Universiteit Leiden
Opleiding Informatica

Mining a scientific conference

Name: S. Kanhai

Date: 27/02/2015

1st supervisor: Dr. S.G.R. Nijssen
2nd supervisor: Dr. H. Blockeel

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Contents

Chapter 1

Abstract

In this thesis we apply data mining to data from *ECMLPKDD 2013*, a scientific conference in machine learning and data mining. We will evaluate whether it is possible to predict which papers are accepted into the conference or which average score a paper will be given by its reviewers. To mine our data we use knowledge about a paper that does not involve its text, such as the number of authors. Our results with the data mining toolkit WEKA do not show improvements over the results of a baseline method that performs majority class predictions. We investigate the use of autoWEKA, an extension of WEKA to optimize the parameters of learning algorithms automatically. The experiments with this toolkit also do not show any improvement over the baseline. Consequently, we did not find indications that it can be predicted whether a paper is going to be accepted in *ECMLPKDD 2013* or which average score its reviewers will give it.

Chapter 2

Introduction

Data mining is the process of extracting new, nontrivial knowledge from large volume of data. In this thesis we apply data mining to analyse data collected from *ECMLPKDD 2013*, a scientific conference. The *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD 2013)* [14] is a European machine learning and data mining conference held in Prague from September 23th to 27th, 2013. During a scientific conference its participants present, discuss and peer-review their research results as written down in papers. A subset from these papers is subsequently accepted and represents the (future) direction for the field center to the conference. Scientific conferences are interesting to study as they form the primary methods of scholarly interaction among researchers together with academic journals.

In this thesis we evaluate if it is possible to create a *predictive model* for *ECMLPKDD 2013* that predicts which what papers are accepted into the conference or at least what score the reviewers of a paper have given that paper. The goal of a **predictive model** is to predict the value for a distinctive attribute, *feature*, in our data. This is our **target feature**. A predictive model is created by running a learning algorithm over one or more feature(s). Here we gather our features from knowledge about a paper that does not involve its text. The knowledge regarding the conference is gathered from its conference management system. Examples of this knowledge include the number of authors of a paper and its primary subject area, main theme. In future references, these features are called the **general characteristics** of our paper. It is interesting to study these general characteristics, because they consist out of the essential information of a scientific conference and they can therefore be extracted from a wide variety of scientific conferences.

In Chapter III we give a description of the conference and related work to our project. We will then, in Chapter IV, discuss the structure and contents of the data. In Chapter V we describe the features we have extracted from our

data and here we also discuss the results of learning algorithms applied to these features. We will finally conclude this thesis in Chapter VI.

Chapter 3

Background

In this chapter we discuss the organizational structure of the conference by means of discussing the different groups of the participants in the conference, who from now on are referred to as *users*. We also discuss the functions and interrelationships of these users. Following this discussion we illustrate the chronological proceedings of the conference with a *use case* diagram for reviewing of a paper. In this chapter we also explore previous work done on our subject.

3.1 Conference

In *ECMLPKDD 2013* we recognize the following groups for the users: *PC chair, metareviewer, reviewer, submission owner, authors* and *bidder* where the group of metareviewers, reviewers and bidders are determined while reviewing a paper. The positions of PC chair, submission owner and author are predefined. For this project we do not investigate the PC Chairs, since we have gathered our data from one of the PC Chairs and are therefore limited to the data level access of this user.

The groups of users and their respective functions are:

PC chair

- assigns 1 metareviewer to each paper to supervise the review process of that paper.
- assigns 3 or 4 reviewers to review a paper. The assignment is based on, but not limited to, the bid of interest of a user on that paper.
- decides whether a paper is accepted into the conference. Here the PC Chair takes the score and reviews of the metareviewer and reviewers of a paper as advise for his decision to accept a paper into the conference.

- supervises the metareviewers

Metareviewer supervises the reviewers and review process of a paper. He also determines what the collective final score of the reviewers of a paper will be in conjunction with these reviewers. A paper can get one of four final scores listed left to right from the best to the worst score: *strong accept, weak accept, weak reject* and *strong reject*.

Reviewer reviews the paper they are assigned to and individually scores this paper with one of the scores as described for the metareviewer.

Bidder evaluates the title, abstract and author(s) from a paper and decides his level of interest to review that paper. He expresses his interest in reviewing this paper by bidding on the paper. The possible levels of willingness to review a paper are: *eager, willing, in-a-pinch* or *not willing* where a user with an *eagar* bid is the most willing to review the paper. For our project we consider a user who bids *not willing* to review that paper to less interested in reviewing this paper in comparison to a user who has abstained from bidding on that paper.

Submission owner submits the paper he has written into the conference.

Author has written one or more paper(s) sent into the conference, but is not necessarily the submission owner of one or more of these papers.

With a **use case** diagram we will discuss the process of a paper sent into the conference, reviewed and then accepted or rejected into the conference. In this use case we first give a short description of this process. We follow this description by denoting its primary and secondary actor(s). A primary actor is the primarily responsible party for starting of the process and a secondary actor has an assisting role in the process. Subsequently, we list the preconditions, main flow and post-conditions of this process in the use case diagram. Finally, any alternatives for the actions described are also denoted in the use case. Our use case diagram is as follows:

Use Case: Review a paper
ID: 1
Description: A paper is sent into the conference, reviewed and then accepted or rejected.
Primary actor: Submission owner
Secondary Actor: Reviewers, metareviewers and PC Chairs
Preconditions: None
Main flow: • The submission owner sends the paper into the conference. • The metareviewers and reviewers evaluate the abstract, title and author(s) of the paper. • The metareviewers and reviewers bid on the paper, automatically making them a bidder on the paper. • The PC chair assign the paper to 3 or 4 users to review . • The PC chair assign the paper to 1 metareviewer to coordinate the review process of the paper. • The reviewers review the paper. • The metareviewer in conjunction with the reviewers determine the final score of the paper. • The PC chair accept the paper. [1]
Postcondition: The paper has been reviewed.
Alternative flow: [1] The PC chair reject the paper .

Table 3.1: In this use case diagram we list the complete process from a paper being sent into the conference to it being accepted into or rejected from the conference.

3.2 Related work

In [20] 410 reviews from Epinions ranging from reviews of automobiles, banks, movies, and travel destinations are evaluated to identify emotions in mostly unstructured text. This is often called *sentiment analysis*. Here predictions are made by determining the *semantic orientation* of the phrases in the adjectives or adverbs from a review with the goal to classify a review of that particular product or service as recommends or not recommends it, binary classification. The simple unsupervised learning algorithm in this paper has proven an average accuracy of 74% with an accuracy of 84% for automobile reviews and an accuracy of 66% for movie reviews. Equivalent research to predict the recommendation of a review by looking at its contents is done for electronics in [16], and for the reviews of mobile devices in [15]. Similar recommendation classification has also been performed in [19]. However, here a review is not classified as recommending or not recommending, but on a five-star rating system. The methods described in these papers are non-optimally applicable to our project, since our reviews are structured. In *ECMLPKDD 2013* a reviewer needs to answer an questionnaire regarding the paper he has reviewed. Since this limits the use of adjectives or adverbs in our reviews the application of sentiment analysis to our data is not optimal.

In [21] a framework for analyzing and comparing opinions is proposed for products discussed in online customer reviews. With this framework the general characteristics of these products are extracted to visualize their strengths and weaknesses. These strength and weaknesses are subsequently used to enable visual side-by-side and feature-by-feature comparisons for these products and so aid consumers in their decision to buy that product. Here, the general characteristics are visualized to aid the consumer, while we use these features to perform classification to determine if a paper is accepted or rejected from the conference.

All-in-all, the contents of reviews have been used to perform classification [20], [16], [15] and [19] and general characteristics have been used to perform visual side-by-side and feature-by-feature comparisons [21], while there has not been any research that considers to perform classification, or even regression, with the general characteristics of product or service. In the thesis we will consider if it is possible to create a predictive model that predicts if a paper is accepted or rejected from a scientific conference. To the best of our knowledge, apart from our subject, any research with a scientific conference as its dataset also seems non-existent. Our main contributions are therefore our unique application of the use of general characteristics and the data we use for our thesis, the data from a scientific conference.

Chapter 4

Description of the data

A Microsoft conference management toolkit (CMT) was used to log all the data of the conference. This system enables simultaneous online availability of the data of the conference to all its users. To mine the data from the conference we needed to make it available locally. Since the CMT only allows us to extract the data from the conference over different files in different file types we decided to create a relational database from the data of the conference. We will use an SQLite3 relational database. The conversion of our data into a relational database makes our data set homogeneous and therefore is allows for easy manipulation and extraction of the data, especially when we consider the use of SQL queries [11]. It is also important to note that we will only use a subset of the total amount of data that is available for the PC Chair in the CMT.

In this chapter we will initially discuss the structure and contents of the data as we have extracted it from the CMT, then we discuss our steps of converting the extracted data into a relational database and finally we will give some technical specifications of our conversion process.

4.1 The data

The data extracted from the CMT is structured into several folders where each folder has one or more files containing actual data from the conference. The files in these folders were of the *.html*, *.xml* or *.xls* format. Each of these files is listed below with an example entry and an explanation of the contents from the file. For each attribute in these files which is not self-explanatory or previously explained we have also added an explanation:

- *Bids.xml* lists for every user in the conference the papers they bid on and what they bid. Based on these bids users are assigned by the PC Chairs to the papers they need to review.

```
<reviewer firstName="John" lastName="Doe" email="j.doe@umail
    .leidenuniv.nl" organization="Leiden University">
   <submission submissionId="1" title="Data mining a
       conference" track="ECMLPKDD2013 Main"
       primarySubjectArea="Data mining" bid="2 - Willing"
       bidValue="2" />
```

The attributes are:

- *submissionId*: an unique identification code that is assigned to a paper when it is uploaded to the CMT of the conference;
- *track*: within the *ECMLPKDD* conference papers could be uploaded to a *journal* track, *proceedings* track, *industrial* track and *nectar* track. We will only consider the papers from the *proceedings* track which are denoted as *ECMLPKDD2013 Main* in the CMT;
- *primarySubjectArea*: from the 45 predefined subject areas the submission owner has chosen this subject area to be the most relevant to this paper;
- *bid*: the numerical value of the bid of a metareviewer or reviewer on the concerning paper combined with a textual description of this paper;
- *bidValue*: only the numerical value of the bid a user has made on the concerning paper.

- *Assignment By Paper.xml* has different versions for the metareviewer and the reviewer who were assigned to a paper in these files. Each of the files has their own structure:

  For the metareviewer we list for every paper in the conference the metareviewer that is assigned to that paper.

  ```
  <submission submissionId="2"><metareviewer email="p.
      doe@umail.leidenuniv.nl" />
  ```

  For the reviewer we lists for every paper in the conference the 3 or 4 reviewers that are assigned to review that paper.

  ```
  <submission submissionId="3">
    <reviewer email="Jane.doe@umail.leidenuniv.nl" />
    <reviewer email="puck.vd.petteflat@gmail.com" />
    <reviewer email="jantje_berentse@outlook.com" />
  </submission>
  ```

- *Reviewer Subject Areas.xls* is an excel sheet with two worksheets, one for the *subject area distribution* and one for the *subject area selection*. For both the metareviewers and reviewers there is a separate excel file. Since the structure of both files are the same we only give one example entry and we will do so for a metareviewer:

– *Subject Area Distribution* lists for every subject area in the conference the number of times this subject area has been selected as the primary subject area for the total number of metareviewers in the conference. While it also lists the number of times this subject area has been selected as either the primary or secondary subject area for the total number of metareviewers in the conference.

Subject Area Name	Selected as Primary	Selected as Primary or Secondary
Active learning	0	5

– *Subject Area Selection* lists for every metareviewer in the conference some general details such as his email and affiliated organization as seen in the header of this excel sheet. This file also lists the subject areas this metareviewer has tagged to himself. For each of these subject areas he also needs to denote if he tagged this subject area as his primary subject area or as one of his secondary subject areas. A metareviewer needs to tag exactly one primary subject area to himself and at least one secondary subject area. Each of the subject areas listed to our metareviewer has a separate entry in the Excel sheet.

First name	Last Name	Email	Organization
Jan	Wu	j.wu@gmail.com	Leiden University

Selected Subject Area	Primary Or Secondary
Reinforcement Learning	Primary

- *Reviews, Discussions, Author Feedback and Meta-Reviews.html*, this file lists for every paper in the conference some general information from the paper and remarks about the paper from the metareviewer and the reviewers of the paper. We have chosen not to give any example data, since the data value for these attributes are trivial:

  – The section for the general characteristics of the paper:

    Paper ID, title, track name

    With the attributes:

    * *Paper ID*: this is exactly the same attribute as *submissionId* only with a different name;
    * *track name*: this is exactly the same attribute as *track* only with a different name.

  – The section for the metareviewer of the paper lists the final score the metareviewer and reviewers have decided for a paper. Here, this is called this score is called a *recommendation*. There are also arguments given for the final score and it is also possible to leave any comments to the author of the paper:

recommendation, arguments for recommendation, comments to the authors

l

– The section for the reviewers of the paper lists the answers to a questionnaire of eleven questions which embodies the review from a reviewer who has been assigned to review this paper:

summary of paper, contributions of paper, readability of paper, three strong point of paper, three weak points of paper, confidence in review, rough ranking of paper amongst other assigned papers, recommendations to area chair, confidential comments to the Area Chair, detailed comments justifying your evaluation of the paper

.

- *Papers.xls* lists general information about each paper in the conference such as its title and abstract. The contents for the general information of each paper is listed in the header of this table:

Paper ID	Title	Track Name
1	Data mining a conference	ECMLPKDD2013

Abstract	Author Names	Author Emails
N/A	John Doe*, LU	j.doe@umail.leidenuniv.nl

Subject Areas
Social Network Mining*; Graph and Tree Mining; Graphical Models

Conflict Reasons	Files
-	ECMLPKDD_Paper1_V2.pdf (271,571 bytes)

Supplementary File
-

With the attributes:

– *Conflict Reasons*: this attribute allows the authors of the paper to list any conflict they have with another member in the conference such as a student-mentor relationship between one of the authors of the paper and one of the members in the conference.

- *Paper Meta Data.html* lists the same data as *Papers.xls* only with an extra entry for submission questions.

Paper ID, Title, Track Name, Abstract, Author Names, Author Emails, Subject Areas, Conflict Reasons, Files, Supplementary File, Submission Questions

With the attributes:

– *Submission Questions*: these questions are asked to the authors of the paper by its reviewers.

- *View Paper Meta Data - accepted.html* is exactly the same as *Paper Meta Data.html* with the sole exception that we have only listed the papers which are accepted into the conference in this file. Due to limitations of the CMT we were only able to extract the id's of the papers which are accepted into the conference in such a manner.

- *time.html* lists for each paper in the conference the times a document has been uploaded for that particular paper.

```
<td>1</td><td>Data mining a conference</td><td>John</td><td>
    Doe</td><td>j.doe@umail.leidenuniv.nl</td><td>Leiden
    University</td><td>Author</td><td>7/1/2013 1:17:08 AM</td
    ><td>UPLOAD_FILE</td><td>1</td><td></td>
```

- *Users.xls* lists some general information of every user in the conference such as their first and last name, affiliated organization and also what positions a user has in the organization:

FirstName	MiddleInitial	LastName	Email
John		Doe	j.doe@umail.leidenuniv.nl

Organization	LastLoginDate	IsAuthor	IsAssociateChair
Leiden University	4/8/2013 5:45:01 PM	Yes	No

IsReviewer	IsExternalReviewer	IsMetaReviewer	IsSubmissionOwner
No	No	No	Yes

ConflictDomainsNotEnteredForSubmissionPapers	IsChair
Yes	No

IsProceedingsEditor
No

  With the attributes:

  - *IsX* checks for each of the positions listed on X if that particular user has that position in the conference$X$;
  - *ConflictDomainsNotEnteredForSubmissionPapers* is an attribute that is not clearly defined in the conference; we will therefore not use this attribute for our project.

Contents of the data

All-in-all, our data from the conference can be separated into several different categories. In the *bids* category we list for every user in the conference how interested they were in reviewing a paper by denoting their bid on the paper. These bids of interest are subsequently used by the PC Chairs in the conference to assign the users as metareviewers or reviewers to the papers in the conference.

For these assignments we recognize their own category, the *assignment by paper* category of data.

The *reviewer subject area* data category lists for every user in the conference the subject areas they have tagged to themselves as their primary subject area and the subject areas they have tagged to themselves as their secondary subject area. A user can only tag one subject area as his primary subject area, while he is unrestricted in the number of subject areas he tags to himself as a secondary subject area as long as he selects at least one subject area as his secondary subject area. This data category also includes a list of all the available subject areas in the conference where for each subject area we have listed the number of times this subject area has been tagged as a primary subject area and the number of times a subject area has been tagged as either a primary or secondary subject area by the users of the conference.

In the *reviews, discussions, author feedback and meta-reviews* data category we list the reviews of the reviewers of the conference together with the discussions between the metareviewer and reviewers regarding the final score on a paper. The review of a paper consist out of answering an eleven question long questionnaire about the paper.

