Advanced Feature Engineering II

shkim.hi@gmail.com 김수현

Advanced feature engineering I

학습 목표

1. **Regularize mean encodings**
2. **Extend mean encodings**
3. **Summarize the concept of mean encodings**

shkim.hi@gmail.com 김수현

Advanced feature engineering II

학습 목표

1. Understand matrix factorization technique for feature extraction
2. Create features based on tSNE
3. Explain concept of feature interactions
4. Generate and make up new features based on statistics and nearest neighbors

shkim.hi@gmail.com 김수현

Advanced Feature Engineering II

- **Contents**
  - **Statistics and distance based features**
  - Matrix Factorizations
  - Feature Interactions
  - t-SNE

shkim.hi@gmail.com 김수현

Statistics and distance based features

➢ Some data for CTR task

	User_id	Page_id	Ad_price	Ad_position
0	4	6	165.977125	Bottom_right
1	4	6	34.5395640	Bottom_right
2	4	6	29.1963786	Bottom_left
3	4	6	79.4373729	Bottom_left
4	4	6	290.534595	Bottom_right
5	4	6	314.412660	Bottom_right
6	4	6	138.9007639	Bottom_right
7	4	6	107.4711914	Bottom_right
8	4	6	242.1089786	Bottom_left
9	4	7	27.16719836	Bottom_left
10	4	7	413.5421978	Bottom_right

shkim.hi@gmail.com 김수현

Statistics and distance based features

➢ Useful features

	User_id	Page_id	Ad_price	Ad_position	Max_price	min_price	Min_price _position
0	4	6	95.874252	Bottom_right	474.63772	73.711548	Bottom_left
1	4	6	215.751007	Bottom_right	474.63772	73.711548	Bottom_left
2	4	6	474.637726	Bottom_left	474.63772	73.711548	Bottom_left
3	4	6	73.711548	Bottom_left	474.63772	73.711548	Bottom_left
4	4	6	79.288841	Bottom_right	474.63772	73.711548	Bottom_left
5	4	6	271.391785	Bottom_right	474.63772	73.711548	Bottom_left
6	4	6	296.529053	Bottom_right	474.63772	73.711548	Bottom_left
7	4	6	96.030029	Bottom_right	474.63772	73.711548	Bottom_left
8	4	6	130.175064	Bottom_left	474.63772	73.711548	Bottom_left
9	4	7	35.465202	Bottom_left	121.54219	35.465202	Bottom_left
10	4	7	121.542191	Bottom_right	121.54219	35.465202	Bottom_left

shkim.hi@gmail.com 김수현

Statistics and distance based features

➢ Useful features: implementation

```
In [22]: gb = df.groupby(['user_id','page_id'],as_index=False).agg(
             {'ad_price':{'max_price':np.max, 'min_price':np.min}})
         gb.columns = ['user_id','page_id','min_price','max_price']
         gb
```

Out[22]:

	user_id	page_id	min_price	max_price
0	4	6	73.711548	474.637726
1	4	7	35.465202	121.542191

```
In [23]: df = pd.merge(df,gb,how='left',on=['user_id','page_id'])
```


shkim.hi@gmail.com 김수현

Statistics and distance based features

- **More features**
  - 사용자 및 페이지 단위로 계산되지 않은 다른 유용한 피처들을 추가할 수 있음
  - How many pages user visited
  - 사용자가 특정 세션 동안 방문한 페이지 수
  - Standard deviation of prices
  - Most visited page
  - 해당 페이지를 방문한 사용자 수 등
  - Many, many more

  - **새로운 정보를 도입함으로써 모델의 품질을 크게 향상 시킬 수 있음**

shkim.hi@gmail.com 김수현

Statistics and distance based features

- **Neighbors**
  - Explicit group is not needed
  - More flexible
  - Much harder to implement

  - **그룹핑을 사용할 피처가 없다면, 이런 경우에는 그룹화 작업을 가까운 이웃들을 찾는 것으로 대체할 수 있음**

shkim.hi@gmail.com 김수현

Statistics and distance based features

- **Neighbors**
  - 임대 가격 예측 사례
    - → 바닥 공간 크기, 방의 개수, 버스 정류장의 유무등에 대한 피처가 있을 것임
  - 정말 좋은 모델을 만드려면 그 이상의 무언가가 필요할 것임
    - → Number of houses in 500m, 1000m,..
    - → Average price per square meter in 500m, 1000m,..
    - → Number of schools/supermarkets/parking lots in 500m, 1000m,..
    - → Distance to closest subway station

shkim.hi@gmail.com 김수현

Statistics and distance based features

- **Neighbors**
  - Explicit group is not needed
  - More flexible
  - Much harder to implement

