A Semi-Supervised Recommender System to Predict Online Job Offer Performance

Julie Séguéla[1,2] and Gilbert Saporta[1]

[1]CNAM, Cedric Lab, Paris

[2]Multiposting.fr, Paris

October 29th 2011, Beijing

Theory and Application of High-dimensional Complex and Symbolic Data Analysis

Outline

Introduction

✓ Context and objectives

✓ Recommender systems

✓ Data complexity

Methodology

✓Data handling

✓Similarity computing between job postings

✓Return estimation and system evaluation

Experiments: job board recommendation for job postings

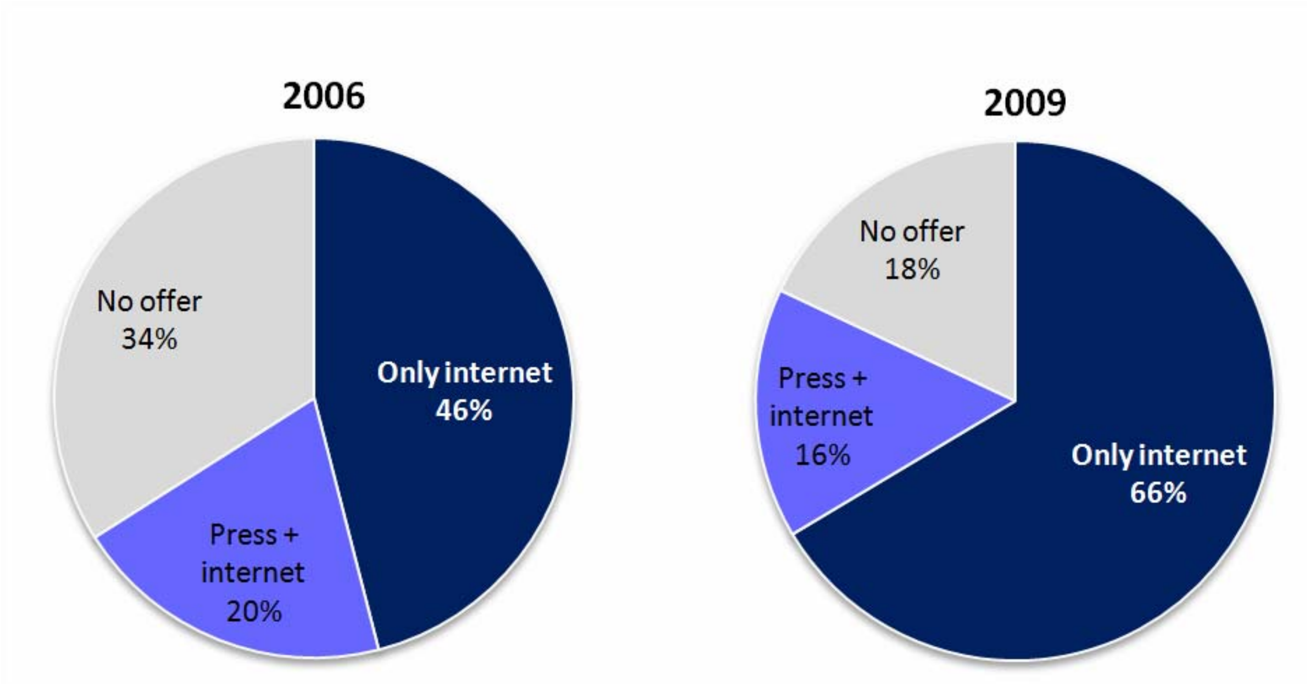
✓Data description

✓Experiments and results

Conclusions and future work

Context: Internet recruitment in France

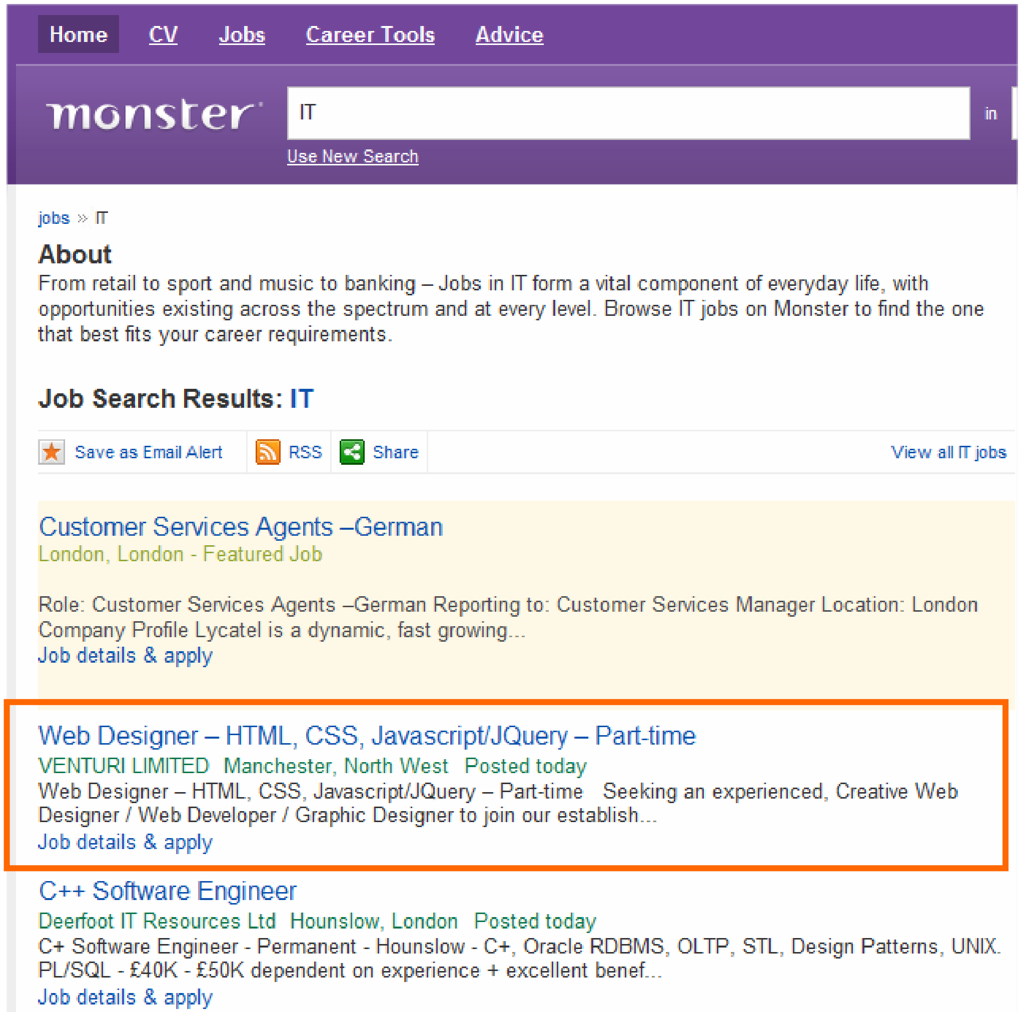
Proportion of job offers (source: APEC)



In 2009, 82% of vacancies were published on the internet (66% percent in 2006)

