.

- Past Working Time: This is an aggregated feature. For each past valid company, employee's working time at that company is calculated. Then minimum, maximum and average of these calculated past working times are put to feature

vector as *minPastWorkTime, maxPastWorkTime, avgPastWorkTime*.

- Experience: Experience (in months) involving all professional experiences at valid past and current companies is calculated.

- External Turnover Rate: This is count of the employee's company changes. The left company should be "valid" in terms of degree threshold and also unemployment at that company should be within last $5$ years from the given year. We limit the past time to $5$ years in order to compare different year's data (training and testing). Otherwise, towards *2010*, external turnover rate would increase with the more experience information of the employee.

- Internal Turnover Rate: Inversely, this is number of the internal position changes of the employee within the same "valid" company. Again, last $5$ years from the given year is considered. Internal job changes are thought to contribute to turnover problem as well as external ones.

- Number of Industry Change: This is the number of industry changes by the employee within last $5$ years. Start or end company should have enough transitions at given year graph, in order to count this change into industry change feature.

- Number of Jobs: Number of "valid" jobs until the end of the given year is calculated for each employee.

- Number of Universities: Number of educational institutions (mostly universities) that the employee declares in educational experience is calculated.

#### 4.4.2.2 Employee Features from Companies

Each company feature explained in Section 4.4.1 is calculated as employee feature with the help of aggregated functions *minimum, maximum, average*. But aggregated functions are used only for the companies in the past. This is because we study on the employees currently worked for only one company.

For each employee and company feature definition, *four* company-related employee features are calculated. For example, for company *annual stock price change* feature,

*curr-stockChange, past-stockChange-min, past-stockChange-max, past-stockChange-avg* features are inserted into feature vector of an employee considering employee's past and current companies.

In fact, count of feature definitions is only $27$. However, with the aggregated company-related employee features and employee's own features, size of the feature vector for each employee and each given year becomes $95$. While some of them are put for the parameter selection, most of them comes from aggregated functions. Therefore, feature selection phase plays an important role to select "best" features.

Full lists of the features before and after feature selection process can be seen in Appendix A.1.

### 4.4.3 Class Label

For each feature vector of the employee, one class label should be assigned indicating the turnover status of the employee. In our study, we have two classes; "turnover" as $1$ and "no-movement" as $0$. For calculating class label, only "external" turnovers are taken into consideration. This means that we are only interested in "company" lefts during our study.

The situations below are *not considered as turnover* (in terms of $X$ company):

- Working at $X$ company for $A$ and $B$ positions, then leaving $A$ position and only proceeding with $B$ position
- Leaving $A$ position at $X$ company and starting to work for $B$ position at the same company (internal - position change)
- Without leaving $A$ position at $X$ company, starting to work for $B$ position at $Y$ company

The situations below are *considered as turnover* (in terms of $X$ company):

- Leaving $A$ position at $X$ company and starting to work for $B$ position at $Y$ company (external - company change)
- Leaving $A$ position at $X$ company and stopping working for some time period

Apart from above situations, there can be date related issues for internal position changes.

- No turnover situation: Leaving current position at *April 2003* and starting another position at the same company at *May 2003*
- Turnover situation: Leaving current position at *April 2003* and starting another position at the same company at *June 2003*

Considering above two situations, no turnover occurs for the first situation because there is no gap between two positions. For the second one, turnover occurs at *April 2003* and class "1" is assigned to the employee. Because in this case *May 2003* is not known time period.

Class label can not be calculated by only given year. In order to determine the turnover, a specific time period is also needed. For this purpose, *future window parameter* is defined in years. If the employee works for a company at given year $y$, we investigate the turnovers within the later $t$ years which $t$ is future window parameter. If employee leaves current company until year $y+t$, then class label of this employee feature vector will be "1" for the year $y$ and the future window $t$. Otherwise, "0" is determined.

In this study, future window of 1, 2, 3 and 5 years are tried. Since class label is determined by two parameters *year* and *future window*, experiments are conducted for these two parameters. The experiments and experimental results associated with each year and future window parameter are illustrated in Chapter 5.

After the construction of feature vectors for each pair of employee and future window, feature selection methods should be applied to our datasets.

## 4.5 Feature Selection

For each employee, feature vector size becomes too large (95) after the extracted features with aggregated company features and employee's own features as explained in Section 4.4.2. Before classification phase, dimensionality reduction process should

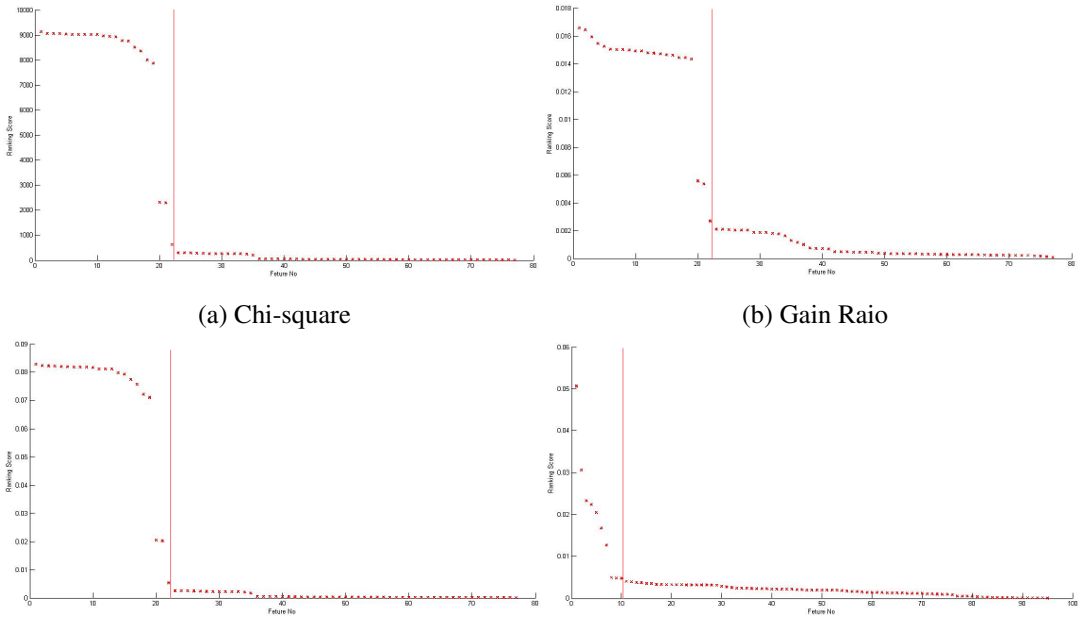
be employed to reduce redundant or irrelevant features. In this phase, feature selection techniques are applied to select a minimum set of "relevant" features contributing mostly to class labels.

As a dimensionality reduction technique, feature extraction method which transforms features into a reduced representation of features [3] is not preferred, because for our problem training and testing data is taken from different sources (years) (Section 5.1). In feature extraction, it is necessary to apply the same transformation formulas to both training and testing data. However, for our case, it is so complicated situation which requires saving and restoring transformations several times.

Feature evaluation methods which rank the importance of features according to the class label are listed and explained in Section 3.3. We apply these methods on training data through WEKA and LibSVM tools. All methods perform cross-validation on data when calculating the rankings to give more accurate results. While 5-fold cross validation is used for F-score based feature selection in LibSVM, other evaluation methods in WEKA (Chi-square, Gain Ratio and Information Gain based) use 10-fold cross validation.

By using four different methods, several thresholding experiments are conducted on different years' data (*2000* and *2005*) with different future window parameters that determine the class labels (future window of $1$ and $5$). Each experiment's ranking scores assigned to the features are presented in a sorted way on the graphical interface and a threshold for each experiment is determined. The features that meet a certain threshold for the given experiment are taken into consideration and included in the "candidate" feature set. Thresholding experiment results for year *2005* and future window of $5$ years are shown below.

According to Figure 4.4, x-axis represents the feature number in a sorted way, y-axis represents the ranking score given by the evaluation method. Scores are sorted in decreasing order and labeled with the red points. The red line indicates the determined threshold. Features corresponding to the score points on the left side of the threshold line are inserted into candidate feature set. In this figure, about $22$ features are included in the "candidate" feature set. Results from the experiments on year *2000* with future window of $1$ & $5$ years are almost the same as the above results.



(a) Chi-square

(b) Gain Raio

(c) Information Gain

(d) F-score

Figure 4.4: Thresholding experiments for *2005* and future window of 5 years

Combining different "candidate" feature sets from different experiments, the last feature subset is composed. Last feature set after feature selection phase contains 24 features. Feature sets before and after feature selection can be seen in Appendix A.1.

While decreasing the feature set size from 95 to 24, some findings emerge:

- Among employee features, features from employee's past companies are ignored and not included in the last feature set. This means that they do not provide enough information about the employee turnover. These features are basically aggregated centrality values, turnover rates and annual stock price changes of the past companies. Moreover, employee's working time in past companies is discarded.

- For eigenvector centrality and Katz centrality features of the companies, two versions are calculated treating the graphs as directed and undirected. According to feature selection results, directed centrality values are observed as more important than the undirected ones in the context of the turnover. Therefore, only directed centrality values are included in the last feature set.

- PageRank centralities of the company nodes are calculated for both $d = 0.5$

and $d = 0.85$ separately. After the feature selection phase, PageRank feature with parameter $d = 0.5$ is selected only.

- Number of jobs/positions worked for and number of educational institutions declared so far are eliminated after feature selection.

After generating the last feature set of size $24$, data of the years from *2000* to *2010* including only $24$ features is ready for classification.

### 4.6 Classification

Main focus in this study is to classify employee profiles into two classes; employees who leave their current companies (*external turnover*) and employees who continue to work for current company (*no-movement*). We have to look at the turnovers within the specific year period called *future window*. By considering $x$ years later which $x$ is future window parameter, class label of each employee feature vector is calculated as "1" for turnover or "0" for no-movement.

In this study, different future window parameters (1, 2, 3 and 5 years later) are used. For each future window, models of different "trainable" years are trained separately and then models are tested on different "testable" years for the same future window. The details of the experiments, training and testing data considerations and experimental results can be found in Chapter 5.

Support Vector Machine (SVM) algorithm (explained in Section 3.4.1) is used as main classification algorithm during our study. We mainly choose SVM because;

- It is only directly applicable for two-class tasks. Since our problem is binary classification problem, it directly fits to our problem.
- SVM requires each data instance is represented as a vector of real numbers. This also applies to our problem, we have only numerical attributes. We don't have to make any conversion to our already calculated attributes.
- According to experimental results, other tried classifiers do not give as successful results as SVM.

LibSVM software [21] is used for applying Support Vector Machine algorithm. To get acceptable results, proposed procedure explained in the guide of LibSVM [36] is taken into account. As a first step, calculated feature vector values are scaled to the range $[0, 1]$. Scaling factor is obtained from the year *2009* that has the largest cumulative employee profiles. Then, same scaling factor is applied to both training and testing data of all years.

In addition to linear classification, SVM can efficiently perform a non-linear classification using the kernel trick. The choice of kernel is also a very important decision, since it affects the whole process. In this study, we choose non-linear classification for our problem because linear classification does not give the desired results. As a kernel function, Gaussian RBF kernel is selected as a reasonable first choice because RBF kernel has fewer numerical difficulties/parameters compared to other kernels and it is generalized version of other kernels. RBF kernel nonlinearly maps data into a higher dimensional space and can handle the nonlinear relation between class labels and attributes.

