

# Vector model

## Distributive semantics

## Dimension reduction

Stanislav Protasov

# Agenda

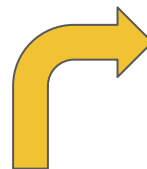
- Vector interpretation of boolean query
- Distributive semantics
- Dimension reduction and LSA

| Term document matrix |           |           |            |
|----------------------|-----------|-----------|------------|
| words\documents      | Document1 | document2 | query term |
| cat                  | 1         | 1         | 0          |
| runs                 | 1         | 1         | 0          |
| behind               | 1         | 1         | 0          |
| rat                  | 1         | 0         | 1          |
| dog                  | 0         | 1         | 0          |

# Term-document matrix

*TDM* — describes the **frequency of terms** that occur in a collection of documents.

**term-document matrix** ... **documents are the columns** and **terms are the rows** (Wiki)



| Term \ Document | information technology | information system | communication technology | software application | telecommunication | computer science |
|-----------------|------------------------|--------------------|--------------------------|----------------------|-------------------|------------------|
| 1               | 0.77                   | 0.55               | 0.45                     | 0.13                 | 0.14              | 0.15             |
| 2               | 0                      | 0                  | 0.13                     | 0.53                 | 0.15              | 0.75             |

In a *document-term matrix* ( $DTM = TDM^T$ ), rows correspond to documents in the **collection** and **columns correspond to terms**

- Column is a description (**vector**, BoW) of a document
- Row is a vector representation of a word
- **Sparse** for short texts

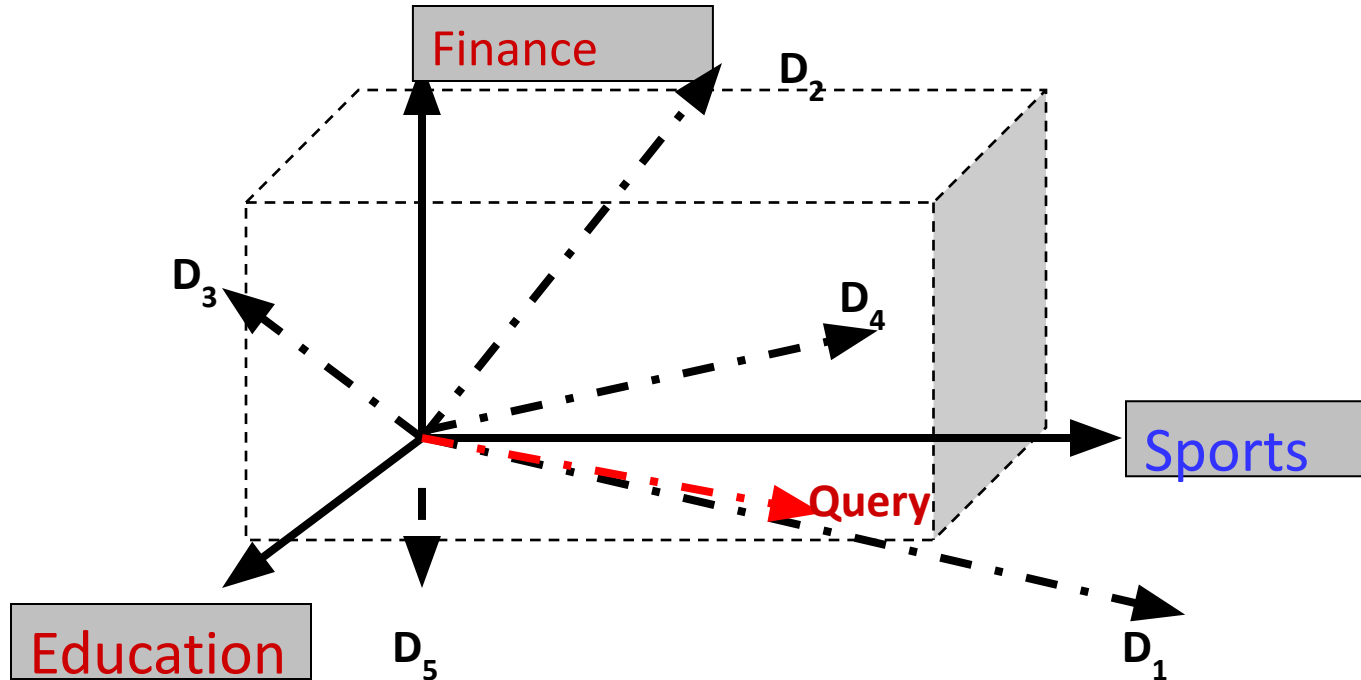
| Venue     | CAMAD | EUNICE | HAISA | HPCC-ICESS | IJESMA | ISCA | KMIS | NMR | SPRINGL | SSV |
|-----------|-------|--------|-------|------------|--------|------|------|-----|---------|-----|
| algorithm | 2     | 8      | 0     | 24         | 0      | 5    | 0    | 2   | 1       | 1   |
| cellular  | 2     | 1      | 0     | 1          | 0      | 0    | 0    | 0   | 0       | 0   |
| game      | 1     | 1      | 0     | 1          | 0      | 0    | 1    | 1   | 0       | 0   |
| hardwar   | 1     | 0      | 1     | 4          | 0      | 18   | 0    | 0   | 1       | 0   |
| internet  | 2     | 6      | 2     | 0          | 2      | 0    | 0    | 0   | 0       | 0   |
| mobil     | 10    | 8      | 0     | 6          | 17     | 5    | 2    | 0   | 2       | 0   |
| network   | 58    | 60     | 4     | 38         | 2      | 25   | 12   | 0   | 3       | 0   |
| search    | 0     | 1      | 0     | 1          | 2      | 4    | 1    | 0   | 0       | 0   |
| secur     | 4     | 4      | 29    | 5          | 1      | 12   | 3    | 0   | 4       | 0   |
| web       | 0     | 2      | 0     | 3          | 3      | 1    | 13   | 0   | 2       | 0   |

