Learning Global Term Weights for Content-based Recommender Systems

Yupeng Gu¹, Bo Zhao², David Hardtke², Yizhou Sun¹

¹Northeastern University
²LinkedIn Corp.

April 14, 2016
Given a LinkedIn member, we aim to find the jobs that he/she is most interested in.

Figure: Job recommendation panel on www.linkedin.com
Given a LinkedIn member, we aim to find the jobs that he/she is most interested in.

Figure: Job recommendation panel on www.linkedin.com

What information is available for members and jobs?
Yupeng Gu

Research Assistant at Northeastern University
Boston, Massachusetts | Computer & Network Security

Experience

Research Assistant
Northeastern University
September 2013 – Present (2 years 7 months) | Greater Boston Area

Software Engineer Internship
LinkedIn
May 2015 – August 2015 (4 months) | Sunnyvale

Education

Northeastern University
Doctor of Philosophy (PhD), Computer Science
2013 – 2018

University of Science and Technology of China
BS, Applied Mathematics
2009 – 2013

Skills

- Object Oriented Design
- Data Mining
- Git
- Bash
- Shell Scripting
- Distributed Systems
- Subversion
- Machine Learning
- OOP
- Cryptography
<table>
<thead>
<tr>
<th><strong>Member Example</strong></th>
</tr>
</thead>
</table>

**Yupeng Gu**

**Research Assistant** at Northeastern University

**Boston, Massachusetts | Computer & Network Security**

---

<table>
<thead>
<tr>
<th><strong>Experience</strong></th>
</tr>
</thead>
</table>

**Research Assistant**
Northeastern University
September 2013 – Present (2 years 7 months) | Greater Boston Area

---

**Software Engineer Internship**
Linkedin
May 2015 – August 2015 (4 months) | Sunnyvale

---

<table>
<thead>
<tr>
<th><strong>Education</strong></th>
</tr>
</thead>
</table>

**Northeastern University**
Doctor of Philosophy (PhD), Computer Science
2013 – 2018

---

**University of Science and Technology of China**
BS, Applied Mathematics
2008 – 2013

---

<table>
<thead>
<tr>
<th><strong>Skills</strong></th>
</tr>
</thead>
</table>

- Object Oriented Design
- Data Mining
- Git
- Bash
- Shell Scripting
- Distributed Systems
- Subversion
- Machine Learning
- OOP
- Cryptography
Senior Software Engineer - Front-End

LinkedIn
Mountain View

Job description

LinkedIn was built to help professionals achieve more in their careers, and every day millions of people use our products to make connections, discover opportunities, and gain insights. Our global reach means we get to make a direct impact on the world’s workforce in ways no other company can. We’re much more than a digital resume – we transform lives through innovative products and technology.
Job Example

Senior Software Engineer - Front-End
LinkedIn
Mountain View

Job description
LinkedIn was built to help professionals achieve more in their careers, and every day millions of people use our products to make connections, discover opportunities, and gain insights. Our global reach means we get to make a direct impact on the world’s workforce in ways no other company can. We’re much more than a digital resume – we transform lives through innovative products and technology.
Overview of the Model

(member's description)

(3, 0, 9, 0, 0, ..., 0, 0, 21, 2, 0)

(member's skills)

(0, 0, 1, 0, 2, ..., 0, 7, 0, 15, 2)

(job's skills)

(0, 0, 7, 0, 4, ..., 1, 1, 0, 0, 0)
Overview of the Model

- Member's description: (3, 0, 9, 0, 0, ..., 0, 0, 21, 2, 0)
- Member's skills: (0, 0, 1, 0, 2, ..., 0, 7, 0, 15, 2)
- Job's skills: (0, 0, 7, 0, 4, ..., 1, 1, 0, 0, 0)
Overview of the Model

(3, 0, 9, 0, 0, ... , 0, 0, 21, 2, 0)
member's description

(0, 0, 1, 0, 2, ... , 0, 7, 0, 15, 2)
member's skills

(0, 0, 7, 0, 4, ... , 1, 1, 0, 0, 0)
job's skills

similarility = 0.9
A Simple Idea

- For each (member field $s$, job field $t$), calculate the similarity score between two feature vectors.
A Simple Idea

- For each (member field $s$, job field $t$), calculate the similarity score between two feature vectors.
- Aggregate the scores of all field pairs $(s, t)$. 
Text features: different terms should have different weights in similarity calculation.

