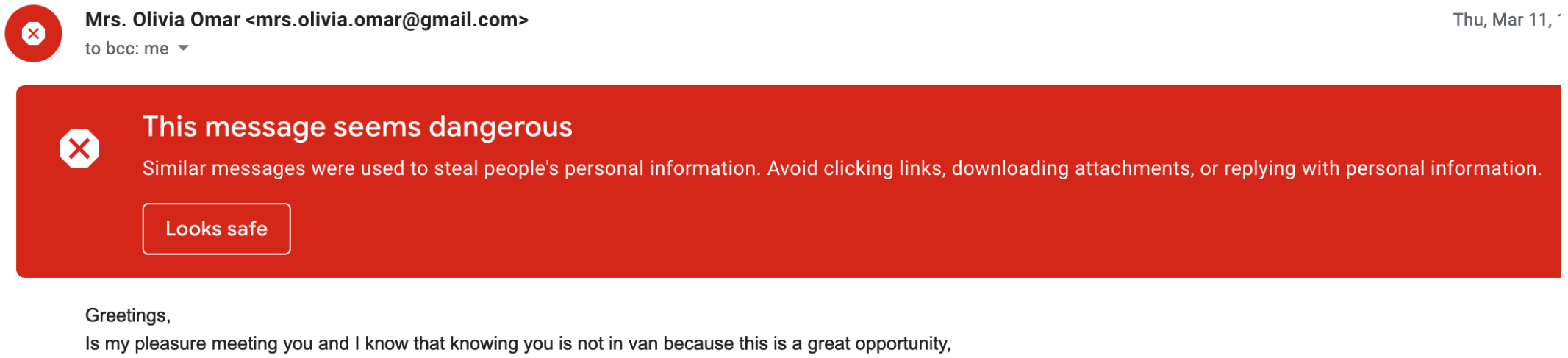
Document Analysis

Spring 2024

Hongchang Gao

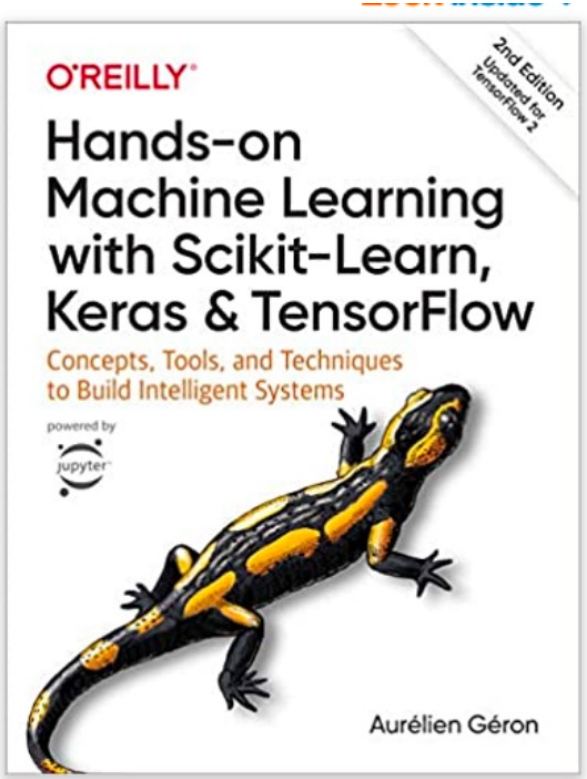
Introduction

- Spam detection



Introduction

- Review Classification



★☆☆☆☆ **Content seems good. Printing quality is not satisfactory.**

Reviewed in the United States on February 1, 2021

Verified Purchase

The content of the book seems pretty good, although I have not gone thoroughly through it. However, the quality of the printing is not satisfactory. In just a quick first inspection I found many blurry pages throughout the book, including figures, and at least 5 pages that were not printed at all, just like that, with the corresponding scripts and pieces of information missing. I tend to buy less books through Amazon day by day, since the quality of books leaves me almost 100% of the times dissapointed.