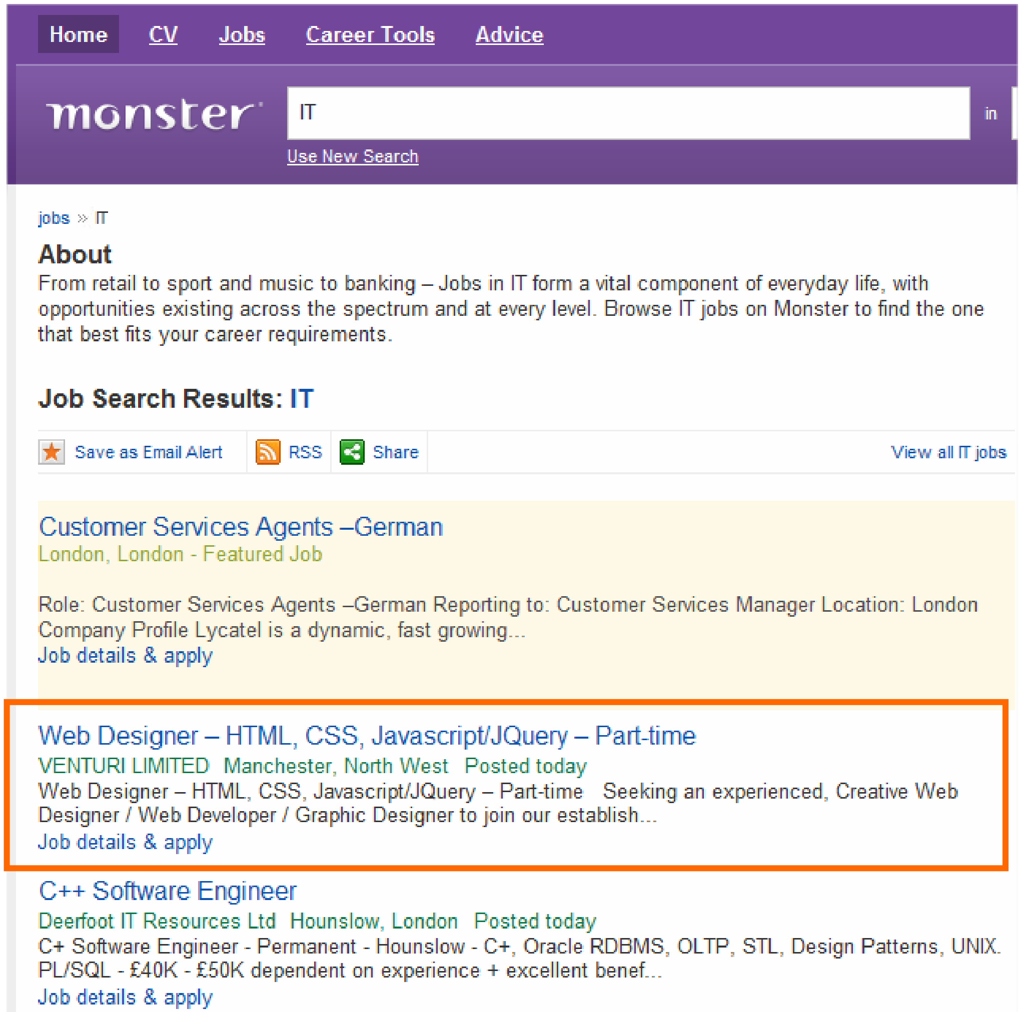
Context: A job posting on a job board

Job list



Context: A job posting on a job board

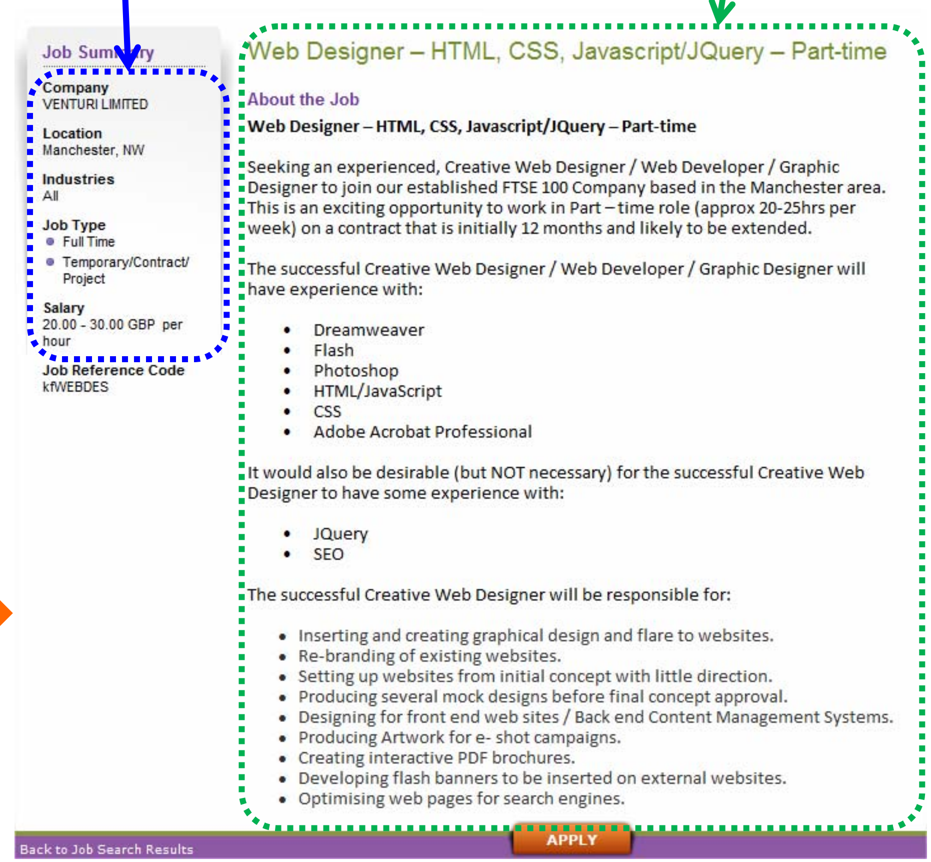
Job list



Structured data

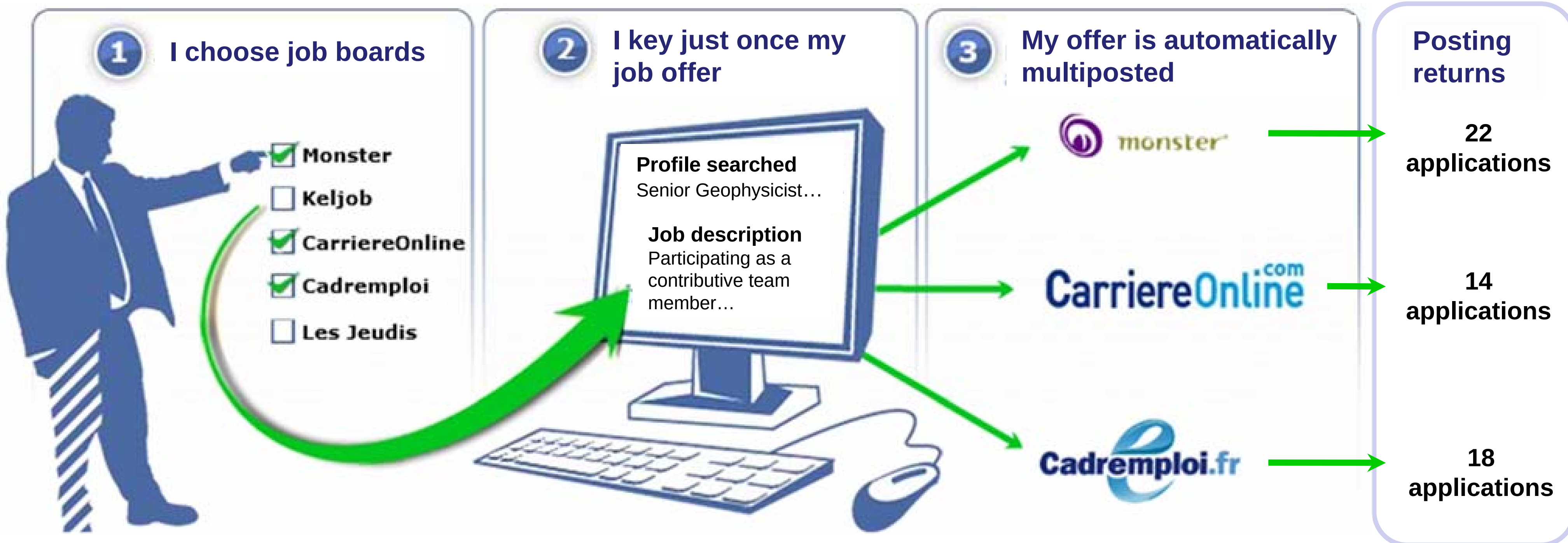
Unstructured data

Job offer



Context: Multiposting of a job offer

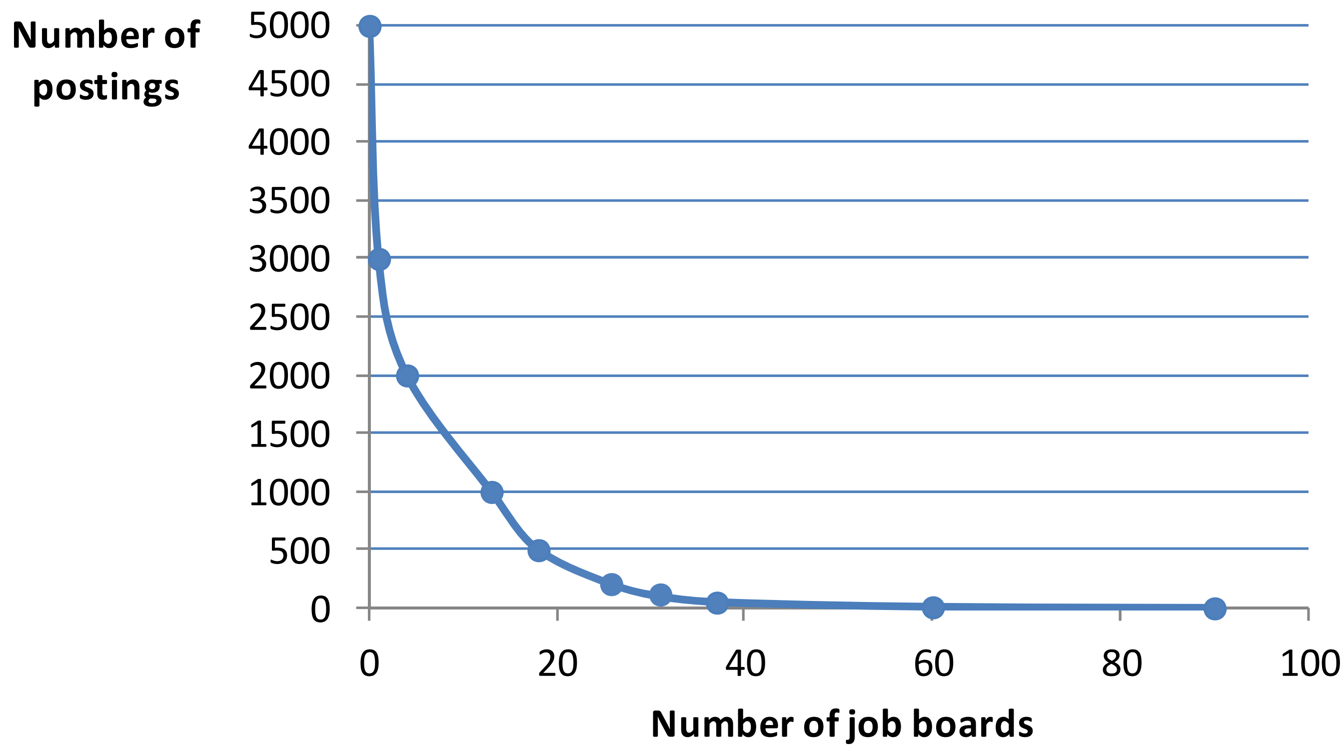
Illustration of multiposting



Our data are provided by Multiposting.fr, an online job posting solution

Context: A hundred of job boards

Number of job boards which have at least « X » postings



- Ex: 13 job boards have 1000 postings or more

Objectives

- With internet expansion, the number of potential job boards is exponentially growing
- It is now necessary to understand job board performances in order to make adequate choices when posting a job on internet
- Develop a predictive algorithm of job posting performance on a job board
- Develop an intelligent tool which recommends the best job boards according to the job offer
- We present here a recommender system predicting the ranking of job boards with respect to job posting returns

Outline

Introduction

✓ Context and objectives

✓ Recommender systems

✓ Data complexity

Methodology

✓Data handling

✓Similarity computing between job postings

✓Return estimation and system evaluation

Experiments: job board recommendation for job postings

✓Data description

✓Experiments and results

Conclusions and future work

General idea: the aim of a recommender system is to help **users** to find **items** from huge catalogues that they should appreciate and that they have not seen yet

Illustration with a movie recommender system

User	Harry Potter	The Chronicles of Narnia	Terminator	Rambo	The Lord of the Rings
Alice	**4**	**5**	1	**?**	**?**
Bob	**5**	**4**	2	1	**5**
Cindy	3	**5**	**?**	2	**4**
David	1	**?**	5	4	2

Fragment of a rating matrix

? = unknown rating

✓ What movie should be recommended to Alice?

- Bob and Cindy like the same movies as Alice
- So we should recommend to Alice an other movie that they liked:

« The Lord of the Rings »

✓ This is a **collaborative system** (based on ratings and no use of descriptive variables)