We can also recognize the *paper* and *user* categories in the data. In the *paper* category we include the authors, title, primary and secondary subject areas tagged to the paper, and the upload times for the submission of the paper. The upload times concern the initial upload, the upload of a revision and the upload of a supplementary file for the submission. In this category it is also noted if a paper has been accepted into the conference. The *user* category contains all the general information of the users in a conference such as their name, email, affiliated organization and role(s) within the conference.

4.2 Relational database design

In the previous sections of this chapter we have described the structure and contents of each of the data files we have extracted from the CMT of the conference and we have also discussed what categories we recognize in our data. In this section we will discuss the relationships between the different categories by creating an *entity-relationship model*. An **entity-relationship model**, also called an *ER diagram*, is a visual data model that displays the relationships between the entities within the data. In this particular case the relationships between the entities are displayed for a single paper.

In our data we only recognize the *paper* and *user* categories as entities in our ER diagram. All the information in the *paper* and *user* categories are listed as attributes for their respective entities in the ER diagram. We list the positions of *submission owner, reviewer* and *metareviewer* in the conference as entities

which are a subset from the *user* entity. Since these positions are classified as a subset of the *user* entity they have the same characteristic as an *user* in the conference. Every other category we have recognized in the contents of our data is either a relationship between two entities in our ER diagram or the attribute of a relationship between the entities in our ER diagram. Due to practical reasons we have also decided to create an separate *subject area* entity. This is due to observation that both the papers and users in the conference each have a subject area tagged to them as a primary subject area and they are also tagged with one or more subject areas as their secondary subject area. The *subject area* entity is therefore connected via a one-to-one relationship, *selected as primary*,to the *paper* and *users* entities. While these entities are also connected as a one-to-many relationship, *tagged as secondary* for the subject areas which are connected as secondary subject areas to a paper or a user.

In the ER diagram in Appendix A we can also recognize a *bids_ on* relationship between the *user* and *paper* entities where the value of the bid of interest from the user is denoted in an *value* attribute from this relationship. The data for this relationship is gathered from the *bids* category. We have used the *assignment by paper* category in the same way to create the relationship *assigned_ to* between the *paper* entity and the *reviewer* and *metareviewer* entities. Here the relationship between the reviewer and a paper is denoted as many-to-one relationship and the relationship between the metareviewer and a paper as a one-to-one relationship. These entities are also connected via the contents from the *reviews, discussions, author feedback and meta-reviews* category. For the *review* relationship between a reviewer and a paper, a one-to-one relationship, we have listed each of the eleven questions as a single attribute of *review*. The same is done for the *review* relationship between the metareviewer and paper, again a one-to-one relationship, where each of the answers to the questions asked to the metareviewer about the paper are also listed as the attributes for the *review* relationship. The final score of a paper is part of the answers on these questions.

With the data extracted from the CMT of the conference mapped to a ER diagram we have created the following tables in our SQLite3 database. For each of these tables we have listed one example entry and a short description of the contents of the table. The tables which start with *R_* are intermediate tables that are created a practical point of view.

One of our intermediate tables is for example the *R_ bids_ missing_ when_ somebody_ omitted_ to_ look_ at_ paper* table which lists for all the users in the conference the paper they bid on and what their bid was.

email	paperid	bid_value
j.doe@gmail.com	52	2

However, since we consider a user more interested in reviewing a paper if he has abstained himself from bidding on this paper instead of bidding that he is *not willing* to review that particular paper we had to create a table which listed

all the available papers in the conference and set the default interest value of a reviewer for the papers to 1. We call that table the *bids_complete* table. By default, a user would then abstained himself from bidding on that paper, while the value of 0 of a bid on a paper represents the *not willing* bid of a user on a paper.

bids_complete lists for every metareviewer and reviewer all sent in papers in the conference. Here, we set the accompanying numerical value of a bid to the paper if the metareviewer or reviewer has bid on the paper. If a metareviewer or reviewer did not bid on a paper, we sets their bid on this paper to 1, the default value. We are required to do so, because we consider making a *not willing* (0) on a paper worse than abstaining (1) from bidding on that paper.

email	paperid	bid_value
j.doe@gmail.com	53	0

continents_per_paper lists the corresponding continents to every top-level domain we have gathered from the email addresses from the users in the conference. Here we have also created a fictitious continent *OP*, since email addresses with obne of these top-level domains: *.com, .org* and *.net*, are usually from a third-party email-provider, such as Gmail.

paperid	continent
1	EU

We had to create the following intermediary tables:

- *R_authors_per_paper* lists for every paper in the conference the email addresses of its author(s).

paperid	email
1	j.doe@umail.leidenuniv.nl

- *R_end_of_domains* lists all the available top-level domains of the email addresses of the users in the conference.

top-level
nl

- *R_countries_with_continent* lists for each of the top-level domains of the email addresses in the conference the corresponding continents.

country	continent
nl	EU

For each metareviewer in the conference we need to not the papers they are assigned to, to overview the reviews of these papers. While we also need to know what primary and secondary subject areas each of the metareviewers has

tagged to himself and what the distribution of subject areas was among the metareviewers.

- *metareviewer_assignments_by_paper* lists for every metareviewer in the conference the papers they are assigned to overview.

email_metareviewer	paperid
j.doe@umail.leideinuniv.com	52

- *metareviewer_subject_area_selections* lists for every metareviewer in the conference the subject area he has assigned to himself and whether he has assigned this subject area as a primary or secondary subject area to himself.

email_metareviewer	subject_area_as_primary
j.doe@umail.leidenuniv.nl	Reinforcement Learning

subject_area_as_primary_or_secondary
Primary

- *metareviewer_subject_area_distribution* lists for every subject area in the conference the number of times this subject area has been chosen as the primary subject area of a metareviewer and the number of times it has been chosen as either the primary or secondary subject area by a metareviewer.

subject_area	subject_area_as_primary
Active Learning	0.0

subject_area_as_primary_or_secondary
5.0

We have stored exactly the same data for the reviewers in the tables *reviewer_assignments_by_paper*, *reviewer_subject_area_selections* and *reviewer_subject_area_distribution*.

For each paper in the conference we also need to store the author(s) of that paper, its subject area, the last time a file for the submission of the paper has been uploaded, the final score its metareviewer and reviewers have decided to score it and finally whether or not the paper has been accepted into the conference.

- *papers_authors* lists for each paper who its authors are. Each author of the paper gets a separate data entry in the table. This is done for practical reasons, since the number of authors per paper is not predefined.

paperid	email_author
1	j.doe@umail.leidenuniv.nl

- *papers_subject_areas* lists for every paper what subject area is tagged as its primary subject area and what subject area are tagged as its secondary subject area(s) by the submission owner of the paper. The subject area that is tagged as the primary subject area to the paper has an asterisk (∗) added to its name, while the secondary subject area(s) do not.

paperid	subject_area
1	Social Network Mining

- *upload_time* lists for every paper each time an upload has been made for that particular paper. This table contains for every file upload of a file for a submission of a paper the date and time. In this table we have listed the last time an upload for a submission of a paper has been done as the top entry for that paper in the table and the first time this has been done as the bottom entry .

paperid	date_and_time
678	7/1/2013 1:15:59 AM

- *review_judgment* lists for every paper its final score as given by the metareviewer and reviewers whom were assigned to this paper.

paperid	judgment
13	Weak Reject

- *papers_accepted_id* lists for every paper in the conference it the paper is accepted into the conference. This is done by omitting the paperid's from the paper that are not accepted into the conference from this table.

paperid
1

Finally, we lists all the users of the conference with their email, affiliated organization and whether a user has a particular role within the organization in the *users* table. Since the selection of these roles was limited by the CMT some of these are redundant for our conference, but still denoted in this table. Because we can not conclusively determine which roles are listed due to this limitation of the CMT we have listed them just as the CMT does.

email_user	organization	is_author
j.doe@umail.leidenuniv.nl	Leiden University	Yes

is_associatechair	is_reviewer	is_metareviewer	is_submission
No	No	No	No

conflict_domains_not_entered_for_submitted_papers	is_chair
Yes	Yes

is_proceedings_editor
No

4.3 Converting the data

To convert our data to a relational database we use the programming language Python. We chose Python, because it is an easily usable interpreted language that has a large library of *modules*[10]. A **module** [6] is pre-programmed Python file that serves a specific function.

For the conversion of the multiple data files of different file types into a single relational database we three different modules to read our data and one module to write the data. We read *.xls* files with the *xlrd* module, *.xml* files with the *minidom* [3] and *.html* files with the *beautifulsoup* [4] module. The *sqlite3* [5] module was finally used to store the data we gathered from the data files into the relational database.

Chapter 5

Mining our data

For this thesis our goal is to create a model capable of predicting the values for our target feature. We mine our data with the data mining toolkit *WEKA* which requires its input to be in the *Attribute-Relation File Format (ARFF)* we will therefore first discuss this file format. Since the terms *attribute* and *feature* are synonymous they can be used interchangeably. To create a predictive model with WEKA we run one of its prepackaged learning algorithms over features extracted from our relational database. We recognize two kinds of features: *direct* and *indirect*. **Direct** features involve a single summation such as the number of authors of a paper or get a data entry from the database such as its primary subject area. **Indirect** features involve all features requiring more processing such as normalization one of these features is for example the level of interest in a paper from its reviewers in comparison to their interest to review other papers in the conference. For each feature we give its definition, its meaning it it not trivial, and mathematical notation if useful. The mathematical formula of a feature is denoted as F_p where F is the value of the feature and $_p$ the unique id given to a paper as it is sent into the conference, its *paperid*. Followed by the methodology used to extract its data from our relational database and a hypothesis for the expected behaviour of the feature with an evaluation of this hypothesis. During the evaluation of the hypothesis we disregard every data point which represents less than 1% of the papers in the conference. So we ignore all the data points represented by less than or equal to 4 papers and we perform linear regression if the data points for the analyzed feature does not show any immediate relation. In table B.1 we list the minimum, maximum, mean and correlation coefficient with the target feature for each feature.

With WEKA we perform *classification* and *regression*. We perform **classification** if our target feature is defined as a finite and explicitly defined set of labels and **regression** if the target feature can be any value in a range of continuous numerical values. For this project of the best case scenario is the ability to create a predictive model that predicts if a paper is accepted into the conference. Since the values for our target feature are then set to: *accepted* or *not*

accepted, a finite and explicitly defined set of labels. We perform classification. In this chapter we will show that the predictive models we create for this target feature are not more accurate than our baseline, *ZeroR*. This classifier ignores every features we extract from our dataset, except for the target feature it sets its predictions for this feature in the testing data to its majority value from the training data. With the observation of these poor results we want to create a model that predicts the average score as given by its reviewers for a paper, here we perform regression. We assume that we can make more accurate predictions for this target feature, since the decision-making process of PC Chairs to accept or reject a paper is undocumented, while the reviews by the reviewers of a papers clearly are well-documented. However, as our results below will show we still do not make any more accurate predictions than the predictive model created by our baseline.

In our final effort to successfully create a predictive model capable of predicting either if a paper is going to be accepted into the conference or capable of predicting the average score given by its reviewers we use *autoWEKA*. This data mining toolkit is an extension of WEKA that can automatically optimize the parameters of classifiers, select the most optimal classifier given multiple classifiers and apply attribute selection. Here we will not use any attribute selection as it allows for selecting another target feature and so invalidating this predictive models for our thesis. We define 2^3 experiments for autoWEKA as we are able to vary the dataset, selection of classifiers and target feature. In this chapter we will finally show that regardless of the optimization time we allow for these experiments: 1, 2 or 4 hours, we are not able to create and find a predictive model more accurate than our baseline. It is important to note that we abstain from attribute selection, since we are unable to guarantee that our target feature is not changed.

5.1 ARFF file

WEKA reads ARFF files. An ARFF file has the following structure:

```
@RELATION <relation-name>

@ATTRIBUTE <attribute-name>    <datatype>

@DATA
20.0
```

It has two distinct sections. The first section is the *Header* that is followed by the *Data* section. Lines that begin with a % are comments.

5.1.1 Header section of the ARFF file

On the first line of the **Header** of an ARFF file we declare the name of the relation it represents by denoting `@RELATION` followed by `<relation-name>` at

the top of the ARFF file. The name of the relation, `<relation-name>`, is a string and must be quoted if it includes spaces. The relation declaration is followed by feature declarations. Each feature declaration is stated on its own line starts with `@ATTRIBUTE` followed by its (unique) name and datatype. Features can have one of four datatypes:

- *Numeric*: *real* or *integer* numbers.

```
@ ATTRIBUTE <attribute-name> numeric
```

- *Nominal*: an explicit and finite set of labels as specified by the user.
- *String*: arbitrary text:

```
@ATTRIBUTE LLC    string
```

- *Date*: date and time in the format as declared with our date statement such as yyyy-MM-dd'T'HH:mm:ss

```
@ ATTRIBUTE <name> date [<date-format>]
```

5.1.2 Data section of the ARFF file

In the data section of our ARFF file we list each paper in the conference on a seperate line. In each line we list the feature declared as the nth feature in the header as the nth field in this line. For each paper the value for the features is gathered. We indicate the start of the data section by `@DATA`.

5.2 Extracting direct features from the relational database

In this section we discuss the direct features we extract from our relational database.

How many authors does the paper have?

We calculate the number of authors of a paper by counting the number of times we find its paperid in the *paper authors* table of our relational database. We hypothesize that papers with a certain number of papers are more likely to be accepted into the conference. Using this feature, we will check our hypothesis. In table B.1 we list this feature as *amount_of_authors*. Mathematically, we can denote this as follows:

$$F_p = \sum_{i=0}^{U} A_{ip} \tag{5.1}$$

$$A_{ip} = \begin{cases} 0 & \text{if user } i \text{ is not author of paper } p \\ 1 & \text{if user } i \text{ is the author } p. \end{cases}$$

U ranges over all the users in the database where $1 \leq i \leq U$.

In figure 5.1 we note an increase in the percentage of papers accepted into the

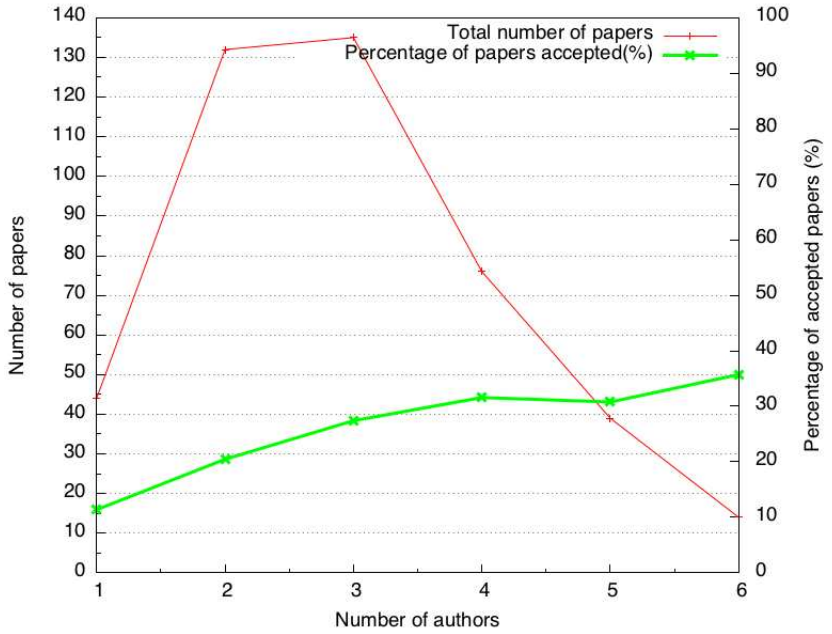

Figure 5.1: Here we plot for each possible number of authors of a paper the total number of papers with that number of authors in the conference and percentage of papers that are accepted into the conference.

conference as the number of authors of the paper increases. This increase does fade as the number of authors of a paper grows. We ignore data from papers with more than 6 authors, since the number of papers with this many authors is less than 1% of the total number of papers in the conference, 5 papers. From now on, we ignore data points with such small sample sizes. In figure 5.1 we also note that most papers have either 2 or 3 authors and that the number of papers related to the number of authors steeply declines as the number of authors increases beyond 3 authors.

How many bids are there on the paper?

After reviewing the author(s), title and abstract of a paper a user can express his interest to review a paper by bidding on it. We hypothesize that as the number of bids on a paper increases, so does its probability to be accepted into the conference. Using this feature, we will check this hypothesis. The reason of our hypothesis is that since users are not assigned to bid on a predefined set or number of papers they need to be genuinely attracted to or repulsed by the papers they decide to bid on. We gather the number of bids on a paper by

counting the number of occurrences of its paperid in the *bids* table. In table B.1 we list this feature as *amount_of_bids*. Mathematically, this feature can be denoted as:

$$F_p = \sum_{i=0}^{B} B_{ip} \tag{5.2}$$

$$B_{ip} = \begin{cases} 0 & \text{if bid } i \text{ is not a bid on paper } p \\ 1 & \text{if bid } i \text{ is a bid on paper } p. \end{cases}$$

$$B$$

and $1 \leq i \leq S$ ranges over all the *bids* in the database.