★★★★★ **Single handedly one of the best ML books on the market**

Reviewed in the United Kingdom on January 4, 2020

Verified Purchase

If you have the budget to only buy one ML book, I would suggest going for this one. It covers most of the field in one book. Get a datacamo subscription too and you can break into the DS career.

Introduction

- What is the subject of this document?

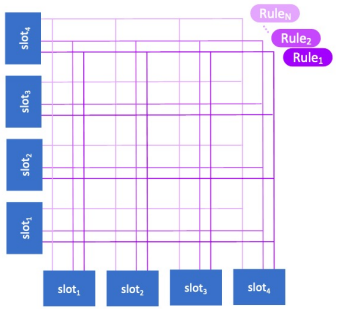


Figure 1. **Rule and slot combinatorics.** Condition-action rules specify how entities interact. Slots maintain the time-varying state of an entity. Every rule is matched to every pair of slots. Through key-value attention, a goodness of match is determined, and a rule is selected along with its binding to slots.

themselves. The ability to represent abstract knowledge allows for the transfer of learning across different environments as long as they fit within the conditions of the given production rule.

Production rules are sparse. In order that production rules have broad applicability, they involve only a subset of entities. This assumption imposes a strong prior that dependencies among entities are sparse. In the context of visual reasoning, we conjecture that this prior is superior to what has often been assumed in the past, particularly in the disentanglement literature—independence among entities (Higgins et al., 2016; Chen et al., 2018).

for instance the frames in a video, are processed by a neural encoder (Burgess et al., 2019; Greff et al., 2019; Goyal et al., 2019b; 2020) applied to each $\boldsymbol{x}^t$, to obtain a set of M entity representations $\{\boldsymbol{V}_1^t, \ldots, \ldots, \boldsymbol{V}_M^t\}$, one for each of the M slots. These representations describe an entity and are updated based on both the previous state, $\boldsymbol{V}^{t-1}$ and the current input, $\boldsymbol{x}^t$.

NPS consists of N separately encoded rules, $\{\boldsymbol{R}_1, \boldsymbol{R}_2, .., \boldsymbol{R}_N\}$. Each rule consists of two components, $\boldsymbol{R}_i = (\vec{\boldsymbol{R}}_i, MLP_i)$, where $\vec{\boldsymbol{R}}_i$ is a learned rule embedding vector, which can be thought of as a template defining the condition for when a rule applies; and MLP_i, which determines the action taken by a rule. Both $\vec{\boldsymbol{R}}_i$ and the parameters of MLP_i are learned along with the other parameters of the model using back-propagation on an end-to-end objective.

As we describe in detail in the next section, rules are selected and applied one at a time, with a sequence of K rule applications (*stages*) per time step. In the general form of the model, each rule has conditions and actions that are specified on a pair of entities, meaning that there are NM^2 possible rule-slot bindings to consider at each stage. To reduce the complexity of the search, we assume that the two slots are asymmetric: one slot is *primary* in that it is used both to match the rule condition and it is acted on by the rule; the other slot is *contextual* in that it determines how the primary slot is acted upon. With this set up, we perform selection in two operations, first considering all combinations of {rule, primary slot}, making a selection, and then conditioned on the selection, choosing a contextual slot. The resulting search is reduced to $NM + M$ combinations.

- Machine Learning
- Computer Vision
- Natural Language processing
- Robotics
- …

Introduction

- Document Classification
  - Spam detection
  - Review classification
  - Sentiment analysis
  - …
- Document Clustering
  - Topic discovery
  - …