Hybrid system?

About recommender systems

Prediction are based on ratings obtained by the most similar items with respect to rating vectors → **Collaborative filtering**

Prediction are based on item features (recommends items similar to those that the user liked in the past) → **Content-based filtering**

Hybrid system
(a system which combines collaborative and content-based approaches)

Our system as a particular case of recommender system

Usual recommender objectives / issues

• Recommendation of items (= postings) to users (= job boards) according to the expected rating (= return)

• Unlimited number of potential items

• Sparse matrix: a lot of items, for each item few ratings are known

• Similarity between items is based on the ratings given by users

Our additional issues

• We are interested in predicting ratings only for « new items »: no rating, only descriptive variables

• It is not possible to obtain ratings for new items because this is a « one shot » recommendation

• Posting return is more complex than a rating (usually between 0 and 5): much variability within and between users

• We need to understand posting return variability

Outline

Introduction

- ✓ Context and objectives
- ✓ Recommender systems
- ✓ Data complexity

Methodology

- ✓Data handling
- ✓Similarity computing between job postings
- ✓Return estimation and system evaluation

Experiments: job board recommendation for job postings

- ✓Data description
- ✓Experiments and results

Conclusions and future work

Complexity of our data and issues

Which factors are relevant to explain job posting performance ?

- Identification of potential factors (job characteristics, job board, job market, etc.), coming from different sources (job offer, demographic data source, firm data, etc.)

- Use of Text mining techniques to extract relevant descriptors from the job offer

High dimensional data

- We are working with **structured** and **unstructured** data which have to be handle simultaneously

- Job postings are described by thousands of features

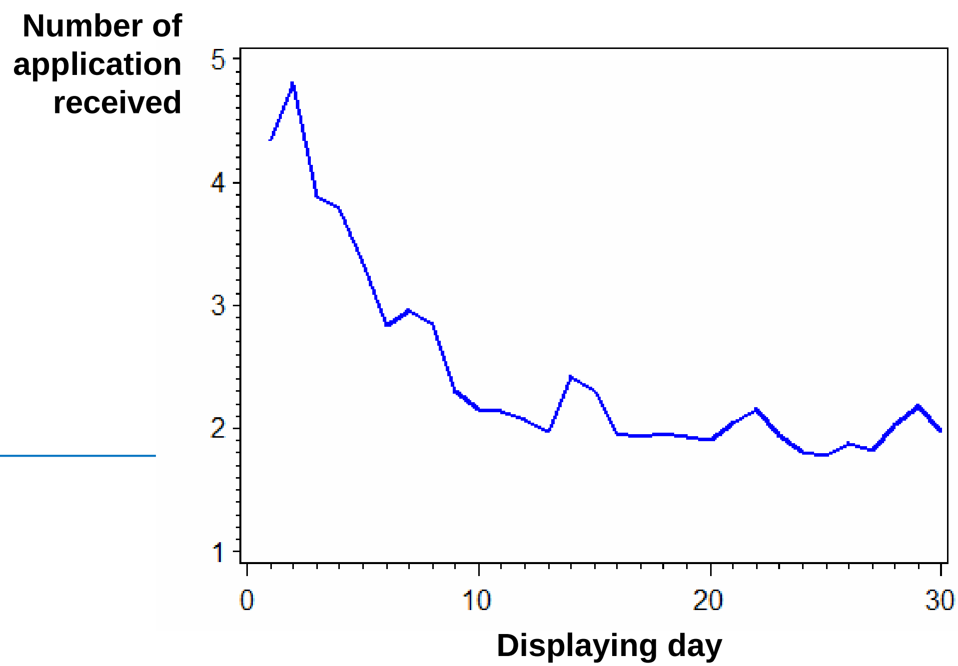
- Features have to be weighted in the algorithm according to their power of explanation