From figure 5.2 we note that most papers are bid on between 25 to 55 times,

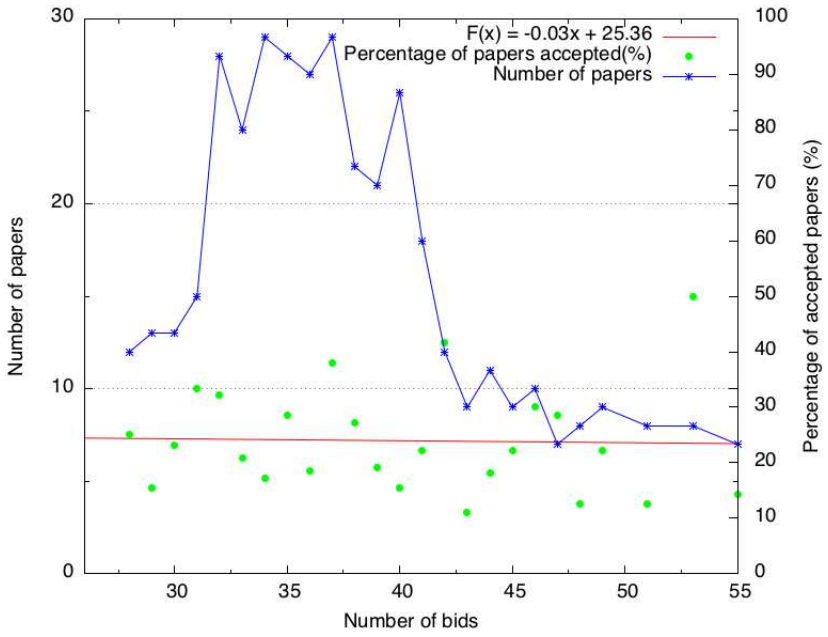

Figure 5.2: Here we plot for each possible number of bids on a paper the total number of papers with that number of bids in the conference and percentage of papers that are accepted into the conference.

where the large majority of these papers have been bid on between 30 and 40 times. While there are outliers, papers which are bid on just 10 times or all the way up to 80 the number of papers with these number of bids have a sample size of less than 4 papers and these data points are therefore ignored. While the percentage of papers that are accepted into the conference is scattered between 12% and 100% without any particular pattern abstracting these values with

linear regression shows that there is no relation between the number of bids on a paper and the likeliness of a paper to be accepted into the conference. We only note a decrease of 3% as the number of bids on a paper increases. We can therefore refute our hypothesis.

What is the primary subject area of the paper?

For each paper in *ECMLPKDD 2013* one of the 46 available subject areas in the conference needs to be selected as its primary subject area bu the submission owner of the paper. The primary subject area of a paper indicates its main theme. With this feature we intend evaluate if a paper selected with a certain subject area as its primary subject are is more likely to be accepted into the conference. Our hypothesis is that papers have a higher probability to be accepted into the conference with a subject area that is relatively popular as a primary subject area. We get the primary subject area of our paper by selecting the subject area with tagged with the $(*)$ in the *papers_subject_area* table. In table B.1 we list this feature as *primary_subject_area*. In figure 5.3 we note

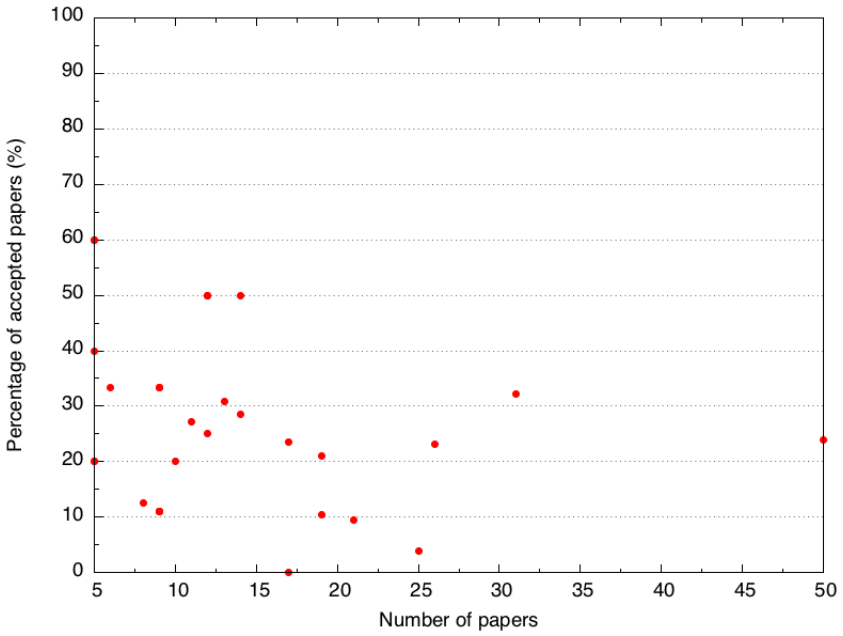

Figure 5.3: Here we plot for each available subject area in the conference the number of papers with that subject area as its primary subject area and the percentage of papers that are accepted into the conference.

that there is a decrease in the number of papers accepted into the conference as the popularity of the primary subject area of a paper increases. We can therefore refute our hypothesis. It is important to note that have omitted the

titles of the subject areas corresponding to our data point as it is not the goal of this feature to analyze the type of subject area and its success rate. Just as with the previous features we have omitted the data point with less than 4 papers due to their small sample sizes.

What subject areas are tagged as a secondary subject area to the paper?

The secondary subject area(s) of a paper indicate what other theme(s) the paper discusses apart from its main theme as indicated by its primary subject area. For each paper its submission owner needs to select at least one of the 46 available subject areas as its secondary subject area. As opposed to the previously evaluated features that are represented as a single attribute in the ARFF file this feature is represented by 46 attributes in our ARFF file where there is one attribute for each subject area. For each of the 46 subject area we then check if they are listed in the *paper_ subject_ area* table in our database with the paperid of our paper and without an $(*)$, since that would indicate that the subject area is the primary subject area of our paper. If the subject area is found to be a secondary subject area of our paper the value of its attribute in the ARFF file is set to `True`. If it is not the case however, the attribute is set to `False`. For this feature we expect to see that the percentage of papers accepted in the conference is higher for the subject areas which are less popular to be selected as secondary subject areas of the papers. Using this feature, we want to evaluate this hypothesis. In table B.1 we list this feature for each of the 46 available subject areas in the conference from *Active_ Learning* till *Web_ Mining*.

In figure 5.4 we note far more data points than in figure 5.3. This is to be expected as each paper is limited to having just a single primary subject area, while it can have multiple secondary subject areas. Here we note a similar relation as in figure 5.3: the number of papers accepted into the conference decreases at the popularity of its (one of) its corresponding secondary subject areas increases. We can therefore confirm our hypothesis. Here we also note more homogeneity in the data, since most secondary subject areas sit in the cluster where they are selected for 15 to 45 papers and of these papers between 15% and 45% is accepted into the conference.

How many subject areas are selected to the paper?

Apart from knowing what subject areas are selected as the secondary subject areas of a paper it is also interesting to determine if a paper with a certain number of secondary subject areas has a higher probability to be accepted into the conference. We calculate the number of secondary subject area(s) selected to a paper by counting the number of occurrences of our paperid in the *papers_ subject_ area* table for subject areas without an $(*)$. In table B.1 this feature is listed as *amount_ of_ tagged_ subject_ areas*. We can mathematically

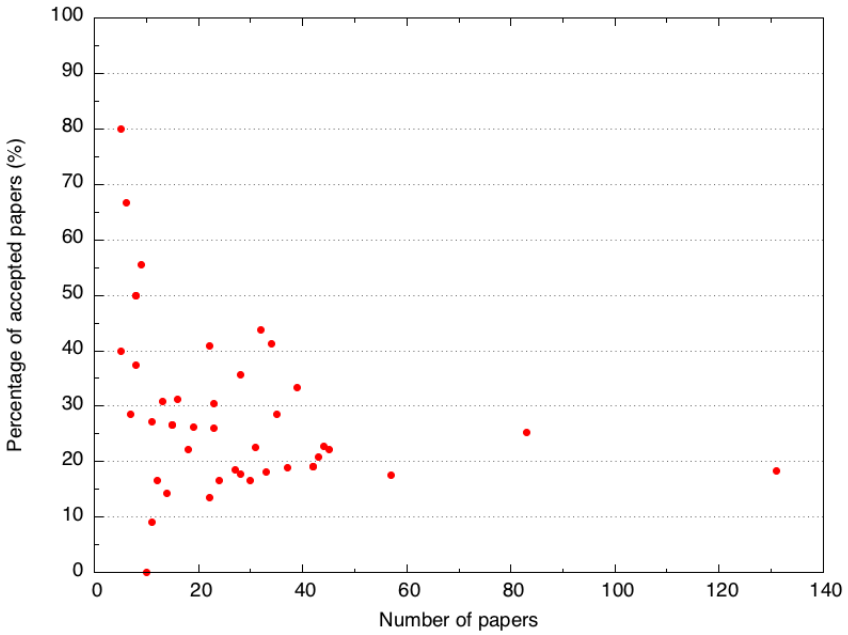

Figure 5.4: Here we plot for each available subject area in the conference the number of papers with that subject area as one of its secondary subject area and the percentage of papers that are accepted into the conference.

denote this calculation as follows:

$$F_p = \sum_{i=0}^{S} S_{ip} \tag{5.3}$$

$$S_{ip} = \begin{cases} 0 & \text{if subject area } i \text{ is not a subject area of paper } p \\ 1 & \text{if subject area } i \text{ is a subject area of paper } p \end{cases}$$

and $1 \leq i \leq S$ ranges over all the *subject areas* in the system.

In figure 5.5 we note that for any number of secondary subject areas between 1 and 6 the percentage of papers accepted into the conference is between 20% and 25% where it consistently alternates between a peak for an uneven number of selected subject areas and a dip for an even number of selected subject areas. However, since these differences in the percentage of accepted papers is so insignificant between an uneven and even number of selected subject areas this observation is not worth investigating. We can therefore refute our hypothesis.

At what date was the last file for a paper uploaded?

For every submission of a paper in the conference its submission owner needs to upload the paper itself on the CMT and possibly also any supplementary

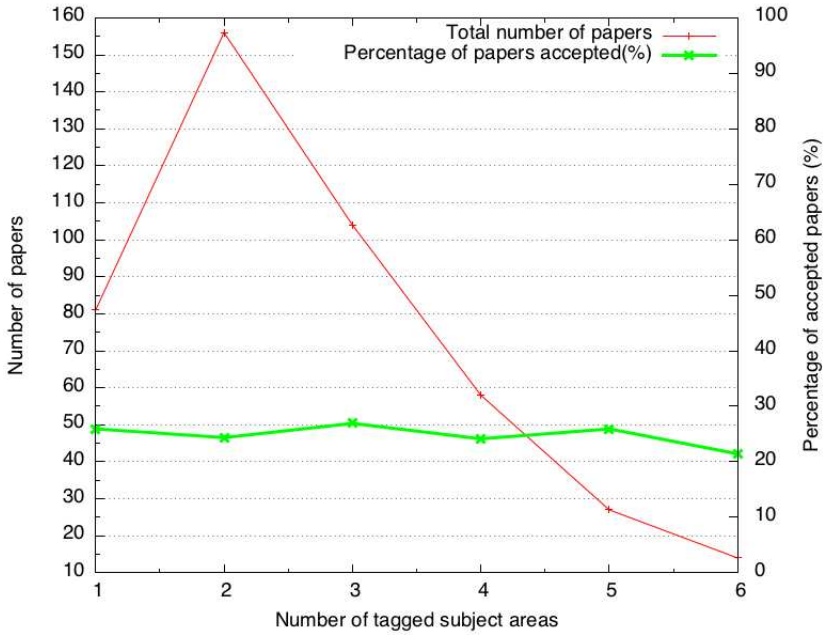

Figure 5.5: Here we plot for each available number of secondary subject areas tagged to a paper the total number of papers and percentage of accepted papers with that number of secondary subject areas.

files for the submission. All these upload activities are logged by the CMT. We get the data for this feature by finding the date with the paperid of our paper that is listed as much as possible to the top of the *upload_time* table, since the *upload_time* table orders the dates from top to bottom from the most recent upload to oldest. We hypothesize that papers are less likely to be accepted into the conference if their final upload for the submission of the paper is closer to the end of the submission window of the papers. Using this feature, we will evaluate this hypothesis. It is important to note that we disregard the uploads of the supplementary files as only 0.03% of the submissions also contains the upload of supplementary files. In table B.1 this feature is listed as *d_t_date*. In figure 5.6 we note that most papers have been uploaded in the second to last week of the submission period for *ECMLPKDD 2013*. We also note that the papers for which their final upload has been made in this week are more likely to be accepted into the conference. Furthermore, we also that as the number of final uploads increases so does the percentage of papers accepted into the conference. So the relation we detect in this figure can be most likely be attributed to this relation. We can therefore refute our hypothesis.

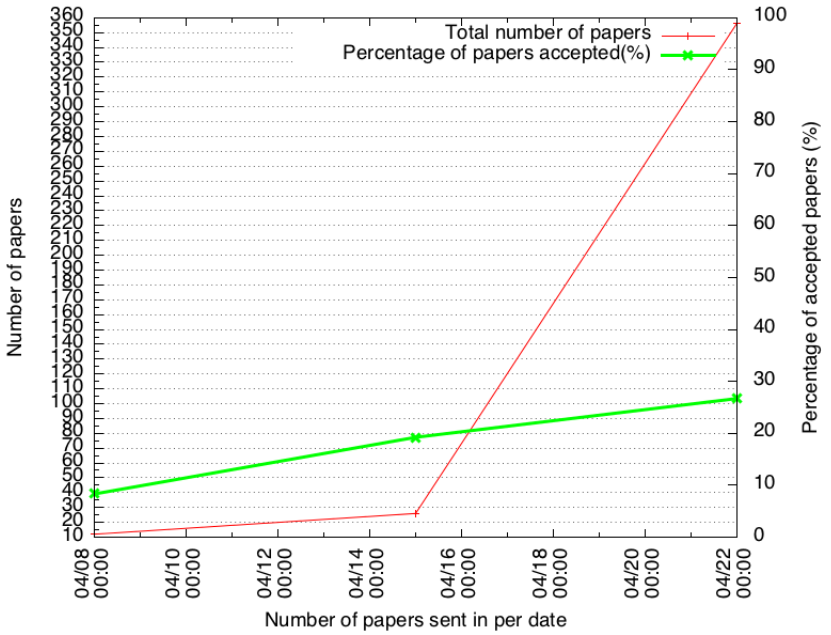

Figure 5.6: Here we plot per week what number of submission made their final upload in that particular week. For these final submission we also note what percentage of papers has been accepted into the conference. Here we only plot the period from the second to the last week of April 2013 as these were the only weeks in which the final upload were made.

5.3 Mining the direct features

We use WEKA to create a predictive models from our data to such a model we need to define our dataset, target feature and classifier. Here we use the direct features as our dataset. We split this dataset into training (75%) and testing (25%) data: we use every 4^{th} data entry as our testing data. We stratify our testing data, since each paper is assigned to its unique id, *paperid*, when they are submitted into the conference to omit any unforeseen we therefore stratify our testing data. With our predictive model we want to predict the values for this target feature:

Is this paper going to be accepted into the conference?

Each paper sent into the conference is reviewed and then accepted or rejected. If a paper is accepted into the conference we set this feature to `True` and if it is rejected we set it to `False`. Since we recognize a finite and explicitly defined set of values for our target feature we perform classification. The data for this feature is gathered by checking if the id of our paper is listed in the

paper_ accepted table.

As it is unclear to us what classifier will be able to create the most accurate predictive model from our features we have decided to use the two most popular classifiers from each *base* types of classifiers. A **base** classifier is a classifier that is capable of creating a model from features on its own, unlike the *meta* and *mi* type of classifier that require one base classifier and the *ensemble* type of classifier that requires at least one base, meta or mi type of classifier to create a predictive model. We recognize 5 types of *base* classifiers: *bayes, functions, lazy, rules* and *trees*. From these types of classifiers we select the *Logistic, Votedperceptron, IBk, KStar, ConjunctiveRule, DecisionTable, J48* and *REPTree*. Each of these classifiers is used with their default parameter settings. We do not use any bayes type of classifiers as these have proven incompatible with our dataset: some of our features have a standard deviation of 0.0. While the features with such a standard deviation are useless we abstain from using attribute selection in this thesis and therefore keep these features in our dataset.