The effectiveness of SVM depends on also the kernel's parameters, and soft margin parameter $C$. When training an SVM with RBF kernel, two parameters $C$ and $\gamma$ are considered (Section 3.4.1). Good selection of these parameters may make the training very successful, whereas poor selection may cause it to fail. Therefore, the best combination of $C$ and $\gamma$ is often selected by a "grid search" with exponentially growing sequences of $C$ and $\gamma$. Each combination of parameter choices is checked using cross validation, and the parameters with best cross-validation accuracy are picked. LibSVM provides a script "grid.py" to extract best $C$ and $\gamma$ values by grid search. This script is used in our study with default exponential growing sequences of $C \in \{2^{-5}, 2^{-3}, ..., 2^{13}, 2^{15}\}$ and $\gamma \in \{2^{-15}, 2^{-13}, ..., 2^{1}, 2^{3}\}$ and with 5-fold cross validation. For each pair of trainable year and future window, best $C$ and $\gamma$ values are extracted for models to be trained.

After the grid search, using the selected best parameters (no cross-validation), the final model for each year associated with future window parameter is trained on the whole training data that is composed for the given year and future window. So, many models are created which are used for testing later. Experiments with both model and

test years are shown in experiments tables in Section 5.1. These tables illustrate all experiments that are conducted during our study.

Apart from SVM algorithm, to get comparative results, Decision Table/Naive Bayes Hybrid Classifier (DTNB) and Neural Network algorithms are also tried. But, not all models are trained with these algorithms. Only for the best model year(s) obtained from SVM experiments, models are trained with these algorithms and results of two are compared to SVM ones.

DTNB algorithm [31] explained in Section 3.4.2 is applied through WEKA tool's [32] command line option. Default parameters are used and as a performance evaluation measure for selecting attributes, accuracy is used. Using the forward selection search in the algorithm, at each step selected features from our data (almost quarter of all features) are modeled by decision table and the rest by Naive Bayes. A few DTNB models are trained for the best SVM model years. However, comparisons show that DTNB is not as successful as SVM. It may be caused by the nature of the algorithm, use of decision tables and probabilistic models, which may not be suitable to our all numeric data.

For Neural Network algorithm, Multilayer Perceptron classifier is applied through again WEKA's command line. Since our data is scaled, no normalization of class value and attributes are applied again. In default mode, $14$ hidden layers are created for the network by the algorithm. Learning through adjusting weights in each layer (backpropagation), a few Neural Network models are trained for the best SVM model years. For the given years, Neural Network algorithm does not give promising results as SVM. There are many parameters that should be adjusted to compose a successful neural network which is a very complex issue.

In Chapter 5, mostly Support Vector Machine model results are shown. Comparative results of all classification algorithms for the best SVM model years are also shown in Section 5.4.6.

In this study, training and testing data is separated automatically by the years. Training data for each year are composed as having balanced class distribution while testing data have two versions, one for balanced and one for preserved original class

distribution. Training and testing data details are explained when experimental details are given in Section 5.1.

When applying Support Vector Machine algorithm in classification phase, some other tested points are listed as follows:

- Trainings of the models are tried with original class distributed training data. These models fail for predicting employee turnovers.
- Linear classification using LibLINEAR is also tried but successful results are not obtained.
- Since "0" classes dominate over "1" classes in number, weights for "1" classes are given more than "0" classes. But this try does not give promising results during training process.

Since the above points do not give desired results, we continue the use of SVM with balanced class distributed training data, non-linear classification with RBF kernel and unweighted turnover classes in training data.

# CHAPTER 5

# RESULTS AND DISCUSSIONS

In this chapter, details of the experimental settings, baselines for our problem, evaluation metrics and results of classification experiments are presented.

## 5.1 Experimental Setup and Details for Experiments

Before showing the experimental results, it is useful to give details about the training and testing data and how experiments are conducted.

In this study, we split the original dataset into years and graphs, and compose the feature vectors of the employees for each year from *2000* to *2010* as explained previously in Chapter 4. In summary, our problem is to predict whether the employee will leave his/her company until year $y + t$ if we are given an employee profile working for a company at year $y$. To determine the (external) turnover, we should *train our model with the features from one year and test this model for the next years*. Here there is a limitation factor, future window parameter $t$, which determines the class label of the given feature vector (Section 4.4.3). In our study, four different future window parameters/years $1$, $2$, $3$ and $5$ are used. This means that we try to predict the turnovers that can occur within the period of $1$, $2$, $3$ or $5$ years after the given year.

Training and testing years are limited due to future window parameter. For example, we cannot create a model of *2006* year for future window of $5$ years because data of *2011* ($5$ years later) is not available and turnover (class label) cannot be determined. Additionally, we cannot model *2005* year again with future window of $5$ years due

to the absence of testing year. We should have *2006* or later years for testing but no class labels can be assigned for these years considering $5$ years later.

Experiments are performed with these future window parameters separately. In Table 5.1, Table 5.2, Table 5.3 and Table 5.4, experiments that can be conducted are shown with the corresponding model year in row and test year in column for future window of $1$, $2$, $3$ and $5$ years respectively.

Another detail is that if the model is trained for one future window, this model cannot be tested on data having class labels determined by different future window. This is obvious result because class labels are not calculated within the same period.

Table 5.1: Experiments for future window of $1$ year

| | | Test Year | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 |
| Model Year | 2000 | X | X | X | X | X | X | X | X | X |
| | 2001 | | X | X | X | X | X | X | X | X |
| | 2002 | | | X | X | X | X | X | X | X |
| | 2003 | | | | X | X | X | X | X | X |
| | 2004 | | | | | X | X | X | X | X |
| | 2005 | | | | | | X | X | X | X |
| | 2006 | | | | | | | X | X | X |
| | 2007 | | | | | | | | X | X |
| | 2008 | | | | | | | | | X |

Table 5.2: Experiments for future window of $2$ years

| | | Test Year | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 |
| Model Year | 2000 | X | X | X | X | X | X | X | X |
| | 2001 | | X | X | X | X | X | X | X |
| | 2002 | | | X | X | X | X | X | X |
| | 2003 | | | | X | X | X | X | X |
| | 2004 | | | | | X | X | X | X |
| | 2005 | | | | | | X | X | X |
| | 2006 | | | | | | | X | X |
| | 2007 | | | | | | | | X |

Table 5.3: Experiments for future window of 3 years

| | | Test Year | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 |
| Model Year | 2000 | X | X | X | X | X | X | X |
| | 2001 | | X | X | X | X | X | X |
| | 2002 | | | X | X | X | X | X |
| | 2003 | | | | X | X | X | X |
| | 2004 | | | | | X | X | X |
| | 2005 | | | | | | X | X |
| | 2006 | | | | | | | X |

Table 5.4: Experiments for future window of 5 years

| | | Test Year | | | | |
|---|---|---|---|---|---|---|
| | | 2001 | 2002 | 2003 | 2004 | 2005 |
| Model Year | 2000 | X | X | X | X | X |
| | 2001 | | X | X | X | X |
| | 2002 | | | X | X | X |
| | 2003 | | | | X | X |
| | 2004 | | | | | X |

According to experiments tables, it can be seen that many comparisons can be done with experiment results apart from each experiment's own evaluation. Moving along a row of the tables, model aging can be tested. Moving along a column of the tables, test year's turnover results from different models can be compared. From different perspectives, experiment results are presented in Section 5.4.

We put some limitations to employee profiles in our experiments. These limitations are below:

- First of all, training or testing data for year $y$ should compose of the employee profiles who continue to work for their companies considering the end of the year (*December y*).
- An employee can work for many companies at the same time. For our experiments, we select the employees working for only one company at the end of the year.

- Another limitation to employee profiles for the experiments are made in terms of companies. Top 100 and top 25 companies for each year are extracted from our data in terms of node pageRank and harmonic closeness centrality values (see Appendix A.2 for the full list of top 100 companies for years *2000* and *2009*). Employees currently working for a top company at the given year are considered for training or testing. In other words, in our study we analyze the employees from only top companies in terms of turnover.

  In Table 5.5, numbers of the currently working employees at top companies are shown for different years. Employee counts generally increase towards the year *2009*.

Table 5.5: Numbers of currently working employees at top 100 and top 25 companies for different years

| | Top 100 Companies | Top 25 Companies |
|---|---|---|
| 2000 | 106622 | 49227 |
| 2001 | 110409 | 51153 |
| 2002 | 113224 | 55454 |
| 2003 | 119332 | 57682 |
| 2004 | 127805 | 60612 |
| 2005 | 137186 | 65290 |
| 2006 | 149164 | 70334 |
| 2007 | 158344 | 74690 |
| 2008 | 162290 | 77324 |
| 2009 | 157171 | 78514 |

In addition, when we create the training and testing data from composed data structures respecting the above limitations, we take several points into consideration:

- When creating training data, distribution of class labels is an important issue. For our data, "0" classes are generally dominant over "1" classes in number. Therefore, we prefer randomly picking employee profiles such that training data for each pair of year and future window contains balanced employee profiles in terms of their turnovers (class labels).
- Even if we put the company limitation, there are still many employee profiles that can be analyzed. In order to reduce training time, 25000 employ-

ees working at top 100 companies ($12500$ turnover and $12500$ no-movement) and $15000$ employees working at top 25 companies ($7500$ turnover and $7500$ no-movement) for the given year are taken randomly into training data of that year.

- In terms of testing data, two versions are used. One is the testing data having preserved original class distribution and without any limitation on profiles count. As an another version, if we test the year $y$ model, training data (balanced and having limited number of employees) generated for the following years are used for testing purpose because they are not used in the training of $y$ model. Therefore, there are titles "Model Year" and "Test Year" in the above experiments tables.

As a result, there are two sets of the training and testing data that contain employees from top 100 companies and top 25 companies. Models are trained with both sets as explained previously in Section 4.6 and experiments are conducted, but turnover analysis for employee profiles from top 25 companies is more successful than top 100 ones. Hence, experimental results of top 25 companies' employees are presented in Section 5.4.

In this thesis, Support Vector Machine (SVM) algorithm is used mainly for classification and training of models. The classification performance under Decision Table/-Naive Bayes Hybrid Classifier (DTNB) and Neural Network algorithms is also analyzed, but these algorithms do not give promising results. Therefore, we continue with the SVM algorithm and results of SVM algorithm are shown in Section 5.4. The comparison between SVM and different classifiers for the best experimental setup is also shown in Section 5.4.6.

## 5.2 Baseline

For each experiment conducted as explained in the previous section, a simple baseline model should be taken as a point of reference. In this study, we choose three different baselines from employee historical features (Section 4.4.2.1):

1. **Average current working time (in months):** Current working times of the employees, who are known as involved in turnover within given period (class label "1"), are averaged out for each year. Calculated average value for each pair of year and future window is set as baseline value.

2. **Average experience (in months):** Professional experiences of the employees, who are known as involved in turnover within given period (class label "1"), are averaged out for each year. Calculated average value for each pair of year and future window is set as baseline value.

3. **External turnover rate of** 1**:** At least one external turnover rate for an employee is considered and set as baseline value for all years and future windows.

Table 5.6: Current working time averaged baseline values (in months)

| | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 |
|---|---|---|---|---|---|---|---|---|---|
| | Top 100 companies | | | | | | | | |
| fw1,2,3 | 49 | 51 | 51 | 51 | 51 | 52 | 53 | 57 | 37 |
| fw5 | 49 | 54 | 57 | 60 | 58 | | | | |
| | Top 25 companies | | | | | | | | |
| fw1,2,3 | 53 | 55 | 55 | 54 | 52 | 54 | 50 | 56 | 53 |
| fw5 | 54 | 57 | 59 | 61 | 59 | | | | |

Table 5.7: Experience averaged baseline values (in months)

| | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 |
|---|---|---|---|---|---|---|---|---|---|
| | Top 100 companies | | | | | | | | |
| fw1,2,3 | 76 | 77 | 78 | 80 | 82 | 87 | 94 | 103 | 94 |
| fw5 | 76 | 80 | 85 | 91 | 93 | | | | |
| | Top 25 companies | | | | | | | | |
| fw1,2,3 | 77 | 78 | 81 | 83 | 84 | 89 | 92 | 102 | 93 |
| fw5 | 75 | 80 | 84 | 90 | 93 | | | | |

Calculated (average) baseline values for current working time and experience features are shown in Table 5.6 and Table 5.7. For each year and future window, different baseline values are calculated because employee profiles can change by year and class labels (turnovers) can change by future window parameter. Moreover, baseline values depend on top 100 or top 25 companies' employees. For future window of 1,

2 and 3 years, calculated baseline values are almost equal, so one value is set for all of them. Since 5 year period is a large period, baseline values for future window of 5 years are usually larger.