  - 가장 간단한 사례는 이웃들을 지리적 공간으로 계산하는 것임
  - 이 방법을 추상적이거나 익명화된 피처 공간에 적용하는 것도 시도 가능하며, 시도에 두려워 하지 마라

shkim.hi@gmail.com 김수현

Statistics and distance based features

KNN features in Springleaf

1. Mean encode all the variables
   → 균일 피처 공간이 생성되므로 각 피처의 scaling과 중요도를 걱정하지 않아도 됨
2. For every point, find 2000 nearest neighbors using Bray-Curtis metric

$$\sum |u_i - v_i| / \sum |u_i + v_i|$$

3. Calculate various features from those 2000 neighbors

Bray–Curtis dissimilarity

→ 생태학 과 생물학 에서 J. Roger Bray 와 John T. Curtis의 이름을 딴 Bray-Curtis의 비 유사성 은 각 사이트의 수에 따라 두 개의 서로 다른 사이트 간의 구성 비 유사성을 정량화하는 데 사용되는 통계임

→ https://en.m.wikipedia.org/wiki/Bray–Curtis_dissimilarity

shkim.hi@gmail.com 김수현

Statistics and distance based features

- **KNN features in Springleaf**
  - Mean target of nearest 5, 10, 15, 500, 2000 neighbors
  - Mean distance to 10 closest neighbors
  - Mean distance to 10 closest neighbors with target 1
  - Mean distance to 10 closest neighbors with target 0

  - 이들은 잘 작동을 하였음

  - **결론적으로, 그룹 별 및 KNN 방법의 주요 아이디어를 수용하고 실제로 적용 할 수 있기를 바람**

  - **from sklearn.neighbors import NearestNeighbors**

shkim.hi@gmail.com 김수현

Advanced Feature Engineering II

- **Contents**
  - Statistics and distance based features
  - **Matrix Factorizations**
  - Feature Interactions
  - t-SNE

shkim.hi@gmail.com 김수현

Matrix Factorizations for Feature Extraction

- **Matrix Factorization : 선형 차원 축소**
  - PCA (Principal Component Analysis)
  - SVD (Singular Value Decomposition)
  - TruncatedSVD
  - NMF (Non-negative Matrix Factorization)

  - https://scikit-learn.org/stable/modules/decomposition.html

  - from scipy.linalg import svd
  - from sklearn.decomposition import PCA, TruncatedSVD, NMF
  - https://scikit-learn.org/stable/modules/classes.html#module-sklearn.decomposition

shkim.hi@gmail.com 김수현

Matrix Factorizations for Feature Extraction

➢ Recommendation Problem, Matrix Completion

- **matrix completion 문제 해결 방법 중 하나 → Matrix Factorization**
- **사용자 u가 아이템 i를 얼마나 좋아할 것인지 나타내는 값을 rating R_{ui}**
- 이때 *은 아직 사용자가 평가하지 않은 데이터를 의미함

$$
\begin{array}{c}
movie.1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \\
\begin{array}{l} user1 \\ user2 \\ user3 \\ user4 \\ user5 \\ user6 \\ user7 \\ user8 \end{array}
\begin{bmatrix}
3 & 5 & * & 4 & 1 & * & * & 2 \\
* & 3 & 5 & 1 & 2 & * & * & 3 \\
4 & 1 & * & 4 & 1 & * & 3 & 2 \\
5 & 2 & * & * & 2 & 3 & * & * \\
* & 2 & 4 & 2 & * & * & 1 & 2 \\
5 & * & * & 5 & 4 & * & * & 4 \\
1 & * & 5 & 2 & 3 & 1 & 5 & 3 \\
* & 3 & 2 & 1 & 4 & * & * & *
\end{bmatrix}
\end{array}
$$