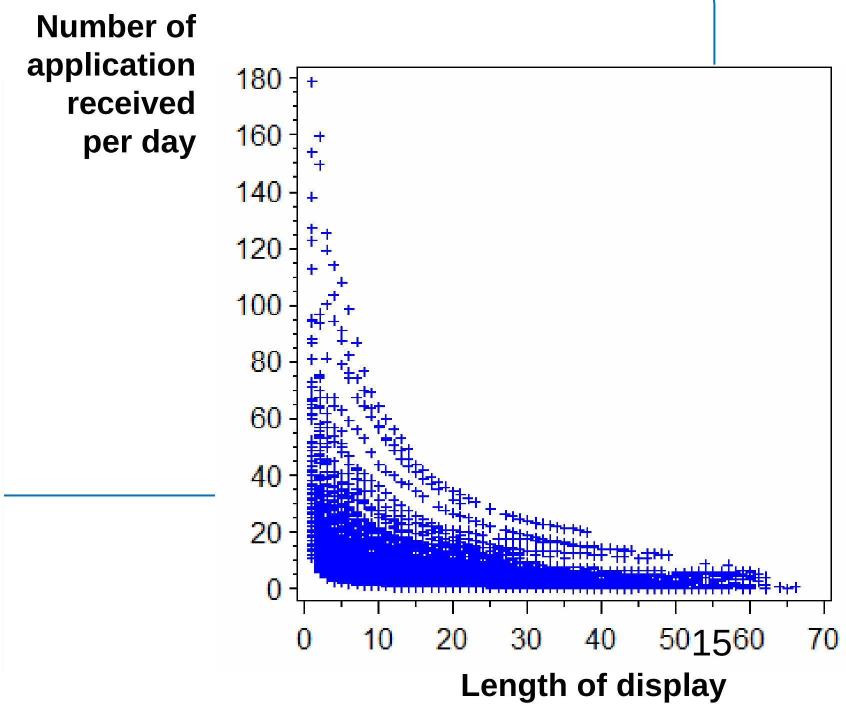
Complexity of our data and issues: display length

Irregular flow of applications and different display length because:

- Each job board has a specific length of display
- Some job postings are stopped before their end

We have to predict posting daily performance for a given time





Outline

Introduction

- ✓ Context and objectives
- ✓ Recommender systems
- ✓ Data complexity

Methodology

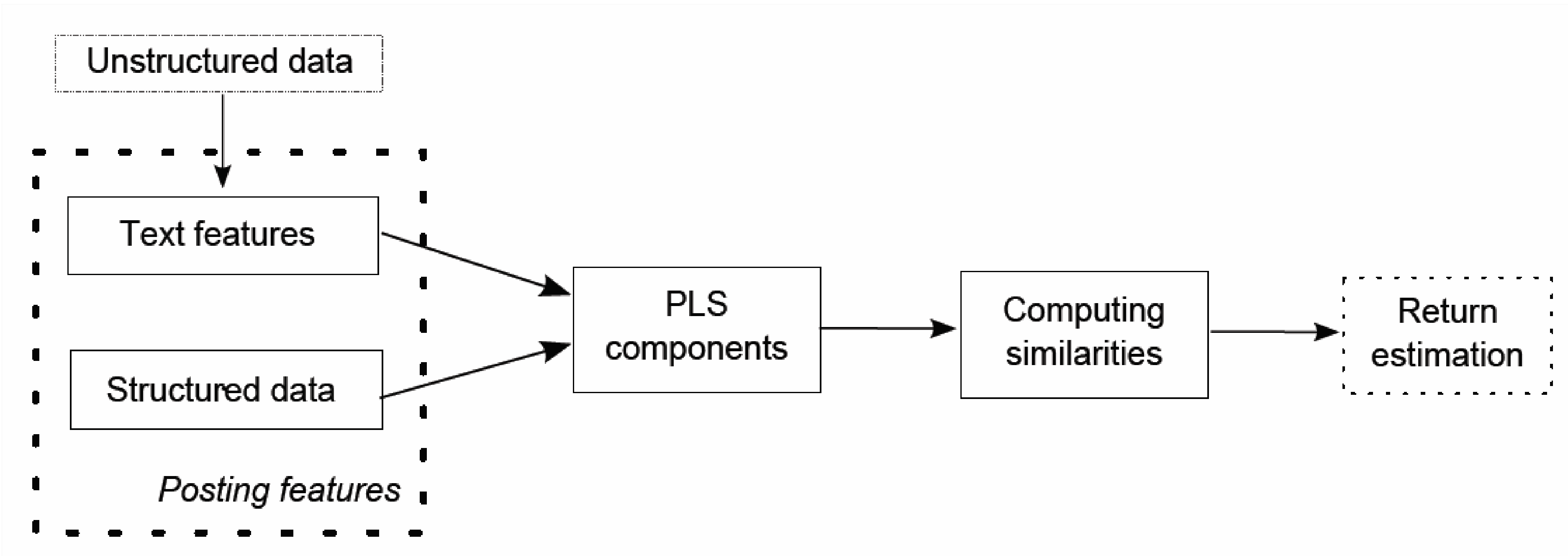
- ✓Data handling
- ✓Similarity computing between job postings
- ✓Return estimation and system evaluation

Experiments: job board recommendation for job postings

- ✓Data description
- ✓Experiments and results

Conclusions and future work

Methodology: General overview of the recommender system



Methodology: Handling of structured data

Categorical variables

- contract type
- education level
- career level
- location (region)
- job category (occupation)
- Industry
- Type of recruiter (company, recruitment agency, etc.)
- year
- month

Quantitative variables

- Location (city, employment area) demographic characteristics:

-Population

-Unemployed people

-Working people

- **Displaying time**

Categorical variables are recoded into dummy variables

Handling of unstructured data: job offer text representation

Latent Semantic Indexing (LSI) with TF-IDF weighting

1) Document-term matrix

$$T = \begin{pmatrix} & \vdots & \\ \dots & f_{ij} & \dots \\ & \vdots & \end{pmatrix}$$

2) Weighting

$$T_W = \begin{pmatrix} & \vdots & \\ \dots & l_{ij}(f_{ij}) \cdot g_j(f_{ij}) & \dots \\ & \vdots & \end{pmatrix}$$

3) SVD

$$T_W = U \Sigma V'$$

4) Document coordinates in the latent semantic space:

$$C = U_k \Sigma_k$$

Local weighting:
TF (Term Frequency)

$$l_{ij}(f_{ij}) = f_{ij}$$

Global weighting:
IDF (Inverse Document Frequency)

$$g_j(f_{ij}) = 1 + log(\frac{n}{n_j})$$

n : number of documents

n_j : number of documents in which term j occurs

Outline

Introduction

✓ Context and objectives

✓ Recommender systems

✓ Data complexity

Methodology

✓Data handling

✓Similarity computing between job postings

✓Return estimation and system evaluation

Experiments: job board recommendation for job postings

✓Data description

✓Experiments and results

Conclusions and future work

Methodology: Computing of PLS components

Why PLS?