For each baseline model, if an employee's corresponding feature value is greater than the baseline value, employee is predicted as involved in turnover (class label "1"). Otherwise, class label "0" is assigned for this employee. For example, if an employee is working for a company from top 25 list for 60 months considering the end of *2004*, according to current working time baseline values in Table 5.6, class label "1" (turnover) is assigned to this employee with respect to 1, 2 or 3 years later. Because currently working time value 60 is greater than the baseline value 54 for *2004* and future window of 1, 2 and 3 years. The same analogy applies to other baselines.

Results of each baseline model are compared to our proposed models' results in experiments (Section 5.4). Experience baseline results are almost the same as the results of current work time baseline. Therefore, only external turnover and current working time baselines are used for comparison in our experiments.

## 5.3 Evaluation Metrics

Collected test results of our experiments are evaluated and compared on the basis of certain statistical terms; accuracy, precision, recall and F1 score. True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) terms used in binary classification are described in Table 5.8 as confusion matrix. In our case, Positive Instance corresponds to an instance having class label "1" (turnover) and Negative Instance corresponds to an instance having class label "0" (no-movement).

*Accuracy* is simply the ratio of correctly classified instances (employee profiles) over all instances. It is the degree of closeness of a measure to actual value. It can be calculated using Equation 5.1.

*Precision* is the probability that a (randomly selected) positively classified instance is indeed positive. Perfect precision value 1.0 means that every instance classified as positive is indeed a positive instance, but says nothing about whether all positive

instances are retrieved. The precision equation is in Equation 5.2.

*Recall* is the probability that a (randomly selected) actual positive instance is correctly classified. Perfect recall value 1.0 means that all positive instances are classified as positive, but says nothing about how many negative instances are also classified as positive (False Positive). The recall value can be calculated with Equation 5.3.

*F1 score* (balanced F-score) is a popular measure which is the harmonic mean of precision and recall. F1 score can be calculated using precision and recall as shown in Equation 5.4.

Table 5.8: Confusion matrix description

| | Positive Instance | Negative Instance |
|---|---|---|
| Classified as Positive | True Positive | False Positive |
| Classified as Negative | False Negative | True Negative |

$$Accuracy = \frac{TruePositive + TrueNegative}{|PositiveInstance| + |NegativeInstance|} \tag{5.1}$$

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{5.2}$$

$$Recall = \frac{TruePositive}{|PositiveInstance|} \tag{5.3}$$

$$F_1\ score = 2\ \times\ \frac{precision\ \times\ recall}{precision\ +\ recall} \tag{5.4}$$

Even if the experiment results are compared according to all of these metrics, accuracy and F1 score are the ones that we take into consideration more than others.

## 5.4 Experimental Results

In our study, the aim is to classify working employees to two classes; employees who leave their current companies (external turnover) and employees who continue to work for current company (no-movement) within specific period. After the classification phase and trained models, experiments/tests are conducted according to exper-

iments tables (Table 5.1, Table 5.2, Table 5.3 and Table 5.4) as explained in Section 5.1.

Experimental results for the employee profiles from top 25 companies are shown in first four subsections. Support Vector Machine (SVM) is the main classification algorithm during our study, hence results of SVM generated models are usually presented in this section. For testing, two versions of testing data (balanced class distributed and preserved original class distributed) are used. Each experimental result is compared to two baselines (current work time and external turnover baselines) with respect to evaluation metrics. In Section 5.4.6, from different perspectives, the results are compared to each other (different classifiers results comparison, employee profiles from top 100 companies, model aging etc.).

Since one model can be tested on many years according to experiments tables, experimental results of each model on all testable years are averaged out and a single average value is obtained for a year model for a future window. These values are calculated for all evaluation metrics and are used to compare and evaluate the models. Similarly, for each baseline model, results on different years are averaged out for each future window. There is only one baseline model for a specific feature definition that uses a simple comparison rule, so a single average value is obtained for each baseline model without a model year concern. Therefore, a baseline model's results (from model year perspective) are the same for all model years for a future window.

### 5.4.1 Results for Future Window of 1 year

For the experiments with future window of 1 year, we aim to predict the turnovers within one year period by considering employee features of the given year. The results are in Figure 5.1, Figure 5.2 and Figure 5.3.

According to accuracy results, tests with balanced class distributed data (with limited 15000 employee profiles) have about stable accuracy results for all models while test results of original class distributed data (without limitation on number of employee profiles) fluctuate between model years. Hence, at first glance, it can be said that balanced test data gives more robust results.


Figure 5.1: Average accuracy results of the models for future window of 1 year

For balanced test data, accuracy results are above the current work time baseline accuracies for all models. Mostly results are also above or close to the external turnover baseline results. In terms of accuracy, the most successful result is obtained with *2005* model. For all models, accuracies are above $50\%$.

According to precision results on Figure 5.2, balanced test data results are above the precision values of two baselines for all models. Tests on original class distributed data have very low precision values meaning that mostly positively classified instances (saying turnovers) are not actual positives/turnovers.

Recall results in Figure 5.2 for future window of 1 year show that except *2002* and *2003* models, all model tests have the recall values above the baselines. This means that most of the actual turnovers are predicted correctly with our models. According to recall values, almost all models are successful but the most successful one is *2001* model and the failed model is *2003*. An interesting point about the recall values is that test results on original class distributed data are almost the same as corresponding model's test results on balanced data. This indicates that actual turnovers (1 classes) can be predicted correctly on both original and balanced data.

F1 scores are considered more important than precision and recall values for evaluation. According to F1 scores for the experiments with future window of 1 year (Figure 5.3), except for *2002* and *2003* models, all models are successful having F1

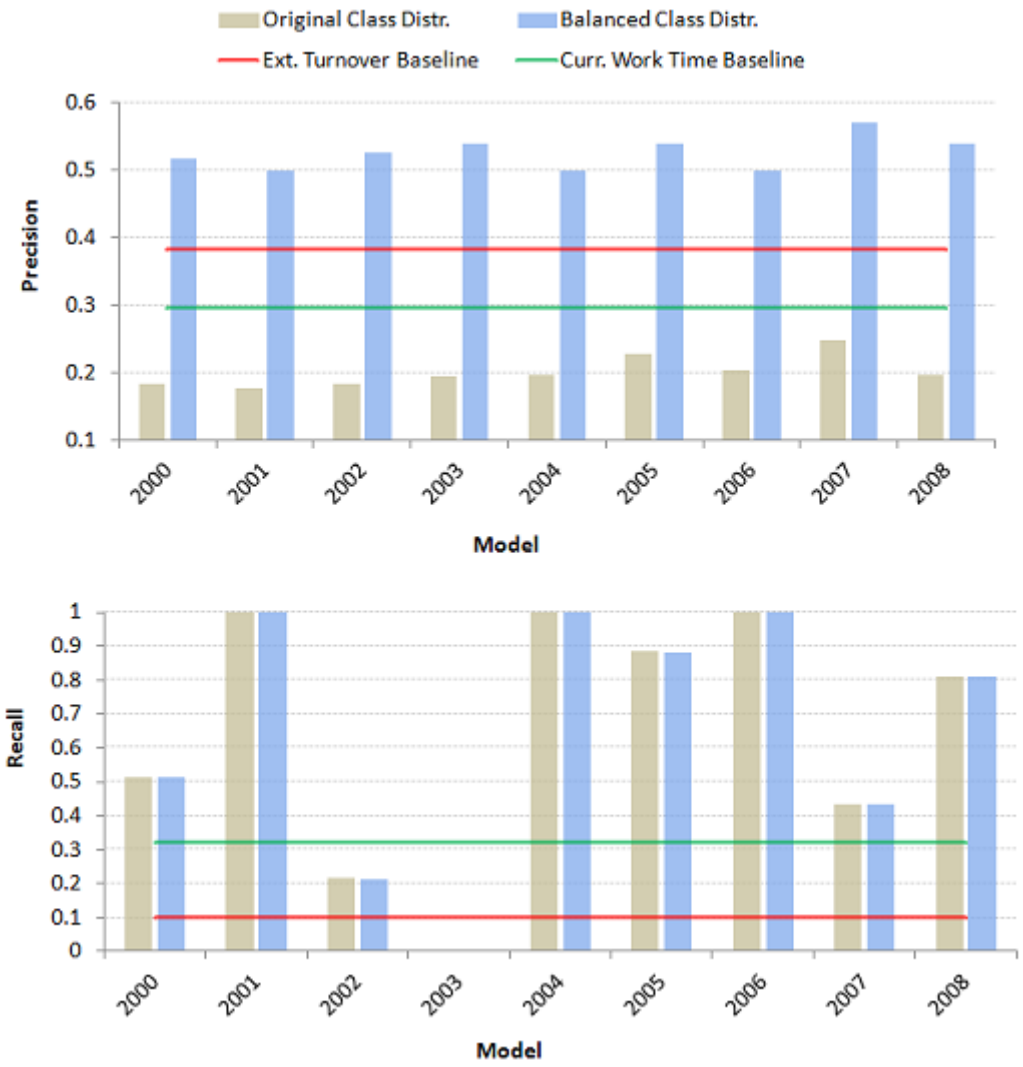



Figure 5.2: Average precision and recall results of the models for future window of 1 year

scores above baseline F1 scores. The most successful model with respect to F1 score is *2005*.

Except for model *2003*, average of evaluation metrics are calculated for all experiments performed for future window of 1 year and shown in Table 5.9. According to the difference between our model and baseline average metrics, our proposed models are considerably successful over baseline models for predicting turnovers within 1 year period. The largest difference occurs with the external turnover baseline considering recall and F1 scores. Also, difference between precision and recall scores of external turnover baseline is higher compared to current work time baseline. Hence, it can be said that external turnover baseline is worse than current work time baseline.


Figure 5.3: Average F1 scores of the models for future window of 1 year

Table 5.9: Average evaluation of all experiments conducted for future window of 1 year

| | Our Model | Ext Turnover Baseline | Diff | Curr Work Time Baseline | Diff |
|---|---|---|---|---|---|
| Accuracy | 0.53 | 0.51 | 0.02 | 0.46 | 0.07 |
| Precision | 0.52 | 0.38 | 0.14 | 0.29 | 0.23 |
| Recall | 0.73 | 0.10 | 0.63 | 0.32 | 0.41 |
| F1 Score | 0.57 | 0.15 | 0.42 | 0.28 | 0.28 |

In this study, we take accuracy and F1 score evaluation metrics into consideration more than others. Therefore, according to these metrics, the most successful model is *2005* while the worst ones are *2002* and *2003*. The failure of the model *2003* (and maybe *2002*) may be caused by data itself for these years because economy in developed countries is under the effects of early *2000*s recession [2] at these years. Due to this economical complication, employment/unemployment may depend on more complex factors and may not be predicted.

Another important result from our experiments is that tests on balanced class distributed data become more successful than tests on original class distributed data. This can become a "normal" outcome as models are trained on balanced data.

