Data Mining in Action

Recommender Systems II

26.03.16

How to personalise recommendations?

- Item-to-Item:
  Given a page **all users** observe **the same set** of recommendations on this page.

- **Question**: how to make personalisation?

Читайте также

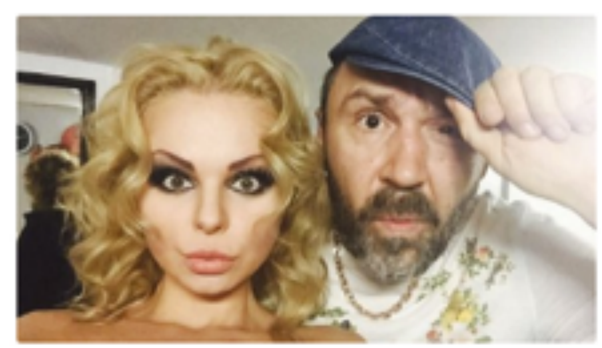
10 минут, которые помогают стать умнее

Где найти бесплатную и свободную музыку для своих проектов?

На чём заработать студенту: 5 нестандартных способов обзавестись деньгами

Секретные места Киева, которые вы не найдёте в типичном путеводителе

Как начать питаться правильно, не изменяя своим привычкам



«Ленинград» расстался с исполнительницей «Экспоната» Алисой Вокс

tjournal.ru

How to personalise recommendations?

- **Approach 1**: Content-Based algorithm for each user.

  Problem: need to store too many records in database (for each pair [page,user]).

How to personalise recommendations?

- Approach 2: User clustering.

  Determine cluster of the user and get recommendations for [page, cluster].

- **Question**: How to clusterize users?

How to personalise recommendations?

Item-to-Item Algorithm

```
For each item in product catalog, I1
  For each customer C who purchased I1
    For each item I2 purchased by
       customer C
      Record that a customer purchased I1
       and I2
  For each item I2
    Compute the similarity between I1 and I2
```

Iterate C only within one cluster!

Clustering for personalisation

- Need to represent users as vectors.
- Can’t compute user-item matrix.
- But we can represent items as vectors!
  (TFIDF, Bag-of-Words)
- If we clusterize items, we can see at user history and determine vector of cluster weights.

Clustering for personalisation

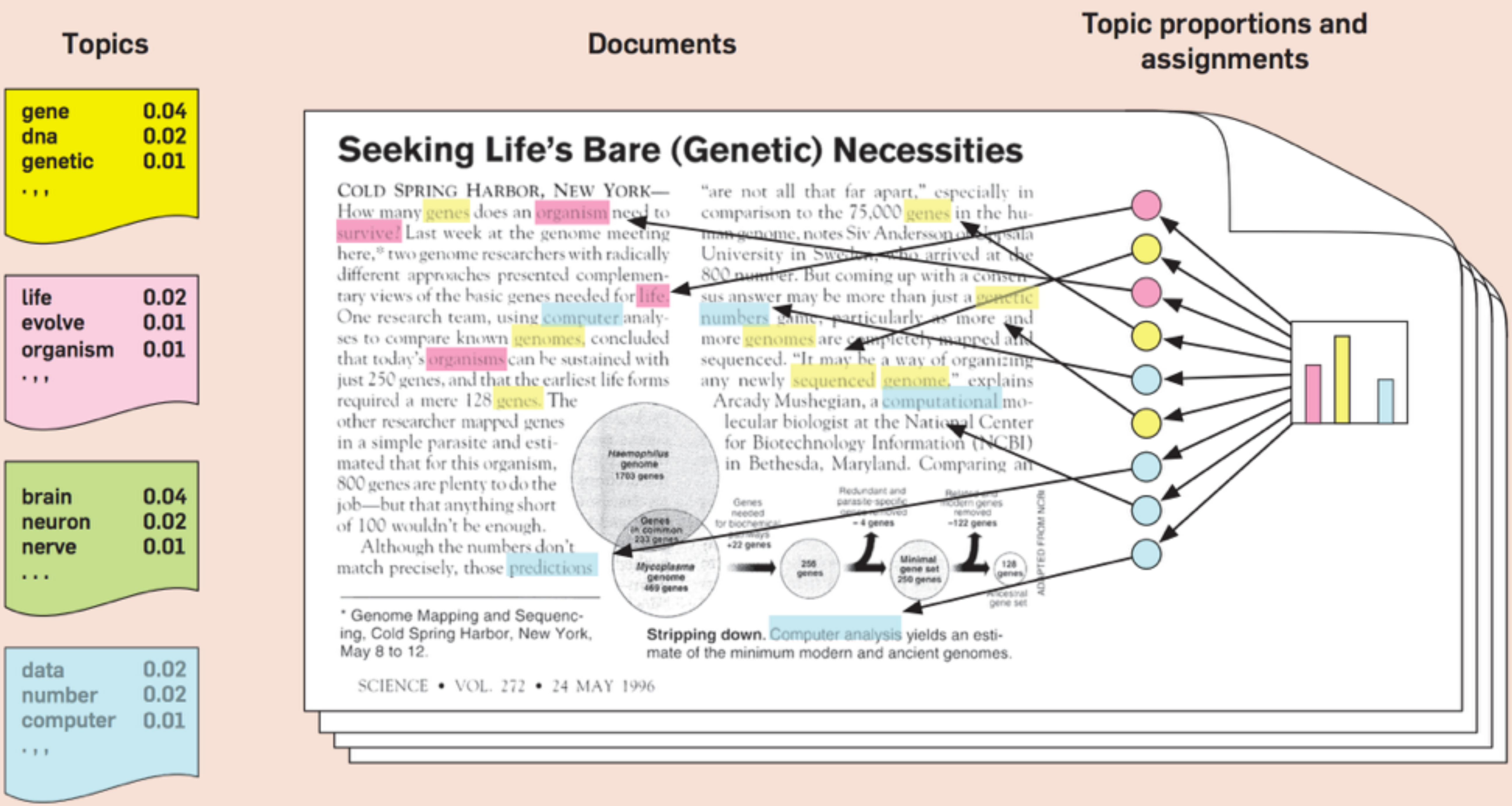
- K-Means
- Hierarchical Clustering
- etc.
- **Problems?**