Document Classification

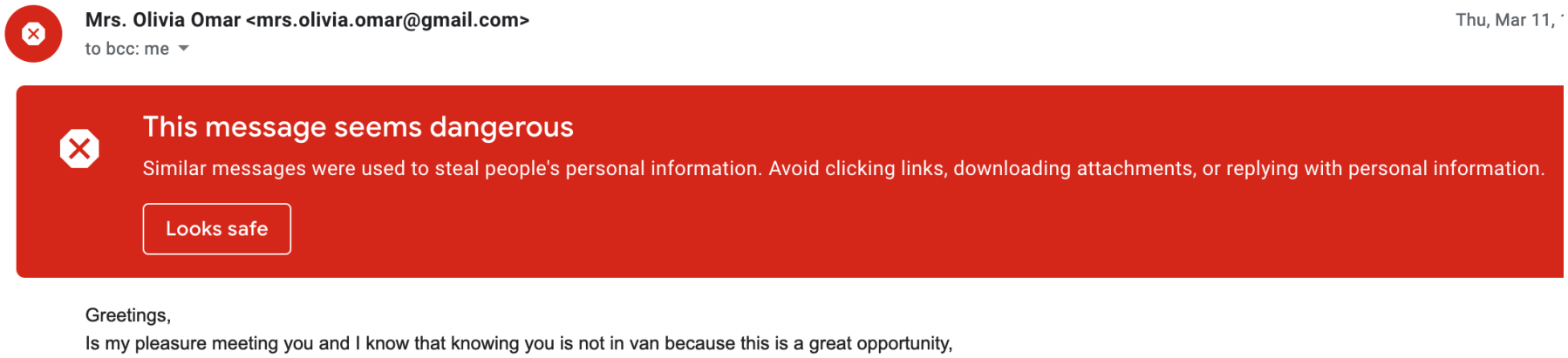
- Given n samples: $\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$

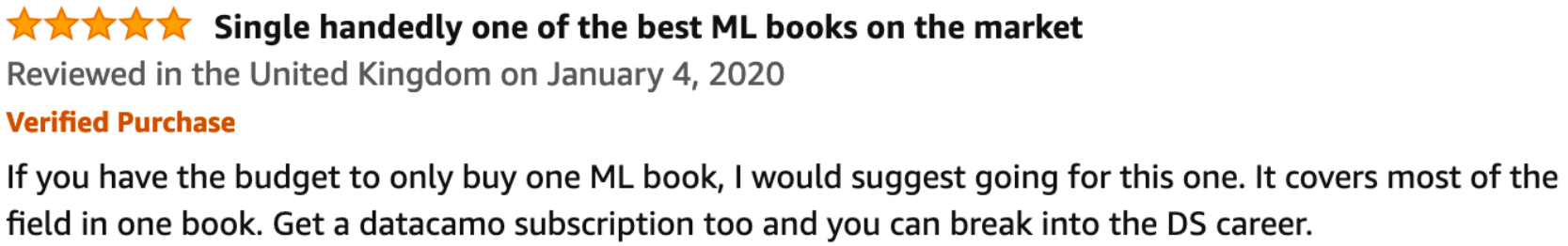
  Learn a mapping function $x_i \xrightarrow{f(x)} y_i$

- Each sample could be:
  - An email (spam detection)
  - A paragraph (review classification)
  - An article (topic classification)
  - ….

How to represent the text?

- The words in a sentence are not numerical values!





How to represent the text?

- Convert categorical features into numerical values
  - Label encoding
  - One-hot encoding
  - Ordinal encoding

- How to convert words into numerical values?
  - Each sentence/paragraph/article contains multiple words
  - Bag-of-words!

The Bag-of-Words Representation

- BoW can represent a sentence/paragraph/article as a bag of words vector.

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…

The Bag-of-Words Representation

- 1. Build the vocabulary/dictionary from the given dataset
  - Get all the unique words in the given dataset
  - Each word in the vocabulary has an index

It was the best of times,
it was the worst of times,
it was the age of wisdom,
it was the age of foolishness,

The given dataset.
(Each sentence is a sample)

Get unique words

- "it"
- "was"
- "the"
- "best"
- "of"
- "times"
- "worst"
- "age"
- "wisdom"
- "foolishness"

Vocabulary/dictionary
(unique words in the given dataset)

The Bag-of-Words Representation

- 2. Represent each sentence/paragraph/article with the vocabulary
  - Use a vector whose dimensionality equals to the size of the vocabulary
  - If the word appears, add 1 to the corresponding element in the vector