### 5.4.2 Results for Future Window of 2 years

Classification results for future window of 2 years are not as successful as the future window of 1 year results. This situation can be caused by randomly selected 15000 profiles data. Here, we expand the period to 2 years that employee turnovers can occur. The experimental results of future window of 2 tests can be seen in Figure 5.4, Figure 5.5 and Figure 5.6.

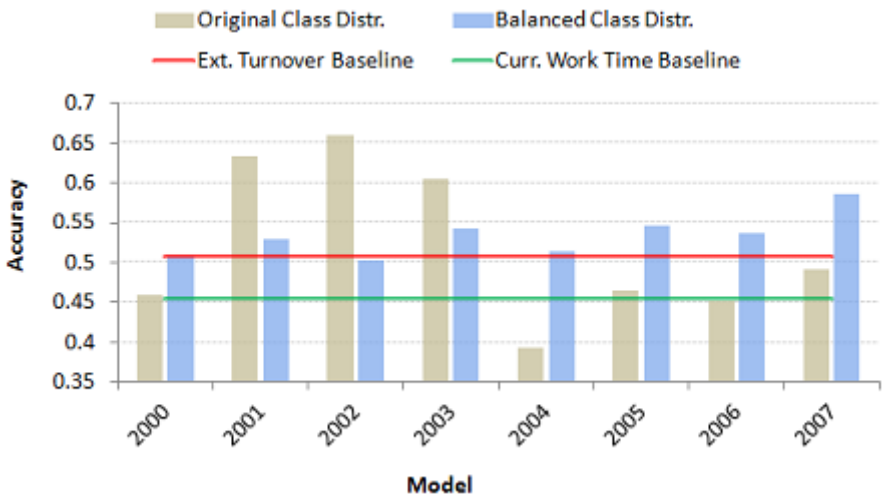



Figure 5.4: Average accuracy results of the models for future window of 2 years

According to accuracy results, all test accuracies on balanced data are above the best baseline (current work time baseline) accuracies. Original class distributed test data results again do not give reliable results. The best model with respect to accuracies is *2007* with accuracy of 58.65%.

Precision results are again above the current work time baseline precisions for all model tests on balanced data. But the difference between our results and baseline scores is not as large as the results of future window of 1. This is because of the fast increase of baseline scores. For example, current work time baseline precision increases from 0.29 to 0.48 compared to future window of 1.

Considering baseline models, successful recall results are obtained for all models except for *2001*, *2002* and *2003* years. Two baseline recall results are almost equal for future window of 2 experiments. The best model considering recall values is *2004* with recall of 0.96.

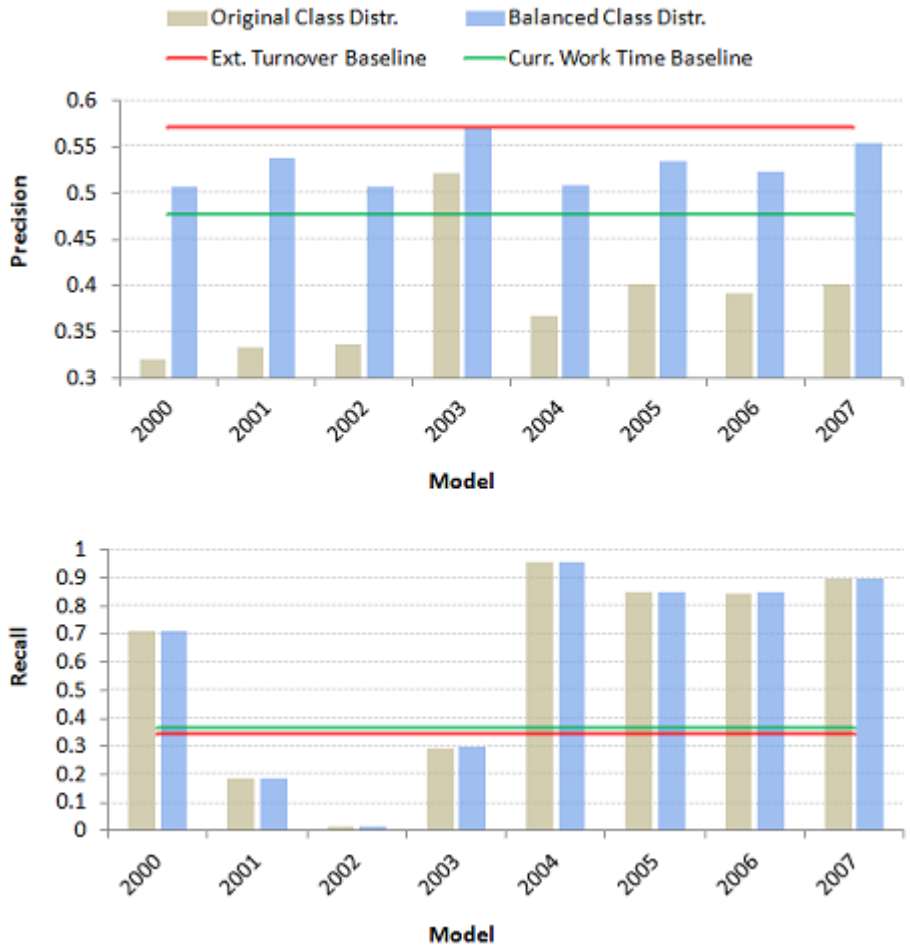



Figure 5.5: Average precision and recall results of the models for future window of 2 years

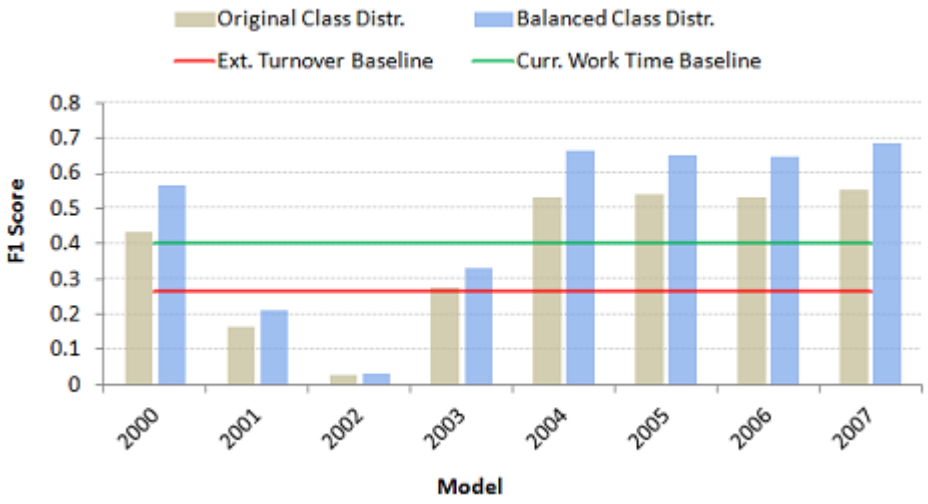



Figure 5.6: Average F1 scores of the models for future window of 2 years

More reliable F1 scores tell us almost the same results as the recall values. Except for *2001*, *2002* and *2003*, experiments on balanced data for all models are above the external turnover and current work time baselines with the difference of $0.27$ and $0.14$ in average, respectively. The model *2002* is unsuccessful with respect to F1 score.

Except for the worst model *2002*, evaluations of the results are averaged out for all experiments of future window of $2$ (Table 5.10). According to Table 5.10, our model's average accuracy increases compared to future window of $1$ results. But other metrics remain the same while baseline metrics increases. Although results of our proposed models are still more successful than baseline models, differences between baseline metrics and ours are not as large as the previous experiments (future window $1$).

Table 5.10: Average evaluation of all experiments conducted for future window of $2$ years

| | Our Model | Ext Turnover Baseline | Diff | Curr Work Time Baseline | Diff |
|---|---|---|---|---|---|
| Accuracy | 0.54 | 0.51 | 0.03 | 0.45 | 0.09 |
| Precision | 0.53 | 0.57 | -0.04 | 0.48 | 0.06 |
| Recall | 0.68 | 0.34 | 0.33 | 0.37 | 0.31 |
| F1 Score | 0.54 | 0.26 | 0.27 | 0.40 | 0.14 |

Considering $2$ years later, turnovers can be predicted more accurately with *2004* and *2007* models. *2001*, *2002* and *2003* models (especially *2002*) fail to predict turnovers. Their failure again may be caused by the data and economical issues at these years. As a result, future window of $2$ experiments do not give as successful results as future window $1$ ones. Another reason for this situation is that the increase rate of baseline scores for future window of $2$ is very fast compared to future window of $1$ baseline scores.

Other findings (more successful performance of tests on balanced data than original class distributed data, and the success of current work time baseline over external turnover baseline) are still applicable to future window of $2$ experiments.

### 5.4.3 Results for Future Window of 3 years

The success of our models increases for future window of 3 experiments. This means that we predict the employee turnovers that occur within the period of 3 years more successfully than smaller periods. The results are in Figure 5.7, Figure 5.8 and Figure 5.9.

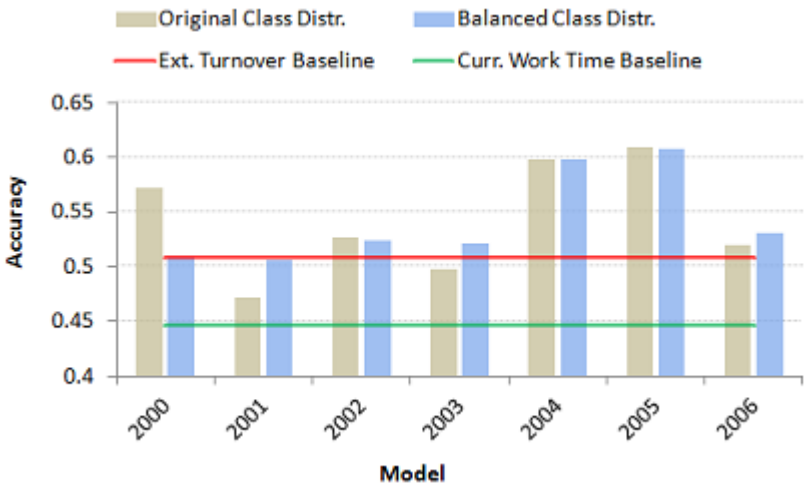



Figure 5.7: Average accuracy results of the models for future window of 3 years

Compared to small future window parameters, accuracies and accuracy differences to current work time baseline increase with future window of 3 years. The best model considering accuracy is *2005* with accuracy of 60.77%. Baseline accuracies are almost the same as the ones for the previous future window of 2 years.

Whereas *2000* model has high precision value, its recall is very low (0.07) causing its F1 score to become below baselines. Therefore, *2000* model fails for predicting turnovers within 3 years. Except for *2000* model, other models are successful in terms of all evaluation metrics and always have results above baseline metrics.

Average of evaluation metrics of our results and difference values can be seen in Table 5.11. A large average recall (0.79), a large average F1 score (0.63) and also large differences are obtained for future window parameter of 3 years. Baseline metrics for future window of 3 are almost the same when compared to future window of 2.