- 이제 recommendation problem은 이 매트릭스의 비어있는 부분의 값을 예측하는 문제로 바꿔서 생각할 수 있음
- 이를 수식으로 표현하면 아래와 같은 형태로 표현하는 것이 가능함

$$\min_{\hat{R}} \|\hat{R} - R\|_F^2$$

shkim.hi@gmail.com 김수현

Matrix Factorizations for Feature Extraction

Matrix Factorization

- Matrix Factorization 방법은 모든 사용자와 상품에 대해 다음 오차 함수를 최소화하는 요인 벡터를 찾아냄
- 즉 다음과 같은 행렬 P, Q를 찾음

$$R \approx PQ^T$$

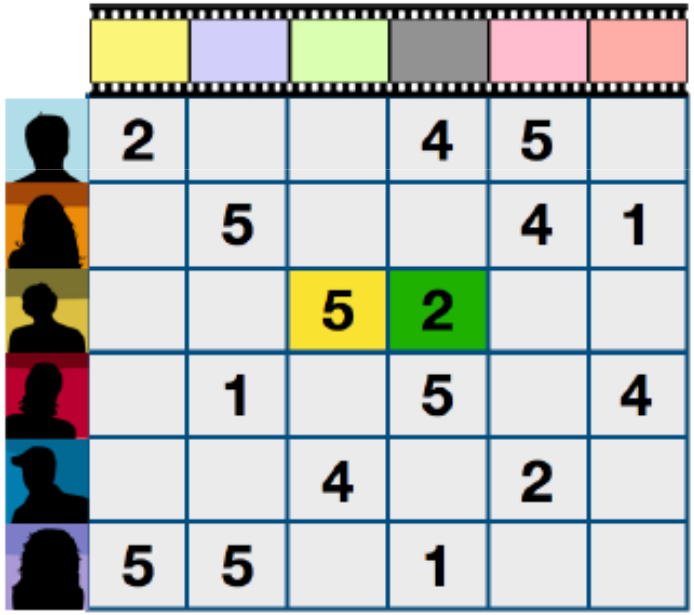
- $R \in \mathbf{R}^{m \times n}$: m 사용자와 n 상품의 평점 행렬
- $P \in \mathbf{R}^{m \times k}$: m 사용자와 k 요인의 관계 행렬
- $Q \in \mathbf{R}^{n \times k}$: n 상품의와 k 요인의 관계 행렬

shkim.hi@gmail.com 김수현

Matrix Factorizations for Feature Extraction

➢ Example : recommandations

- 사용자 정보가 있음
  - → age, region, interest and items like gender, year length
  - → ratings that users gave to some items

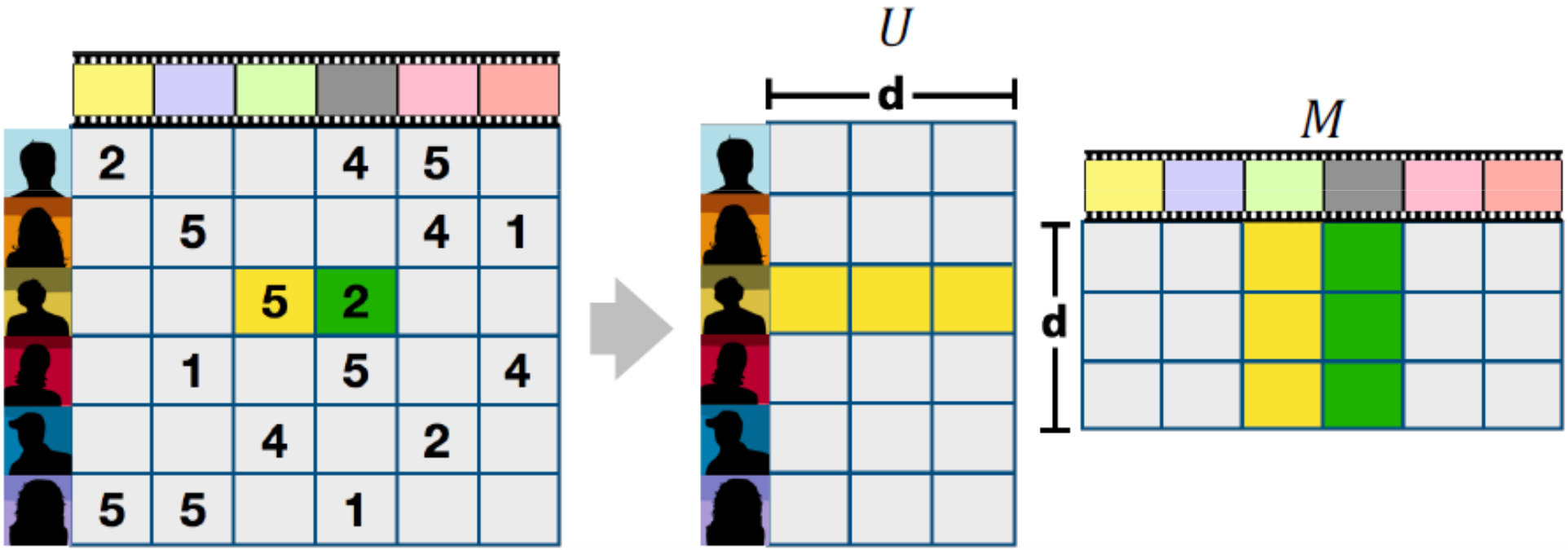


shkim.hi@gmail.com 김수현

Matrix Factorizations for Feature Extraction

Example : recommandations

- 사용자 정보가 있음
  - → age, region, interest and items like gender, year length
  - → ratings that users gave to some items