- The number of predictors can be large compared to the number of observations
- Components are independent and highly correlated with the dependent variable
- Dimensionality reduction

Method:

- Extraction of PLS components: NIPALS algorithm
- Number of components chosen by cross-validation
- Selection of relevant predictors thanks to VIP indicator (> 0.8)
- Computing of PLS components based on the predictors kept

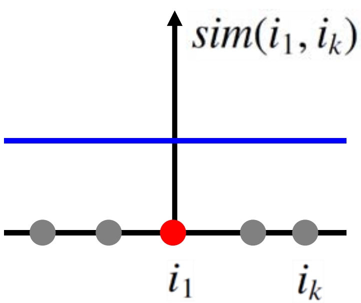
Methodology: Similarity measures

- Computing of new posting similarity with respect to all past postings

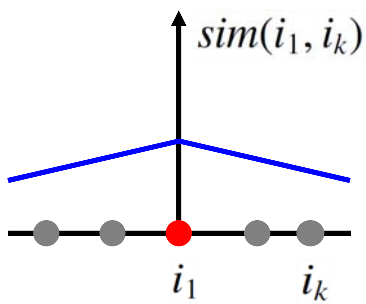
- It supposes that similar items regarding to their PLS components should have similar returns for a given job board

Method:

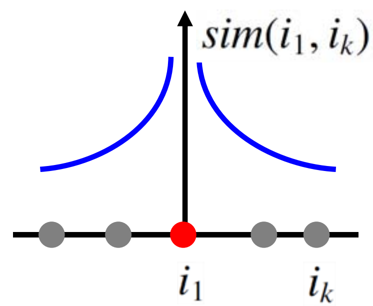
- Computation of euclidean distances between posting coordinates
- Similarity is a decreasing function of euclidean distance:



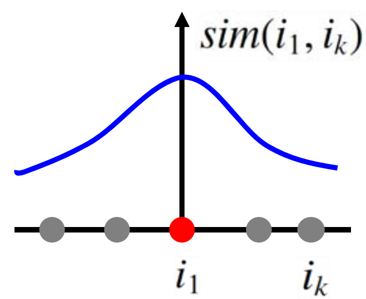
Mean



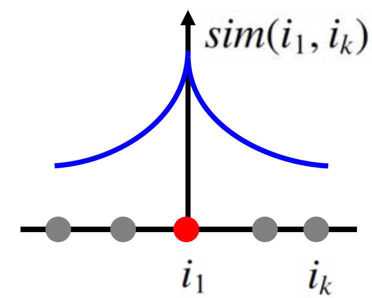
Distance max - distance



Inverse distance



Gaussian function



Exponential function

Outline

Introduction

- ✓ Context and objectives
- ✓ Recommender systems
- ✓ Data complexity

Methodology

- ✓Data handling
- ✓Similarity computing between job postings
- ✓Return estimation and system evaluation

Experiments: job board recommendation for job postings

- ✓Data description
- ✓Experiments and results

Conclusions and future work

Methodology: Return estimation

- Expected return of an item (posting) i_1 is estimated thanks to an aggregating function computed on item neighborhood

- Neighborhood is defined by the $|K|$ nearest neighbors of item i_1 with respect to the used similarity measure

- R_{u,i_1} = expected return of item i_1 for user u (job board)

- r_{u,i_k} = return of item i_k for user u

$$R_{u,i_1} = \frac{\sum_{i_k \in K} sim(i_1, i_k) \times r_{u,i_k}}{\sum_{i_k \in K} sim(i_1, i_k)}$$

Methodology: Other approaches for comparison

1 - Comparison with PLS regression (model-based recommendation)

- Computing of PLS components (method was described before)
- Regression of PLS components on the dependent variable
- Prediction by 10-fold cross validation

2 - Comparison with a non-supervised system based on text features (heuristic-based recommendation)

- LSI with TF-IDF weighting and 50 dimensions
- Similarity measures are computed directly on LSI coordinates
- Same measures as those used in the semi-supervised system
- Same estimation technique

Advantages and weaknesses of the three approaches

	Linearity constraint	Risk of overfitting	Interpreting	Weight fitting
PLS-R	yes	yes	yes	yes
Non supervised system	no	no	no	no
Semi-supervised system	**no**	**low**	**yes**	**yes**

Methodology: System evaluation

- *U = set of job boards*
- D_u = set of postings with an observed return for job board *u*
- $r_{u,i}$ = return of posting *i* on job board *u*
- $p_{u,i}$ = predicted return of posting *i* on job board *u*

Mean Absolute Error (mean error per job board)

$$\overline{MAE} = \frac{1}{|U|} \sum_{u \in U} \frac{\sum_{i \in D_u} |p_{u,i} - r_{u,i}|}{|D_u|}$$