According to F1 scores and accuracy values, there is no single successful model. For

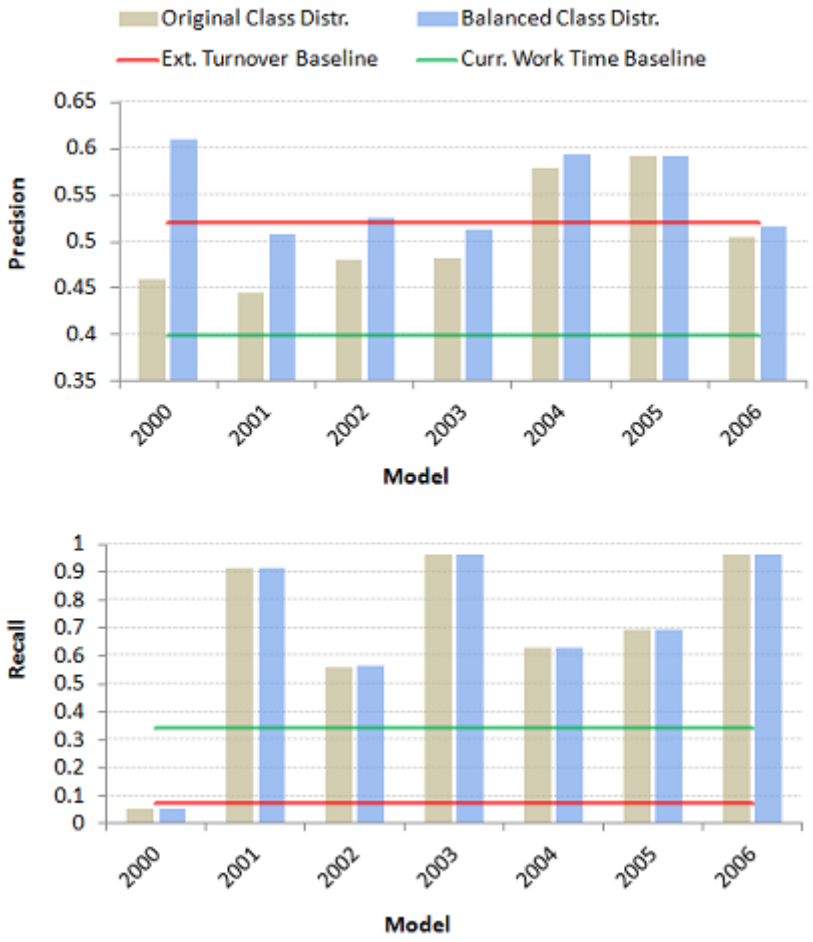



Figure 5.8: Average precision and recall results of the models for future window of 3 years

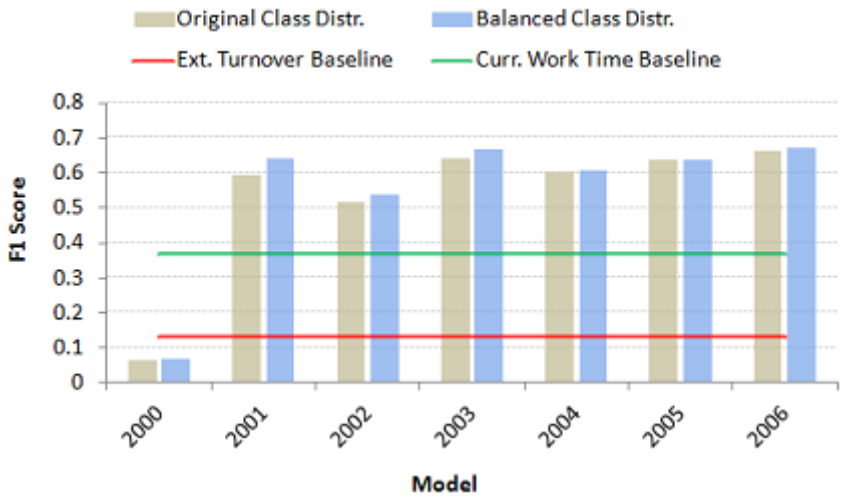



Figure 5.9: Average F1 scores of the models for future window of 3 years

Table 5.11: Average evaluation of all experiments conducted for future window of 3 years

| | Our Model | Ext Turnover Baseline | Diff | Curr Work Time Baseline | Diff |
|---|---|---|---|---|---|
| Accuracy | 0.55 | 0.51 | 0.04 | 0.45 | 0.10 |
| Precision | 0.54 | 0.52 | 0.02 | 0.40 | 0.14 |
| Recall | 0.79 | 0.07 | 0.71 | 0.34 | 0.44 |
| F1 Score | 0.63 | 0.13 | 0.50 | 0.37 | 0.26 |

F1 score, *2003* and *2006* models are the most successful ones for predicting 3 years later. With respect to accuracy, *2004* and *2005* models are the best. Success of the results of these models are on testing data having balanced class distribution.

### 5.4.4 Results for Future Window of 5 years

The largest turnover period is tested in experiments for future window of 5 years. The results in Figure 5.10, Figure 5.11 and Figure 5.12 indicate that future window of 5 experiments are the most successful ones compared to previous future window experiments.

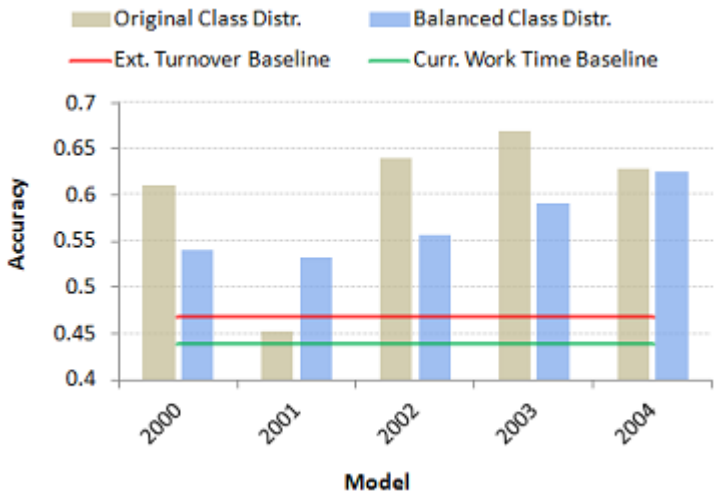



Figure 5.10: Average accuracy results of the models for future window of 5 years

Calculated accuracy values for future window of 5 years show us that tests on balanced data have higher accuracies than baseline accuracies for all models. The most successful model considering turnovers within 5 years is *2004* model with accuracy

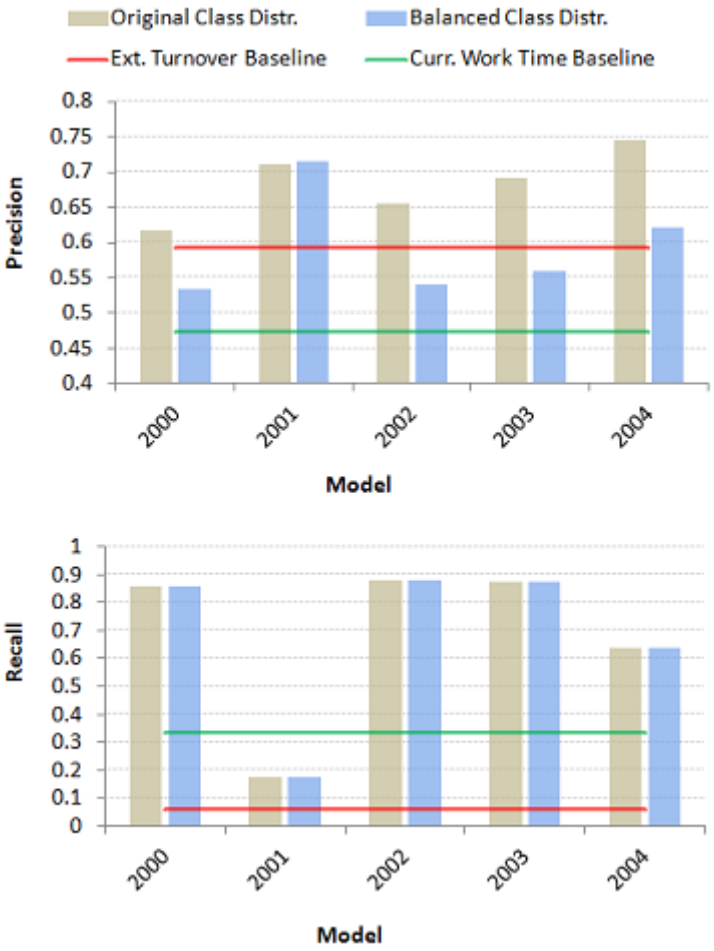



Figure 5.11: Average precision and recall results of the models for future window of 5 years

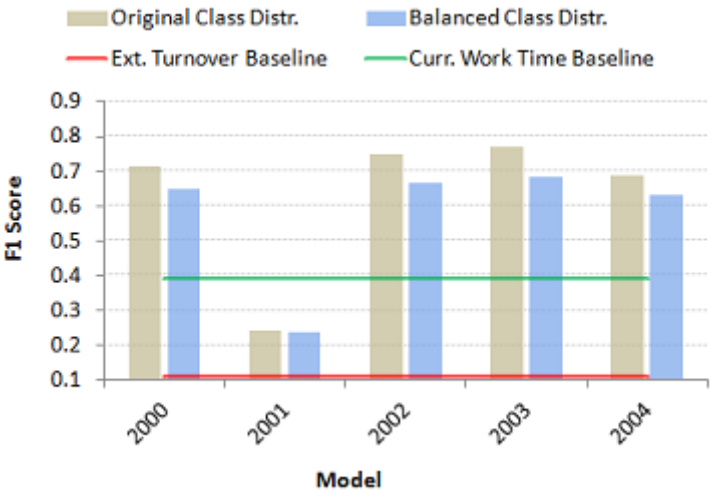



Figure 5.12: Average F1 scores of the models for future window of 5 years

of $62.5\%$. The results of original class distributed data experiments again fluctuate between model years but this time they have generally higher accuracies. Baseline accuracies drop a little for future window of $5$ compared to future window of $3$ experiments.

Precision and recall values of future window of $5$ experiments have higher values than the best baseline metrics (current work time baseline). While *2001* model has the largest precision value of $0.71$, its recall value is very low $(0.17)$. Thus, assigned F1 score for *2001* model is below the F1 score of current work time baseline. Apart from *2001* model, all other models are considered successful in terms of precision, recall and F1 score.

External turnover baseline's F1 score is very low $(0.11)$ meaning that this baseline for future window of $5$ become worse for predicting turnovers than previous future window experiments.

The average results of all evaluations made for models in future window of $5$ experiments are calculated and shown in Table 5.12. According to this table, for future window of $5$ experiments, evaluation metrics of our models reach the largest values along with the largest difference values when compared to previous future window experiments. In terms of accuracy, recall and F1 score, considerable success is achieved by our models compared to baselines.

Table 5.12: Average evaluation of all experiments conducted for future window of $5$ years

| | Our Model | Ext Turnover Baseline | Diff | Curr Work Time Baseline | Diff |
|---|---|---|---|---|---|
| Accuracy | 0.58 | 0.47 | 0.11 | 0.44 | 0.14 |
| Precision | 0.56 | 0.59 | -0.03 | 0.47 | 0.09 |
| Recall | 0.81 | 0.06 | 0.75 | 0.33 | 0.48 |
| F1 Score | 0.66 | 0.11 | 0.55 | 0.39 | 0.27 |

Although other metrics except precision remain almost the same or dropped a little for baseline models, precision values of the baseline models increase compared to future window of $3$ years. Therefore, increased precision value of our model is about the same as the precision values of baseline models. Hence, there is no enough

contribution of our model in terms of precision with future window of 5 years.

In terms of accuracy and F1 score on balanced test data for future window of 5, the most successful models are *2003* and *2004*. *2001* model can be considered as unsuccessful for predicting turnovers within 5 years.

### 5.4.5 General Evaluation of Future Window Experiments

The experiments are conducted for different future window parameters 1, 2, 3 and 5, and results are presented in the previous subsections. To sum up all future window experiments, evaluation metrics of all experiments performed within each future window are averaged out and illustrated in Table 5.13.

According to this table, average accuracy value of our models is 55% which is not so high value but compared to baselines, it is considerably successful. The largest difference between our models and baseline models exists in recall values and F1 scores. This means that with high average recall value of 0.75, our models are good at predicting actual turnovers (1 classes) correctly. Average precision value of our models is 0.54 which is average indeed, but it is still higher than the baseline precision values. This shows us that our models have average success for predicting "no-movements" of the employees (0 classes).

