Fast Supervised Dimensionality Reduction Algorithm with Applications to Document Categorization and Retrieval

George Karypis
Eui-Hong (Sam) Han
University of Minnesota

Presented by Jason Keller
Contents

• Representation of documents
• Dimensionality reduction via clustering
• Interpretation as Concept Indexing
• Experimental results
• Conclusions
Vector Space Modeling of Documents

• *tf-idf* representation
  – Represent each document as a vector of term frequencies, $tf_i$
  – Multiply by $\log(N / df_i)$, where $N$ is the total number of documents, and $df_i$ is the document frequency
  – $d_{tf-idf} = (tf_1 \log(N / df_1), \ldots, tf_n \log(N / df_n))$
  – Normalize all vectors to length 1
Vector Space Modeling of Documents

• Given a set $S$ of documents and corresponding vector representations,

$$\vec{C} = \frac{1}{|S|} \sum_{d \in S} \vec{d}$$

is the centroid of supporting set $S$.

• Document similarity is measured by cosine correlation

• Major result: $\|\vec{C}\|_2^2$ measures pairwise similarity between documents in $S$
Unsupervised Dimensionality Reduction

- Partition the collection of documents into \( k \) disjoint sets
- Find the centroid for each disjoint set
- Scale the centroids to unit length
- Let \( C \) be the matrix in which the \( i \)th column corresponds to the \( i \)th centroid
- Apply \( C \) to the document vectors to project them into \( k \)-dimensional space
Supervised Dimensionality Reduction

- Modification of unsupervised dimensionality reduction
- Given $j$ document classes, compute a $j$-way clustering using the classes
- If $k$ dimensions are desired for the dimensionality reduction and $k > j$, clusters will be partitioned in increasing order of similarity
- Agglomerative clustering can be used if $k < j$, but may combine unlike concepts
Concept Indexing

- Largest values in centroids correspond to the most important terms associated with a concept
Observations

• Few high-weight terms in each centroid; these terms can act as keywords to describe concepts
• High-weight terms are synonymous or closely related to the classes they represent
• Terms in each document vector correspond to how well the document matches the information in each centroid
• Representation captures latent associations between terms that describe concepts
Experiments

- Multi-class Categorization: CI versus higher-dimensional document space
- Single-class Categorization: Compares CI with Naïve Bayes, LSI, and higher-dimensional document space representation
- Query Retrieval: CI versus LSI
Results: Multi-Class Categorization

- Measured in terms of microaveraged Precision/Recall Breakeven Point

<table>
<thead>
<tr>
<th>Topic</th>
<th>kNN</th>
<th>CI-kNN</th>
<th>SVM</th>
<th>CI-SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>earn</td>
<td>97.10</td>
<td>97.40</td>
<td>98.46</td>
<td>98.45</td>
</tr>
<tr>
<td>acq</td>
<td>91.00</td>
<td>92.60</td>
<td>92.89</td>
<td>92.35</td>
</tr>
<tr>
<td>money-fx</td>
<td>77.40</td>
<td>82.10</td>
<td>76.26</td>
<td>82.32</td>
</tr>
<tr>
<td>grain</td>
<td>85.40</td>
<td>89.20</td>
<td>92.66</td>
<td>93.83</td>
</tr>
<tr>
<td>crude</td>
<td>85.50</td>
<td>88.60</td>
<td>87.83</td>
<td>88.76</td>
</tr>
<tr>
<td>trade</td>
<td>74.80</td>
<td>81.80</td>
<td>76.32</td>
<td>80.00</td>
</tr>
<tr>
<td>interest</td>
<td>72.10</td>
<td>78.40</td>
<td>68.80</td>
<td>76.07</td>
</tr>
<tr>
<td>ship</td>
<td>81.30</td>
<td>85.60</td>
<td>83.79</td>
<td>87.20</td>
</tr>
<tr>
<td>wheat</td>
<td>80.30</td>
<td>80.00</td>
<td>83.33</td>
<td>87.14</td>
</tr>
<tr>
<td>corn</td>
<td>78.40</td>
<td>78.90</td>
<td>85.15</td>
<td>84.87</td>
</tr>
<tr>
<td>microaverage</td>
<td>83.13</td>
<td>86.10</td>
<td>85.15</td>
<td>87.62</td>
</tr>
</tbody>
</table>

Table 2: Precision/Recall breakeven point on the ten most frequent Reuters topics and microaveraged performance over all Reuters topics.
Results: Single-Class Categorization

- \( k \)-NN and C4.5 using CI performed about 3-7% better in terms of accuracy than using higher-dimensional space
- CI clearly outperformed LSI
- Was comparable in accuracy to Naïve Bayes
### Results: Single-Class Categorization

Table 4: The classification accuracy of the original and reduced dimensional data sets.
Query Retrieval

• Found the $k$ nearest neighbors for each document $d$ in original and reduced space (LSI and CI)
• Counted the number of neighbors belonging to the same class as $d$
• Summed up counts over all documents in each class
• Compared retrieval improvements: ratio of recalled documents in reduced space to recalled documents in original space
Results: Query Retrieval

• CI improved retrieval and outperformed LSI in all classes
• CI performed well regardless of class size; LSI did worse with smaller classes
Table 5: The per-class RI measures for various data sets for supervised dimensionality reduction. The first column shows the number of documents in each class.
Conclusions

• CI provides a method of dimensionality reduction which performs better in classification accuracy than traditional methods
• CI also improves recall performance
• For more information:
  – www.cs.umn.edu/~karypis
  – Karypis has a longer paper on CI: www-users.cs.umn.edu/~karypis/publications/Papers/PDF/ci.pdf