	w1	w2	w3
i1	5	1	0
i2	3	0	1
i3	2	2	4
i4	1	2	1

Topic Modeling

- **Hypothesis**: there are some topics in our corpus.
- Each topic has distribution over documents $p(t|d)$.
- Each word has distribution over topics $p(w|t)$.

Figure 1. The intuitions behind latent Dirichlet allocation. We assume that some number of "topics," which are distributions over words, exist for the whole collection (far left). Each document is assumed to be generated as follows. First choose a distribution over the topics (the histogram at right); then, for each word, choose a topic assignment (the colored coins) and choose the word from the corresponding topic. The topics and topic assignments in this figure are illustrative—they are not fit from real data. See Figure 2 for topics fit from data.



Topic Modeling

D – corpus of documents.

W – vocabulary.

T – latent variables (topics).

n_{dw}– frequency of word w in document d.

$n_d \equiv \sum_w n_{dw}$– total length of document d.

Topic Modeling

Let's admit *conditional independence hypothesis*: $p(w|t,d) = p(w|t)$.

Using law of total probability:

$$p(w|d) = \sum_{t \in T} p(w|t,d)p(t|d) = \sum_{t \in T} p(w|t)p(t|d) = \sum_{t \in T} \phi_{wt}\theta_{td}. \tag{1}$$

We have matrix $F = (\hat{p}(w|d))_{W \times D}$ – matrix of $p(w|d)$ estimations.