Table 5.13: Average evaluation of all experiments conducted for all future window of 1, 2, 3 and 5 years

| | Our Model | Ext Turnover Baseline | Diff | Curr Work Time Baseline | Diff |
|---|---|---|---|---|---|
| Accuracy | 0.55 | 0.50 | 0.05 | 0.45 | 0.10 |
| Precision | 0.54 | 0.52 | 0.02 | 0.41 | 0.13 |
| Recall | 0.75 | 0.14 | 0.61 | 0.34 | 0.41 |
| F1 Score | 0.60 | 0.16 | 0.43 | 0.36 | 0.24 |

Apart from these, the most successful future window parameter is 5 with the highest scores above the general average values. And then future window of 3 and 1 come. Future window of 2 experiments are not as successful as the other ones, but there is still contribution over baselines. These results indicate that the success for predicting

the turnovers increase with the larger period.

The results of experiments performed so far show that tests on data having balanced class distribution gives more robust and reliable results than tests on data having its original class distribution. This result can be interpreted as "normal" due to the training of the models on balanced data also.

According to results, the most reliable and strong baseline model is the one that predicts turnovers in terms of employee's current work times. External turnover baseline model, which says "turnover" for the employees having at least one external turnover in past, does not reveal so successful results.

In the following section, evaluation of the results continue with the comparison of the results from different perspectives.

### 5.4.6 Other Results

The results presented so far are the experimental results of the Support Vector Machine (SVM) generated models. Among these results, the most successful results are obtained for future window of $5$ years (predicting turnover within $5$ years), and for *2003* and *2004* models. With these best models, we try different classifiers other than SVM and compare their results with SVM. One classifier is Decision Table/-Naive Bayes Hybrid Classifier (DTNB) which is the main classifier used in reference study [60]. The other one is Neural Network learning algorithm that makes use of neural networks of connected input/output units and weights associated with them (see Section 3.4 for details of classifiers).

*2003* and *2004* models are separately trained with SVM, DTNB and Neural Network algorithms for future window of $5$. The results of models from different classifiers are illustrated in Figure 5.13 for Accuracy and F1 score metrics. According to figure, for both models and for both metrics, models of Support Vector Machine algorithm give the most successful results. Since other two algorithms do not give promising results for the "best" SVM models, they are not chosen generally. Support Vector Machine is more suitable learning algorithm for our problem (see Section 3.4).

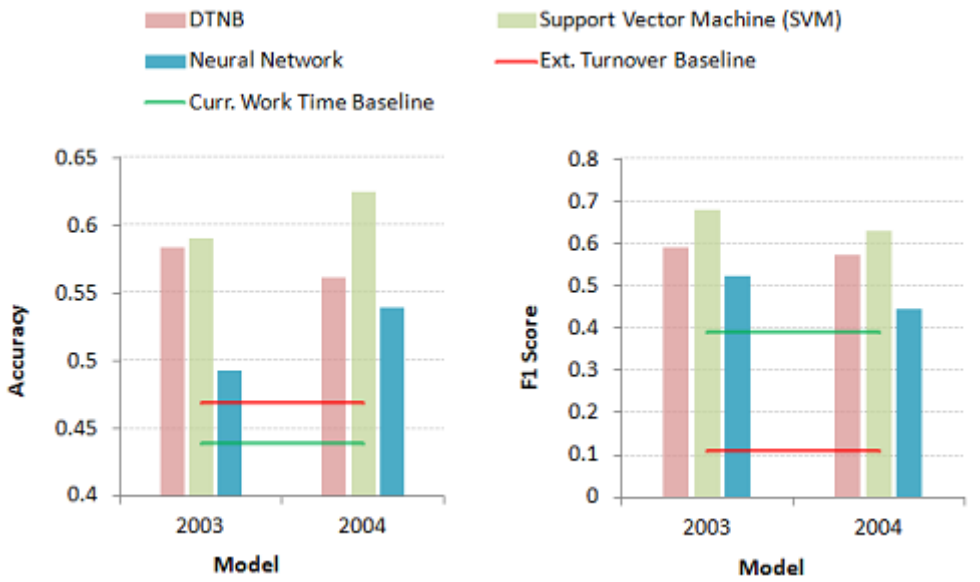



Figure 5.13: Comparative results of different classifier models for future window of 5 years

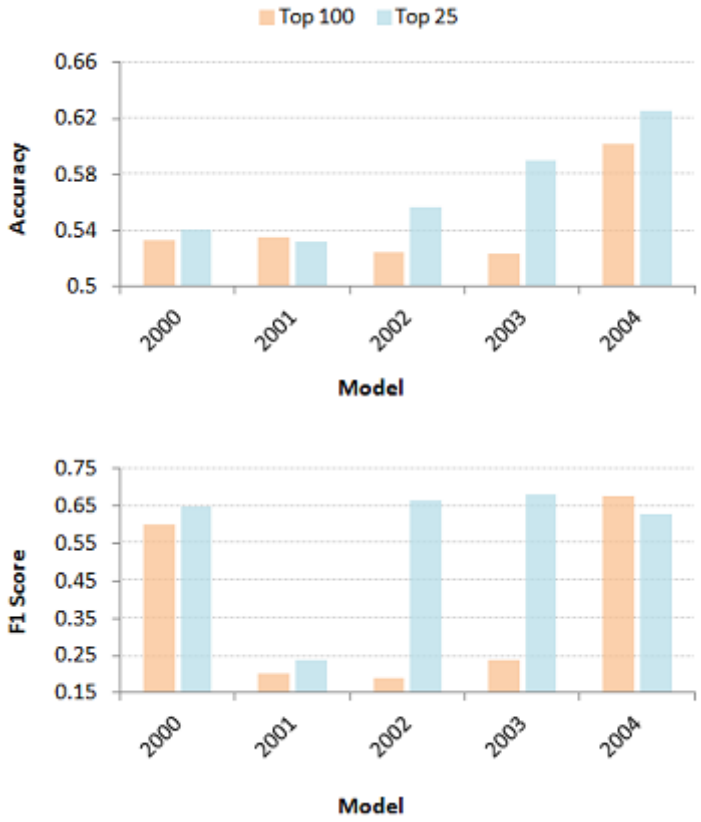



Figure 5.14: Comparative analysis on Top 100 vs Top 25 companies' employee profiles for future window of 5 years

During our study, turnover analysis on employee profiles from both top 100 companies and top 25 companies are studied. Since the analysis of top 25 company employees give more successful results, we present the results of only top 25 companies related study in the previous sections. In Figure 5.14, comparative results of top 100 and top 25 analysis are shown for the best future window parameter of 5 years.

According to Figure 5.14, top 25 analysis have higher accuracy values and F1 scores in general. Especially for F1 scores, *2002* and *2003* models, the difference between top 100 and top 25 analysis results can be seen clearly. Therefore, top 25 company employees analysis is chosen to present.

In our study, model aging can also be experimented. Each model is created with the given year's features and testable on different years. However, employee features may change by the year(s) and models may become old and unsuccessful to analyze the turnovers by using the changing features in later years. In Figure 5.15, this situation can be seen for *2002* model for predicting turnovers within period of 1 year. According to this figure, predictive capability of *2002* model decreases by the years towards present because accuracy and F1 score values drop each year, especially at *2006*. This situation may or may not occur, but it can be said that to get more robust and reliable results, model's testable years range shouldn't be kept so large.

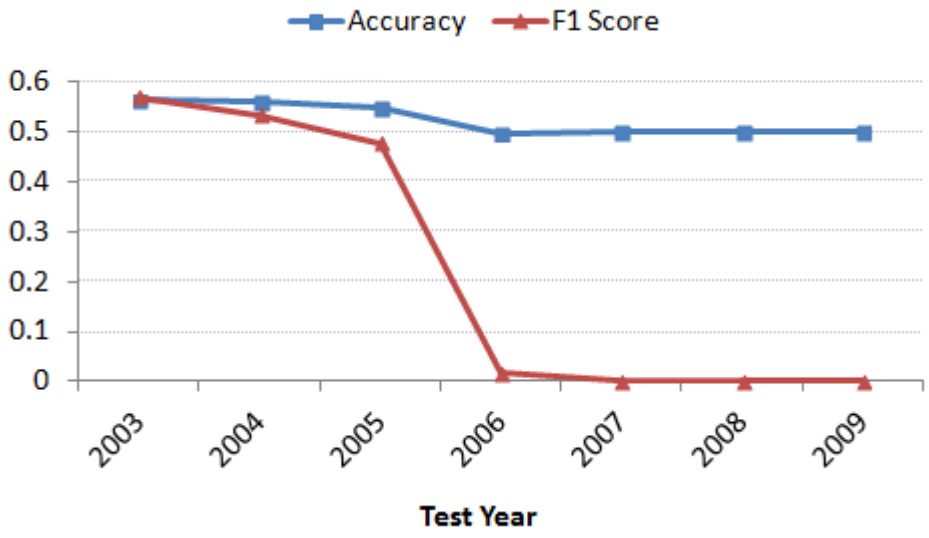



Figure 5.15: Model aging for *2002* model with respect to future window of 1 year

So far, we make analysis from models perspective. From test years perspective, many predictions are made on the same test years with different future window parameters and different models. In Figure 5.16, test years and results of experiments conducted at these years are illustrated. In terms of accuracy and F1 score, the most successful prediction is made on year *2001* within the period of 5 years.

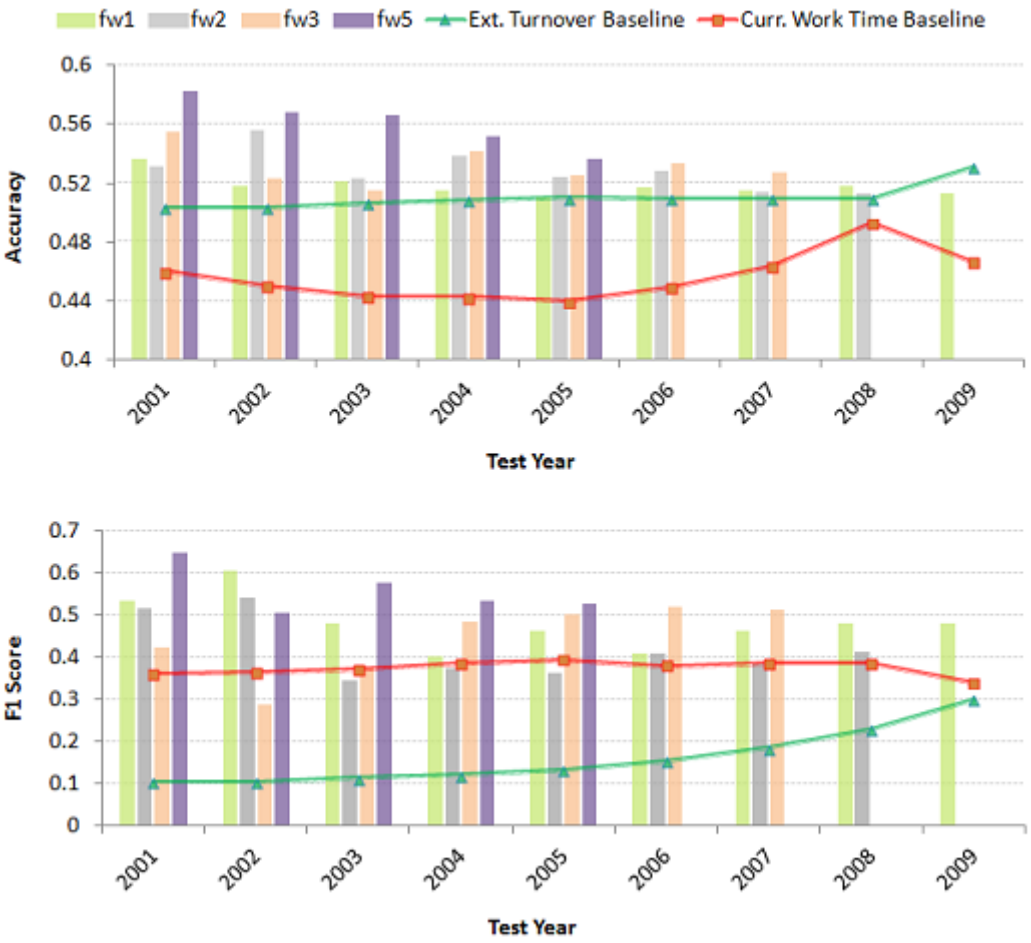



Figure 5.16: Experimental results from test years perspective

According to Figure 5.16, it may seem that with the larger future window experiment, the prediction on the test year become more successful. Although there are many exceptional cases for this statement, still future window of 5 experiments give more successful results than other future window experiments on the same test years. Furthermore, almost all models from different future windows make successful predictions on almost all test years considering baseline values for the given test years.