We measure the predictions of our models among other metrics with the **Correctly Classified Instances (CCI)**. This performance measure calculates the percentage of correctly classified instances. The **Kappa Statistic (KS)**. The point of KS is that unlike CCI, it is chance corrected and sensitive to class distribution. Thus, as disagreement increases KS will decline more quickly than CCI will, because it is chance corrected and sensitive to class distribution. KS is defined as:

$$KS = \frac{\text{observed accuracy - expected accuracy}}{(1 - \text{expected accuracy})}$$

A KS closer to 1 is considered better where a KS of 0.40 is already considered a very good result in machine learning. For KS with a negative value we note that the agreement between our predictions and the actual values occur are worse than it predictions made on chance alone. The **F-Measure** combines two measuring methods : *precision* and *recall*. Mathematically, we denote this metrics as:

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall}$$

With the **precision** we calculate what percentage of papers we found as accepted were actually accepted, while with **recall** we calculate what percentage of the total number of papers accepted we actually found. More formally, we define these metrics as:

$$Precision = \frac{TP}{TP + FP}$$
$$Recall = \frac{TP}{TP + FN}$$

where TP, FP and FN represent the true positives, false positives and true negatives, respectively. Finally, we use the **Receiver Operator Characteristic**

Classifiers	Direct features				Direct and Indirect features			
	CCI	KS	F-Measure	ROC	CCI	KS	F-Measure	ROC
ZeroR	**78.38**	0.00	0.69	0.50	**78.38**	0.00	0.69	0.50
Logistic	64.86	−0.05	0.65	**0.53**	59.46	−0.03	0.62	0.51
VotedPerceptron	**78.38**	0.00	0.69	0.50	**78.38**	0.00	0.69	0.50
IBk	65.77	−0.01	0.66	0.50	63.96	0.03	0.65	0.51
KStar	21.62	0.00	0.08	0.50	21.62	0.00	0.08	0.50
ConjunctiveRule	**78.38**	0.00	**0.70**	0.50	**78.38**	0.00	0.69	0.51
DecisionTable	77.48	0.03	**0.70**	**0.53**	77.48	0.03	**0.70**	**0.53**
J48	74.77	0.02	**0.70**	0.50	72.97	−0.01	0.68	0.49
REPTree	72.07	**0.08**	**0.70**	**0.53**	71.17	**0.06**	**0.70**	0.52

Table 5.1: The three leftmost columns of this table contain our results for classification with only the direct features, while the three rightmost columns are our results for classification with the direct and indirect features.

(ROC) area. For this metric we draw the FP and TP as x and y respectively on both horizontal and vertical lines ranging from 0 to 1. With this graph we depict trade-offs between the TP, benefits, and FP, costs. A ROC area closer to 1 is considered the most accurate prediction, while any ROC area equal to or below 0.5 is considered worse than random.

We compare the result for these metrics of our predictive models with the predictive model created with our baseline; **ZeroR** from the rules type of classifiers. We use this classifier as our baseline as it ignores every features we extract from our dataset and only considers the target feature. This classifier sets its predictions for the target feature in the testing dataset to the majority label it has found for the target feature in the training data [7].

In the four leftmost columns of table 5.1 we list all the results of our predictive models for the previously mentioned metrics. From these results we note that none of the 8 predictive models have a CCI higher than the CCI of our baseline, 78.38%. From this table we also note that both VotedPerceptron and ConjunctiveRule have a similar CCI as our baseline. We note an expectant KS of 0.00 for our baseline as this metric is chance corrected and sensitive to class distribution. The most accurate predictive model as measured per this metric was REPTree with a KS of 0.08, but this still is a very small improvement over our baseline. For Logistic and IBk we note negative KS values indicating that the predictions made by these classifiers are even worse than random guessing, after all even our baseline has a higher KS. The results of the F-Measure also prove that our predictive models are not significantly better than our results for this metric for the baseline where our result for the baseline is 0.69. Our results for the F-Measure are at best 0.70 for some of our classifiers. An improvement of 1.44% is not really a (significant) improvement over the baseline. For the

ROC area we note that the result for our baseline for this metric is equal to a ROC area for random predictions, 0.50. For the remaining classifiers we note a similar ROC area values, except for Logistic, DecisionTable and REPTree which show an improvement of 6.00% for this metric in comparison to baseline. However, this improvement for the ROC are can be disregarded as they are not significant enough.

5.4 Extracting indirect features from the relational database

In this section we discuss the indirect features we extract from our relational database

What is the average bid on the paper?

By comparing the average value of the bids on a paper we can compare the general interest of users in the conference to review a paper. A user can express its interest in reviewing a paper in one of four kinds of bids. We list these bids as follows with their numerical values: *not willing (0)*, *in-a-pinch* (2), *willing* (3) or *eager* (4) , or the user can abstain from bidding on a paper (1). We note that a user is positively interested in reviewing a paper if his bid is higher than equal to 2. To calculate this feature we summarize every bid in the *bids* table that has our paper and divide this value by the number of occurrences of our paperid in this table. Our hypothesis is that papers with higher average values for their bids are more likely to be accepted into the conference. In table B.1 we list this feature as *average_ score_ of_ bids*. Mathematically, this calculation is denoted as:

$$F_p = \frac{\sum_{i=0}^{B} Bv_{ip}}{\sum_{i=0}^{B} B_{ip}} \tag{5.4}$$

$$Bv_{ip} = \begin{cases} 0 & \text{if bid } i \text{ is not a bid on paper } p \\ value\ of\ bid & \text{if bid } i \text{ is a bid on paper } p \end{cases}$$

$$B_{ip} = \begin{cases} 0 & \text{if bid } i \text{ is not a bid on paper } p \\ 1 & \text{if bid } i \text{ is a bid on paper } p \end{cases}$$

and $1 \leq i \leq B$ ranges over all the *bids* in the system.

In figure 5.7 we note two peaks one at the average bid of 0.75 and the other at the average bid of 1.75 these peaks indicate that there were quite a lot of papers not bid on by the reviewers and metareviewers, since we gave papers a default value of 1 if they had not been bid on by a reviewer or metareviewer. Since these peaks are not at 1, exactly, we can note even though some papers were not bid on by most reviewers and metareviewers at least they were bid on by some of them. With regard to the percentage of accepted papers we note that most of them are scattered between 15% and 40% as the average value of

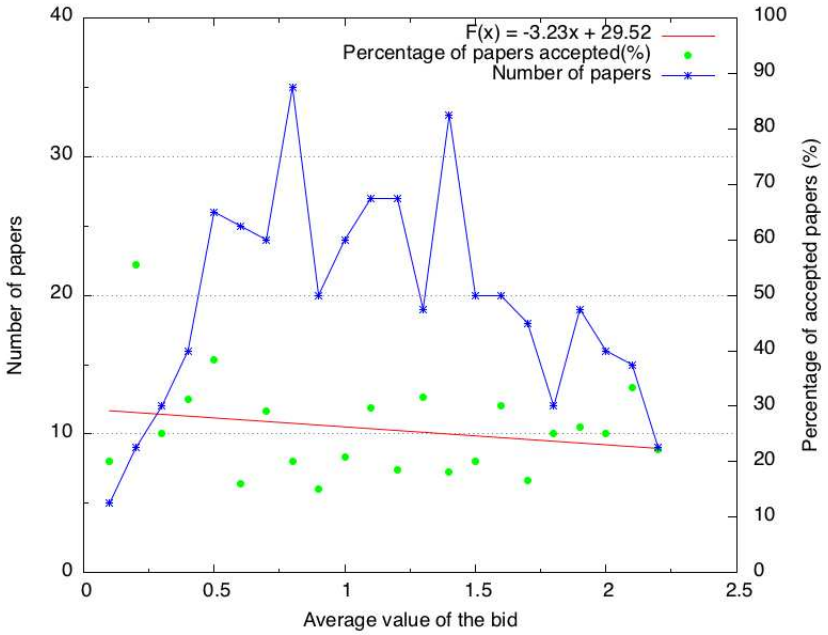

Figure 5.7: Here we plot for each possible average bid on a paper the number of papers with that average value of bids and the percentage of papers that are accepted into the conference.

the bids on a paper increases we note that the percentage of papers that are accepted decreases. With a decrease of about 5% it does not really seems to be a significant relation. We can therefore refute our hypothesis.

Count the other roles which every user who is related to the paper can have.

In *ECMLPKDD* a user could be related to a paper in four ways. A user could have written of a paper (author), bid on a paper (bidder), review a paper(reviewer) or supervise the review process of a paper (metareviewer) where a user could also sent the paper in(submission owner). Here we note several excluding relationships. The author of a paper can not be its reviewer or metareviewer and the reviewer of a paper can not be its metareviewer. Finally, a user could also have a role in the organization of the conference such as *associate chair, chair, external reviewer* or *proceedings officer*. For this feature we are evaluating for each primary relation of a user with a paper what influence this would have on accepting a paper if the user also has an organizational role in the conference. Subsequently, we get combinations of positions such as *author_is_associate_chair*. We will also evaluate what influence it has if a user has two different kinds of primary relationships with two different paper.

A user could for example be the author of one paper, while also being one of the reviewers of another paper. This results in the *reviewer_ is_ author* relationship.

We get the data for each of the 26 entries as seen in table B.1 for this feature in similar ways. Here we discuss what we do to get the value for the combination textttauthor_is_associate_chair. We gather the email addresses of the author(s) from the *paper_ authors* table and then check in the *users* table if an authors is also an associate chair. For each author that has such position we add 1. Finally, we divide this value by the number of authors for our paper. So if we have 4 authors for our paper whereof 3 are also an associate chair we would get a value for this feature of 0.75. With regard to the relatively low standard deviation values for the entries of this feature in B.1 we can conclude that there are no interesting hypothesis to think of for this feature. Mathematically, we can denote this relation as follows:

$$F_p = \frac{\sum_{i=0}^{U}(U_i * U_{ip})}{\sum_{i=0}^{U} U_{ip}} \tag{5.5}$$

$$U_i = \begin{cases} 0 & \text{if user } i \text{ is not a reviewer} \\ 1 & \text{if user } i \text{ is a reviewer} \end{cases}$$

$$U_{ip} = \begin{cases} 0 & \text{if user } i \text{ is not an author of paper } p \\ 1 & \text{if user } i \text{ is an author of paper } p \end{cases}$$

and $1 \leq i \leq U$ ranges over all the *users* in the database.

What is the average number of papers the reviewers of this paper need to review?

Each paper in the conference is assigned to a group of three or four users to review this paper. Each of these reviewers can be assigned to also review other papers. The total number of papers such reviewer is assigned to, is called his workload. To calculate the value of this feature we need to calculate the workload for each reviewer of our paper, summarize these workloads and divide them by the number of reviewers of our paper. We calculate the workload of each reviewer by getting his email address from the *reviewer_ assignments_ by_ paper* table and subsequently count the number of occurrences of this email address in this table. Our hypothesis is that a paper assigned to a group of reviewers with a relatively low average workload are more likely to be accepted into the conference. In table B.1 this feature is listed as *average_ amount_ of_ papers_ assigned_ to_ reviewers*. Mathematically, we can

denote this feature as follows:

$$F_p = \frac{\sum_{i=0}^{U} \left(\left(\sum_{p'=0}^{P} P_{ip'} \right) * U_{ip} \right)}{\sum_{i=0}^{U} U_{ip}} \tag{5.6}$$

$$U_{ip} = \begin{cases} 0 & \text{if user } i \text{ was not assigned to paper } p \\ 1 & \text{if user } i \text{ was assigned to paper } p \end{cases}$$

$$P_{ip'} = \begin{cases} 0 & \text{if paper } p\text{' was not assigned to user } i \text{ to review} \\ 1 & \text{if paper } p\text{' was assigned to user } i \text{ to review} \end{cases}$$

and $1 \leq i \leq U$ ranges over all the users in the database and $1 \leq p' \leq P$ ranges over all the papers in the database. In figure 5.8 we note that a slightly higher

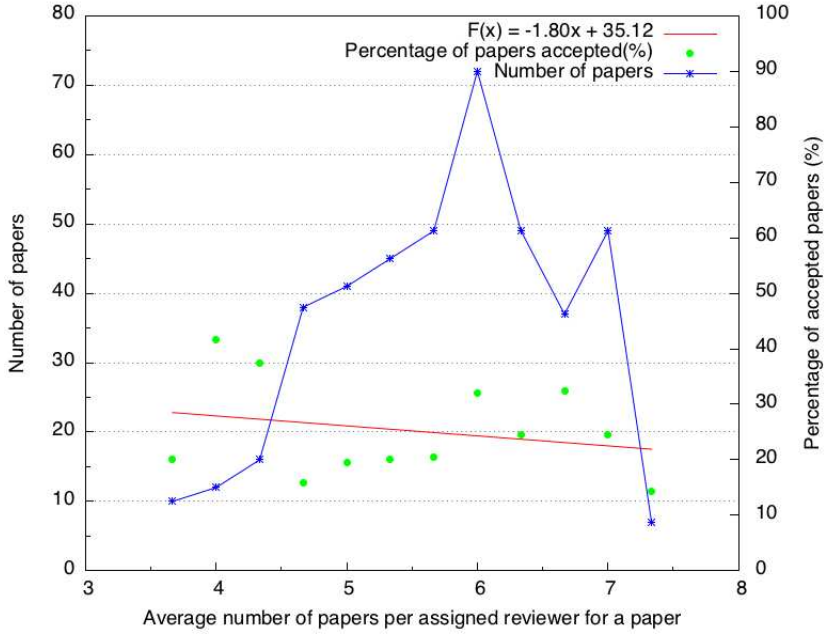

Figure 5.8: Here we plot for each average number of papers assigned to the reviewer of a paper the total number of papers and percentage of accepted papers with that number.

percentage of papers is accepted into the conference as the reviewers of a paper have a lower workload. Overall, we note that this is decrease of about 2.25% in the number of papers accepted into the conference as we move from an average workload of 3.5 papers to a workload of approximately 7 papers. Of course, since we are dealing with averages for the papers it is possible to express the workload in fractions. Regardless, of the linear relation between these variables we note that the data points corresponding to a workload between 6 and 7 papers are

at least 5% and at most 15% better than those data points corresponding to a workload between 4.5 and 5.5 papers. So, while the overall linear relation between the variables in our figure indicate that we can refute our hypothesis it is entirely possible that more data points for a higher average workload could indicate as a confirmation of our hypothesis: which still remains a far more plausible observation for this feature.

Were the reviewers of this paper more than average interested in reviewing this paper?

For this feature we determine what the average bid of interest on our paper is by all the users in the conference. Subsequently, we determine what the average bid of interest is of the reviewers on our paper. By dividing the average bid of interest of all the users on our paper by the average bid of interest of all its reviewers we can establish the relative level of interest of our reviewers in comparison to all the users in the conference. It is important to note than even abstinence of bidding on a paper is numerically represented, see section 3.1. We calculate this feature by dividing the average bids of interest of the reviewers of our paper by the average bids of interest of all the users in the conference. The average bids of interest of the reviewers is calculated by getting the email addresses of the group of reviewers of our paper from the *reviewer_ assignments_ by_ paper* table and subsequently finding their bid on our paper in the *bids_ complete* table. These bids are then averaged. From the *bids_ complete* table we summarize all the bids on our paper and divide that number by the number of bids on our paper to compute the average bid of interest from all the users in the conference on our paper. We hypothesize that papers are more likely to be accepted into the conference if its group of reviewers have a relatively higher level of interest in reviewing the paper than all the users in the conference. Using this feature, we want to evaluate our hypothesis. In table B.1 this feature is listed as *higher_ bid_ from_ reviewer_ than_ average_ bid*. Mathematically, this feature can be denoted as:

$$F_p = \frac{\sum_{i=0}^{U} \frac{Bv_{ip}}{\left(\frac{\sum_{p'=0}^{P} Bv_{ip'}}{\sum_{p'=0}^{P} Bb_{ip'}}\right)}}{\sum_{i=0}^{U} Bb_{ip}} \tag{5.7}$$

$$Bv_{ip} = \begin{cases} 0 & \text{if user } i \text{ did not bid on paper } p \\ \textit{value of bid} & \text{if user } i \text{ did bid on paper } p, \text{ and the value of this bid is at least 1} \end{cases}$$

$$Bb_{ip} = \begin{cases} 0 & \text{if } \mathrm{Bv}_{ip} = 0 \text{ (so a reviewer who has not bid on the paper will not be counted)} \\ 1 & \text{otherwise} \end{cases}$$

and $1 \leq i \leq U$ ranges over all the bids in the database and $1 \leq p \leq P$ ranges over all the papers in the database. In figure 5.9 we note a decrease in the percentage of papers accepted into the conference as the average bid of the reviewers of a paper increases. This decrease indicates that a paper reviewed by reviewers who have a more than average interested in reviewing

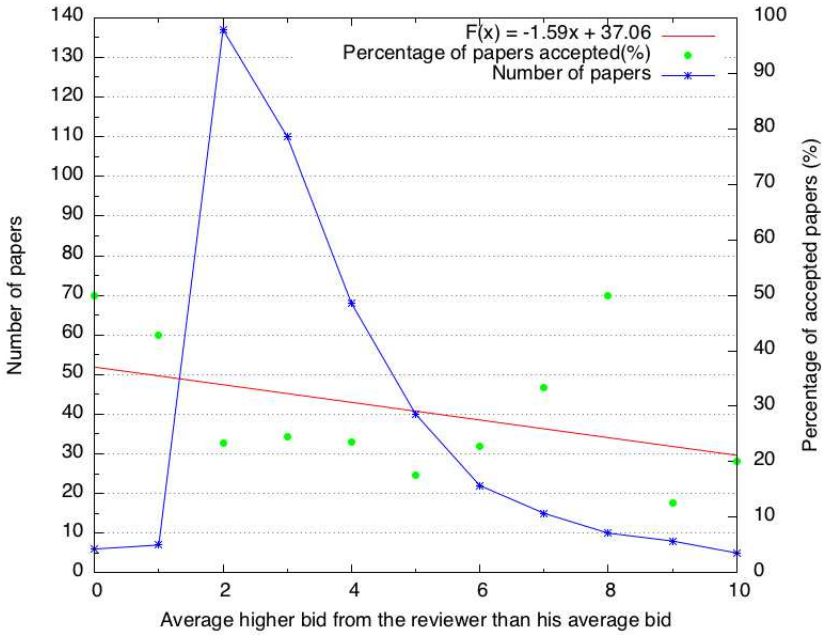

Figure 5.9: Here we plot for each number of times a bid from a user on our paper fits into his average bid value on all the paper the total number of papers and percentage of accepted papers with that number.

this paper than to review other papers in the conference are more likely to not be accepted into the conference. However, to counterweight this observation we note a steep increase in the percentage of papers accepted into the conference as the average bid on a paper goes from 6 to 8. However, the percentage of papers accepted into the conference for an average bid of 8.5 shows a steep decline in comparison to this percentage for an average bid of 8. Since this percentage increases again as the average bid increases the average bids between 8.5 and 10 could be anomalies, however due to a lack of data points with a sufficiently large sample size for larger average bids we can not test this observation. So, for now we can conclude that we refute our hypothesis.