# Vector space model

- Represents both *doc* and *query* by “concept **vectors**”
  - Each concept defines one **dimension**
  - $K$  concepts define a high-dimensional space
  - Element of vector corresponds to concept weight
    - E.g., for  $d = (x_1, \dots, x_k)^T$ ,  $x_i$  is the “weight” of concept  $i$  (e.g. TF-IDF)
- Measures relevance approximation
  - **Distance** between the query vector and document vector in this concept space
  - $\text{relevance} \approx \text{similarity} = 1 - \text{distance}$
  - How can we define distance?

# VS Model: an illustration

- Which document is the closest to the query?

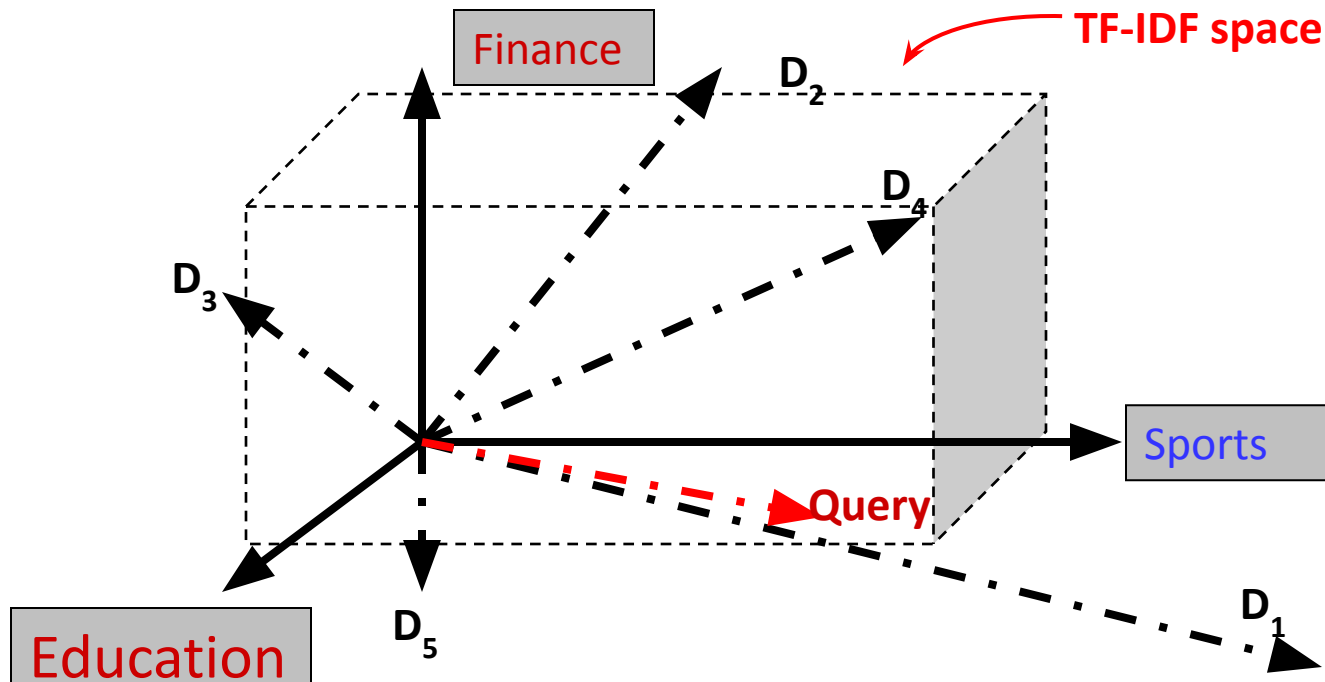


# What the VS model doesn't say

- How to **define**/select the “basic **concept**”