Outline

Introduction

✓ Context and objectives

✓ Recommender systems

✓ Data complexity

Methodology

✓Data handling

✓Similarity computing between job postings

✓Return estimation and system evaluation

Experiments: job board recommendation for job postings

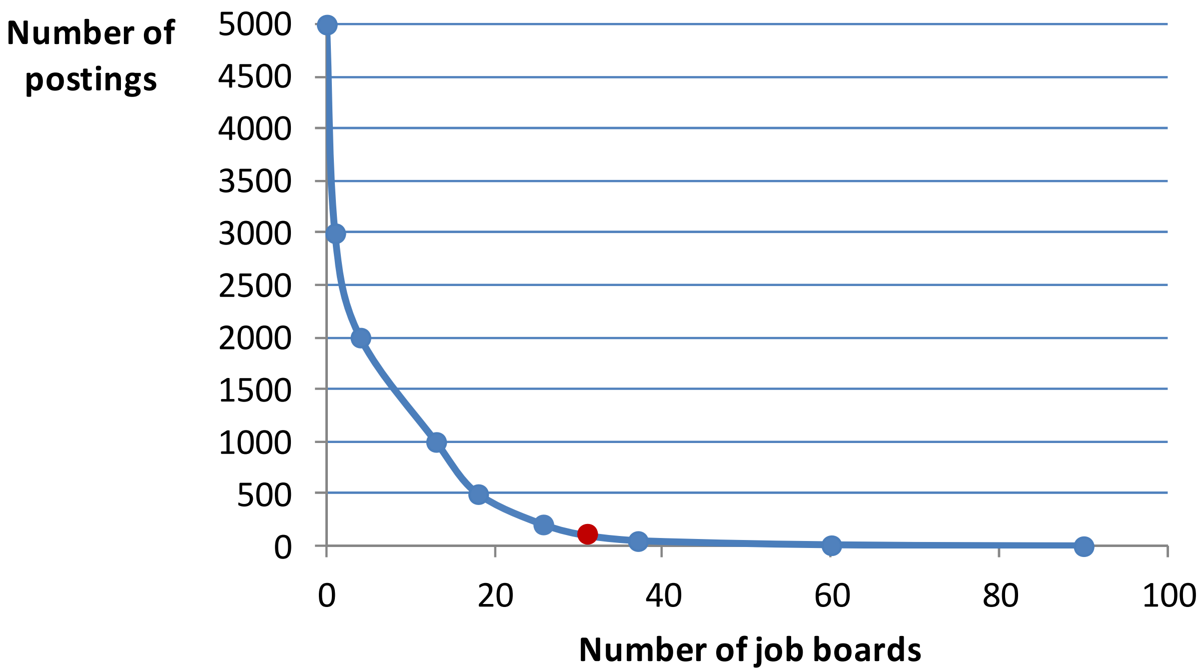
✓Data description

✓Experiments and results

Conclusions and future work

Experiments: Data perimeter

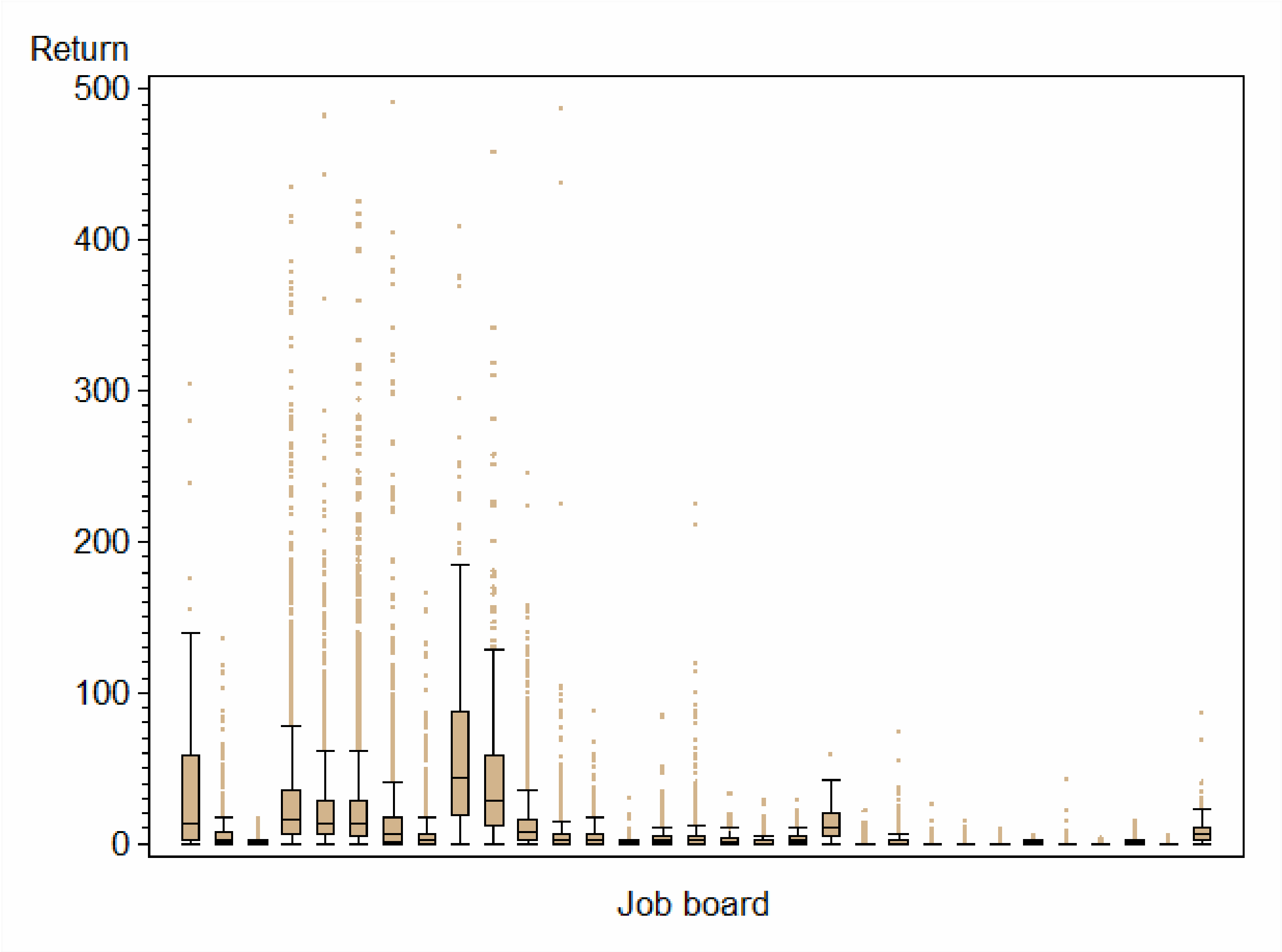
- **Objective:** predict the number of applications received for a new posting on a job board
- We keep in the sample job boards with at least 100 postings
- **Dependent variable**: number of applications / display length



- 31 job boards
- 14 334 postings
- 30875 returns

Comparison of job board returns

Illustration of return variability in and between job boards (one boxplot by job board)



Outline

Introduction

- ✓ Context and objectives
- ✓ Recommender systems
- ✓ Data complexity

Methodology

- ✓Data handling
- ✓Similarity computing between job postings
- ✓Return estimation and system evaluation

Experiments: job board recommendation for job postings

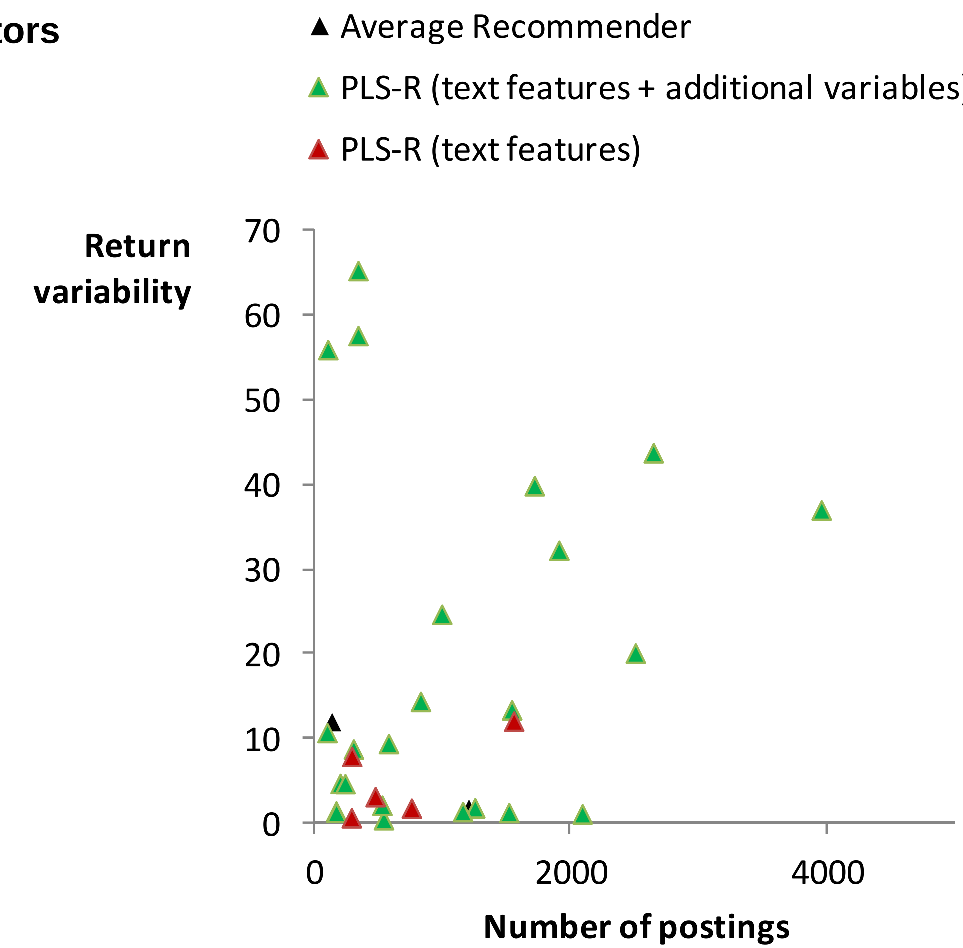
- ✓Data description
- ✓Experiments and results

Conclusions and future work

Results: Introducing of new relevant descriptors

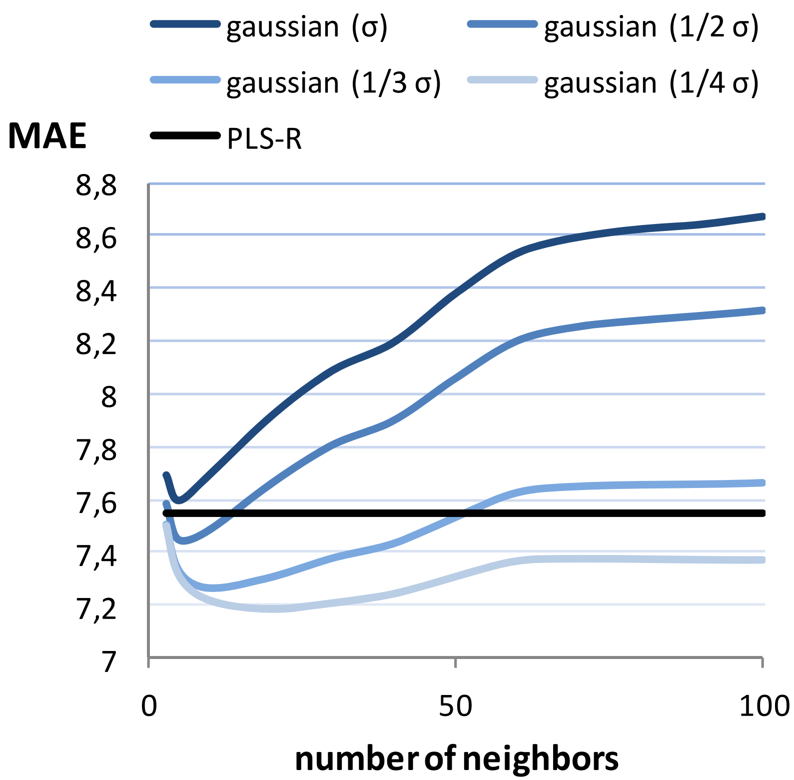
Improving results by adding relevant descriptors