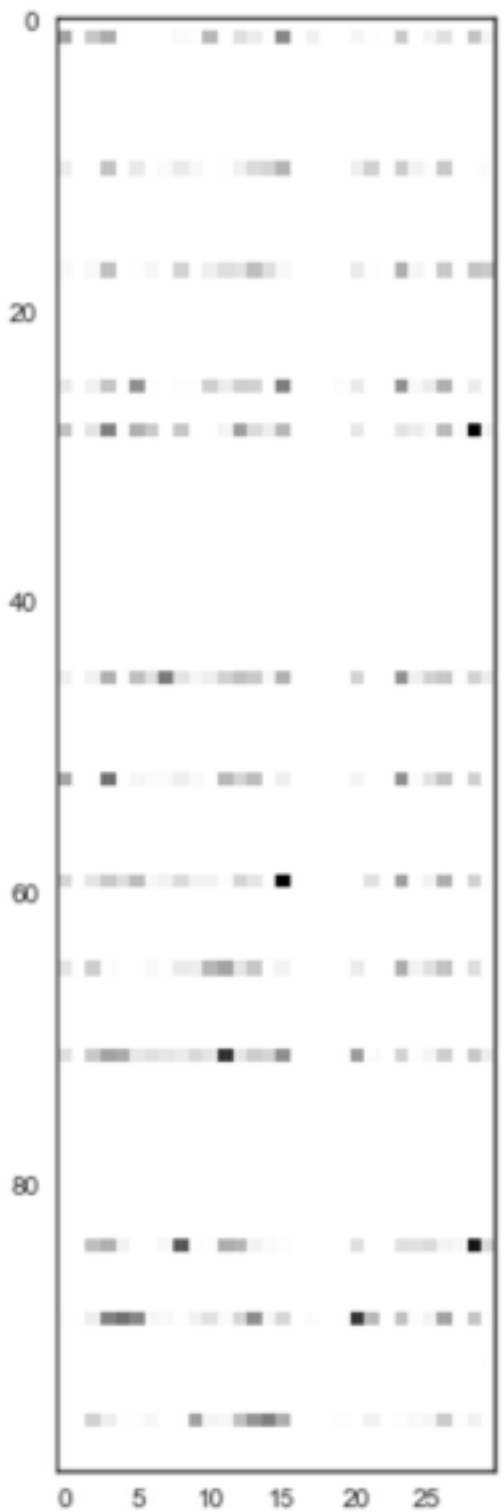
Let $\Phi = (\phi_{wt})_{W \times T}$, $\Theta = (\theta_{td})_{T \times D}$

We need to find Φ and Θ given F.

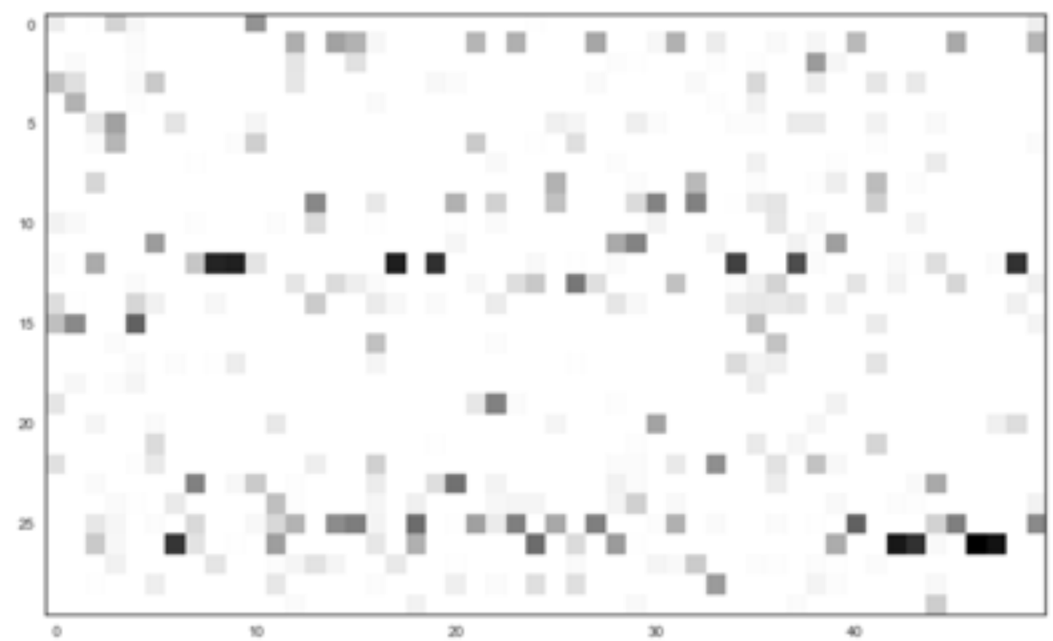
$$\log L(\Phi, \Theta) = \sum_{d \in D} \sum_{w \in W} n_{dw} \log \sum_{t \in T} \phi_{wt}\theta_{td} \longrightarrow \max_{\Phi,\Theta} \tag{2}$$

To maximize objective EM-algorithm can be used.
This algorithm is called PLSA – Probabilistic Latent Semantic Analysis.