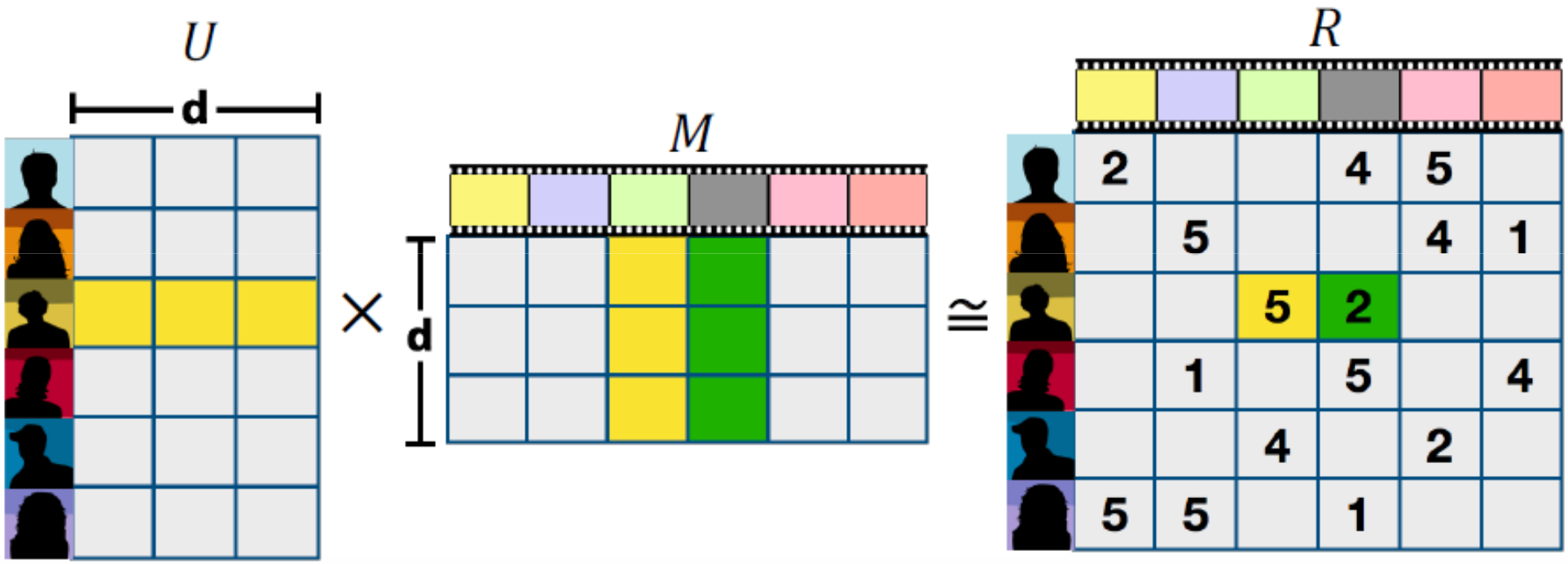


shkim.hi@gmail.com 김수현

Matrix Factorizations for Feature Extraction

➢ What is Matrix Factorization?

- 매트릭스 U와 M의 곱은 매트릭스 R을 생성함
- Matrix Composition



shkim.hi@gmail.com 김수현

Matrix Factorizations for Feature Extraction

- **Documents\words example**

	cat	is	dog	animal	nature
Cat is animal	1	0	0	1	0
Dog is animal	0	1	1	1	0
Cat is dog	1	1	1	0	0

shkim.hi@gmail.com 김수현

Matrix Factorizations for Feature Extraction

- **Documents\words example**
  - **Matrix Factorization 적용**
    - → **Dimension Enery Reduction**
    - → 피쳐 메트릭의 크기를 줄이고 분류 기능에서 real value 피쳐를 추출하는 데 매우 효율적임

	cat	is	dog	animal	nature
Cat is animal	1	0	0	1	0
Dog is animal	0	1	1	1	0
Cat is dog	1	1	1	0	0