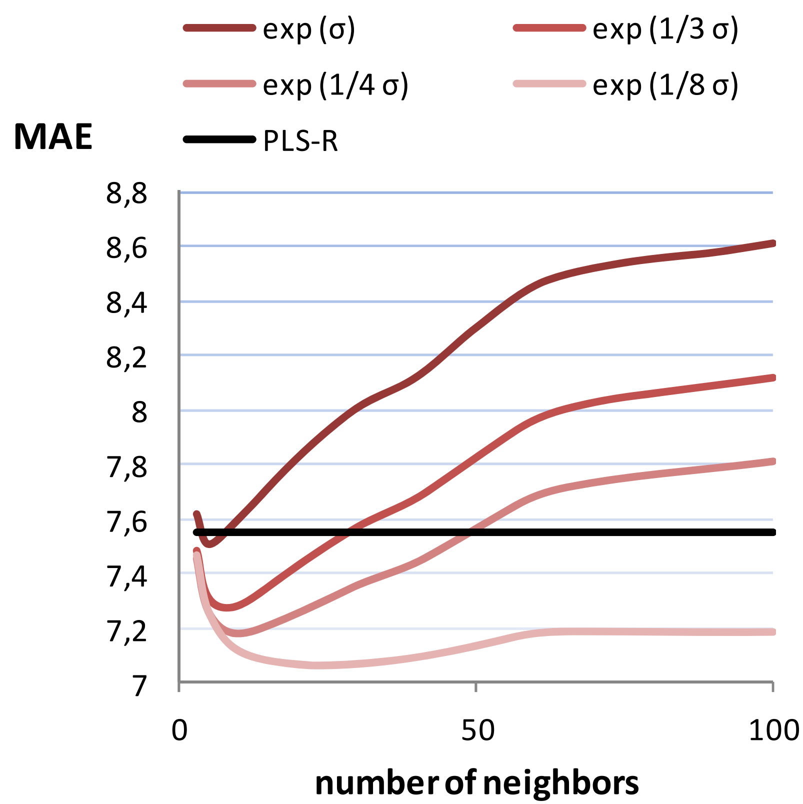
System	MAE	Best on how many job boards?
Average Recommender	10.2	2
PLS-R **text features**	8.0	5
PLS-R **text features +** **job characteristics +** **location characteristics**	**7.5**	**24**



Non-supervised approach: Discussion about parameters

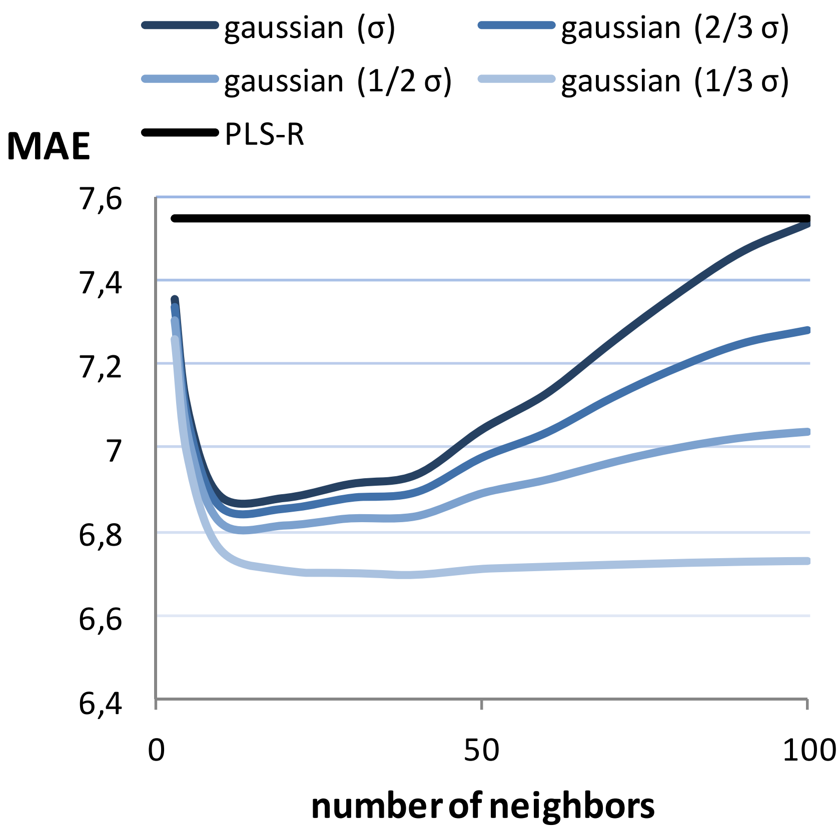
MAE according to the number of neighbors and parameter in gaussian and exponential functions