- "it"
- "was"
- "the"
- "best"
- "of"
- "times"
- "worst"
- "age"
- "wisdom"
- "foolishness"

```
"it was the worst of times" = [1, 1, 1, 0, 1, 1, 1, 0, 0, 0]
"it was the age of wisdom" = [1, 1, 1, 0, 1, 0, 0, 1, 1, 0]
"it was the age of foolishness" = [1, 1, 1, 0, 1, 0, 0, 1, 0, 1]
```

The Bag-of-Words Representation

- Example

- Review 1: This movie is very scary and long
- Review 2: This movie is not scary and is slow
- Review 3: This movie is spooky and good

Term	Review 1	Review 2	Review 3
This	1	1	1
movie	1	1	1
is	1	2	1
very	1	0	0
scary	1	1	0
and	1	1	1
long	1	0	0
not	0	1	0
slow	0	1	0
spooky	0	0	1
good	0	0	1

The Bag-of-Words Representation

- Properties:
  - Cannot preserve the ordering of the words
  - High dimensional
  - Very sparse
  - Some words are too common for all documents
    - The, and, it, to, …..

	the	and	it	to	biology	computer	economy	sports
Doc 1	3	5	7	7	5	0	0	0
Doc 2	4	7	5	7	0	3	0	0

Term Frequency-Inverse Document Frequency

- Term Frequency-Inverse Document Frequency (TF-IDF)
  - Reflect how important a word is to a document in a collection

- Definition

$$TF(t,d) = \frac{\#t \text{ in document } d}{\#words \text{ in document } d} \qquad IDF(t) = \log \frac{\#documents}{\#documents \text{ containing } t}$$

$$TF\_IDF = TF(t,d) \times IDF(t)$$

Term Frequency-Inverse Document Frequency

- Term frequency (TF)
  - Measure the frequency of a word in a document

$$TF(t, d) = \frac{\#t \text{ in document } d}{\#words \text{ in document } d}$$

- Review 1: This movie is very scary and long
- Review 2: This movie is not scary and is slow
- Review 3: This movie is spooky and good

Term	Review 1	Review 2	Review 3
This	1	1	1
movie	1	1	1
is	1	2	1
very	1	0	0
scary	1	1	0
and	1	1	1
long	1	0	0
not	0	1	0
slow	0	1	0
spooky	0	0	1
good	0	0	1

Term Frequency-Inverse Document Frequency

- Inverse Document Frequency (IDF)
  - Measure the rareness of a word in all documents
  - The more documents a word appears in, the less valuable that word is as a signal to differentiate any given document.

$$IDF(t) = \log \frac{\#documents}{\#documents\ containing\ t}$$

Term	Review 1	Review 2	Review 3
This	1	1	1
movie	1	1	1
is	1	2	1
very	1	0	0
scary	1	1	0
and	1	1	1
long	1	0	0
not	0	1	0
slow	0	1	0
spooky	0	0	1
good	0	0	1

- IDF('movie',) = log(3/3) = 0
- IDF('is') = log(3/3) = 0
- IDF('not') = log(3/1) = log(3) = 0.48
- IDF('scary') = log(3/2) = 0.18
- IDF('and') = log(3/3) = 0
- IDF('slow') = log(3/1) = 0.48

Term Frequency-Inverse Document Frequency

- TF-IDF
  - Words with a higher score are more important

$$TF\_IDF = TF(t, d) \times IDF(t)$$

Term	Review 1	Review 2	Review 3
This	1	1	1
movie	1	1	1
is	1	2	1
very	1	0	0
scary	1	1	0
and	1	1	1
long	1	0	0
not	0	1	0
slow	0	1	0
spooky	0	0	1
good	0	0	1

- IDF('movie',) = log(3/3) = 0
- IDF('is') = log(3/3) = 0
- IDF('not') = log(3/1) = log(3) = 0.48
- IDF('scary') = log(3/2) = 0.18
- IDF('and') = log(3/3) = 0
- IDF('slow') = log(3/1) = 0.48