  - Concepts are assumed to be **orthogonal**
- How to assign **weights**
  - *Weight in a query* indicates importance of the concept for a query
  - *Weight in a doc* indicates how well the concept characterizes the doc
- How to define **distance** measure?

# How to define a good similarity measure?

- Euclidean distance?

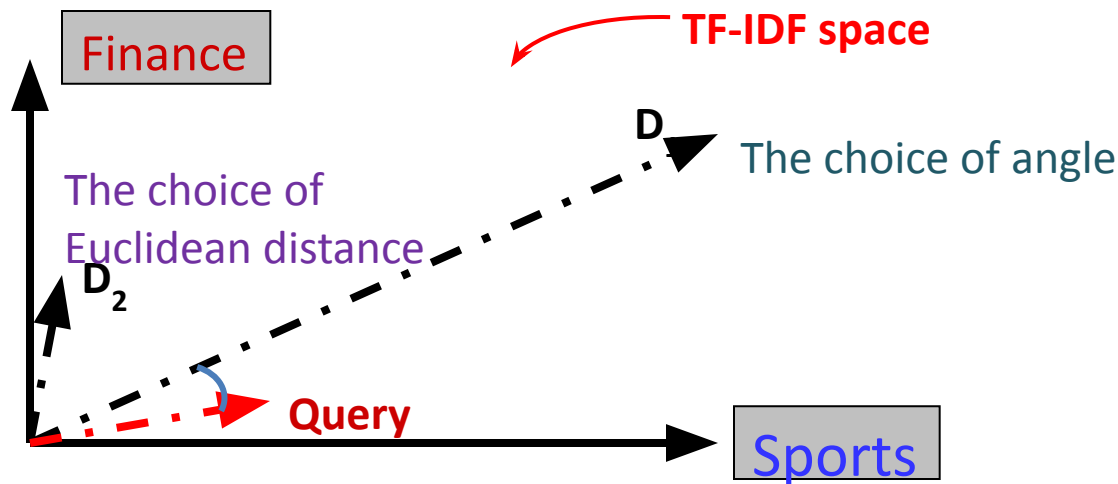


# From distance to angle

**Cosine similarity** – projection of one vector onto another

- $\pm 1$  if vectors are collinear
- 0 if vectors are orthogonal

$$\text{similarity} = \cos(\text{doc}, \text{query}) = \frac{\overrightarrow{\text{doc}} \cdot \overrightarrow{\text{query}}}{\|\overrightarrow{\text{doc}}\| * \|\overrightarrow{\text{query}}\|}$$