What is the average bid on our paper if we normalize the bid of a user by the number of bids this user made?

Here we take the bid of a user on a paper and divide it by the total number of bids of interest this user has made on all the papers in the conference. With this feature we want to give more weight to a bid of interest from a user who bids on less papers, since we hypothesize that such a user makes his bids with more care. We calculate this feature by getting the value of the bid and email address of a user who bid on our paper from the bids_complete table. His bid

is then divided by the number of occurrences of his email address in this table. This value is computed for each user who has bid on our paper. The normalized bids of each user are subsequently averaged over all the users who have bid on our paper to get the value of this feature. We hypothesize that papers with a higher normalized average bid have a higher probability to be accepted into the conference. Using this feature, we want to evaluate this hypothesis. In table B.1 this feature is listed as *bids_ normalized_ per_ amount_ of_ bidder*. Mathematically, this feature can be denoted as:

$$F_p = \frac{\sum_{i=0}^{B} \left(Bv_{ip} * (\frac{\sum_{j=0}^{U} U_{ji}}{\sum_{k=0}^{P} P_{kj}}) \right)}{\sum_{i=0}^{B} B_{ip}} \tag{5.8}$$

$$Bv_{ip} = \begin{cases} 0 & \text{if bid } i \text{ is not a bid on paper } p \\ \textit{value of bid} & \text{if bid } i \text{ is a bid on paper } p \end{cases}$$

$$U_{ji} = \begin{cases} 0 & \text{if user } j \text{ has not made bid } i \\ 1 & \text{if user } j \text{ has made bid } i \end{cases}$$

$$P_{kj} = \begin{cases} 0 & \text{if user } j \text{ has not bid on paper } p \\ 1 & \text{if user } j \text{ has bid on paper } p \end{cases}$$

$$B_{ip} = \begin{cases} 0 & \text{if bid } i \text{ is not a bid on paper } p \\ 1 & \text{if bid } i \text{ is a bid on paper } p \end{cases}$$

and $1 \leq i \leq B$ ranges over all the *bids*, $1 \leq i \leq U$ over all the *users* and $1 \leq i \leq P$ over all the *papers* in the conference.

In figure 5.10 we note that an overall increase in the percentage of papers accepted into the conference as the values of the normalized bids increase. We can therefore confirm our hypothesis. However, from the data points we note that this is not a completely linear relationship, since the percentage of papers accepted into the conference decreases between normalized bids averages of 0.00 to 0.03, while this percentage remains largely consistent between bid values between 0.03 and 0.06. A large increase in the percentage of papers accepted in the conference happens for the bid values from 0.06 and beyond. Furthermore, we note that most papers have a bid value between 0.02 and 0.04.

What is the average bid on our paper if we normalize the bid of a user by the average value bids this user made?

If we divide the bid of a user on our paper by the average bid of this user on all the papers in the conference, we can determine on average to what extent such user is interested in reviewing our paper as opposed to the other papers in the conference. For > 1 the level of interest of this user to review our paper is higher

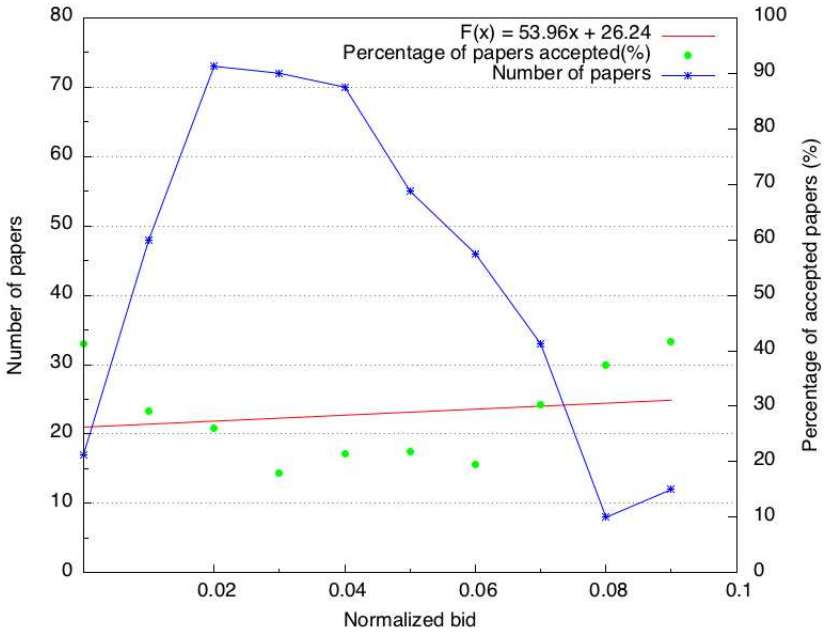

Figure 5.10: Here we plot for each normalized bid the total number of papers and percentage of accepted papers with that normalized bid. A normalized bid is the bid from a user on our paper divided by his average bid on all the papers in the conference. Since we only consider the bids of this user where he actually bid on a paper we disregard bids with value 1 as this is the default value for a bid.

than his average level of interest to review the other papers in the conference. We calculate this feature just as the previous feature with the notable exception that we now normalize his bid by his average bid on all the papers in the conference. We get his average bid on all the papers in the conference by taking his email address from the bids_complete table and summarizing the bids from this table with this email address. Finally, this sum is divided by the total number of bid this user made. We compute this for every user who bid on our paper and divide by the total number of user who bid on our paper to get the value for this feature. Our hypothesis is that papers with a higher normalized average bid have a higher probability to be accepted into the conference. In table B.1 this feature is listed as *bids_normalized_per_average_of_bidder*. Mathematically, this feature can be denoted as:

$$F_p = \frac{\sum_{i=0}^{B} \left(Bv_{ip} * \frac{\sum_{j=0}^{U} U_{ji}}{\left(\frac{\sum_{k=0}^{P} Pv_{kj}}{\sum_{k=0}^{P} P_{kj}} \right)} \right)}{\sum_{i=0}^{B} B_{ip}} \tag{5.9}$$

$$Bv_{ip} = \begin{cases} 0 & \text{if bid } i \text{ is not a bid on paper } p \\ \textit{value of bid} & \text{if bid } i \text{ is a bid on paper } p \end{cases}$$

$$U_{ji} = \begin{cases} 0 & \text{if user } j \text{ has not made bid } i \\ 1 & \text{if user } j \text{ has made bid } i \end{cases}$$

$$Pv_{kj} = \begin{cases} 0 & \text{if user } j \text{ has not bid on paper } p \\ \textit{value of bid} & \text{if user } j \text{ has bid on paper } p \end{cases}$$

$$P_{kj} = \begin{cases} 0 & \text{if user } j \text{ has not bid on paper } p \\ 1 & \text{if user } j \text{ has bid on paper } p \end{cases}$$

$$B_{ip} = \begin{cases} 0 & \text{if bid } i \text{ is not a bid on paper } p \\ 1 & \text{if bid } i \text{ is a bid on paper } p \end{cases}$$

and $1 \leq i \leq B$ ranges over all the *bids*, $1 \leq i \leq U$ over all the *users* and $1 \leq i \leq P$ over all the *papers* in the conference.

In figure 5.10 we note that the percentage of papers accepted into the conference increases as the values of the normalized bids increase. Furthermore, we note a mostly even distribution surrounding our regression line with most data points sitting between a percentage of 20% to 40% of papers accepted into the conference. We can therefore confirm our hypothesis. In this figure we also note that most papers have a normalized bid of either 0.25 or 0.75.

What is the popularity of the group of subject area selected for our paper?

Each paper has one primary subject and one or more secondary subject areas. Here we count for this group of subject areas the number of times they have been selected as primary subject areas for all the papers in the conference. We also do this for the number of times they have been selected as secondary subject areas or as either primary or secondary subject areas. We average the popularity of this group of subject areas, since the number of secondary subject areas selected for a paper can vary among the papers in the conference and this could influence our results. We calculate these features by getting the subject areas of our paper from the *subject_ area_ per_ paper* table and subsequently count their number of occurrences as primary or secondary subject areas in this table. For each of the three kinds of feature we discuss

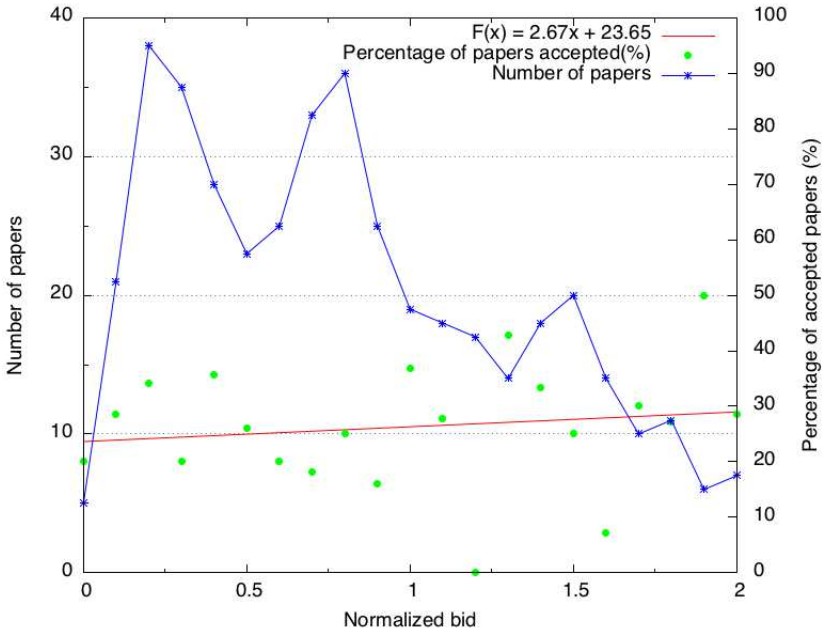

Figure 5.11: Here we plot for each normalized bid the total number of papers and percentage of accepted papers with that normalized bid. A normalized bid is the bid from a user on our paper divided by the number of papers this user has bid on. This also includes the bids this user did not explicitly bid on so the bids with the default bid value of 1.

here we summarize their required number of occurrences to average it and the value for the respective feature. We hypothesize that the papers with the more popular group of subject areas are more likely to be accepted into the conference. We expect to see this kind of relation more clearly where we measure the popularity as the number of times the subject areas have been selected as primary subject areas as opposed to secondary subject areas. In table B.1 these features are represented as *average_pri_popularity_subject_area_paper*, *average_sec_popularity_subject_area_paper* and *average_pri_or_sec_popularity_subject_area_paper*, respectively. Mathematically, we denote this feature as follows for the popularity of the group of subject areas as the primary subject areas, since the other features are denoted in a similar fashion they are omitted:

$$F_p = \sum_{i=0}^{S}(S_{ip} * \sum_{j=0}^{P}(S_{ji}) \tag{5.10}$$

$$S_{ip} = \begin{cases} 0 & \text{if subject area } i \text{ is not a subject area of paper } p \\ 1 & \text{if subject area } i \text{ is a subject area of paper } p \end{cases}$$
$$S_{ji} = \begin{cases} 0 & \text{if subject area } i \text{ is not a subject area of paper } j \\ value & \text{if subject area } i \text{ is a primary subject area of paper } j \end{cases} \tag{5.11}$$

and $1 \leq i \leq S$ ranges over all the *subject areas* and $1 \leq j \leq P$ over all the papers in the conference.

In figure **??** we note a consistent relationship between the percentage of papers

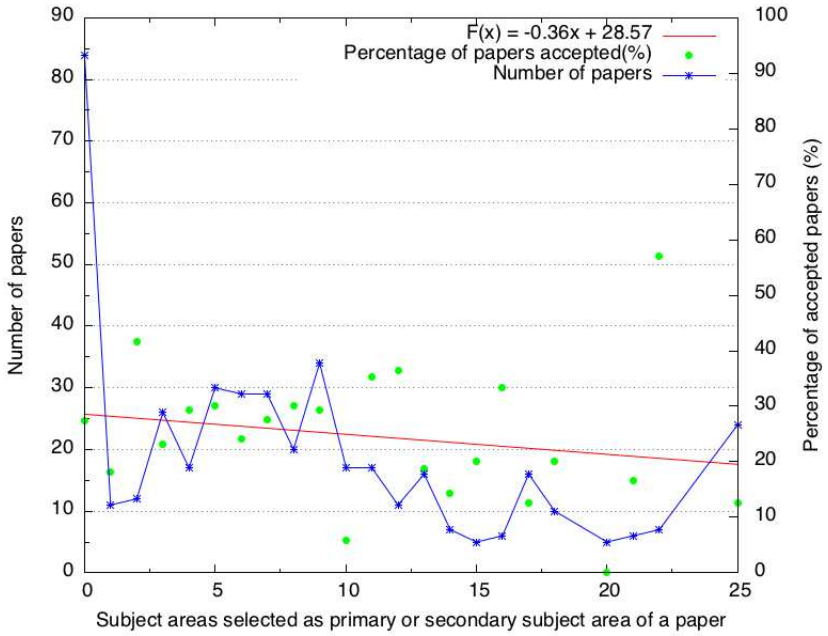

Figure 5.12: Here we plot for each popularity value of a subject area, either as primary or secondary subject area, the total number of papers and percentage of accepted papers with that popularity value.

accepted into the conference and the overall popularity of its group of selected subject areas as the primary subject area for all the papers in the conference. We note for these groups of subject areas that most of them are on average only 30 times selected as primary subject areas for the papers in the conference where the majority of papers accepted into the conference is between 10% and 40%. Completely in agreement with out linear relation we do not notice any particular pattern in the distribution of the data points. We can therefore refute our hypothesis that papers where their group of selected subject areas is more popular as primary subject areas is more likely to be accepted into the conference. In figure 5.13 we note that the probability of a paper to be accepted into

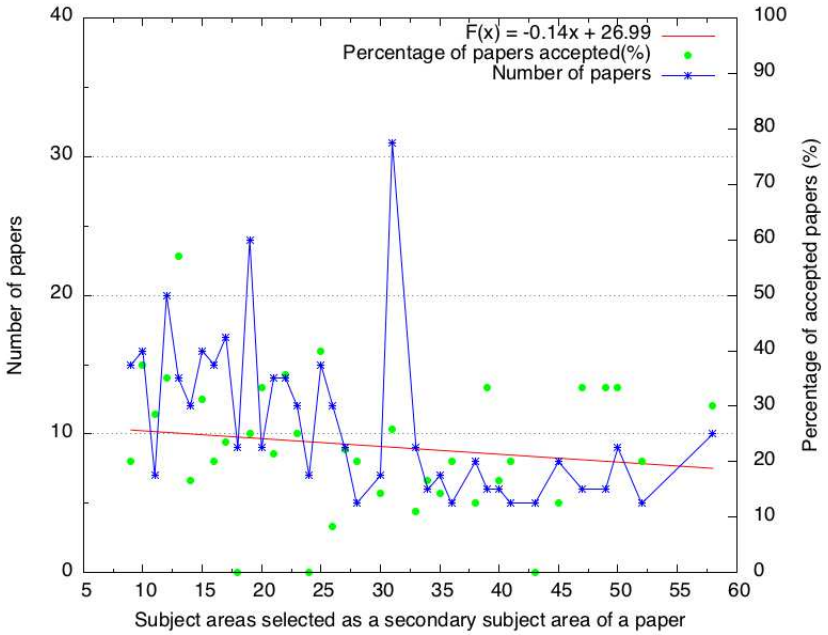

Figure 5.13: Here we plot for each popularity value of a subject area, either as primary or secondary subject area, the total number of papers and percentage of accepted papers with that popularity value.

the conference decreases as the popularity of its group of selected subject areas increases. More specifically, here we are talking about the popularity of this group of subject areas as the number of times these subject areas are selected as the secondary subject areas for the papers in the conference. Since we did not see any relationship in figure 5.13 and our hypothesis was that the relation in that figure is a more clear version of the relation in figure 5.13 we refute this hypothesis. In figure 5.13 we also note the majority of papers accepted into the conference is between 10% and 40%, but now we do recognize an effective linear relation for this percentage. In both graph we also see a somewhat similar distribution between the absolute number of papers corresponding to the number of times our group of subject areas have been selected as, respectively, primary or secondary subject areas. To be concise we also plot the relation between the percentage of papers accepted into the conference and the popularity of the group of subject areas selected for a paper. Here their popularity is measured as the number of times these subject areas have been selected as either the primary or secondary subject areas of the papers in the conference. We plot this relationship in figure 5.14. In this figure we note a decline in the percentage of papers accepted into the conference as the popularity of our subject area rises. The decline we measure here is larger than the decline we have measure in figure 5.13 this is unexpected behaviour, since figure 5.14 is a combination of

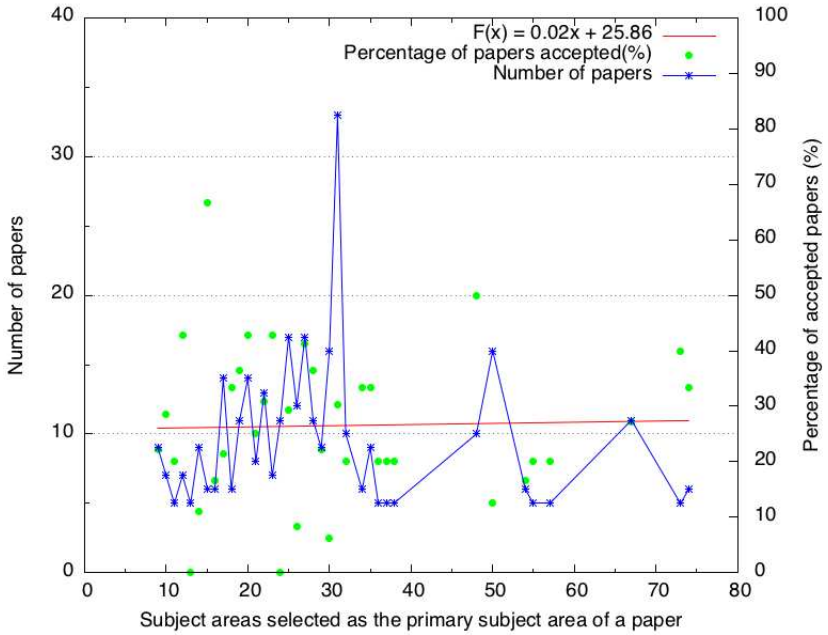

Figure 5.14: Here we plot for each popularity value of a subject area, either as primary or secondary subject area, the total number of papers and percentage of accepted papers with that popularity value.

concisive

figure 5.12 and 5.13. We do note that the majority of papers accepted into the conference is between 10% and 40% as we would expect.