Topic Modeling



X



Number of topics = 30

Topic Modeling

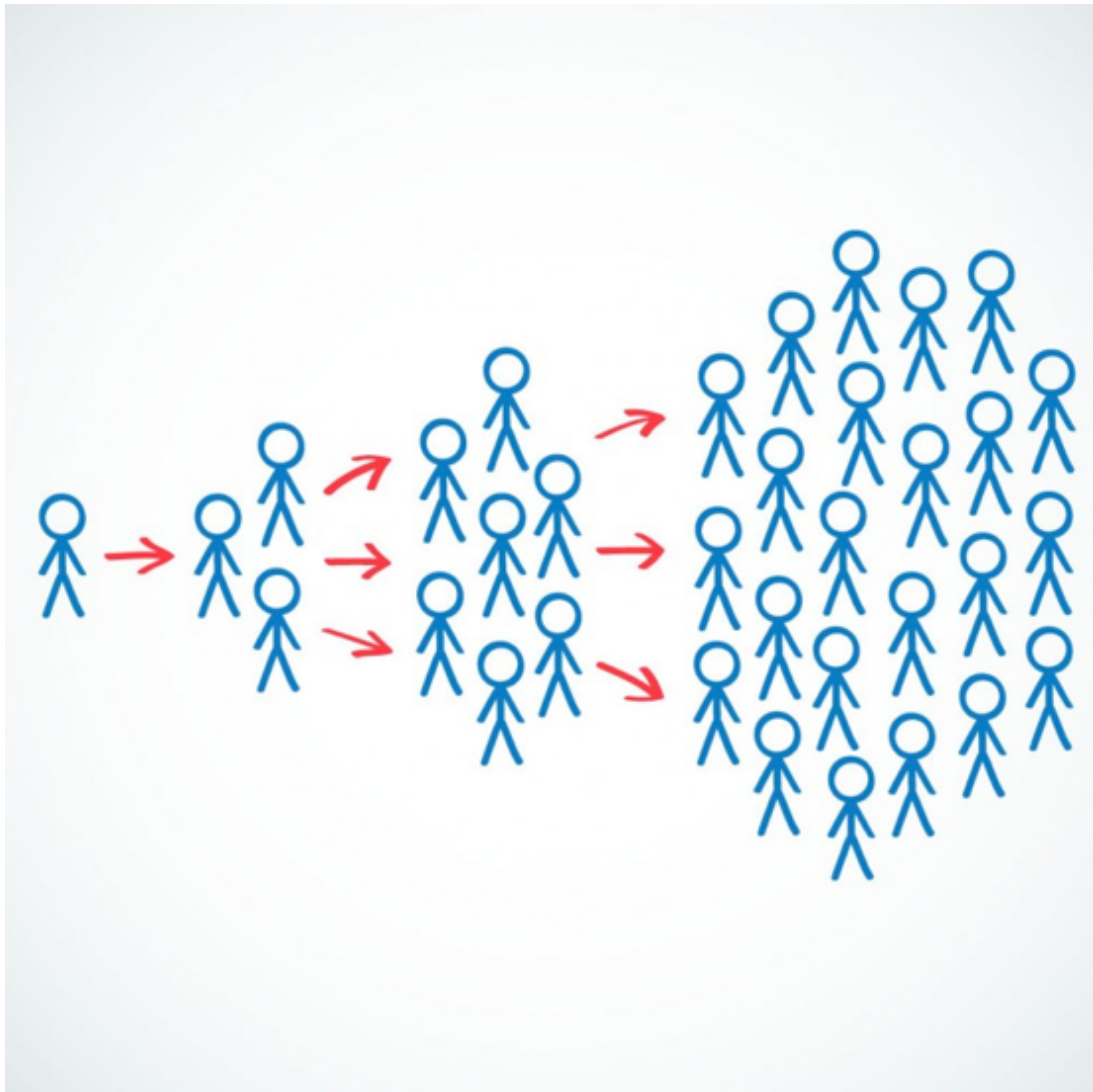
$$\text{Item} = \big(p(t_1|\text{Item}), ..., p(t_T|\text{Item})\big)$$

$$\text{User}_u = (w_1, ..., w_T)$$

$$w_j = \tfrac{1}{|I_u|} \textstyle\sum_{Item \in I_u} p(t_j|\text{Item})$$

Now we can apply clustering algorithm to user vectors.

Virality Prediction



Virality Prediction

- Want to predict viral content
- What is viral?
    - Total number of views is big
    - Number of views within N hours after publication is big

Virality Prediction

- Article-specific features:
    - Publication date (month, day-of-month)
    - Length
    - TF-IDF features
    - ?
- Domain-specific features
    - Average number of views on this domain
    - ?