Cat is animal			
Dog is animal			
Cat is dog			

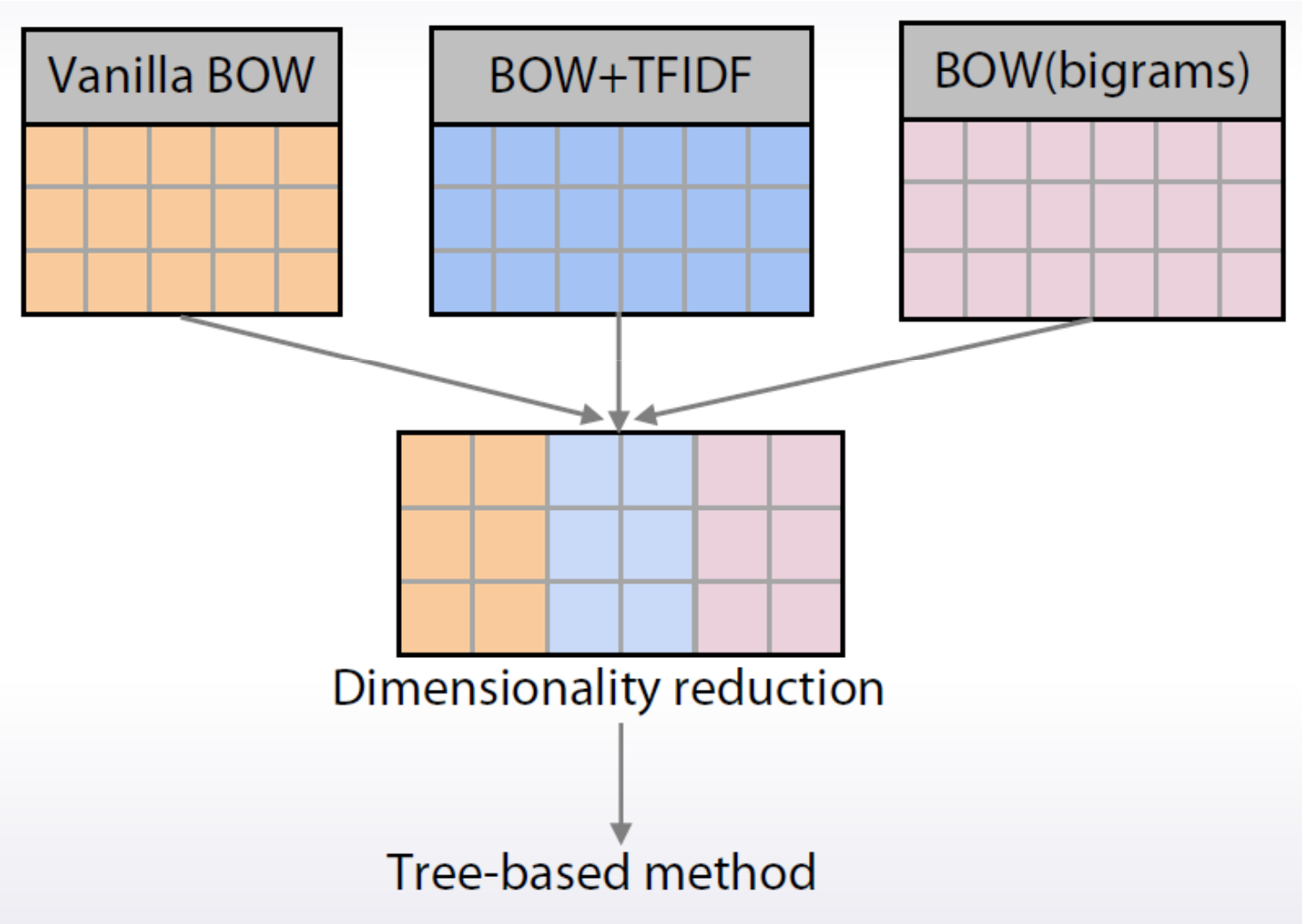
cat			
is			
dog			
animal			
nature			

shkim.hi@gmail.com 김수현

Matrix Factorizations for Feature Extraction

➢ Example of feature fusion

- Matrix Optimization 기술 사용



shkim.hi@gmail.com 김수현

Matrix Factorizations for Feature Extraction

- **Notes about Matrix Factorization**
  - Can be apply only for some columns
  - Can provide additional diversity
    - → Good for ensembles
  - It is a lossy transformation. Its' efficiency depends on:
    - → Particular task
    - → Number of latent factors
      - ☞ Usually 5-100