- TF-IDF('movie', Review 2) = 1/8 * 0 = 0
- TF-IDF('is', Review 2) = 1/4 * 0 = 0
- TF-IDF('not', Review 2) = 1/8 * 0.48 = 0.06
- TF-IDF('scary', Review 2) = 1/8 * 0.18 = 0.023
- TF-IDF('and', Review 2) = 1/8 * 0 = 0
- TF-IDF('slow', Review 2) = 1/8 * 0.48 = 0.06

Document Classification

- When there are only two classes
  - Binary classification → logistic regression, KNN, etc
  - e.g. spam detection, review classification, …
- When the number of classes is greater than 2
  - Multi-class logistic regression, KNN, etc
  - e.g. topic classification, …

Multi-class Classification

- Logistic regression for binary classification

$$p(y=1|\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1+\exp(-\mathbf{w}^T\mathbf{x})} = \frac{\exp(\mathbf{w}^T\mathbf{x})}{1+\exp(\mathbf{w}^T\mathbf{x})}$$

$$p(y=0|\mathbf{x}) = 1 - p(y=1|\mathbf{x}) = 1 - \frac{1}{1+\exp(-\mathbf{w}^T\mathbf{x})} = \frac{\exp(-\mathbf{w}^T\mathbf{x})}{1+\exp(-\mathbf{w}^T\mathbf{x})} = \frac{1}{1+\exp(\mathbf{w}^T\mathbf{x})}$$

Multi-class Classification

- Softmax regression for multi-class classification

$$p(y = 1|\mathbf{x}) = \frac{e^{\mathbf{w}_1^t \mathbf{x}}}{\sum_{c=1}^{K} e^{\mathbf{w}_c^T \mathbf{x}}},$$

$$p(y = 2|\mathbf{x}) = \frac{e^{\mathbf{w}_2^t \mathbf{x}}}{\sum_{c=1}^{K} e^{\mathbf{w}_c^T \mathbf{x}}},$$

$$\dots$$

$$p(y = K|\mathbf{x}) = \frac{e^{\mathbf{w}_K^t \mathbf{x}}}{\sum_{c=1}^{K} e^{\mathbf{w}_c^T \mathbf{x}}},$$

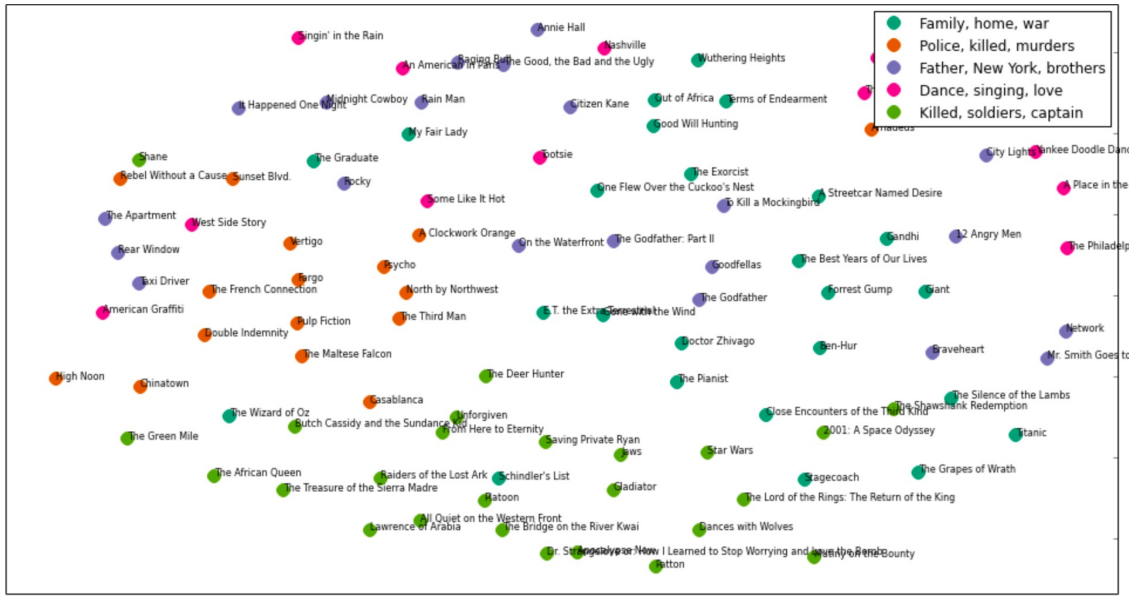
Evaluation of multi-class classification