For term $t$ and a collection of documents $D = \{d\}$, idf score is defined as

$$\text{idf}(t, D) = \log \frac{1 + |D|}{1 + |\{d \in D : t \in d\}|}.$$ 

Higher IDF usually indicates more importance. TF × IDF is used as feature for each term.
Motivation

Text features: different terms should have different weights in similarity calculation.
  - Use TF-IDF (Term Frequency - Inverse Document Frequency)?
Text features: different terms should have different weights in similarity calculation.

- Use TF-IDF (Term Frequency - Inverse Document Frequency)?
- For term $t$ and a collection of documents $D = \{d\}$, idf score is defined as

$$idf(t, D) = \log \frac{1 + |D|}{1 + |\{d \in D : t \in d\}|}.$$
Text features: different terms should have different weights in similarity calculation.

- Use TF-IDF (Term Frequency - Inverse Document Frequency)?
- For term $t$ and a collection of documents $D = \{d\}$, idf score is defined as

$$idf(t, D) = \log \frac{1 + |D|}{1 + |\{d \in D : t \in d\}|}.$$ 

- Higher IDF usually indicates more importance.
Text features: different terms should have different weights in similarity calculation.

- Use TF-IDF (Term Frequency - Inverse Document Frequency)?
- For term $t$ and a collection of documents $D = \{d\}$, idf score is defined as

$$idf(t, D) = \log \frac{1 + |D|}{1 + |\{d \in D : t \in d\}|}.$$

- Higher IDF usually indicates more importance.
- TF $\times$ IDF is used as feature for each term.
Example: Lower IDF Terms Can be Predictive

High IDF term: government   Low IDF term: machine learning

Member (description)

I have enrolled in a project which provides users with a visualization of government financial statistics using machine learning techniques ...

Recommended Jobs (description)

Job 1
We are a managed services provider and we support many projects with government agencies and non-profit organizations.

Job 2
You will apply machine learning algorithms to analyze large data sets ...
Example: Lower IDF Terms Can be Predictive

High IDF term: government  Low IDF term: machine learning

Member (description)
I have enrolled in a project which provides users with a visualization of government financial statistics using machine learning techniques ...

Recommended Jobs

Job 1
We are a managed services provider and we support many projects with government agencies and non-profit organizations.

Job 2
You will apply machine learning algorithms to analyze large data sets ...

Apply?
No

Yes
Example: Lower IDF Terms Can be Predictive

High IDF term: government ↓  Low IDF term: machine learning ↑

Member (description)
I have enrolled in a project which provides users with a visualization of government financial statistics using machine learning techniques ...

Recommended Jobs
Job 1
We are a managed services provider and we support many projects with government agencies and non-profit organizations.

Apply?
No

Job 2
You will apply machine learning algorithms to analyze large data sets ...

Yes
Problem

- Learn the optimal global term weights for each user text section and item text section (e.g., importance of “machine learning” in job skills)
Learn the optimal global term weights for each user text section and item text section (e.g., importance of “machine learning” in job skills)
Learn the weights of multiple content matching features between user and item profiles (e.g., user skills vs. job skills, user titles vs. job skills)
Problem

- Learn the optimal global term weights for each user text section and item text section (e.g., importance of “machine learning” in job skills)
- Learn the weights of multiple content matching features between user and item profiles (e.g., user skills vs. job skills, user titles vs. job skills)
- Model: MLRM (Multi-layer Logistic Regression Model).
First Layer

\[\begin{align*}
&\mathbf{u}_{s1} \quad \mathbf{u}_{s2} \quad \ldots \quad \mathbf{u}_{sV} \\
&\text{Member Field } S \\
&
\mathbf{v}_{t1} \quad \mathbf{v}_{t2} \quad \ldots \quad \mathbf{v}_{tV} \\
&\text{Job Field } t
\end{align*} \]
First Layer

\[ u^{(1)}_{s1} w^{(1)}_{s1} \quad u^{(1)}_{s2} w^{(1)}_{s2} \quad \ldots \quad u^{(1)}_{sV} w^{(1)}_{sV} \]