How many *authors* from the paper are from a specific continent?

Here we want to evaluate if there is a relationship between the origin of an author and the probability of his paper to be accepted into the conference. Since the nationality of a user is not explicitly defined we get their nationality by the finding the corresponding nationality to the top-level domains of their email address. With this method we get the nationality of their email provider in the worst case and the nationality of the organization they are affiliated to in the best case depending on the kind of email address of the user. Since most international organizations and email providers such as *gmail.com* use the *.com, .org* and *.net* top-level domains we created a fictitious continent, *OP*, email addresses with these top-level domains are counted for the fictitious continent. With the relatively low number of user in the conference of 2072 user we expect the diversification of nationalities to be low and therefore evaluate the nationalities of the authors on a continent level. When evaluating the data for North-American continent we need to take this into account. As the number of

authors per continent is counted this feature is represented with 7 entries in table B.1: *AF, AS, AU, EU, NA, OP* and *SA*. We calculate this feature by extracting the email addresses of the author(s) of our paper from the *authors_per_paper* table and getting their top-level domains, to connect these to their respective continent as listed in `domains_with_continents.txt`. Finally, we run to each available continent and add 1 to its value if one or more of the authors of our paper originates from this continent. Since *ECMLPKDD* is an European conference we hypothesize that most authors are from Europe. We do not hypothesize however that there the origin of the authors of a paper influences its probability to be accepted into the conference. Using this feature, we evaluate this hypothesis. Mathematically, we can denote this feature as follows:

$$F_{continent_p} = \bigcup_{i=0}^{U} U_{ip} * \sum_{j=continent_1}^{C} U_{ji} \tag{5.12}$$

$$U_{ip} = \begin{cases} 0 & \text{if user } i \text{ is not an author of paper } p \\ 1 & \text{if user } i \text{ is an author of paper } p \end{cases}$$
$$U_{ji} = \begin{cases} 0 & \text{if user } i \text{ is not from continent } j \\ 1 & \text{if user } i \text{ is from continent } j \end{cases} \tag{5.13}$$

and $1 \leq i \leq C$ ranges over all the continents and $1 \leq i \leq U$ over all the users. In figure 5.15 we note that most authors either use the email address from the international organizations they are affiliated to or they use an email providers such as *gmail.com* with the number authors origination from Europe coming in as a close second. Since we do know the actual nationality of the authors corresponding to this fictitious continent, *OP*, we omit drawing conclusions from this continent. We ignore the data from the African continent, *AF*, as it has a sample size of just 2 paper. In figure 5.15 it also appears that while the number of authors from the different continents does strongly deviate the percentage of papers that are accepted does keep ranging between 20% and 40% with no particular pattern in sight. Since most people originate from the factitious continent Europe and there is no relation between the origin of the authors of a paper and its probability we therefore confirm our hypothesis.

How many *reviewers* from the paper are from a specific continent?

This feature is similar to the feature that evaluates if there is a relationship between the origin of an author and the probability of his paper to be accepted into the conference. Only now we want to determine if there is a relationship between the origin of a reviewer and the likeliness that a paper he has reviewed is accepted into the conference. This feature is also calculated similarly to the

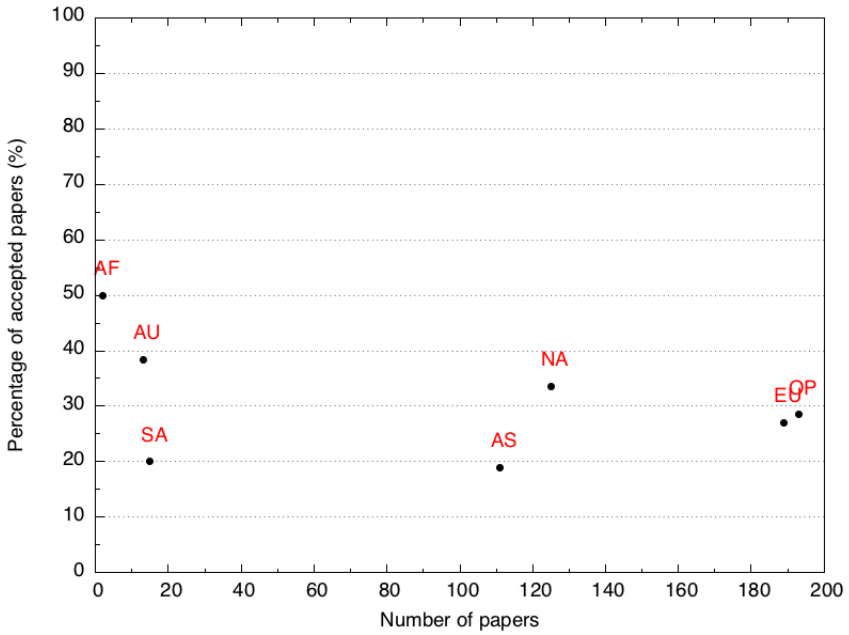

Figure 5.15: Here we plot the total number of authors originating for a continent for each paper and also plot the percentage of author for each paper where the paper is accepted into the continent.

feature that evaluates if there is a relationship between the origin of an author and the probability of his paper to be accepted into the conference only know we get the email address from *reviewers_ assignments_ by_ paper*. We hypothesize that most reviewer are from Europe. We do not hypothesize however that there the origin of the reviewer and the likeliness that a paper he has reviewed is accepted into the conference. Using this feature, we evaluate this hypothesis.

In figure 5.16 we note a somewhat similar pattern as figure 5.15. Yet again, Europe is the continent with the largest number of authors and here the result from the African continent, *AF*, since it is not the identity of any reviewer in the conference. The percentage of papers accepted into the conference is also in the range between 20% and 40% with no particular pattern in sight. With its relatively small sample size the deviating result for South-Asia is more of the exception than the rule we. We can therefore confirm our hypothesis.

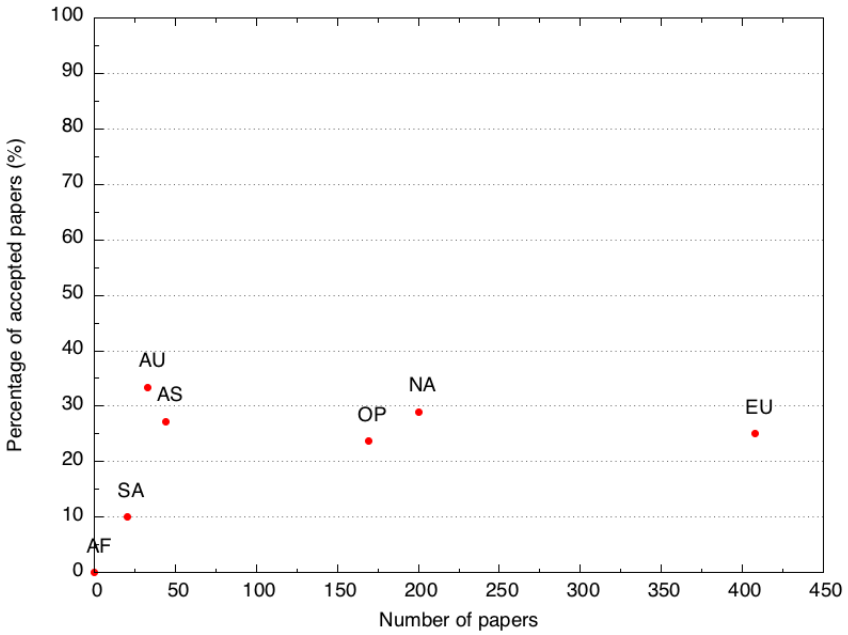

Figure 5.16: Here we plot the total number of reviewers originating for a continent for each paper and also plot the percentage of reviewer for each paper where the paper is accepted into the continent.

5.5 Mining the indirect features

Now that we have gathered additional features to create our predictive model we repeat our previous attempts to create such a model with WEKA. We will use both the direct and indirect features as our dataset, while our target feature, classifiers and baseline remain the same.

From the four rightmost columns in table 5.1 we note that the results of the predictive model created by our baseline remains consistent. We expected such behaviour from our baseline as it ignores every feature, except for the target feature which has remained consistent throughout these experiments. For the results of our predictive models for the CCI metric we note most of these results remain consistent, while Logistic, IBk, J48 and REPTree notably get worse results. This is possibly due to the fact that some of indirect features are derivatives from the direct features that could an increase in overfitting to these particular features. For KS we do not note consistent performance changes for the results of the predictive models. There is a not a consist improvement or deterioration in the results for the dataset with both the direct and indirect features in comparison to the results for just the direct features dataset. However, since the changes in the predictions are at most of 0.03 these are not significant

enough to be taken into consideration, regardless of them being improvements of deteriorations. With the F-Measure metric we note similar behaviour in comparison to the results gathered for KS. Finally, we also note for the ROC area metric that the predictive models for some classifiers have improved, while for others it has decreased Our best ROC area still remains 0.53 which still is not a significant enough improvement over the ROC area of our baseline.

A possible explanation for our incapability to predict if a paper is accepted or rejected is due to the way we have gathered our data: we are given the data from *ECMLPKKD 2013* by one of its PC Chairs. As each member of the conference is (at least) restricted from accessing the data from its peers our dataset does not include data regarding the PC Chairs. While the PC Chairs do carry the final responsibility to accept or reject a paper into conference the score given by the reviewers of a paper is just used as advise. Since the PC Chairs are able to access the reviews of the reviewers of a paper it is more likely that we are capable to predict what average score they give a paper. It is important to note that in the conference a paper is given one of the four predefined scores: 0, 1, 2 and 3, by joint decision between its metareviewer and reviewers. However, since this is another poorly documented decision we have decided to predict the average score of the reviewers of a paper, since we have the corresponding reviews for the score of a reviewer available. Our target feature therefore becomes:

What is the average score of a *paper* given by its metareviewer and reviewers?

Each paper is individually scored by its reviewers with a (integer) score between 0 and 3. Here a score given by the reviewer closer to 3 is considered better. We gather the data for this feature by averaging every score for our paper from the *review_judgment* table. Mathematically, this feature can be denoted as:

$$F_p = \frac{\sum_{i=0}^{J} Jv_{ip}}{\sum_{i=0}^{J} J_{ip}} \tag{5.14}$$

$$Jv_{ip} = \begin{cases} 0 & \text{if judgment } i \text{ is not a judgment on paper } p \\ \textit{value of judgment} & \text{if judgment } i \text{ is a judgment on paper } p \end{cases}$$

$$J_{ip} = \begin{cases} 0 & \text{if judgment } i \text{ is not a judgment on paper } p \\ 1 & \text{if judgment } i \text{ is a judgment on paper } p \end{cases}$$

and $1 \leq i \leq J$ ranges over all the *judgment scores* in the database. Of course, it is important to evaluate the relation between the average score of its reviewers on a paper and whether it is accepted or rejected in figure 5.17 we plot this relation. From this figure we note a steep increase in the number of accepted as the average score of the reviewers of a paper increases from 1.5 to 2.5. For each

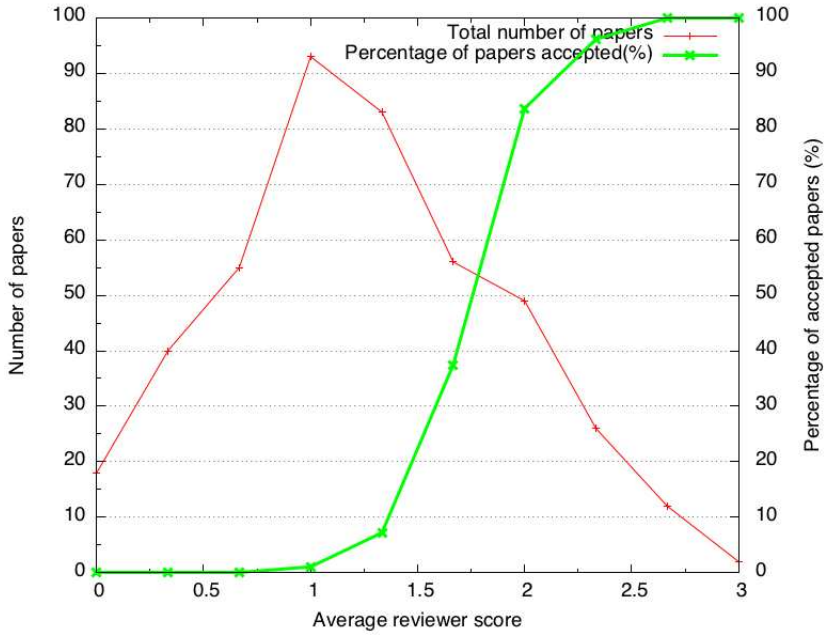

Figure 5.17: Here we plot the total number of authors originating for a continent for each paper and also plot the percentage of author for each paper where the paper is accepted into the continent.

paper with a score higher than 2.5 we note that they are all accepted. Here we also note that the number of papers grows as the score of the reviewers increases up until a score of 1 from there on the number of papers accompanying the score decreases. So from figure we note that there is a clear relation between the score of the reviewers of a paper and its probability to be accepted into the conference.

We also create our predictive models for this target feature with WEKA. Here our datasets remain consistent; we have a dataset with just direct features and a dataset with both direct and indirect features, while we keep as much as possible of the same classifiers. We do replace Logistic with *LinearRegression*, Voted-Perceptron with *MultiPerceptron* and J48 with *M5P* as these classifiers have proven to be incompatible with regression. We measure our predictive models for regression with the *Correlationo Coefficient (CC)*. With CC we measure the strength and direction of a linear relationship between two variables. Here our two variables are the average score of the reviewers of a paper we predict versus its actual values. Generally, a CC closer to 1 is considered very good, while in machine learning this is a CC of 0.40 is already considered very good. CC is

Classifiers	Direct features			Direct and Indirect features		
	CC	MAE	RMSE	CC	MAE	RMSE
ZeroR	0.00	**0.54**	**0.67**	0.00	**0.54**	**0.67**
LinearRegression	**0.21**	0.56	0.71	0.15	0.61	0.82
MultiPerceptron	0.18	0.79	0.99	**0.22**	0.73	0.95
IBk	0.01	0.69	0.84	0.12	0.71	0.87
KStar	0.00	1.21	1.37	0.00	1.21	1.38v
ConjunctiveRule	0.07	0.56	0.70	0.06	0.57	0.70
DecisionTable	0.12	0.55	0.68	0.11	0.56	0.69
M5P	0.12	**0.54**	0.69	0.17	0.58	0.73
REPTree	0.00	**0.54**	**0.67**	0.00	**0.54**	**0.67**

Table 5.2: The three most left columns of this table contain our results for regression with only the direct features, while the three most right columns contain the results for the data set with both direct and indirect features.

defined as follows:

$$CC = \frac{\sum_n^{i=1}(p_i - p)(a_i - a)}{\sqrt{\sum_n^{i=1}(p_i - p)^2 \sum_n^{i=1}(a_i - a)^2}} \tag{5.15}$$

p_i is the predicted value for data instance i

a_i is the actual value for data instance i

p is the average of the predicted values

a is the average of the actual values

Since we are evaluating several pairs numerical values it is interesting to know if we have a large number of small errors or a few big errors. By calculating the **Mean Absolute Error (MAE)** and **Root Mean Squared Error (RMSE** we want to evaluate such relation. MAE and RMSE are defined as:

$$MAE = \sum_{i=1}^{n} |p_i - a_i| \tag{5.16}$$

$$RMSE = \sqrt{\sum_{i=1}^{n} |p_i - a_i|^2} \tag{5.17}$$

With RMSE giving a greater weight to large numerical differences we note that as difference between these metrics increases the number of errors decreases, but the size of these errors increases. For both of these metrics a value closer to 0 is considered better.

