Confidence-Aware Graph Regularization with Heterogeneous Pairwise Features

Yuan Fang
Bo-June (Paul) Hsu
Kevin Chen-Chuan Chang

University of Illinois at Urbana-Champaign
Microsoft Research
University of Illinois at Urbana-Champaign

SIGIR 2012 @ Portland, OR, USA
Outline

• Problem and motivation
• Regularization framework
• Applications in IR
• Experiments
• Conclusion
Classifications in IR

- Many classification tasks in IR
  - Given some objects and a set of classes
  - Some objects are labeled (with known classes)
  - Predict the class of each unlabeled object

- Eg 1. Text categorization
  - Spam detection
  - Information filtering
  - Email organization
  - ...

- Eg 2. Query intent classification
  - Search vertical
  - Ads targeting
  - ...

Challenges

• **Feature sparsity**
  • In our query classification dataset, 95% of queries contain no more than five words

• **Scarcity of labeled data**
  • Especially for IR tasks with a large number of classes
  • Our query classification dataset contains 2000+ fine-grained classes for the shopping domain alone
    • Eg. Inkjet-printer, laser-printer, line printer
Graph Regularization

- Addresses both challenges
- **Feature sparsity**
  - Traditionally features are extracted at object level
  - Features can be potentially extracted from each pair of objects
  - Can be modeled by an undirected graph
    - Vertices: objects
    - Edges: pairwise features
- **Scarcity of labeled data**
  - Neighboring objects on the graph are similar
  - Labels propagate across similar objects
    - “Similar objects share similar labels”
  - Semi-supervised in nature
Key Observation 1

- **Heterogeneous Pairwise Features**
  - Most existing frameworks use a single pairwise feature
  - Heterogeneous features exist
    - Complement each other
    - More robust
  - Eg. in query intent classification
    - Co-clicks
      - If two queries share a common click landing on the same page
        - only about \( \frac{1}{4} \) of the queries have clicks
    - Lexical similarity
      - If two queries contain overlapping words
        - “laptop” vs. “notebook computer” — same products
        - “laptop” vs. “laptop bag” — different products
Key Observation 2

• **Confidence-aware regularization**
  - Existing frameworks regularize based on similarity only
    - “Similar objects share similar labels”
    - More similar $\rightarrow$ higher influence on label

\[
\text{a: a printer} \\
\text{b: more likely a printer} \\
\text{c: less likely a printer}
\]

• **Classification confidence also matters**
  - Some objects are easier to classify than others
  - If we are more confident about the prediction on an object, we expect it to influence its neighbors more

\[
\text{a: a printer (90% confident)} \\
\text{b: a camera (10% confident)} \\
\text{c: more likely a printer than a camera}
\]
Outline

• Problem and motivation
• Regularization framework
• Applications in IR
• Experiments
• Conclusion
Object-Relationship Graph

- Vertices: objects, \( o \)
- Edges: relationships, \( e = (o, o', \tau) \)
  - Have different types \( \tau \) for different pairwise features
  - Can have multiple edges between two objects
  - Weights encode the affinity between objects, \( W(o, o', \tau) \)

\[ \]

![Diagram of Object-Relationship Graph]

- Relationship based on lexical pairwise feature
- Relationship based on co-click pairwise feature
Dirichlet Distribution

- Target classes \(\{1, ..., K\}\)
- Each object has an underlying class distribution over \(\{1, ..., K\}\)
  - Eg. “canon”: (digital-camera:0.3; inkjet-printer:0.2; . . .)
  - Inherently latent
- Model each object \(o\) with a **Dirichlet distribution** \(\text{Dir}(\alpha_o)\)
  - \(\alpha_0 = (\alpha_0[1], ..., \alpha_0[K])\)
  - Describes the distribution over all possible class distributions when class \(i\) has been observed \(\alpha_0[i] - 1\) times
- Interpret the total count of observation as confidence \(\sigma_o\):

\[
\sigma_o \triangleq \sum_{i=1}^{K} (\alpha_o[i] - 1) = \sum_{i=1}^{K} \alpha_o[i] - K
\]
Regularization by Neighbors

Dirichlet prior
$\text{Dir}(\alpha_o)$

Additional multinomial observations

$S(o, o_1) (\alpha_{o_1} - 1)$
$S(o, o_1) (\alpha_{o_2} - 1)$
$S(o, o_1) (\alpha_{o_3} - 1)$

More neighbors $\Rightarrow$ More observations $\Rightarrow$ Higher confidence?

Dirichlet posterior
$\text{Dir}(\tilde{\alpha}_o)$

Overall similarity:

$S(o, o') = \sum_\tau \lambda_\tau W(o, o', \tau)$

$\tilde{\alpha}_o \propto \alpha_o + \sum_{i=1}^3 S(o, o_i) \alpha_{o_i}$
Confidence-Aware Prediction

- Find the posterior mode $\tilde{m}_o$ of the Dirichlet posterior $Dir(\tilde{\alpha}_o)$
  - $\tilde{m}_o$ itself is a distribution over the classes
- Assign labels by:
  - using a cut-off threshold on $\tilde{m}_o$
  - taking top $k$ classes in $\tilde{m}_o$
- Exists a closed form for $\tilde{m}_o$
  - Weighted average of the prior mode of $o$ and its neighbors $N(o)$
  - Weights accounts for both similarity and confidence

$$\tilde{m}_o \propto \sigma_o m_o + \sum_{o' \in N(o)} S(o, o') \sigma_{o', o} m_{o'}$$
Iterative Regularization