# Stop here!

1. We found a way, which allows to **represent any document** (even unseen) **as a vector**.
2. We introduced a **relevance metric** using a simple well-known mathematical concept - **cosine similarity**
3. Still **non-orthogonal** concepts
4. To measure similarity of 2 documents (or doc vs query) we need to do circa **100K arithmetic operations** with floating point numbers

# Reduce dimensions!

- Compression approach #1 — works great for **sparse databases**:

```
max_size = N
```

```
doc_compressed[i % max_size] += doc[i] (or max, or =)
```

- Compression approach #2:
  - [Random projection](#)
  - Or even randomly remove some dimensions!
- Compression approach #3 — [latent semantic analysis](#):
  - [LDA](#), [PCA](#)/SVD, GDA, ...
  - Embedding using encoder networks (BERT, doc2vec, DSSM, ...)

# Distributional semantics

| Term document matrix |           |           |            |
|----------------------|-----------|-----------|------------|
| words\documents      | Document1 | document2 | query term |
| cat                  | 1         | 1         | 0          |
| runs                 | 1         | 1         | 0          |
| behind               | 1         | 1         | 0          |
| rat                  | 1         | 0         | 1          |
| dog                  | 0         | 1         | 0          |

Recall: **word** is just a **vector** in the vector space model

The **distributional hypothesis**:

- linguistic items with similar distributions have similar meanings
- words that are used and occur in the same contexts tend to purport similar meanings
- You shall know a word by the company it keeps (Firth, J. R. 1957)

# Notes on distributional semantics

“*Similar context*” and “*same distribution*” are not well-defined terms. It can go for bag of words model (TDM) or for near-context (as in word2vec).

Distributional hypothesis is a powerful model, which made to happen topic modelling, and all the ML-based NLP.

Consequence:

- If 2 **word**-corresponding **rows** from TDM **correlate**, we fail with orthogonality.
- But we can **infer orthogonal vector** from TDM!

**Hypothesis 2:** Maybe there is a **latent semantic space** of smaller dimension?

# Latent semantic analysis (patent)

Idea: **search for low-rank approximation of TDM!**

What does it mean? By now, query is (assume vectors are **normed**):

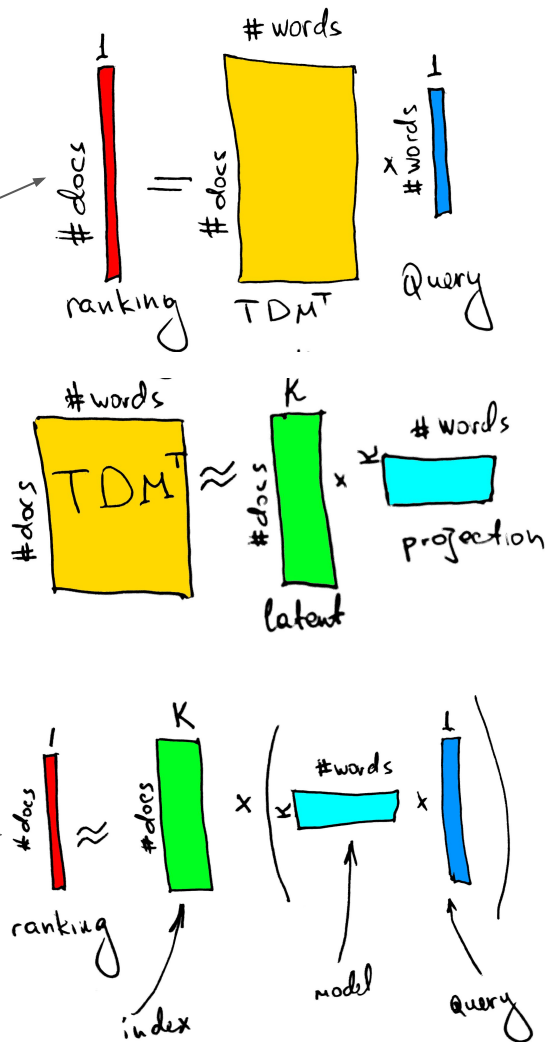
$$\text{Rankings} = \text{TDM}^T * Q_{\text{vector}}$$