shkim.hi@gmail.com 김수현

Matrix Factorizations for Feature Extraction

Implementation

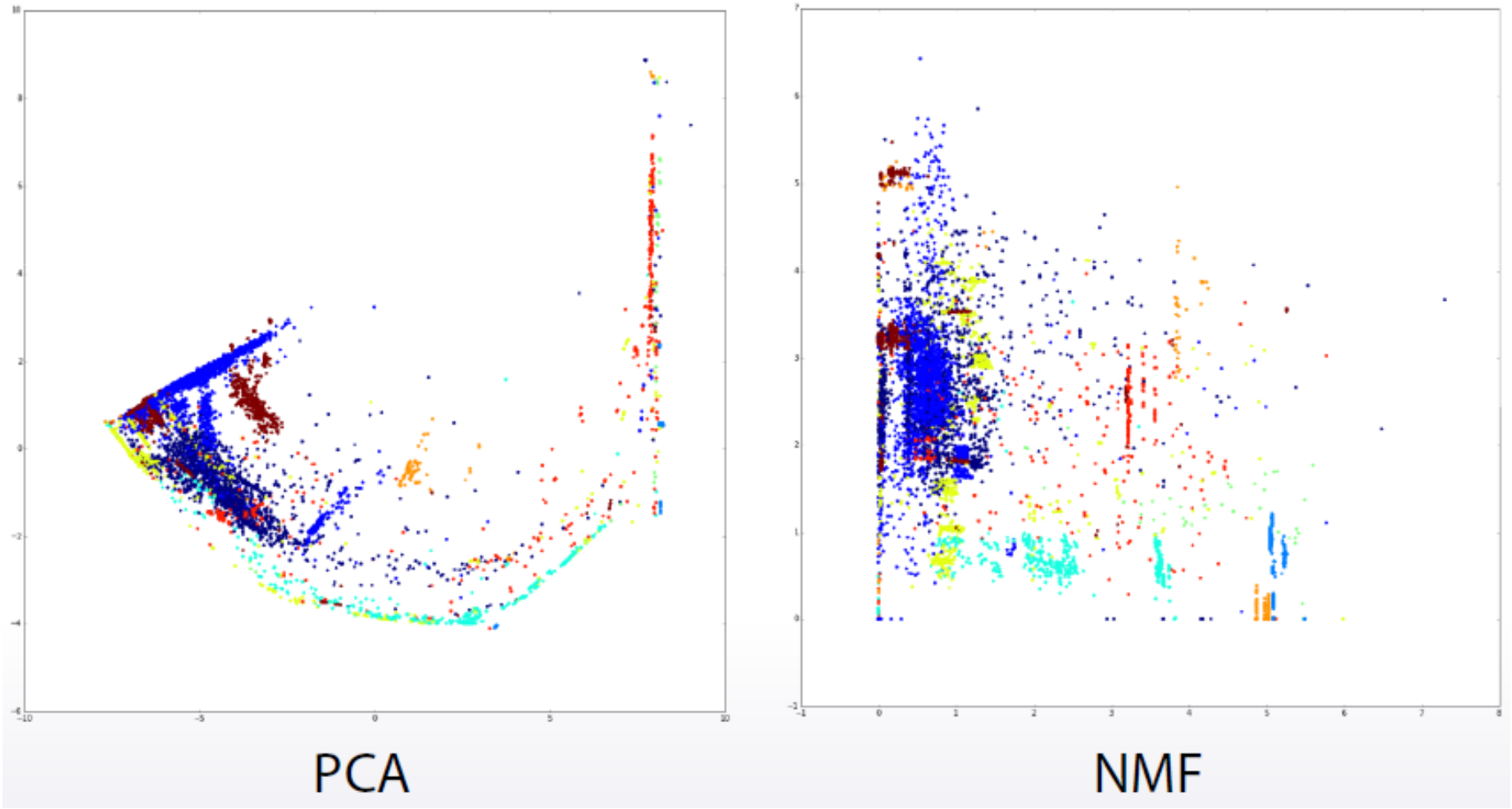
- Several MF methods you can find in sklearn
- SVD and PCA
  - → Standart tools for Matrix Factorization
- TruncatedSVD
  - → Works with sparse matrices
  - → 텍스트 데이터 세트의 경우 매우 편리함
- Non-negative Matrix Factorization (NMF)
  - → Ensures that all latent factors are non-negative
  - → Good for counts-like data

shkim.hi@gmail.com 김수현

Matrix Factorizations for Feature Extraction

➢ NMF for tree-based methods

- 의사 결정 트리에 보다 적합한 데이터를 만드는 방식으로 데이터를 변환함
- 아래는 Microsoft Mobile Classification Challenge에서 만들어진 그림임

