Intelligent Data Analysis

Mining Textual Data

Peter Tiňo
School of Computer Science
University of Birmingham

Representing documents as numerical vectors

Use a special set of terms $\mathcal{T} = \{t_1, t_2, ..., t_T\}$.

Represent documents from $\mathcal{D} = \{d_1, d_2, ..., d_N\}$ by checking, for each document $d_j \in \mathcal{D}$, the "occurrence" of each of terms $t_k \in \mathcal{T}$ in d_j.

The intensity of occurrence of a term t_k in document d_j is quantified by a number (weight) $x_{k,j} \in \mathcal{R}$.

Each document d_j, $j = 1, 2, ..., N$, is then represented by a "document vector" $\mathbf{d}_j$ summarizing contributions of all terms

$$\mathbf{d}_j = (x_{1,j}, x_{2,j}, ..., x_{T,j})' \in \mathcal{R}^T.$$

What are the terms t_k?

Terms are

1. "intelligently" picked words that are useful to check in a given application (Bag of Words).
   - Bag of words can vary from application to application.
   - More sophisticated representations are often not necessary. Bag of words works nicely in many applications. Can you think why?

2. Phrases (rather than individual words). Results not very encouraging. Can you think why?

3. A combination of the two. A promising direction.

Calculating the weights $x_{k,j}$

Usually $x_{k,j} \in [0, 1]$.

1. Binary weights:
   $x_{k,j} = 1$, if term t_k is present in document d_j; $x_{k,j} = 0$, otherwise.

2. Real valued weights:
   - The more frequent t_k is in d_j, the more it is representative of its content.
   - The more documents t_k occurs in, the less discriminating it is.

$$x_{k,j} = \text{tfidf}(t_k, d_j) = N_{k,j} \cdot \log \frac{N}{N_k},$$

where

- N - number of documents in $\mathcal{D}$
- N_k - number of documents in $\mathcal{D}$ in which term t_k occurs
- $N_{k,j}$ - number of times t_k occurs in document d_j

The weights are often normalized to unit length:

$$x_{k,j} \leftarrow \frac{x_{k,j}}{\|x_{k,j}\|}.$$

Now, $x_{k,j} \in [0, 1]$.

Document retrieval/clustering/classification

Work directly in the document representation space!

Document retrieval:
Query of terms: q - view as a "mini document" $\mathbf{q} \in \mathcal{R}^T$.

Compare with documents $\mathbf{d}_j$ in $\mathcal{D}$ along the term directions using e.g. a similarity measure

$$\cos\alpha = \frac{\mathbf{q}'\ \mathbf{d}_j}{\|\mathbf{q}\| \cdot \|\mathbf{d}_j\|}$$

Output the best matching documents from $\mathcal{D}$.

Think of document clustering and classification...

This is all nice, but ...

... the data is extremely sparse!

Typically we can expect 1000's of terms and 10000's of documents.

Think of the situation in terms of a set of N document representations in a T-dimensional space. How can we hope for a reasonable statistics?

In addition, there is yet another problem ...

Synonymy and Polysemy

Synonymy - different authors describe the same idea using different words (terms).

Polysemy - the same word (term) can have multiple meanings.

But maybe we do not need all the terms! Need to detect a smaller set of dominant abstract contexts in which the terms tend to co-occur in the document collection $\mathcal{D}$ and then concentrate only in these ...

... sounds like Principal Component Analysis and dimensionality reduction :-)

Latent Semantic Analysis (LSA)

As usual, collect all the document representations $\mathbf{d}_j$, $j = 1, 2, ..., N$, in a $T \times N$ (design) matrix $\mathcal{X} = (x_{k,j})$.
Documents $\mathbf{d}_j \in \mathcal{R}^T$ (data points) are stored as columns.

Scaled "covariance" (may not be zero-mean!) matrix of documents (in the term space):
$\mathcal{C} = \mathcal{X}\mathcal{X}'$.