Member Field \( S \)

\[ u_{s1} \quad u_{s2} \quad \ldots \quad u_{sV} \]

\[ v^{(1)}_{t1} w^{(1)}_{t1} \quad v^{(1)}_{t2} w^{(1)}_{t2} \quad \ldots \quad v^{(1)}_{tV} w^{(1)}_{tV} \]

Job Field \( t \)

\[ v_{t1} \quad v_{t2} \quad \ldots \quad v_{tV} \]
First Layer

\[
sim_{s,t}
\]

\[
\begin{array}{c}
w^{(1)}_{s1} v_{s1} \\
u_{s1}
\end{array} \quad \begin{array}{c}
w^{(1)}_{s2} v_{s2} \\
u_{s2}
\end{array} \quad \ldots \quad \begin{array}{c}
w^{(1)}_{sV} v_{sV} \\
u_{sV}
\end{array}
\]

\[
\begin{array}{c}
w^{(1)}_{t1} v_{t1} \\
u_{t1}
\end{array} \quad \begin{array}{c}
w^{(1)}_{t2} v_{t2} \\
u_{t2}
\end{array} \quad \ldots \quad \begin{array}{c}
w^{(1)}_{tV} v_{tV} \\
u_{tV}
\end{array}
\]

Member Field \( S \)  
Job Field \( t \)
Second Layer

\[ \text{sim}_{1,1} \quad \text{sim}_{1,2} \quad \ldots \quad \text{sim}_{s,t} \quad \ldots \quad \text{sim}_{M,J} \]

Similarity between member field \( s \) and job field \( t \)
Second Layer

\[ s = \sum_{s,t} w_{st}^{(2)} \cdot \text{sim}_{s,t} + w_0^{(2)} \]

Similarity between member field \( s \) and job field \( t \)
Unified Model

Objective: minimize the logit loss of all \((m_i, j)\) pairs.

\[ L(W^{(1)}, W^{(2)}) = \sum_{i, j} \log(1 + e^{-y_{ij} s_{ij}}) \]

Yupeng Gu
Learning Global Term Weights for Content-based Recommender Systems
Unified Model

Objective: minimize the logit loss of all \((\text{member } i, \text{ job } j)\) pairs.

\[
L(W^{(1)}, W^{(2)}) = \sum_{i,j} \log(1 + e^{-y_{ij}s_{ij}})
\]
Stochastic Gradient Descent (SGD) is used to perform updates on model parameters.
Stochastic Gradient Descent (SGD) is used to perform updates on model parameters.

- For each (member, job) pair, update the corresponding term weights and field-pair weights using back propagation
  - Most commonly used training algorithm for neural networks
  - Errors are propagated from the top layer to the bottom layer (backward)
Stochastic Gradient Descent (SGD) is used to perform updates on model parameters.

- For each (member, job) pair, update the corresponding term weights and field-pair weights using back propagation
  - Most commonly used training algorithm for neural networks
  - Errors are propagated from the top layer to the bottom layer (backward)
- Adaptive learning rates
  - AdaGrad is adopted: frequent terms converge fast; less frequent words will have bigger updates
Vocabulary: 490 thousand unique terms and 75 fields (for members and jobs in total).
Data Description

- Vocabulary: 490 thousand unique terms and 75 fields (for members and jobs in total).
- Data: 3.1 million (member, job) pairs

<table>
<thead>
<tr>
<th></th>
<th>Positive</th>
<th>Feedback Negative</th>
<th>Random Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>50%</td>
<td>25%</td>
<td>25%</td>
</tr>
</tbody>
</table>
Experimental Results

- Case studies
Experimental Results

- Case studies
  - What are the most important terms in member and job fields?
Experimental Results

Case studies

- What are the most important terms in member and job fields?
- What are the most important pairs of member field and job field?
Experimental Results

Case studies
- What are the most important terms in member and job fields?
- What are the most important pairs of member field and job field?