From table 5.2 we note for the dataset with just the direct features, the three leftmost columns, that the CC of our baseline is 0.00. This was a predictable

value, since our target feature can be any value in a range of continuous numerical values, while our baseline still sets its predictions to the majority value in the training data. For CC we note that our best possible value is 0.21 of the predictive model created with LinearRegression. As we have previously noted a CC of 0.40 is considered very good in machine learning with our CC being half we still do think that our predictive models is to accurately predict what the average score of the reviewers for a paper will be. For MAE we note that none of our predictive models are better than our baseline as is also the case for RMSE. For the relation between MAE and RMSE we note that these relations have roughly the same difference for each classifier. These differences do not suggest that we have some big or a lot of small errors, but they are more nuanced. Comparing the results for our dataset with just direct features and for our dataset with both direct and indirect features does not suggest any major differences or anomalies. Some classifiers perform better with the additional features, while others perform worse. However, these differences are not big enough to be considered interesting.

5.5.1 AUTOweka

From table 5.1 we note that we are not able to make more accurate predictions than our baseline as to predicting if a paper will be accepted or rejected. From table 5.2 we make a similar observation only now concerning the average score of a paper given by its reviewers. For both of these cases we used our classifiers with their default settings. However, in [13] empirical research has shown that the optimization of the parameters of a classifier can reduce its test error rate up to 15%. As manual optimization of these parameters often has researchers use their intuition the reproduction of their research is jeopardized. We will therefore use *autoWEKA*. autoWEKA is an extension to WEKA that allows for automatic classifier selection; we select the most accurate classifier for our data mining problem, and parameter optimization. We formally define classifier selection as:

$$\alpha^* \in \min_{\alpha \in A} \frac{1}{k} \sum_{i=1}^{k} L(\alpha, D_{train}^{(i)}, D_{test}^{(i)}) \tag{5.18}$$

In equation 5.18 our dataset is split into k equal-sized partition we use each of these partitions exactly once as testing data, D_{test}, while the other, $k-1$, partitions are used as training data, D_{train}. So, if k is 10 we run our learn and test our classifier ten times, also called *folds*. We call this process of separating, learning and testing our classifier multiple times over the same dataset **cross-validation**. During each fold i we calculate the *loss* function, $\mathcal{L}$. The values of the loss functions are finally averaged for each classifier and these average values are finally compared to select the classifier with the lowest average loss function. This classifier is the most optimal classifier for our data mining problem.

$$\lambda^* \in \min_{\lambda \in \Lambda} \frac{1}{k} \sum_{i=1}^{k} L(\alpha_\lambda, D_{train}^{(i)}, D_{test}^{(i)}) \tag{5.19}$$

Similarly, we need to calculate the most optimal values for our parameters, Λ, of our classifier α. This is formally defined in equation 5.19. Since our goal with autoWEKA is to find our most optimal classifier with its most optimal parameter settings for our data mining problem we need to combine these equations. The combined equations are stated in equation 5.20.

$$\alpha^*\lambda^* \in \min_{\alpha^{(j)} \in A, \lambda \in \Lambda^{(j)}} \frac{1}{k} \sum_{i=1}^{k} L(\alpha_\lambda^{(j)}, D_{train}^{(i)}, D_{test}^{(i)}) \tag{5.20}$$

autoWEKA uses the Bayesian Optimization [18], and in particular the *Sequential Model-based Algorithm Con- figuration (SMAC)* from **Sequential Model Based Optimization (SMBO)** [17] methods provided in autoWEKA to solve the problem of finding the most optimal classifier with its most optimal parameter settings. SMBO, outlined in Algorithm 1, first builds a model $\mathcal{M}_\mathcal{L}$ that captured the dependence of the loss function $\mathcal{L}$ on the parameter settings, λ (line 1 of Algorithm 1). Subsequently, SMBO iterates as long as the time budget has not been exhausted with the $\mathcal{M}_\mathcal{L}$ in line 3 it determines a candidate configuration of the parameters λ to evaluate its loss function (line 4) and update the model with a new data point (λ, c) (lines 5 - 6). Each parameter setting that is not optimized retains its default value.

Algorithm 1 SMBO

1: initialise model $\mathcal{M}_L$; $\mathcal{H} \leftarrow \emptyset$
2: **while** time budget for optimization has not been exhausted **do**
3: $\quad \boldsymbol{\lambda} \leftarrow$ candidate configuration from $\mathcal{M}_L$
4: $\quad$ Compute $c = \mathcal{L}(A_{\boldsymbol{\lambda}}, \mathcal{D}_{\text{train}}^{(i)}, \mathcal{D}_{\text{valid}}^{(i)})$
5: $\quad \mathcal{H} \leftarrow \mathcal{H} \cup \{(\boldsymbol{\lambda}, c)\}$
6: $\quad$ Update $\mathcal{M}_L$ given $\mathcal{H}$
7: **end while**
8: **return** $\boldsymbol{\lambda}$ from $\mathcal{H}$ with minimal c

Using autoWEKA can be broken down into several steps. First, we define our training and testing data. autoWEKA subsequently evaluates the data and suggests a set of compatible classifiers from which the user is prompted to select the classifiers he wants to use. Finally, we define our experiment settings for autoWEKA.

Our experiment settings includes defining our success measure for our classifier and its parameter settings. Here we choose CCI for classification and RMSE for regression. We set the maximum timeout to find the most optimal model for our autoWEKA experiment with *optimization timeout(hours)*, its time-out to optimize the parameters of a single classifier with *training run timeout (minutes)* and we set its maximum memory usage with the *training memory limit*

Experiments	Problem		Features		Classifiers	
	Classification	Regresssion	Direct	Indirect	Subset[1]	All[2]
c_sel_di	X		X		X	
c_sel_indi	X		X	X	X	
c_all_di	X		X			X
c_all_indi	X		X	X		X
r_sel_di		X	X		X	
r_sel_indi		X	X	X	X	
r_all_di		X	X			X
r_all_indi		X	X	X		X

Table 5.3: Here we list the properties for each of our experiments with autoWEKA. Here [1] represents the set of classifiers as specified in table 5.1 and 5.2 for classification and regression, respectfully. [2] represents all available classifiers in (AUTO)WEKA, except for our baseline.

Experiments	CCI	KS	F-measure	ROC area	Classifier (parameters)
c_sel_di_1hrs	72.97	-0.09	0.66	0.48	IBk -E -K 12 -I
c_sel_indi_1hrs	76.58	-0.03	0.68	0.46	IBk -K 10 -X -I
c_all_di_1hrs	**78.38**	0.00	0.69	0.53	AttributeSelectedClassifier -E GainRatioAttributeEval IBk – -K 30 -I
c_all_indi_1hrs	**78.38**	0.00	0.69	0.50	REPTree -M 22 -V 3.36E-5 -L 19
c_sel_di_2hrs	72.97	-0.09	0.66	0.48	IBk -K 12 -I
c_sel_indi_2hrs	76.58	-0.03	0.68	0.46	IBk -E -K 11 -X -I
c_all_di_2hrs	72.97	-0.09	0.66	0.48	IBk -K 46 -X - I
c_all_indi_2hrs	76.58	**0.05**	**0.71**	0.53	RandomTree -M 61 -K 0 -depth 17 -N 0 -U
c_sel_di_4hrs	72.97	-0.09	0.66	0.48	IBk -E -K 12 -I
c_sel_indi_4hrs	76.58	-0.03	0.68	0.46	IBk -E -K 12 -X -I
c_all_di_4hrs	**78.38**	0.00	0.69	**0.57**	IBk -K 53 -I
c_all_indi_4hrs	77.47	-0.08	0.68	0.51	RandomForest -I 245 -K 1 -depth 0

Table 5.4: Experiments for classification with autoWEKA.

(MB).

With autoWEKA we repeat each experiment we have previously performed with WEKA and we also repeat each of these experiment without a limitation of the usable classifiers as we have previously done for our experiments WEKA, except for ZeroR as we do not want our baseline to be inadvertently found as our optimal classifier. After all, autoWEKA should automatically find the most optimal classifiers with its most optimal parameter settings. We list these experiments with their properties in table 5.3. As for our experiment settings we set our optimization timeout to 1, 2 and 4 hours with the assumption that any possible relation detected with the first two optimization timeouts can be confirmed with the third optimization timeout. With the notion our training run timeout is directly dependent from the optimization timeout we set it to 5, 10 and 20 minutes, respectively. We set our training memory limit for each experiment to 1000 MB. We run all our experiments on a Intel(R) CORE(R) CPU I7-860 with 6GB RAM.

From table 5.4 we note that we are not able to create none of the experiments we have defined in table 5.3 to run with autoWEKA are capable of creating a more accurate data model than our baseline. From table 5.4 we note that our maximum CCI is equal to the CCI of our baseline with a value of 78.38. Furthermore, we note for KS that our results are generally negative which indicates that the data models we have created with autoWEKA are worse than simply random *guessing* whether a paper is going to be accepted or rejected from the conference. Our highest KS of 0.05 is not even as good as the highest KS value, 0.08, for WEKA which we got for the predictive model created with just direct features and REPTree classifier. The F-Measure does not display any curious behaviour with values ranging from 0.66 to 0.71 without any particular pattern for these values as the time constraints change. Here the maximum F-Measure does show a slight improvement over our maximum F-Measure of 0.70 for WEKA, but with an improvement of just 1% this is improvement is not significant enough to warrant further investigation. For the ROC area also note a slight improvement when comparing the experiments run with autoWEKA (0.57) to those run with WEKA (0.53). However, since this is also just an improvement of 8% over our maximal ROC area for WEKA and therefore still does not show any significant improvement over our ROC area with regard to our baseline this improvement also does not warrant any further investigation.

With the regard to the chosen classifiers and their most optimal parameters as found in table 5.4 we note that the experiments with autoWEKA where we are limited to the selection classifiers as used with WEKA we continuously find IBk as the most optimal classifier. Compared with the results of this classifiers in table 5.1 for both kinds of datasets we note that the results from IBk with optimized parameter show a big improvement over the results with this classifier with WEKA. However, it is still strange that all of our experiments with autoWEKA give results at least as good as our baseline, since VotedPerceptron gives results as good as baseline and it is also select able in all our experiments with autoWEKA. We note that the results for this classifier stay the same regardless of its given optimization time. IBk does seem to find different optimal parameters with different time constraints, but these do not influence its results. Examples of these optimizations include, but are not limited to, varying the number of neighbours (kNN) between 10 and 12 and using or not using cross-validation. None of these optimization are performed with a specific recognizable pattern. For the experiments where we are not limited in our selection of classifiers we also note that IBk has been selected as the most optimal classifier where we use just the direct feature as our dataset and our optimization time is 2 and 4 hours. For the other experiments where we are not limited to a certain selection of classifiers we do note any particular pattern for the most optimal classifier aside from the observation that three out of four classifier are from the *trees* class of classifiers, while one classifier originates from the *meta* class of classifiers. From our results for our experiments as listed in table 5.4 with autoWEKA for the three different time constraints; 1, 2 and 4 hours, we can also conclude that giving autoWEKA more time to optimize does not give

Experiments	CC	MAE	RMSE	Classifier (parameters)
r_sel_di_1hrs	0.19	**0.53**	0.66	IBk -K 48 -F
r_sel_indi_1hrs	0.00	0.55	0.66	MultilayerPerceptron -L 0.99 -M 0.62 -B -H i -C -R -D -S 1
r_all_di_1hrs	0.06	0.54	0.68	DecisionStump
r_all_indi_1hrs	0.00	0.54	0.67	MultilayerPerceptron -L 0.99 -M 0.19 -H i -C -D -S 1
r_sel_di_2hrs	0.21	**0.53**	**0.65**	IBk -E -K 47 -I
r_sel_indi_2hrs	0.24	**0.53**	**0.65**	IBk -K 58 -F
r_all_di_2hrs	0.11	0.54	0.68	RandomSubSpace -I 2 -P 0.33 -S 1 -W M5P – -M 1
r_all_indi_2hrs	0.19	0.56	0.72	M5P -M 3 -U
r_sel_di_4hrs	0.19	**0.53**	0.66	IBk -K 48 -X -F
r_sel_indi_4hrs	0.22	**0.53**	0.66	IBk -E -K 61 -F
r_all_di_4hrs	0.06	0.54	0.68	DecisionStump
r_all_indi_4hrs	**0.25**	**0.53**	**0.65**	RandomSubSpace -I 57 -P 0.24 -S 1 -W M5P – -M 1

Table 5.5: Experiments for regression with autoWEKA.

us more accurate predictive models for our data mining problem: to predict if a paper is going to be accepted into the conference. In table 5.5 we note that the autoWEKA experiment run with the direct and indirect features as our dataset without any restrictions on our selection of classifiers and a duration of 4 hours results in a CC of 0.25. This is an improvement of 14% over highest CC we have gotten for our experiments with WEKA where our most accurate predictive model was created with MultiPerceptron and direct and indirect features as our dataset. However, even if our maximum CC with autoWEKA is higher than our maximum CC with WEKA it still is nowhere near 0.40 which is the required CC for a predictive model to be considered accurate in machine learning. For the other CC values for our experiments with autoWEKA we note that different values are scattered over different experiments without any particular pattern with regard to the available selection of classifier, features or optimization time. Looking at the values for MAE we note that these range between being a 2% improvement to lowest MAE in table 5.2 to at most performing 4% worse than the lowest MAE in table 5.2. For RMSE we note roughly similar behaviour. More importantly, as expected the relationship between MAE and RMSE remains the same: indicating that we do not have a few big errors or a lot of small errors, but that the deviation between the predictions and actual values are more nuanced.

For our selection of classifiers in table 5.5 we note that most experiments where our selection of classifiers is limited to the classifiers used with WEKA, the classifiers listed in table 5.2, have IBk as their most optimal classifier. Here the predictive models created with IBk with optimized parameters show great improvement in comparison to the results of the predictive models it created as listed in table 5.2 with CC values ranging from 0.19 to 0.24, MAE of 0.53 and RMSE from 0.65 to 0.66. So, for CC we note very high values and for MAE and RMSE we note some of the best values listed in table 5.5 for these metrics with the listed experiments in this table. Again, we note variation in

the number of evaluated nearest neighbours (*kNN*), use of cross-validation and so on. These optimization do not seem to be performed with explicit pattern. *r_sel_indi_1hrs*uses MultilayerPerceptron autoWEKA's choice for this classifier does seem to be strange as it results in a predictive model with a CC of 0.00. Furthermore, this classifier is also chosen for *r_all_indi_1hrs* where we note roughly similar results for the CC, MAE and RMSE as *r_sel_indi_1hrs*. For *r_all_di_1hrs* and *r_all_di_4hrs* we note that autoWEKA has chosen DecisionStump as the most optimal classifier for these experiments, while the MAE and RMSE values are acceptable for the corresponding predicitve model it creates. A CC of 0.06 is rather low which makes us question the optimization process for autoWEKA for these experiments. Finally, autoWEKA has chosen RandomSubSpace as the optimal classifier for the *r_all_di_2hrs* and *r_all_indi_4hrs* experiments, while M5P is chosen as the most optimal classifier for *r_all_indi_2hrs*. We do not notice any patterns in these choices. All-in-all, we see that we are not able to create a predicitve model capable of predicting what the score given by the reviewers of our papers will be. In table 5.5 we also note that increasing the optimization time does not guarantee to improve the accuracy of the predictive model that is created for that particular autoWEKA experiment.

Chapter 6

Conclusions

In this work, we have given empirical proof for some features that certain values for these features improve the probability of these papers to be accepted into the *ECMLPKDD 2013*. With this thesis we have shown also that we are unable to make more accurate predictive models than our baseline, ZeroR, from knowledge surrounding a paper without looking at its contents. Initially, we wanted to predict with our predictive model if a paper is accepted or rejected from the conference. However, as further evaluation of the model showed us incapable of making such a prediction we wanted to predict the average score a paper is given by his reviewers. We were also incapable of making these predictions.

We made our predictive models by converting the separate files of data we had received from *ECMLPKDD 2013* into a single relational database. The data in this relational database was subsequently used to extract general characteristics from the papers in the conference. These general characteristics were then mined with WEKA with the two most popular classifiers from each compatible class of classifiers. We used these classifiers with their default parameters and did not use any attribute selection algorithms. Since we did not yield better predictions for both classification; can we predict if a paper is accepted or rejected from the conference, as regression; can we predict the average score a paper is given by his reviewers. We repeated and expanded our effort to make these predictions with autoWEKA, since the argument in [13] was made that predictive models created with classifiers with optimized parameters yield better results and this tool does exactly that; optimizing the parameters of classifier and if we have multiple classifiers selecting the most optimal classifier for the evaluated data mining problem. None of the experiments we ran with autoWEKA whether it was classification or regression, with or without the indirect features, with a selection of classifiers or with all the classifiers or ran for 1, 2 or 4 hours none of them yielded significantly better results than our baseline.