# CHAPTER 6

# CONCLUSION AND FUTURE WORK

## 6.1 Conclusion

In this thesis work, we focus on the employee turnover problem and try to predict whether the employee will leave his/her current company within the specific time period. We formulate it as a binary classification problem which classifies the employees as who involve in "turnover" and who continue to work for current company ("no-movement"). We apply supervised machine learning algorithms for this classification task. Experimental results are evaluated from different perspectives and compared to different baseline models. The initial results show that our proposed models are considerably successful compared to the baseline models. The main contributions of our approach are the use of machine learning methods and also exploiting the employee job transitions for the solution of the problem.

In summary, this study is mainly composed of five phases. In data preprocessing phase, publicly available employee profiles taken from the Web are analyzed and data preprocessing operations are applied. Job transition graphs construction phase contains the split of the original dataset into years and job transition graphs. Job transitions for each employee are constructed in terms of years from *2000* to *2010* using the algorithm in [60] and stored in the corresponding job transition graphs. Feature vector for each employee is generated in feature extraction phase. Company and employee features of each year are calculated from both relational data and job transition graphs. For companies (also nodes in the graphs), several centrality values are computed for each year graph. After the feature selection phase, in classification phase

different models of different years are trained with mainly Support Vector Machine (SVM) algorithm. Experiments of each model are conducted considering test year and future window parameter that defines the time period of turnover. Experimental results are evaluated with accuracy, precision, recall and F1 score metrics.

In this study, three different baseline models are assumed; current work time baseline, experience baseline and external turnover baseline. Each baseline model predicts the turnover according to a specific feature definition and calculated baseline value. Among baseline models, current work time and experience baselines give almost the same results. While external turnover baseline do not reveal so successful results, the current work time baseline is observed as the most reliable and strong baseline model.

According to the evaluations of SVM generated models, average accuracy value of our all models is $55\%$. Compared to baseline models, it can be seen as the success although the accuracy value is not so high.

Considering strong baselines with low evaluation metrics, this situation shows us the difficulty of the studied problem. In real world, there may be many reasons, especially personal ones, that affect the employees to leave their companies. However, we cannot capture them with only "limited" available information in our dataset.

The largest difference between our models and baseline models occurs in recall values and F1 scores. With high average recall value of $0.75$, our models are good at predicting actual turnovers correctly. Average precision value of our models $0.54$ is indeed average but it is still higher than the baseline precision values. This is an indicator of average success of our models for predicting "no-movements" of the employees.

Among the future window experiments, the larger period of $5$ years become more successful than other future window parameters of $1$, $2$ and $3$ years. Expanding the future window period of turnover increases the success of turnover prediction.

To get comparative results, Decision Table/Naive Bayes Hybrid Classifier (DTNB) and Neural Network algorithms are applied to the best model years obtained from SVM models. The results of comparison show that for our problem SVM is more suitable classification algorithm than others.

In our study, training data having balanced class distributions are used for generating models. Tests on balanced class distributed test data become more successful than original class distributed tests.

Our results in this study indicate that employee's current company features are more important in predicting turnovers than past company features. After the feature selection phase, current company features are ranked high as relevant attributes, whereas mostly past company features are not included in feature vectors.

## 6.2 Future Work

In this thesis, we focus on only "external" turnovers of the employees. "Internal" turnovers (position changes) can also be studied.

In this study, only the employees working at top companies from different industries are analyzed. However, each industry may have a different turnover pattern. As a future work, the problem can be specified as the analysis of the employees from a certain industry or certain company in terms of turnover.

Middle or small scaled companies and their employees can be added to this study to understand their effects on turnover problem.

We use publicly available employee and company information to extract the features and train our models. With new features, even detailed features, our models can be improved. The influence of social connections of the employees for turnover problem can be studied. Additionally, the use of detailed employee features like age, gender, work environment related features would most probably affect the success of the models. The effect or no effect of past company features on employee turnovers can also be studied in detail.

In this work, we study on turnover analysis as a binary classification problem. For the employees who are known as involved in a "turnover" given certain time period, regression problem can also be defined as the prediction of how many months after the given employee will leave his/her current company.

Additionally, we analyze the turnovers from employee perspective. From company perspective, the problem can be stated as prediction of company turnover rate by considering employee turnovers and can be studied as a regression problem.

# REFERENCES


[1] Churn rate. http://en.wikipedia.org/wiki/Churn_rate. Last visited on: June 2014.

[2] Early 2000s recession. http://en.wikipedia.org/wiki/Early_2000s_recession. Last visited on: August 2014.

[3] Feature extraction. http://en.wikipedia.org/wiki/Feature_extraction. Last visited on: July 2014.

[4] Historical stock prices. http://finance.yahoo.com/. Last visited on: June 2014.

[5] Jaccard Index. http://en.wikipedia.org/wiki/Jaccard_index. Last visited on: Aug 2014.

[6] Neo4j: an open-source graph database. http://www.neo4j.org/. Last visited on: May 2013.

[7] Power iteration. http://en.wikipedia.org/wiki/Power_iteration. Last visited on: May 2014.

[8] Michael A Abelson. Examination of avoidable and unavoidable turnover. *Journal of Applied Psychology*, 72(3):382, 1987.

[9] Christian Belzil. An empirical model of job-to-job transition with self-selectivity. *Canadian Journal of Economics*, pages 536–551, 1993.

[10] Luigi Biggeri, Matilde Bini, and Leonardo Grilli. The transition from university to work: a multilevel approach to the analysis of the time to obtain the first job. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 164(2):293–305, 2001.

[11] Paolo Boldi and Sebastiano Vigna. Axioms for centrality. *Internet Mathematics*, (just-accepted):00–00, 2014.

[12] Phillip Bonacich. Power and centrality: A family of measures. *American journal of sociology*, pages 1170–1182, 1987.

[13] Stephen P Borgatti. Centrality and network flow. *Social networks*, 27(1):55–71, 2005.

[14] Stephen P Borgatti and Martin G Everett. A graph-theoretic perspective on centrality. *Social networks*, 28(4):466–484, 2006.

[15] Stephen P Borgatti, Ajay Mehra, Daniel J Brass, and Giuseppe Labianca. Network analysis in the social sciences. *science*, 323(5916):892–895, 2009.

[16] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[17] Ed Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3):186–198, 2009.

[18] Antoni Calvó-Armengol and Yves Zenou. Job matching, social network and word-of-mouth communication. *Journal of urban economics*, 57(3):500–522, 2005.

[19] Shawn M Carraher. Turnover prediction using attitudes towards benefits, pay, and pay satisfaction among employees and entrepreneurs in estonia, latvia, and lithuania. *Baltic Journal of Management*, 6(1):25–52, 2011.

[20] Peter J Carrington, John Scott, and Stanley Wasserman. *Models and methods in social network analysis*, volume 28. Cambridge university press, 2005.

[21] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[22] Yi-Wei Chen and Chih-Jen Lin. Combining svms with various feature selection strategies. In *Feature extraction*, pages 315–324. Springer, 2006.

[23] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[24] John L Cotton and Jeffrey M Tuttle. Employee turnover: A meta-analysis and review with implications for research. *Academy of management Review*, 11(1):55–70, 1986.

[25] Kristof Coussement and Dirk Van den Poel. Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques. *Expert systems with applications*, 34(1):313–327, 2008.

[26] Elizabeth M Daly and Mads Haahr. Social network analysis for routing in disconnected delay-tolerant manets. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 32–40. ACM, 2007.

[27] Andrés Erosa, Luisa Fuster, and Diego Restuccia. Fertility decisions and gender differences in labor turnover, employment, and wages. *Review of Economic Dynamics*, 5(4):856–891, 2002.

[28] Daniel C Feldman and Thomas WH Ng. Careers: Mobility, embeddedness, and success. *Journal of Management*, 33(3):350–377, 2007.

[29] Monica L Forret and Thomas W Dougherty. Networking behaviors and career outcomes: differences for men and women? *Journal of Organizational Behavior*, 25(3):419–437, 2004.

[30] Linton C Freeman. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239, 1979.

[31] Mark Hall and Eibe Frank. Combining naive bayes and decision tables. In *Proceedings of the 21st Florida Artificial Intelligence Society Conference (FLAIRS)*, pages 318–319. AAAI press, 2008.

[32] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[33] Mark C Healy, Michael Lehman, and Michael A McDaniel. Age and voluntary turnover: A quantitative review. *Personnel Psychology*, 48(2):335–345, 1995.

[34] Herbert Gerhard Heneman, Tim Judge, and John D Kammeyer-Mueller. *Staffing organizations*. Mendota House, 2003.

[35] Monica C Higgins. Changing careers: The effects of social context. *Journal of Organizational Behavior*, 22(6):595–618, 2001.

[36] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. A practical guide to support vector classification, 2003.

[37] Herminia Ibarra. Network centrality, power, and innovation involvement: Determinants of technical and administrative roles. *Academy of Management Journal*, 36(3):471–501, 1993.

[38] Herminia Ibarra and PH Deshpande. Networks and identities: Reciprocal influences on career processes and outcomes. *Handbook of career studies*, pages 268–82, 2007.

[39] Elizabeth M Ineson, Eszter Benke, and József László. Employee loyalty in hungarian hotels. *International Journal of Hospitality Management*, 32:31–39, 2013.

[40] Robert W Irving. Matching medical students to pairs of hospitals: a new variation on a well-known theme. In *Algorithms—ESA'98*, pages 381–392. Springer, 1998.

[41] Marjorie Laura Kane-Sellers. *Predictive models of employee voluntary turnover in a North American professional sales force using data-mining analysis*. PhD thesis, Texas A&M University, 2007.

[42] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

[43] S Sathiya Keerthi and Chih-Jen Lin. Asymptotic behaviors of support vector machines with gaussian kernel. *Neural computation*, 15(7):1667–1689, 2003.

[44] Allen I Kraut. Predicting turnover of employees from measured job attitudes. *Organizational Behavior and Human Performance*, 13(2):233–243, 1975.

[45] Janina C Latack and Janelle B Dozier. After the ax falls: Job loss as a career transition. *Academy of Management Review*, 11(2):375–392, 1986.

[46] Victor Lavrenko. Web search 7: sink nodes in pagerank. `http://www.youtube.com/watch?v=_Wc9OkMKS3g`, January 2014. Last visited on: June 2014.

[47] Hsuan-Tien Lin and Chih-Jen Lin. A study on sigmoid kernels for svm and the training of non-psd kernels by smo-type methods. *submitted to Neural Computation*, pages 1–32, 2003.

[48] Nan Lin. *Foundations of social research*. McGraw-Hill New York, 1976.