- Binary classification: imbalanced data
  - Recall
  - Precision
  - F1-socre

- Multi-class classification: imbalanced data
  - Micro/Macro recall
  - Micro/Macro precision
  - Micro/Macro f1-score

Document Clustering

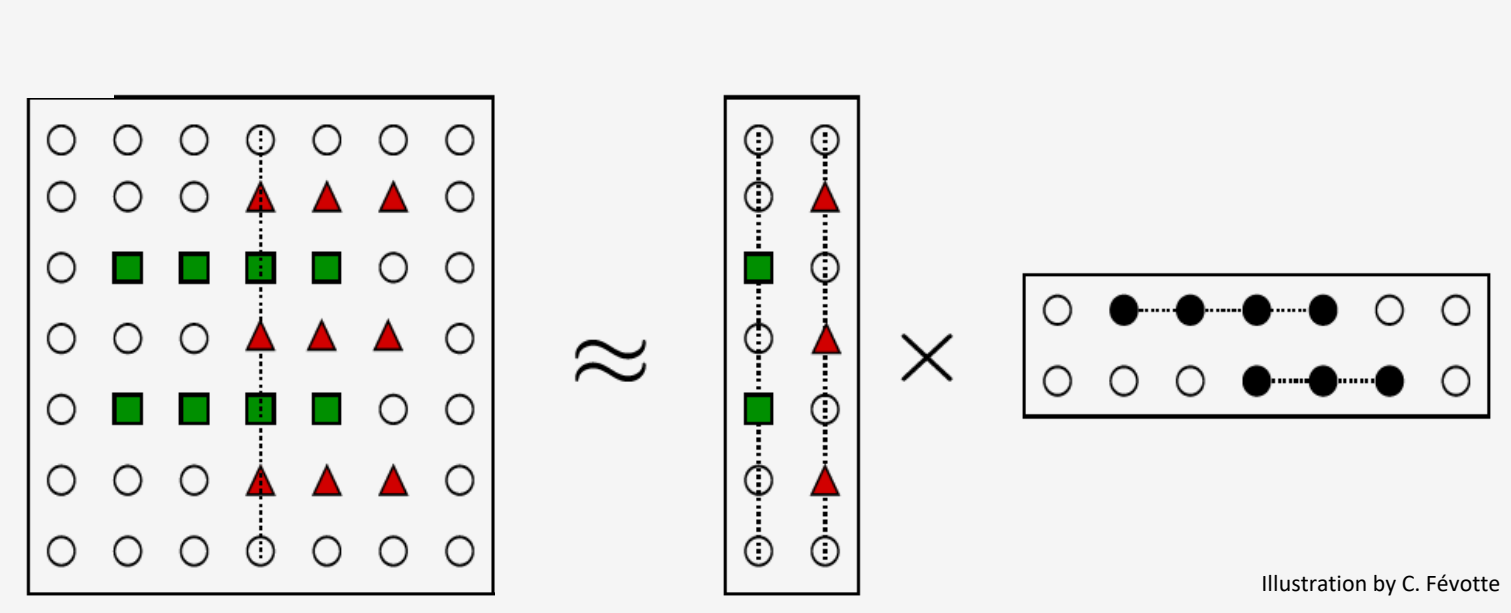
- Document Clustering
  - e.g. Topic discovery, …
  - Kmeans, spectral clustering, agglomerative clustering,



athletics	cricket	football	rugby	tennis
olymp	wicket	chelsea	wale	seed
athlet	cricket	arsen	ireland	63
indoor	test	leagu	robinson	open
athen	pakistan	club	england	64
kenteri	seri	unit	nation	76
thanou	bowl	liverpool	franc	australian
greek	india	mourinho	rugbi	roddick
iaaf	south	football	six	75
drug	onedai	manag	scotland	hewitt
race	africa	manchest	itali	beat