For future work, we test our data model for conference over multiple years, since *ECMLPKDD* is an annual conference and we could then confirm if this

data model can predict the average reviewer score of a paper with great accuracy. We can also take the textual characteristics of a paper into account when we try to make our prediction for such a paper. Finally, we could also perform a more elaborate research on the social network regarding this conference.

Appendices

Appendix A

ER diagram

In this chapter we list our ER diagram:

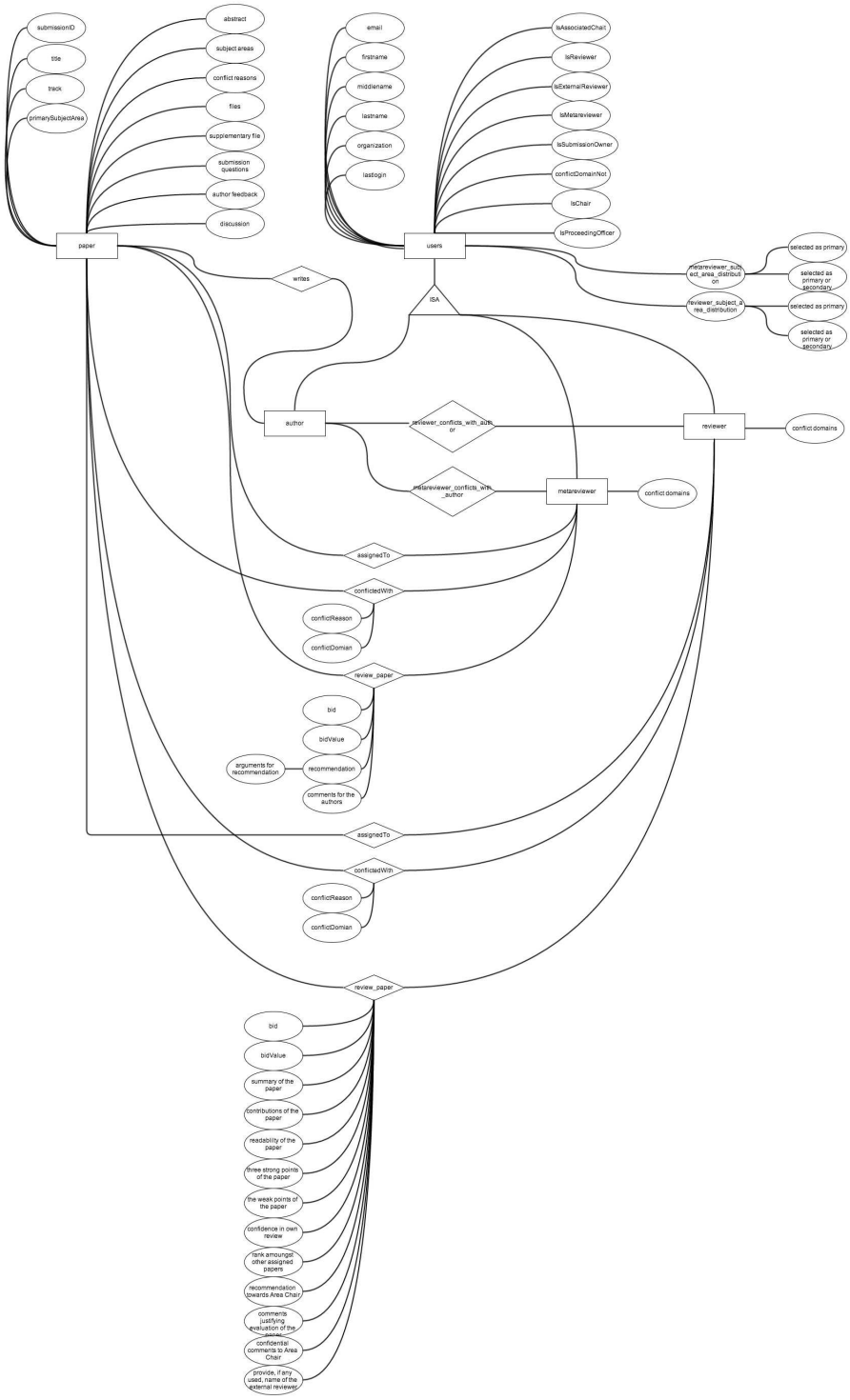

Figure A.1: Here we plot our ER diagram as described in chapter 4.

Appendix B

Table with features

Feature	min	max	mean	std	corr [1]	corr [2]
amount_of_authors	1.0	8.0	2.964	1.266	0.143	0.179
amount_of_bids	11.0	81.0	39.077	8.481	-0.019	-0.032
average_score_of_bids	0.0	2.704	1.169	0.572	-0.027	-0.035
primary_subject_area	'Active Learning'	Clustering	$N \backslash A^3$	$N \backslash A^3$	$N \backslash A^3$	$N \backslash A^3$
amount_of_tagged_subject_areas	1.0	8.0	2.651	1.314	-0.016	-0.038
Active_Learning	0.0	1.0	0.103	0.303	-0.022	-0.013
Association_Rules	0.0	1.0	0.098	0.297	-0.032	-0.09
Bayesian_Learning	0.0	1.0	0.073	0.26	0.12	0.1
Bioinformatics_and_Genomics	0.0	1.0	0.187	0.39	0.002	0.051
Biological_Network_Mining	0.0	1.0	0.055	0.227	-0.047	-0.115
Classification	0.0	1.0	0.1	0.3	-0.018	0.01
Classifier_Evaluation	0.0	1.0	0.059	0.236	-0.039	-0.033

Clinical_ and_ Medical_ Data_ Mining	0.0	1.0	0.296	0.457	-0.102	-0.064
Clustering	0.0	1.0	0.034	0.182	0.007	-0.042
Computational_ Learning_ Theory	0.0	1.0	0.096	0.294	-0.045	-0.032
Cost-Sensitive_ Learning	0.0	1.0	0.096	0.294	-0.045	-0.032
Data_ Mining_ Case_ Studies	0.0	1.0	0.014	0.116	0.112	0.123
Data_ Mining_ Theory_ and_ Foundations	0.0	1.0	0.084	0.278	-0.043	-0.032
Data_ Streams	0.0	1.0	0.048	0.213	0.083	0.114
Dimensionality_ Reduction	0.0	1.0	0.075	0.264	0.107	0.057
Feature_ Selection_ and_ Extraction	0.0	1.0	0.128	0.334	-0.067	-0.075
Ensemble_ Methods	0.0	1.0	0.071	0.256	-0.016	0.015
Frequent_ Sets_ and_ Patterns	0.0	1.0	0.064	0.244	-0.044	-0.017
Graph_ and_ Tree_ Mining	0.0	1.0	0.08	0.271	0.023	0.012
Graphical_ Models	0.0	1.0	0.03	0.17	0.023	-0.043
Inductive_ Logic_ Programming	0.0	1.0	0.068	0.252	-0.053	-0.079
Kernel_ Methods	0.0	1.0	0.034	0.182	0.007	-0.071
Link_ Mining	0.0	1.0	0.021	0.142	0.101	0.096
Matrix_ and_ Tensor_ Analysis	0.0	1.0	0.05	0.218	-0.061	-0.064
Multi-Relational_ Mining_ and_ Learning	0.0	1.0	0.052	0.223	0.029	0.023
Multi-Task_ Learning	0.0	1.0	0.025	0.156	-0.059	-0.011
Natural_ Language_ Processing	0.0	1.0	0.025	0.156	0.008	0.033
None_ of_ the_ above	0.0	1.0	0.032	0.176	-0.045	-0.078
Rankings_ and_ Partial_ Orders	0.0	1.0	0.073	0.26	-0.045	-0.103
Recommender_ Systems	0.0	1.0	0.027	0.163	-0.033	-0.016
Reinforcement_ Learning	0.0	1.0	0.023	0.149	-0.088	-0.083
Rules_ and_ Trees	0.0	1.0	0.089	0.285	0.059	0.105
Semi-Supervised_ and_ Transductive_ Learning	0.0	1.0	0.052	0.223	0.005	0.006
Social_ Network_ Mining	0.0	1.0	0.036	0.187	0.027	-0.004
Statistical_ Methods	0.0	1.0	0.018	0.134	0.039	0.033

Structured_ Data	0.0	1.0	0.043	0.203	0.006	-0.02
Structured_ Output_ Prediction	0.0	1.0	0.007	0.082	0.079	0.041
Subgroup_ Discovery	0.0	1.0	0.018	0.134	0.078	0.0
Supervised_ Learning	0.0	1.0	0.064	0.244	0.064	0.044
Text_ Mining_ and_ Information_ Retrieval	0.0	1.0	0.041	0.198	-0.014	-0.013
Time_ Series_ and_ Temporal_ Data_ Mining	0.0	1.0	0.011	0.106	0.135	0.154
Transfer_ Learning	0.0	1.0	0.016	0.125	0.01	0.033
Unsupervised_ Learning	0.0	1.0	0.007	0.082	0.016	0.087
Visualization_ and_ Visual_ Analytics	0.0	1.0	0.018	0.134	0.078	0.076
Web_ Mining	0.0	1.0	0.011	0.106	0.037	0.056
d_ t_ date	04/03/2013	05/02/2013	$N \backslash A^3$	$N \backslash A^3$	$N \backslash A^3$	$N \backslash A^3$
author_ is_ associate_ chair	0.0	0.5	0.002	0.029	-0.038	-0.013
author_ is_ external_ reviewer	0.0	0.5	0.002	0.034	-0.039	-0.027
author_ is_ submissionowner	1.0	1.0	1.0	0.0	$N \backslash A^4$	$N \backslash A^4$
author_ is_ chair	0.0	0.0	0.0	0.0	$N \backslash A^4$	$N \backslash A^4$
author_ is_ proceedingseditor	0.0	0.333	0.001	0.016	-0.028	0.03
metareviewer_ is_ author	0.0	1.0	0.998	0.048	0.028	-0.005
metareviewer_ is_ associate_ chair	0.0	1.0	0.036	0.187	-0.001	-0.002
metareviewer_ is_ external_ reviewer	0.0	0.0	0.0	0.0	$N \backslash A^4$	$N \backslash A^4$
metareviewer_ is_ submissionowner	0.0	1.0	0.998	0.048	0.028	-0.005
metareviewer_ is_ chair	0.0	1.0	0.692	0.461	0.002	-0.032
metareviewer_ is_ proceedingseditor	0.0	0.0	0.0	0.0	$N \backslash A^4$	$N \backslash A^4$
reviewer_ is_ author	0.0	0.0	0.0	0.0	$N \backslash A^4$	$N \backslash A^4$
reviewer_ is_ associate_ chair	1.0	1.0	1.0	0.0	$N \backslash A^4$	$N \backslash A^4$
reviewer_ is_ external_ reviewer	0.0	0.333	0.015	0.07	-0.026	0.014
reviewer_ is_ metareviewer	0.0	0.0	0.0	0.0	$N \backslash A^4$	$N \backslash A^4$
reviewer_ is_ submissionowner	0.0	1.0	0.302	0.255	-0.008	-0.02
reviewer_ is_ chair	0.0	0.25	0.001	0.012	-0.028	0.018

reviewer_is_proceedingseditor	0.0	0.25	0.001	0.012	-0.028	0.018
bidder_is_author	1.0	1.0	1.0	0.0	$N \backslash A^4$	$N \backslash A^4$
bidder_is_reviewer	1.0	1.0	1.0	0.0	$N \backslash A^4$	$N \backslash A^4$
bidder_is_associate_chair	0.0	0.091	0.005	0.012	-0.045	-0.02
bidder_is_external_reviewer	0.0	0.0	0.0	0.0	$N \backslash A^4$	$N \backslash A^4$
bidder_is_metareviewer	0.047	0.226	0.121	0.034	-0.062	-0.033
bidder_is_submissionowner	0.194	0.479	0.325	0.05	-0.092	-0.116
bidder_is_chair	0.0	0.0	0.0	0.0	$N \backslash A^4$	$N \backslash A^4$
bidder_is_proceedingseditor	0.0	0.0	0.0	0.0	$N \backslash A^4$	$N \backslash A^4$
average_amount_of_papers_assigned_to_reviewers	2.0	8.667	5.699	0.954	0.05	0.025
higher_bid_from_reviewer_than_average_bid	0.0	9.846	3.999	3.378	0.045	0.05
bids_normalized_per_amount_of_bids_bidder	0.0	0.163	0.039	0.024	0.029	0.025
bids_normalized_per_average_of_bids_bidder	0.0	3.421	0.91	0.747	-0.042	-0.039
average_pri_or_sec_popularity_subject_areas_paper	10.0	9.5	33.476	17.927	-0.089	-0.036
average_pri_popularity_subject_areas_paper	0.0	9.75	8.116	7.096	-0.076	-0.034
average_sec_popularity_subject_areas_paper	10.0	9.75	25.36	13.145	-0.081	-0.031
author_AS	0.0	1.0	0.171	0.325	-0.101	-0.156
author_EU	0.0	1.0	0.351	0.438	0.0	0.047
author_OP	0.0	1.0	0.227	0.312	-0.006	-0.05
author_NA	0.0	1.0	0.205	0.363	0.078	0.127
author_AU	0.0	1.0	0.022	0.133	0.077	0.042
author_SA	0.0	1.0	0.024	0.139	-0.035	-0.046
author_AF	0.0	0.333	0.001	0.02	0.027	0.008
reviewer_AS	0.0	0.667	0.035	0.106	0.008	0.006
reviewer_EU	0.0	1.0	0.589	0.298	-0.046	-0.101
reviewer_OP	0.0	1.0	0.149	0.208	-0.031	-0.001
reviewer_NA	0.0	1.0	0.188	0.232	0.087	0.129
reviewer_AU	0.0	0.333	0.024	0.085	0.054	0.059

reviewer_SA	0.0	0.667	0.015	0.073	-0.082	-0.077
reviewer_AF	0.0	0.0	0.0	0.0	$N\backslash A^4$	$N\backslash A^4$

Table B.1: Here we have listed each feature with their minimum , maximum, mean, standard deviation and correlation coefficient with both classification.[1] and regression[2]. For N\A we notice the following distinctions: [3] data instances are strings. [4] the standard deviation is 0.

Bibliography

[1] Advantages SQL database, http://www.cs.iit.edu/~cs561/cs425/VenkatashSQLIntro/Advantages%20&%20Disadvantages.html, 21 7 2014.

[2] Definition of the xlrd module, https://pypi.python.org/pypi/xlrd, 21 7 2014.

[3] Definition of the xml module, https://pypi.python.org/pypi/lxml-wrapper/0.4, 21 7 2014.

[4] Definition of the beautifulsoup module, https://pypi.python.org/pypi/BeautifulSoup/3.2.1, 21 7 2014.

[5] Definition of the sqlite3 module, https://docs.python.org/2/library/sqlite3.html, 21 7 2014.

[6] Definition of a module, https://docs.python.org/2/tutorial/modules.htm, 21 7 2014.

[7] ZeroR classiefier, http://www.saedsayad.com/classification.htm, 30 6 2014.

[8] WEKA homepage, http://www.cs.waikato.ac.nz/~ml/weka/, 6 6 2014.

[9] Website of the ECMLPK2013 conference, http://www.ecmlpkdd2013.org/, 20 3 2013

[10] Library of modules Python, https://pypi.python.org/pypi, 20 3 2014.

[11] Advantages of a SQL database, http://www.cs.iit.edu/~cs561/cs425/VenkatashSQLIntro/Advantages%20&%20Disadvantages.html, 15 6 2014.

[12] Description of the ARFF file, http://weka.wikispaces.com/ARFF+%28book+version%29, 10 4 2014.

[13] C. Thornton, F. Hutter, H. H. Hoos and K. Leyton-Brown, "Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms", `http://www.cs.ubc.ca/labs/beta/Projects/autoweka/papers/autoweka.pdf`, visited on 5 october 2014.

[14] "ECMLPKDD: Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases", `http://www.ecmlpkdd2013.org/`

[15] Satoshi Morinaga,Kenji Yamanishi ,Kenji Tateishi and Toshikazu Fukushima, "Mining product Reputations on the web" presented at the 8th ACM SIGKDD international conference on Knowledge discovery and data mining Edmonton,Alberta,Canada,2002

[16] Kushal Dave,Steve Lawrence ,David Pennock, "Mining the Peanut Gallary:opinion extraction and semantic classification of product reviews" presented at he 12th international conference on www Budapest,Hungary 2003

[17] F. Hutter, H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. *Proc. of LION-5*, pages 507–523, 2011.

[18] E. Brochu, V. M. Cora, and N. de Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. Technical Report UBC TR-2009-23 and arXiv:1012.2599v1, Department of Computer Science, University of British Columbia, 2009.

[19] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales",Proceeding ACL '05 Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics Pages 115-124

[20] P. D. Turney, "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews". Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 417-424.

[21] B. Liu, M. Hu and J. Sheng, "Opinion observer: analyzing and comparing opinions on the Web", Proceeding WWW '05 Proceedings of the 14th international conference on World Wide Web Pages 342-351