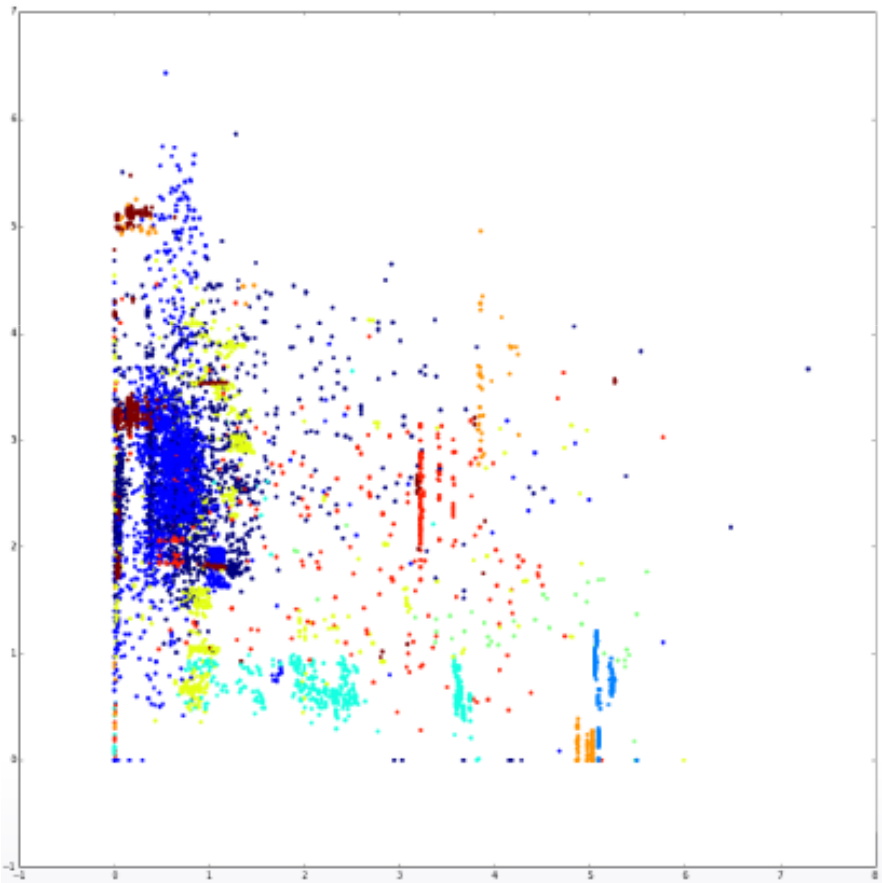


shkim.hi@gmail.com 김수현

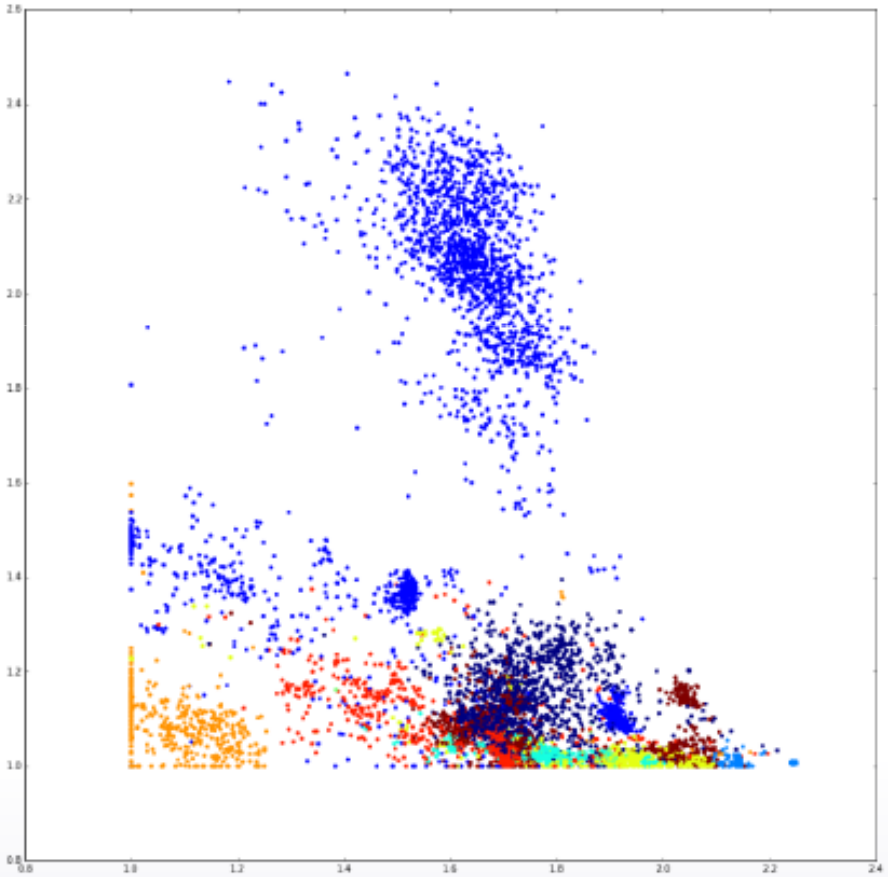
Matrix Factorizations for Feature Extraction