Document Clustering

- Non-Negative Matrix Factorization



Data matrix Basis Coefficient

Non-Negative Matrix Factorization

- “Learning the parts of objects by non-negative matrix factorization”

—Nature 1999

- “Algorithms for non-negative matrix factorization”

—NIPS 2001

- Definition

$$\min \| X - FG^T \|_F^2$$
$$s.t.\ F \geq 0, G \geq 0$$

Interpretation with NMF

$$X \approx FG^T$$

- Columns of F are the underlying basis vectors

$$F = [f_1, f_2, \ldots, f_k]$$

- Rows of G give the weights associated with each basis vector.

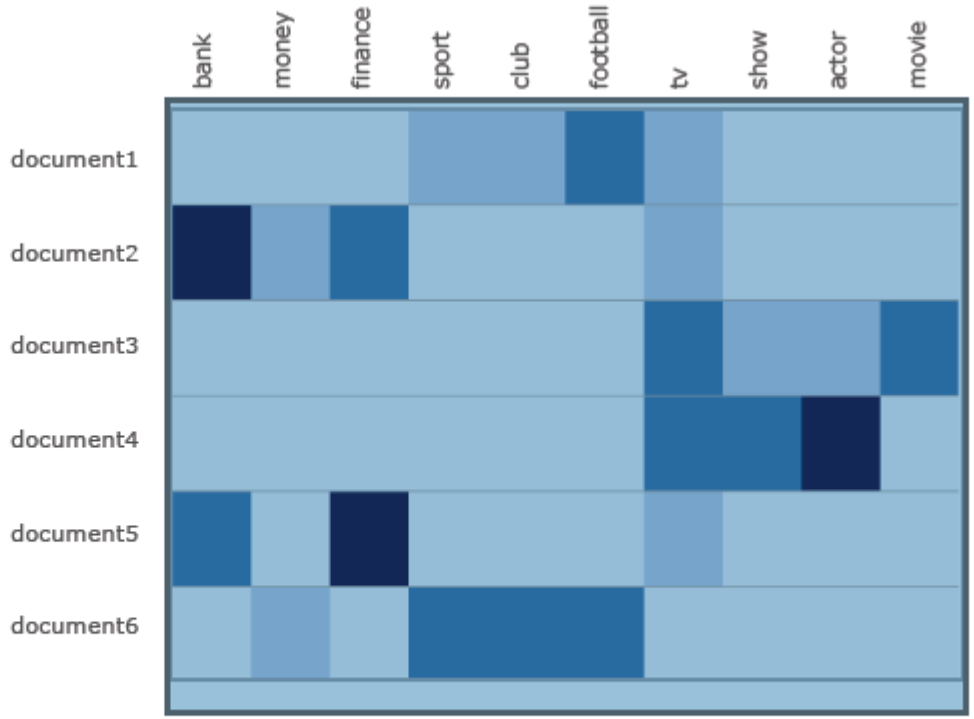
$$[\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n] = [\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_k] \begin{bmatrix} g_{11} & g_{21} & \cdots & g_{n1} \\ g_{12} & g_{22} & \cdots & g_{n2} \\ \vdots & \vdots & \vdots & \vdots \\ g_{1k} & g_{2k} & \cdots & g_{nk} \end{bmatrix}$$

$$\mathbf{x}_i = \mathbf{f}_1 g_{i1} + \mathbf{f}_2 g_{i2} + \cdots + \mathbf{f}_k g_{ik}$$

only additive combinations!!!