Eigenvectors + eigenvalues of $\mathcal{C}$:
$\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_T \in \mathcal{R}^T$
$\lambda_1, \lambda_2, ..., \lambda_T \in \mathcal{R}$

Pick only the ℓ most significant principal (prototypical) directions (documents) $\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_\ell$ (concepts) corresponding to the highest $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_\ell$.

Form a concept matrix $\mathcal{U} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_\ell]$
(principal documents $\mathbf{u}_k$ are columns).

Project each document representation $\mathbf{d}_j \in \mathcal{R}^T$ onto the low(ℓ)-dimensional representation of d_j in the concept space

$$\bar{\mathbf{d}}_j = \mathcal{U}' \ \mathbf{d}_j \in \mathcal{R}^\ell$$

The amount of variation (scale) along projection direction $\mathbf{u}_k$ is proportional to λ_k.

To make up for different scalings we rescale projection along $\mathbf{u}_k$ by λ_k^{-1}:

$$\bar{\mathbf{d}}_j = \Lambda^{-1} \, \mathcal{U}' \, \mathbf{d}_j,$$

where

$$\Lambda = \mathsf{diag}\{\lambda_1, \lambda_2, ..., \lambda_\ell\}.$$

Document retrieval/clustering/classification through LSA

Instead of T-dim document representation space through original terms, represent documents in the ℓ-dim concept space!

Document retrieval:

Query of terms: q - view as a "mini document" $\mathbf{q} \in \mathcal{R}^T$.

Its representation in the concept space: $\bar{\mathbf{q}} = \Lambda^{-1} \mathcal{U}' \mathbf{q}$.

Compare with documents $\bar{\mathbf{d}}_j$ along the concept directions using e.g. a similarity measure

$$\cos\alpha = \frac{\bar{\mathbf{q}}' \bar{\mathbf{d}}_j}{\|\bar{\mathbf{q}}\| \cdot \|\bar{\mathbf{d}}_j\|}$$

Output the best matching documents from $\mathcal{D}$.

Think of document clustering and classification...

Reason about terms

An alternative view:

Given a term t_k, its "meaning" is captured by the document collection $\mathcal{D}$ through the "term vector" $\mathbf{t}_k$ summarizing contributions from all documents

$$\mathbf{t}_k = (x_{k,1}, x_{k,2}, ..., x_{k,N})' \in \mathcal{R}^N.$$

Collect all the term representations $\mathbf{t}_k$, $k = 1, 2, ..., T$, in a $N \times T$ (design) matrix $\mathcal{X}' = (x_{j,k})$.
Terms $\mathbf{t}_k \in \mathcal{R}^N$ (data points) are stored as columns of $\mathcal{X}'$.

Scaled "covariance" matrix of terms (in the document space): $\mathcal{C}' = \mathcal{X}'\mathcal{X}$.

Eigenvectors + eigenvalues of $\mathcal{C}'$:
$\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_N \in \mathcal{R}^N$
$\lambda'_1, \lambda'_2, ..., \lambda'_N \in \mathcal{R}$

Note: because $T < N$,
$\lambda'_j = \lambda_j$ for $j = 1, 2, ..., T$
$\lambda'_j = 0$, for $j = T+1, T+2, ..., N$.

Pick only the ℓ most significant principal (prototypical) directions (term meanings) $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_\ell$ (term concepts) corresponding to the highest $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_\ell$.

Form a term concept matrix $\mathcal{V} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_\ell]$
(principal term meanings $\mathbf{v}_j$ are columns).

Project each term meaning representation $\mathbf{t}_k \in \mathcal{R}^N$ onto the low(ℓ)-dimensional representation of t_k in the term concept space

$$\bar{\mathbf{t}}_k = \Lambda^{-1} \mathcal{V}' \mathbf{t}_k \in \mathcal{R}^\ell$$

It is possible now e.g. to cluster the terms according to their underlying meaning using e.g. a similarity measure

$$\cos\alpha = \frac{\bar{\mathbf{t}}_r \bar{\mathbf{t}}_k}{\|\bar{\mathbf{t}}_r\| \cdot \|\bar{\mathbf{t}}_k\|}$$