What if

$$\text{TDM}^T = \text{LATENT\_MX} * \text{PROJECTION\_MX}$$

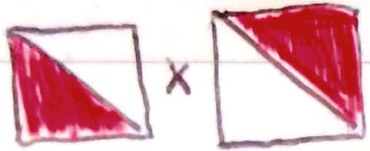
Then

$$\text{Rankings} = \text{LATENT\_MX} * [\text{PROJECTION\_MX} * Q_{\text{vector}}]$$

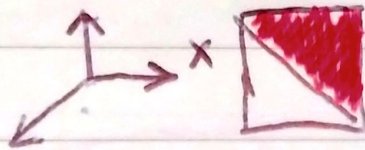


# Decompositions

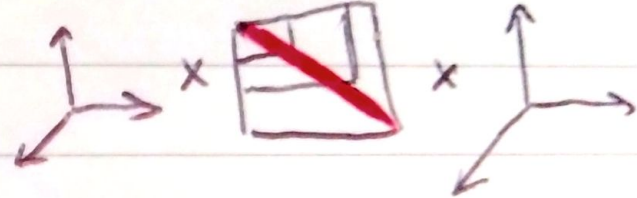
LU



QR



SVD =  $U \Sigma V^T$

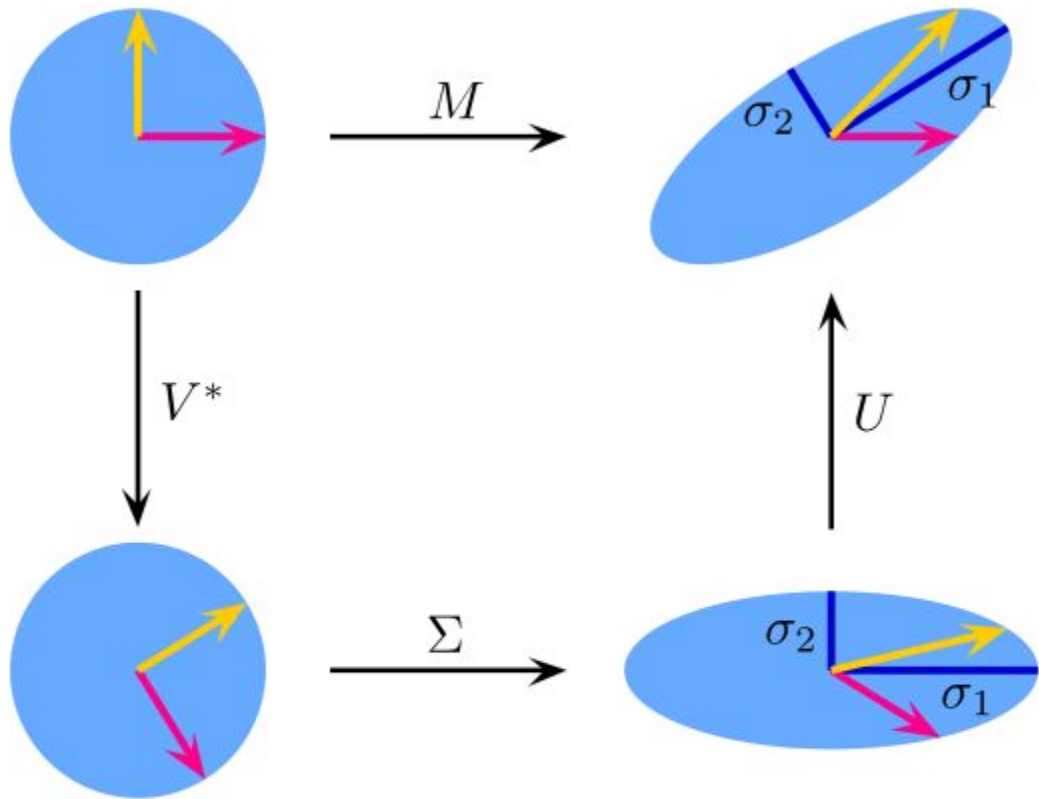


# SVD

$\mathbf{U}$  is eigenvectors for  $\mathbf{M}\mathbf{M}^T$

$\mathbf{V}^T$  is eigenvectors for  $\mathbf{M}^T\mathbf{M}$

$\mathbf{\Sigma}$  is diagonal with square roots of non-negative eigenvalues of  $\mathbf{M}^T\mathbf{M}$