Application Example: Topic Models

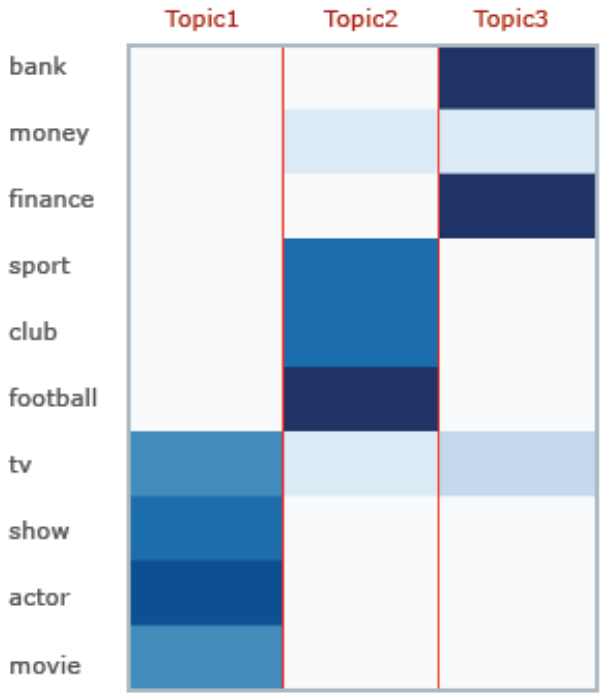
Document-Term Matrix **A**
(6 rows x 10 columns)



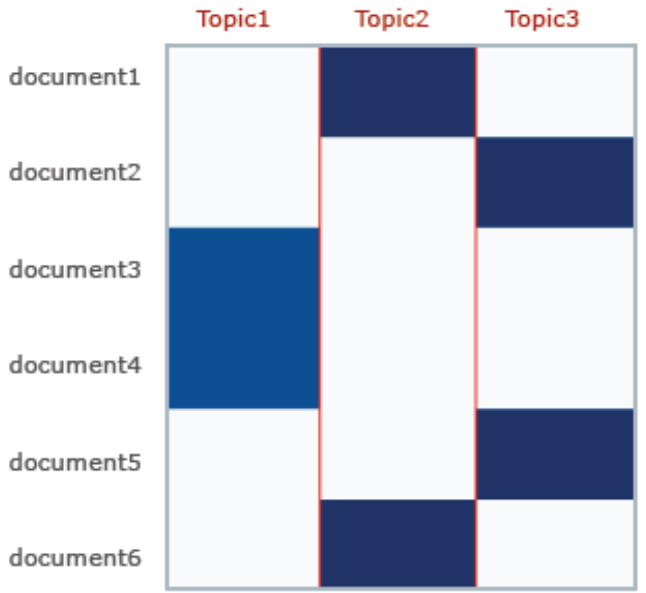
- Apply TF-IDF and unit length normalization to rows of **A**.
- Run Euclidean NMF on normalized **A** (k=3, random initialization).

Application Example: Topic Models

Basis vectors **W**: topics (clusters)



Coefficients **H**: memberships for documents



$$\mathbf{x}_i = \mathbf{f}_1 g_{i1} + \mathbf{f}_2 g_{i2} + \cdots + \mathbf{f}_k g_{ik}$$

Multiplicative Update Method

- The most common used method
  - Proposed by Lee and Seung (2001)
- The update rule:
  - Fix F, solve for G
  - Fix G, solve for F

$$F_{ik} \leftarrow F_{ik} \frac{(XG)_{ik}}{(FG^T G)_{ik}} \qquad G_{jk} \leftarrow G_{jk} \frac{(X^T F)_{jk}}{(GF^T F)_{jk}}$$

Multiplicative Update Method

- Arise from gradient descent method

$$F_{ik} \leftarrow F_{ik} + \varepsilon_{ik}[(XG)_{ik} - (FG^T G)_{ik}]$$

  - Where ε_{ik} is a small positive number.
- Set it as

$$\varepsilon_{ik} = \frac{F_{ik}}{(FG^T G)_{ik}}$$

- Then

$$F_{ik} = F_{ik} \frac{(XG)_{ik}}{(FG^T G)_{ik}}$$

Multiplicative Update Method

Algorithm 2 Algorithm to solve NMF.

Initialize F and G

repeat

Update F:

$$F_{ik} \Leftarrow F_{ik} \frac{(XG)_{ik}}{(FG^TG)_{ik}}$$

Update G:

$$G_{jk} \Leftarrow G_{jk} \frac{(X^TF)_{jk}}{(GF^TF)_{jk}}$$

until Converges