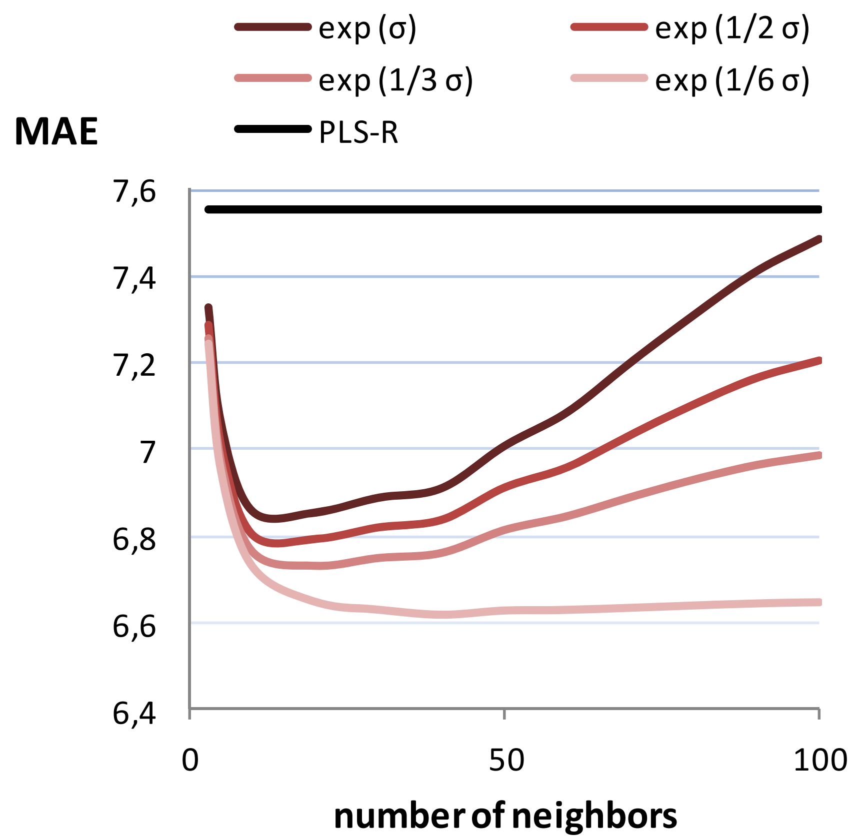




Semi-supervised approach: Discussion about parameters

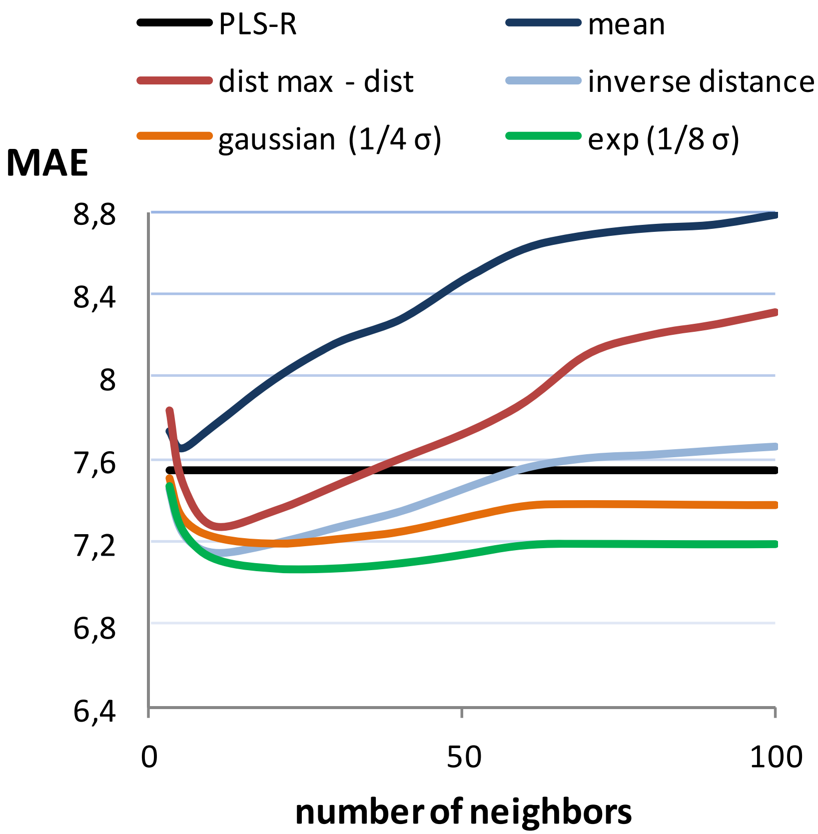
MAE according to the number of neighbors and parameter in gaussian and exponential functions



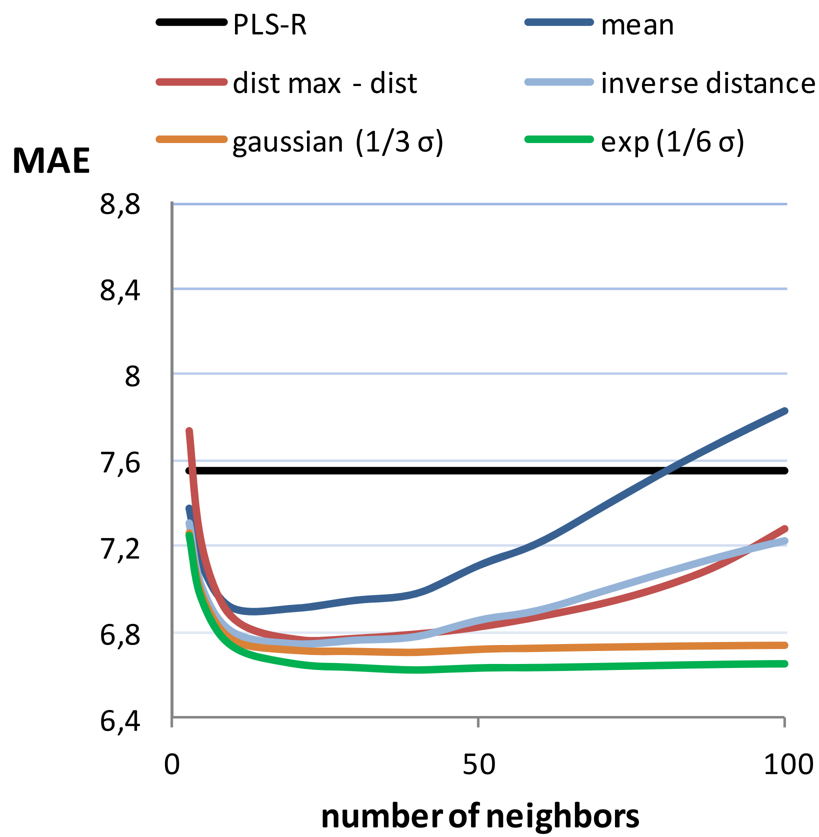


Results: Comparison of similarity functions

Non-supervised approach



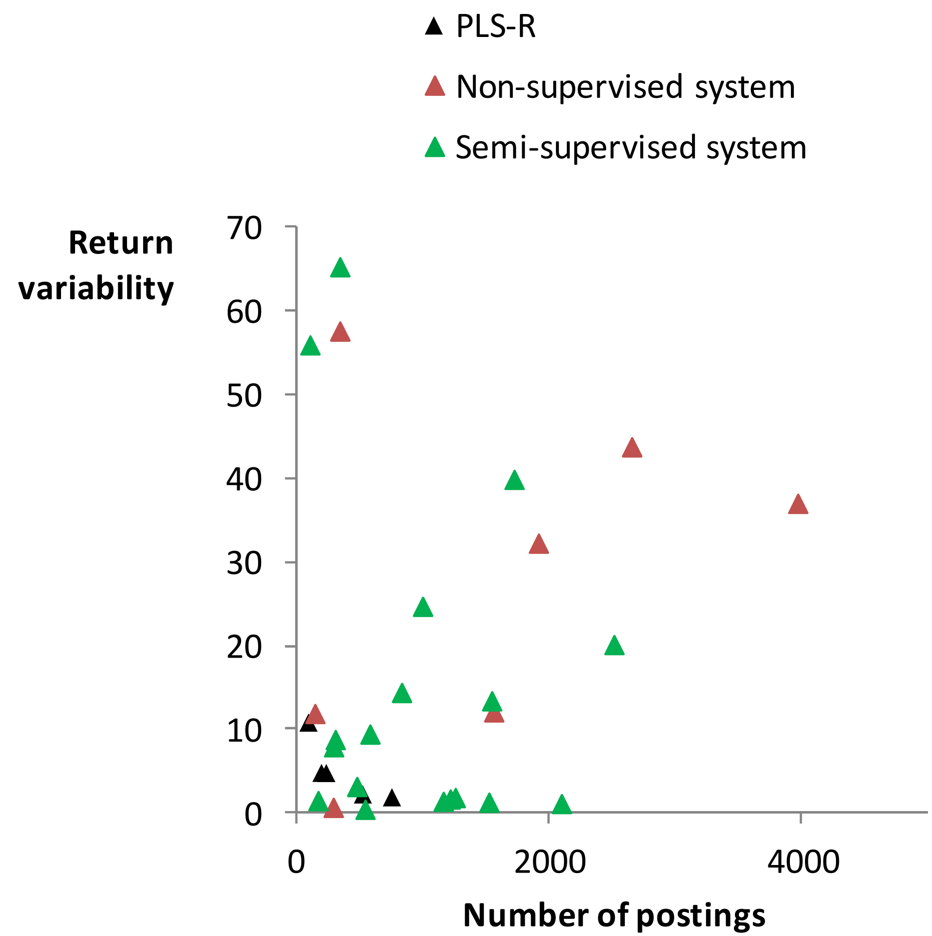
Semi-supervised approach



Results: Summary

Best system of each approach

System	MAE	Best on how many job boards?
Average Recommender	10.2	0
PLS-R	7.5	6
Non-supervised system	7.1	7
Semi-Supervised system	**6.6**	**18**



Outline

Introduction

- ✓ Context and objectives
- ✓ Recommender systems
- ✓ Data complexity

Methodology

- ✓Data handling
- ✓Similarity computing between job postings
- ✓Return estimation and system evaluation

Experiments: job board recommendation for job postings

- ✓Data description
- ✓Experiments and results

Conclusions and future work

Conclusions and future work

Conclusions:

- MAE decreases with the standard deviation parameter in gaussian and exponential functions (but increases if too small)
- In the semi-supervised approach, the optimal parameter implies stability of MAE with the number of neighbors. Select 40 neighbors, and just find the optimal parameter.
- Best results with semi-supervised approach and exponential function
- The system allows introducing of new variables and manage their weight in the model
- Estimation are made on job offers really close to the new offer / the offer studied

Future work:

- Improve the prediction if the posting is in fact « exactly » the same as a previous one
- Manage job boards with very few or no postings

谢谢

Thank you for your attention!