$$M = U \cdot \Sigma \cdot V^*$$

# Matrix approximation

$$M = U_R \Sigma_R V_R^T$$

$$\text{TDM}^T = [U_R \Sigma_R]^* V_R^T$$

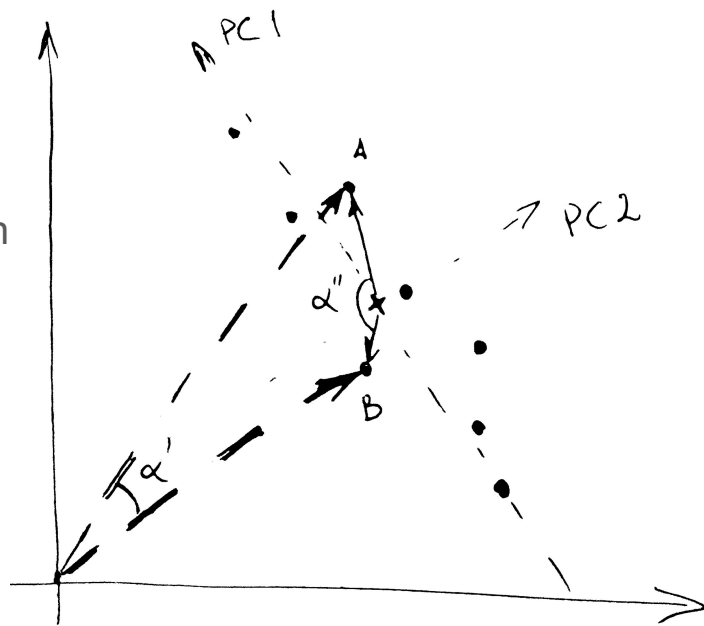


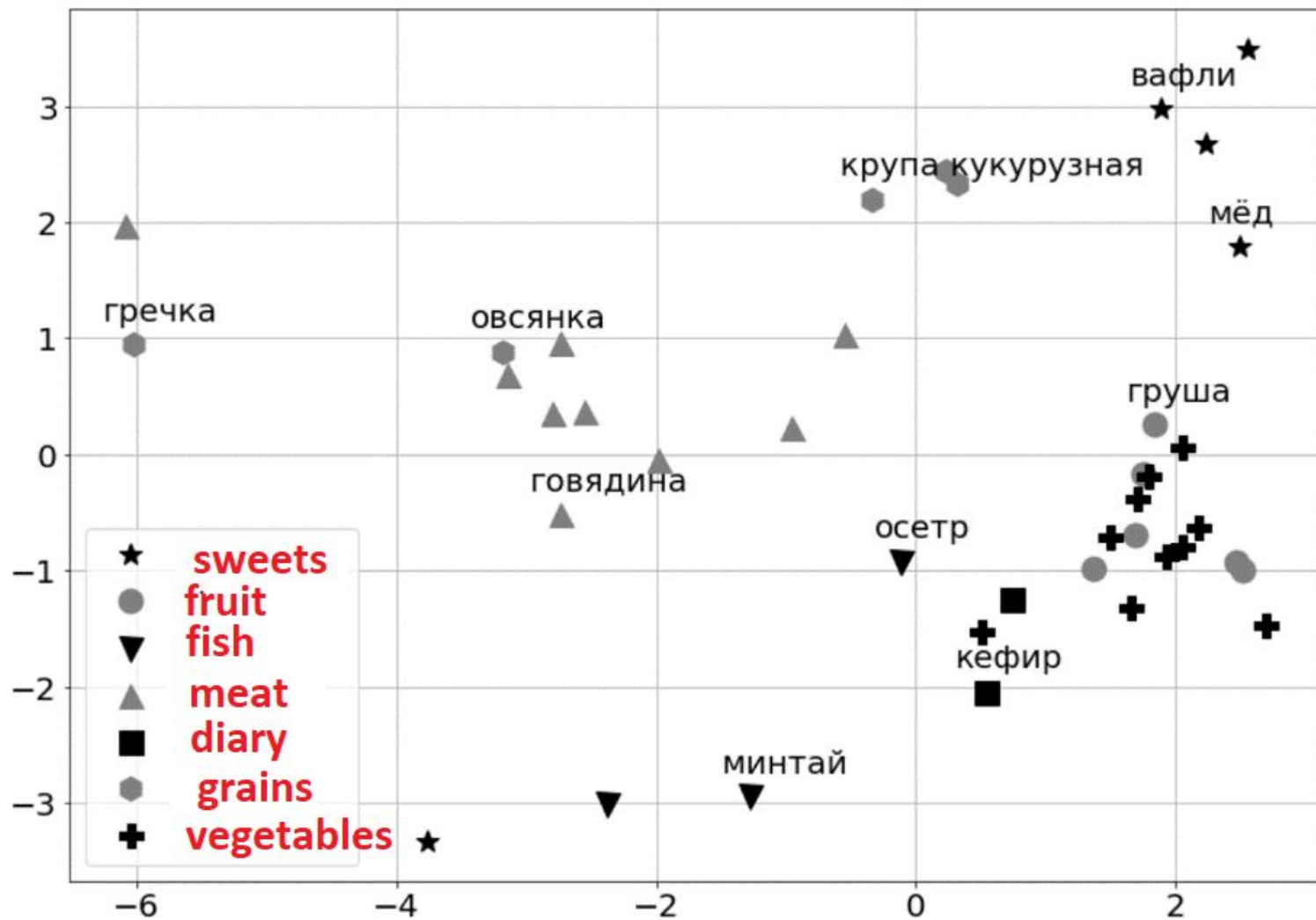
# Principal component analysis (similar idea)

... Convert a set of observations of **possibly correlated variables** (entities each of which takes on various numerical values) **into a set of values** of linearly uncorrelated variables ...  
(wiki)

... the **first principal component** has the **largest possible variance** ... and each succeeding component in turn has the highest variance possible under the constraint that it is orthogon to the preceding components (wiki)

**NB:** Implemented with SVD, PCA requires **data centering** first, which affects the **cosine metric**.





# Stop Here!

1. **Vector space** model is cool, but (1) TDM is sparse (2) concepts are not orthogonal
2. **Distributional hypothesis** gives an insight: words correlate and their distribution defines semantics
3. **Latent semantic analysis** says: yes, and we know that there is a small-dimensional latent space for semantics. TDM is just a **linear projection**
4. **SVD** says: mmmm... We know how this latent space should look like! Orthogonal features + decreasing variance

# Reading

- The Book — chapter 6.2-6.5
- All links in this presentation