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

About
From retail to sport and music to banking – Jobs in IT form a vital component of everyday life, with opportunities existing across the spectrum at all level. Browse IT jobs on Monster to find the one that best fits your career requirements.

Job Search Results: IT

Customer Services Agents –German
London, London - Featured Job
Role: Customer Services Agents –German Reporting to: Customer Services Manager Location: London
Company Profile Lycatol is a dynamic, fast growing...

Web Designer – HTML, CSS, Javascript/ jQuery – Part-time
VENTURI LIMITED Manchester, North West Posted today
Web Designer – HTML, CSS, Javascript/ jQuery – Part-time Seeking an experienced, Creative Web Designer / Web Developer / Graphic Designer to join our establish...

C++ Software Engineer
Dewfoot IT Resources Ltd Hounslow, London Posted today
C++ Software Engineer - Permanent - Hounslow - C++, Oracle RDBMS, OLTP, STL, Design Patterns, UNIX, PL/SQL - £40K - £50K dependent on experience & excellent benef...

Job list

Web Designer – HTML, CSS, Javascript/JQuery – Part-time
VENTURI LIMITED Manchester, North West Posted today
Web Designer – HTML, CSS, Javascript/JQuery – Part-time Seeking an experienced, Creative Web Designer / Web Developer / Graphic Designer to join our establish...
Job details & apply

C++ Software Engineer
dewfoot IT Resources Ltd Hounslow, London Posted today
C++ Software Engineer – Permanent – Hounslow – C++, Oracle RDBMS, OLTP, STL, Design Patterns, UNIX, PL/SQL – £40k – £50k dependent on experience & excellent benet...
Job details & apply

Structured data

Web Designer – HTML, CSS, Javascript/JQuery – Part-time
About the Job
Seeking an experienced, Creative Web Designer / Web Developer / Graphic Designer to join our established FTSE 100 Company based in the Manchester area. This is an exciting opportunity to work in a part-time role (approx 26.25hrs per week) on a contract that is initially 12 months and likely to be extended.

The successful Creative Web Designer / Web Developer / Graphic Designer will have experience with:
- Dreamweaver
- Flash
- Photoshop
- HTML/Javascript
- CSS
- Adobe Acrobat Professional

It would also be desirable (but NOT necessary) for the successful Creative Web Designer to have some experience with:
- jQuery
- SEO

The successful Creative Web Designer will be responsible for:
- Importing and creating graphical design and flare to websites.
- Re-branding of existing websites.
- Setting up websites from initial concept with little direction.
- Producing several mock designs before final concept approval.
- Designing for front end web sites / Back end Content Management Systems.
- Producing Artwork for e- shot campaigns.
- Creating interactive PDF brochures.
- Developing flash banners to be inserted on external websites.
- Optimising web pages for search engines.

Unstructured data
Context: Multiposting of a job offer

Illustration of multiposting

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

<table>
<thead>
<tr>
<th>User</th>
<th>Harry Potter</th>
<th>The Chronicles of Narnia</th>
<th>Terminator</th>
<th>Rambo</th>
<th>The Lord of the Rings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Bob</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Cindy</td>
<td>3</td>
<td>5</td>
<td>?</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>David</td>
<td>1</td>
<td>?</td>
<td>5</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

Fragment of a rating matrix

What movie should be recommended to Alice?
- Bob and Cindy like the same movies as Alice
- So we should recommend to Alice another movie that they liked: « The Lord of the Rings »

This is a collaborative system (based on ratings and no use of descriptive variables)
About recommender systems

Prediction are based on ratings obtained by the most similar items with respect to rating vectors.

Prediction are based on item features (recommends items similar to those that the user liked in the past).

Collaborative filtering

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
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
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

Number of application received vs Displaying day

Number of application received per day vs Length of display
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

Unstructured data

Structured data

Text features

Posting features

PLS components

Computing similarities

Return estimation
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

\[
T = \begin{pmatrix}
\vdots & f_{ij} & \vdots \\
\vdots & \ddots & \vdots \\
\vdots & & \ddots
\end{pmatrix}
\]

2) Weighting

\[
T_W = \begin{pmatrix}
\vdots & l_{ij}(f_{ij}) \cdot g_{j}(f_{ij}) & \vdots \\
\vdots & \ddots & \vdots \\
\vdots & & \ddots
\end{pmatrix}
\]

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)
  
  \[
g_{j}(f_{ij}) = 1 + \log\left(\frac{n}{n_j}\right)
  \]

- \(n\) : 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.

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

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

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

<table>
<thead>
<tr>
<th></th>
<th>Linearity constraint</th>
<th>Risk of overfitting</th>
<th>Interpreting</th>
<th>Weight fitting</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PLS-R</strong></td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td><strong>Non supervised system</strong></td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td><strong>Semi-supervised system</strong></td>
<td>no</td>
<td>low</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>
Methodology: System evaluation

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

**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**
• **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

![Graph showing the relationship between number of job boards and number of postings.]

- 31 job boards
- 14,334 postings
- 30,875 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**

<table>
<thead>
<tr>
<th>System</th>
<th>MAE</th>
<th>Best on how many job boards?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Recommender</td>
<td>10.2</td>
<td>2</td>
</tr>
<tr>
<td>PLS-R text features</td>
<td>8.0</td>
<td>5</td>
</tr>
<tr>
<td>PLS-R text features + job characteristics + location characteristics</td>
<td>7.5</td>
<td>24</td>
</tr>
</tbody>
</table>

---

**Graph: Return variability vs Number of postings**

- ▲ Average Recommender
- ▶ PLS-R (text features + additional variables)
- ▲ PLS-R (text features)
Non-supervised approach: Discussion about parameters

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

- MAE vs number of neighbors for Gaussian (σ), Gaussian (1/2 σ), Gaussian (1/3 σ), and Gaussian (1/4 σ) functions.
- MAE vs number of neighbors for Exponential (σ), Exponential (1/3 σ), Exponential (1/4 σ), and Exponential (1/8 σ) functions.

October 29th - SDA 2011, Beijing
Semi-supervised approach: Discussion about parameters

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

![Graph showing MAE for different parameters and numbers of neighbors for Gaussian and Exponential functions.](image)
Results: Comparison of similarity functions

Non-supervised approach

- PLS-R
- mean
- dist max - dist
- inverse distance
- gaussian (1/4 $\sigma$)
- exp (1/8 $\sigma$)

Semi-supervised approach

- PLS-R
- mean
- dist max - dist
- inverse distance
- gaussian (1/3 $\sigma$)
- exp (1/6 $\sigma$)
## Results: Summary

**Best system of each approach**

<table>
<thead>
<tr>
<th>System</th>
<th>MAE</th>
<th>Best on how many job boards?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Recommender</td>
<td>10.2</td>
<td>0</td>
</tr>
<tr>
<td>PLS-R</td>
<td>7.5</td>
<td>6</td>
</tr>
<tr>
<td>Non-supervised system</td>
<td>7.1</td>
<td>7</td>
</tr>
<tr>
<td>Semi-Supervised system</td>
<td>6.6</td>
<td>18</td>
</tr>
</tbody>
</table>

The chart shows the return variability of each system with respect to the number of postings. The x-axis represents the number of postings, and the y-axis represents return variability. The symbols in the chart correspond to the systems:
- **△** PLS-R
- **△** Non-supervised system
- **△** Semi-supervised system
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
Thank you for your attention!