- An object is directly regularized by its neighbors
- How about neighbors of neighbors?
  - Can be modeled by regularizing the posterior again
  - More generally, iterative regularization
- Posterior is Dirichlet
  - Treat it as the new Dirichlet prior
  - The exact same regularization can be applied
  - Let $\alpha_o^{(0)} = \alpha_o$
  - $\forall t > 0$:

$$\alpha_o^{(t)} - 1 = \frac{1}{S_o} \left( \alpha_o^{(t-1)} - 1 + \sum_{o' \in N(o)} S(o, o') \left( \alpha_{o'}^{(t-1)} - 1 \right) \right)$$
Parameters Learning

- Parameters
  - $T$, number of iterations
  - $\Lambda = \{\lambda_\tau: \forall \tau\}$
- We can minimize a global error function on labeled data
  - Distance between the predicted distribution and the gold standard distribution derived from the labels
  - Expensive to compute for $T \geq 2$
- Use an iterative optimization process instead
  - Dynamically update parameters in each iteration
  - 1) **Regularization step:**
    - Update model using parameters learnt from the previous iteration
  - 2) **Minimization step:**
    - Find parameters by minimizing a local error function

\[ S(o, o') = \sum_{\tau} \lambda_\tau W(o, o', \tau) \]
Outline

• Problem and motivation
• Regularization framework
• Applications in IR
• Experiments
• Conclusion
Realization of Framework

• Requires a vertex model and an edge model

• Vertex model
  • Need an initial Dirichlet prior $\text{Dir} \left( \alpha_o^{(0)} \right)$ for each object at $t = 0$
  • $\alpha_o^{(0)} = \sigma_o^{(0)} \cdot m_o^{(0)} + 1$
  • Can equivalently set $\alpha_o^{(0)}$ by initializing $\sigma_o^{(0)}$ and $m_o^{(0)}$ separately

• Edge model
  • Define an edge weight function for each pairwise feature $\tau$
    \[ W(o, o', \tau) \]
  • Recall that there may exist multiple edges between two objects
Example: query intent

• Query intent classification in the shopping domain
  • Map a query to a predefined product category
• Vertex model
  • Mode initialization
    • Any classification method
    • Unigram model based on a product database (weakly supervised)
      \[ p(\theta_i|q) \propto p(q|\theta_i)p(\theta_i) \]

<table>
<thead>
<tr>
<th>Title</th>
<th>Description</th>
<th>Brand</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD1000 Camera</td>
<td>A digital camera</td>
<td>Canon</td>
<td>digital camera</td>
</tr>
<tr>
<td>15 inch laptop</td>
<td>A laptop for</td>
<td>Dell</td>
<td>laptop</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

• Confidence initialization
  • Background unigram model
  • Heuristic: lower background likelihood \(\rightarrow\) higher confidence
Example: query intent

- Two edge models for two pairwise feature
- Lexical pairwise feature
  - A simple binary similarity
  - 1 if one of the query contains all the words in the other query
  - 0 otherwise
- Co-click pairwise feature
  - More co-clicks $\rightarrow$ higher similarity (like tf)
  - Popular clickthroughs contribute less (like idf)
- Other potential edge models
  - Co-session, search results, user profiles
Outline

• Problem and motivation
• Regularization framework
• Applications in IR
• Experiments
• Conclusion
Experiment Setup

• Query intent classification using a shopping query dataset
  • Map a shopping query to a product category
• Dataset
  • # product categories: 2043
  • # all queries: 4 millions
  • # of labeled training queries: 1K (default)
  • # of labeled testing queries: ≥ 10K
  • # clickthroughs: 11 millions
  • # queries with clicks: 1 million (about ¼)
• Metrics
  • Top-$k$ accuracy
  • Precision-recall plot
  • Optimal f-score
  • Precision at 0.5 recall
Illustrative results

- Classification of two example queries using unigram model

<table>
<thead>
<tr>
<th></th>
<th>Misclassified</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>canon 35</td>
<td>camcorder</td>
<td>camera-lens</td>
</tr>
<tr>
<td>hp laptop hard drive</td>
<td>laptop</td>
<td>hard-drive</td>
</tr>
</tbody>
</table>

- The actual classes can be predicted using their neighbors
  - Look at the lexical neighbors of “canon 35”
    - canon 35 mm lens
    - canon 35 f 2
    - 35 mm wide angle 1.4 canon lens
  - Look at the co-click neighbors of “Hp laptop hard drive”
    - hard drive 1tb
    - seagate harddrive
    - western digital 2tb external
Heterogeneous Pairwise Features

- unigram
- unigram + click
- unigram + lex
- unigram + lex + click

Precision vs. Recall graph with different feature combinations.
Queries without clicks

- “Click” alone has no effect
- “Lex + Click” performs better than “Lex” alone
- **Even queries without clicks can benefit from co-click features**
  - Their lexical neighbors (or neighbors of neighbors) may have clicks
  - Iterative regularization helps propagate the evidence from those clicks

![Bar chart showing top-3 accuracy](image)
Confidence

- **NoConf**: no confidence information
- **Heuristic**: the heuristic method using the background model
- **Simulated**: generate confidence using available labels
Labeled and unlabeled data

- # labeled training queries
- # total queries (using the same 1000 training queries)
Outline

• Problem and motivation
• Regularization framework
• Applications in IR
• Experiments
• Conclusion
Conclusion

- We observe the benefits of:
  - Regularization using heterogeneous pairwise features
  - Regularization with confidence

- We may further improve performance by:
  - Exploring more pairwise features like query sessions, etc.
  - Better confidence estimation

- Can be applied to other classification tasks in IR
  - E.g. Text categorization
  - Using pairwise features such as co-readership, social tagging overlap, document similarity, etc.