[49] Meryl Reis Louis. Career transitions: Varieties and commonalities. *Academy of management review*, 5(3):329–340, 1980.

[50] Jochen Malinowski, Tobias Keim, Oliver Wendt, and Tim Weitzel. Matching people and jobs: A bilateral recommendation approach. In *System Sciences, 2006. HICSS'06. Proceedings of the 39th Annual Hawaii International Conference on*, volume 6, pages 137c–137c. IEEE, 2006.

[51] Douglas C Maynard, Todd Allen Joseph, and Amanda M Maynard. Underemployment, job attitudes, and turnover intentions. *Journal of Organizational Behavior*, 27(4):509–536, 2006.

[52] Jean Marie McGloin and David S Kirk. Social network analysis. In *Handbook of quantitative criminology*, pages 209–224. Springer, 2010.

[53] Jeffrey Mello. *Strategic human resource management*. Cengage Learning, 2014.

[54] Michael C Mozer, Richard Wolniewicz, David B Grimes, Eric Johnson, and Howard Kaushansky. Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry. *Neural Networks, IEEE Transactions on*, 11(3):690–696, 2000.

[55] Thomas WH Ng, Kelly L Sorensen, Lillian T Eby, and Daniel C Feldman. Determinants of job mobility: A theoretical integration and extension. *Journal of Occupational and Organizational Psychology*, 80(3):363–386, 2007.

[56] Nigel Nicholson and Michael West. *Managerial job change: Men and women in transition*. Cambridge University Press, 1988.

[57] Richard J Oentaryo, Ee-Peng Lim, David Lo, Feida Zhu, and Philips K Prasetyo. Collective churn prediction in social network. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, pages 210–214. IEEE Computer Society, 2012.

[58] Evelien Otte and Ronald Rousseau. Social network analysis: a powerful strategy, also for the information sciences. *Journal of information Science*, 28(6):441–453, 2002.

[59] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. 1999.

[60] Ioannis Paparrizos, B Barla Cambazoglu, and Aristides Gionis. Machine learned job recommendation. In *Proceedings of the fifth ACM Conference on Recommender Systems*, pages 325–328. ACM, 2011.

[61] Parag C Pendharkar. Genetic algorithm based neural network approaches for predicting churn in cellular wireless network services. *Expert Systems with Applications*, 36(3):6714–6720, 2009.

[62] John Platt et al. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998.

[63] Geoff Plimmer and Alison Schmidt. Possible selves and career transition: It's who you want to be, not what you want to do. *New Directions for Adult and Continuing Education*, 2007(114):61–74, 2007.

[64] Anne Beeson Royalty. Job-to-job and job-to-nonemployment turnover by gender and education level. *Journal of Labor Economics*, 16(2):392–433, 1998.

[65] Stina Sellgren, Goran Ekvall, and Goran Tomson. Nursing staff turnover: does leadership matter? *Leadership in Health Services*, 20(3):169–183, 2007.

[66] Robert Shimer. The cyclicality of hires, separations, and job-to-job transitions. *Federal Reserve Bank of St. Louis Review*, 87(July/August 2005), 2005.

[67] Alfonso Sousa-Poza and Andrés A Sousa-Poza. The effect of job satisfaction on labor turnover by gender: An analysis for switzerland. *The Journal of Socio-Economics*, 36(6):895–913, 2007.

[68] Toon W Taris, Inge A Bok, and Denise G Caljé. On the relation between job characteristics and depression: a longitudinal study. *International Journal of Stress Management*, 5(3):157–167, 1998.

[69] James R Terborg and Thomas W Lee. A predictive study of organizational turnover rates. *Academy of Management Journal*, 27(4):793–810, 1984.

[70] Noel M Tichy, Michael L Tushman, and Charles Fombrun. Social network analysis for organizations. *Academy of management review*, 4(4):507–519, 1979.

[71] Robert H Topel and Michael P Ward. Job mobility and the careers of young men, 1988.

[72] Stanley Wasserman. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.

[73] Tina Wey, Daniel T Blumstein, Weiwei Shen, and Ferenc Jordán. Social network analysis of animal behaviour: a promising tool for the study of sociality. *Animal Behaviour*, 75(2):333–344, 2008.

[74] Paul Wylleman, Dorothee Alfermann, and David Lavallee. Career transitions in sport: European perspectives. *Psychology of sport and exercise*, 5(1):7–20, 2004.

[75] Guo-en Xia and Wei-dong Jin. Model of customer churn prediction on support vector machine. *Systems Engineering-Theory & Practice*, 28(1):71–77, 2008.

[76] Yu Zhao, Bing Li, Xiu Li, Wenhuang Liu, and Shouju Ren. Customer churn prediction using improved one-class support vector machine. In *Advanced data mining and applications*, pages 300–306. Springer, 2005.

# APPENDIX A

# APPENDIX

## A.1 Features

Table A.1: Feature Set After Feature Selection

| |
|---|
| minCurrWorkTime |
| numPastCompany |
| experience |
| empExtTurnover |
| empIntTurnover |
| empIndustryChg |
| curr-eigenCenDir |
| curr-pageRank-0.5 |
| curr-katzCenDir |
| curr-inDegCen |
| curr-outDegCen |
| curr-harmonicCenIn |
| curr-harmonicCenOut |
| curr-proximityCenIn |
| curr-proximityCenOut |
| curr-extChurnRate |
| curr-intChurnRate |
| curr-extChurnRateDiff |
| curr-intChurnRateDiff |
| curr-ratioOutLink |
| curr-ratioInLink |
| curr-ratioSelfLink |
| curr-stockChange |
| curr-stockChangeDiff |

Table A.2: Feature Set Before Feature Selection

| | | |
|---|---|---|
| minCurrWorkTime | past-eigenCenDir-min | past-harmonicCenIn-min |
| numPastCompany | past-eigenCenDir-max | past-harmonicCenIn-max |
| minPastWorkTime | past-eigenCenDir-avg | past-harmonicCenIn-avg |
| maxPastWorkTime | past-eigenCenUndir-min | past-harmonicCenOut-min |
| avgPastWorkTime | past-eigenCenUndir-max | past-harmonicCenOut-max |
| experience | past-eigenCenUndir-avg | past-harmonicCenOut-avg |
| empExtTurnover | past-pageRank-0.5-min | past-proximityCenIn-min |
| empIntTurnover | past-pageRank-0.5-max | past-proximityCenIn-max |
| empIndustryChg | past-pageRank-0.5-avg | past-proximityCenIn-avg |
| numJob | past-pageRank-0.85-min | past-proximityCenOut-min |
| numUniv | past-pageRank-0.85-max | past-proximityCenOut-max |
| curr-eigenCenDir | past-pageRank-0.85-avg | past-proximityCenOut-avg |
| curr-eigenCenUndir | past-katzCenDir-min | past-ratioOutLink-min |
| curr-pageRank-0.5 | past-katzCenDir-max | past-ratioOutLink-max |
| curr-pageRank-0.85 | past-katzCenDir-avg | past-ratioOutLink-avg |
| curr-katzCenDir | past-katzCenUndir-min | past-ratioInLink-min |
| curr-katzCenUndir | past-katzCenUndir-max | past-ratioInLink-max |
| curr-inDegCen | past-katzCenUndir-avg | past-ratioInLink-avg |
| curr-outDegCen | past-inDegCen-min | past-ratioSelfLink-min |
| curr-harmonicCenIn | past-inDegCen-max | past-ratioSelfLink-max |
| curr-harmonicCenOut | past-inDegCen-avg | past-ratioSelfLink-avg |
| curr-proximityCenIn | past-outDegCen-min | past-extChurnRateDiff-min |
| curr-proximityCenOut | past-outDegCen-max | past-extChurnRateDiff-max |
| curr-extChurnRate | past-outDegCen-avg | past-extChurnRateDiff-avg |
| curr-intChurnRate | past-extChurnRate-min | past-intChurnRateDiff-min |
| curr-extChurnRateDiff | past-extChurnRate-max | past-intChurnRateDiff-max |
| curr-intChurnRateDiff | past-extChurnRate-avg | past-intChurnRateDiff-avg |
| curr-ratioOutLink | past-intChurnRate-min | past-stockChange-min |
| curr-ratioInLink | past-intChurnRate-max | past-stockChange-max |
| curr-ratioSelfLink | past-intChurnRate-avg | past-stockChange-avg |
| curr-stockChange | | past-stockChangeDiff-min |
| curr-stockChangeDiff | | past-stockChangeDiff-max |
| | | past-stockChangeDiff-avg |

## A.2 Top Companies

Table A.3: Top 100 companies for year *2000*

| | | |
|---|---|---|
| ibm | eds | kpmg peat marwick |
| microsoft | procter & gamble | cisco |
| hewlett-packard | ericsson | computer associates |
| us army | arthur andersen | u.s. army |
| sun microsystems | andersen consulting | jpmorgan chase |
| us navy | self employed | deloitte |
| self-employed | merrill lynch | boeing |
| united states air force | american express | atos origin |
| accenture | dell | csc |
| united states marine corps | kpmg | capgemini |
| motorola | ford motor company | ge capital |
| cisco systems | freelance | cap gemini ernst & young |
| at&t | mci | honeywell |
| united states navy | lockheed martin | qwest communications |
| nortel networks | siemens | compuware |
| ibm global services | oracle corporation | computer sciences corp. |
| digital equipment corp. | deloitte consulting | alcatel |
| oracle | xerox | deloitte & touche |
| pricewaterhousecoopers | general motors | deutsche bank |
| usaf | compaq | general dynamics |
| lucent technologies | electronic data systems | raytheon |
| ernst & young | hewlett packard | goldman sachs |
| us air force | united states army | eastman kodak |
| unisys | hp | siebel systems |
| nortel | citibank | u.s. navy |
| general electric | morgan stanley | dupont |
| coopers & lybrand | nokia | emc |
| price waterhouse | usmc | booz allen hamilton |
| sprint | intel corporation | abn amro |
| intel | ncr | 3com |
| fidelity investments | microsoft corporation | alcatel-lucent |
| bank of america | citigroup | ups |
| texas instruments | verizon | pfizer |
| | | wells fargo |

Table A.4: Top 100 companies for year *2009*

| | | |
|---|---|---|
| ibm | siemens | cisco |
| microsoft | fidelity investments | lockheed martin |
| us army | general electric | alcatel-lucent |
| freelance | citigroup | usaf |
| accenture | american express | texas instruments |
| self-employed | nokia | vodafone |
| hewlett-packard | wells fargo | symantec |
| self employed | procter & gamble | sap |
| pricewaterhousecoopers | self employed | saic |
| bank of america | csc | general motors |
| ernst & young | ford motor company | pfizer |
| us navy | intel | nortel networks |
| freelance | emc | mtv networks |
| oracle | lucent technologies | electronic data systems |
| at&t | atos origin | wipro technologies |
| google | sprint | ups |
| ibm global services | thomson reuters | microsoft corporation |
| motorola | citibank | apple |
| dell | deloitte consulting | computer sciences corp. |
| united states air force | target | amazon.com |
| sun microsystems | keller williams realty | verizon |
| cisco systems | hp | electronic arts |
| united states marine corps | xerox | jp morgan chase |
| capgemini | ing | oracle corporation |
| eds | t-mobile | hewlett packard |
| deloitte | arthur andersen | honeywell |
| kpmg | best buy | kaiser permanente |
| merrill lynch | tata consultancy services | nokia siemens networks |
| yahoo! | hsbc | bearingpoint |
| jpmorgan chase | johnson & johnson | mci |
| united states navy | digital equipment corp. | starbucks coffee co. |
| ericsson | logica | independent consultant |
| unisys | morgan stanley | ge healthcare |
| | | booz allen hamilton |