Comparison of AUC and area under precision-recall curves.
- AUC (Area Under the receiver operating characteristic Curve): the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one.
- Precision-recall curve: a precision vs. recall curve generated as the discrimination threshold varies.
### Case Study - Most Important Member-Job Field Pairs

<table>
<thead>
<tr>
<th>Member Fields</th>
<th>Job Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skill ID</td>
<td>Skill ID</td>
</tr>
<tr>
<td>Skill Term</td>
<td>Skill Term</td>
</tr>
<tr>
<td>Summary</td>
<td>Description</td>
</tr>
<tr>
<td>Past Position Summary</td>
<td>Title</td>
</tr>
<tr>
<td>Past Title</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Case Study - Most Important Member-Job Field Pairs

- Skill ID
- Skill Term
- Summary
- Past Position Summary
- Past Title

Member Fields

- Skill ID
- Skill Term
- Description
- Title

Job Fields
Case Study - Most Important Member-Job Field Pairs

Skill ID
Skill Term
Summary
Past Position Summary
Past Title

.:.

Skill ID
Skill Term
Description
Title

.:.

Member Fields

Job Fields
Case Study - Most Important Member-Job Field Pairs

Member Fields

Skill ID
Skill Term
Summary
Past Position Summary
Past Title

Job Fields

Skill ID
Skill Term
Description
Title

...
Case Study - Most Important Member-Job Field Pairs

Skill ID
Skill Term
Summary
Past Position Summary
Past Title

Member Fields

Skill ID
Skill Term
Description
Title

Job Fields
Results

Comparison on test dataset:

<table>
<thead>
<tr>
<th>Method</th>
<th>AUC</th>
<th>AUPRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (tf-idf as feature)</td>
<td>0.692</td>
<td>0.671</td>
</tr>
<tr>
<td>Multi-layer Logistic Regression Model</td>
<td>0.811 (+17.2%)</td>
<td>0.793 (+18.2%)</td>
</tr>
<tr>
<td>Multi-layer Logistic Regression Model (jobs only)</td>
<td>0.792 (+14.5%)</td>
<td>0.771 (+14.9%)</td>
</tr>
</tbody>
</table>
Effectiveness of selecting only the top terms to do recommendation:

<table>
<thead>
<tr>
<th>Method</th>
<th>AUC</th>
<th>AUPRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLRM</td>
<td>0.811 (+17.2%)</td>
<td>0.793 (+18.2%)</td>
</tr>
<tr>
<td>MLRM (top 90%)</td>
<td>0.786 (+13.6%)</td>
<td>0.764 (+13.9%)</td>
</tr>
<tr>
<td>MLRM (top 80%)</td>
<td>0.760 (+9.8%)</td>
<td>0.768 (+14.4%)</td>
</tr>
<tr>
<td>MLRM (top 50%)</td>
<td>0.756 (+9.3%)</td>
<td>0.761 (+13.4%)</td>
</tr>
<tr>
<td>MLRM (top 10%)</td>
<td>0.744 (+7.5%)</td>
<td>0.780 (+16.2%)</td>
</tr>
</tbody>
</table>
We propose MLRM to learn global term weights for content-based recommendation.
Conclusion

- We propose MLRM to learn global term weights for content-based recommendation.
- Text similarity function (cosine) is directly optimized and multiple cosine similarity scores between different sections of user and item profiles are considered holistically.
We propose MLRM to learn global term weights for content-based recommendation.

Text similarity function (cosine) is directly optimized and multiple cosine similarity scores between different sections of user and item profiles are considered holistically.

Our method is efficient in handling large-scale training data generated by production recommender systems.
Conclusion

- We propose MLRM to learn global term weights for content-based recommendation.
- Text similarity function (cosine) is directly optimized and multiple cosine similarity scores between different sections of user and item profiles are considered holistically.
- Our method is efficient in handling large-scale training data generated by production recommender systems.
- We manage to improve AUC of recommendation evaluation by over 17% on real data from LinkedIn job recommendation system.
Thanks!

Q & A