Notes about MF

- Matrix factorization is similar in spirit to linear models.
- You can use the same transformation tricks.



NMF(X)



NMF(log(X+1))

shkim.hi@gmail.com 김수현

Matrix Factorizations for Feature Extraction

➢ Gochas

Wrong way:

```
pca = PCA(n_components=5)
X_train_pca = pca.fit_transform(X_train)
X_test_pca = pca.fit_transform(X_test)
```

Right way:

```
X_all = np.concatenate([X_train,X_test])
pca.fit(X_all)
X_train_pca = pca.transform(X_train)
X_test_pca = pca.transform(X_test)
```


shkim.hi@gmail.com 김수현

Matrix Factorizations for Feature Extraction

Conclusion

- Matrix Factorization is a very general approach for dimensionality reduction and feature extraction
- It can be applied for transforming categorical features into real-valued
- Many of tricks trick suitable for linear models can be useful for MF

shkim.hi@gmail.com 김수현

Advanced Feature Engineering II

- **Contents**
  - Statistics and distance based features
  - Matrix Factorizations
  - **Feature Interactions**
  - t-SNE

shkim.hi@gmail.com 김수현

Feature Interactions

➢ Example: banner selection

- 웹 사이트에 표시할 최상의 광고 배너를 예측하는 모델을 만들고 있다고 가정함

...	category_ad	category_site	...	is_clicked
...	auto_part	game_news	...	0
...	music_tickets	music_news	..	1
...	mobile_phones	auto_blog	...	0

shkim.hi@gmail.com 김수현

Feature Interactions

➢ Example: banner selection

- 웹 사이트에 표시할 최상의 광고 배너를 예측하는 모델을 만들고 있다고 가정함

...	category_ad	category_site	...	is_clicked
...	auto_part	game_news	...	0
...	music_tickets	music_news	..	1
...	mobile_phones	auto_blog	...	0

...	ad_site	...	is_clicked
...	auto_part|game_news	...	0
...	music_tikets|music_news	..	1
...	mobile_phones|auto_blog	...	0

shkim.hi@gmail.com 김수현

Feature Interactions

➢ Example of interactions

f1	f2
A	X
B	Y
B	Z
A	Z

shkim.hi@gmail.com 김수현

Feature Interactions

➢ Example of interactions

f1	f2
A	X
B	Y
B	Z
A	Z

Join →

f_join
A|X
B|Y
B|Z
A|Z

OneHot (f_join) →

A|X	B|Y	B|Z	A|Z
1			
	1		
		1	
			1

shkim.hi@gmail.com 김수현

Feature Interactions

➢ Example of interactions

f1	f2
A	X
B	Y
B	Z
A	Z

OneHot(f1),
OneHot(f2)

A	B
1	
	1
	1
1	

X	Y	Z
1		
	1	
		1
		1

Pairwise columns multiplications

AX	AY	AZ	BX	BY	BZ
1					
				1	
					1
		1			

shkim.hi@gmail.com 김수현

Feature Interactions

➢ Example of interactions

f1	f2
1.2	0.0
3.4	0.1
5.6	1.0
7.8	-1.0

Mul →

f_join
0.0
0.34
5.6
-7.8

shkim.hi@gmail.com 김수현

Feature Interactions

- **Frequent operations for feature interaction**
  - Multiplication
  - Sum
  - Diff
  - Division

shkim.hi@gmail.com 김수현

Feature Interactions

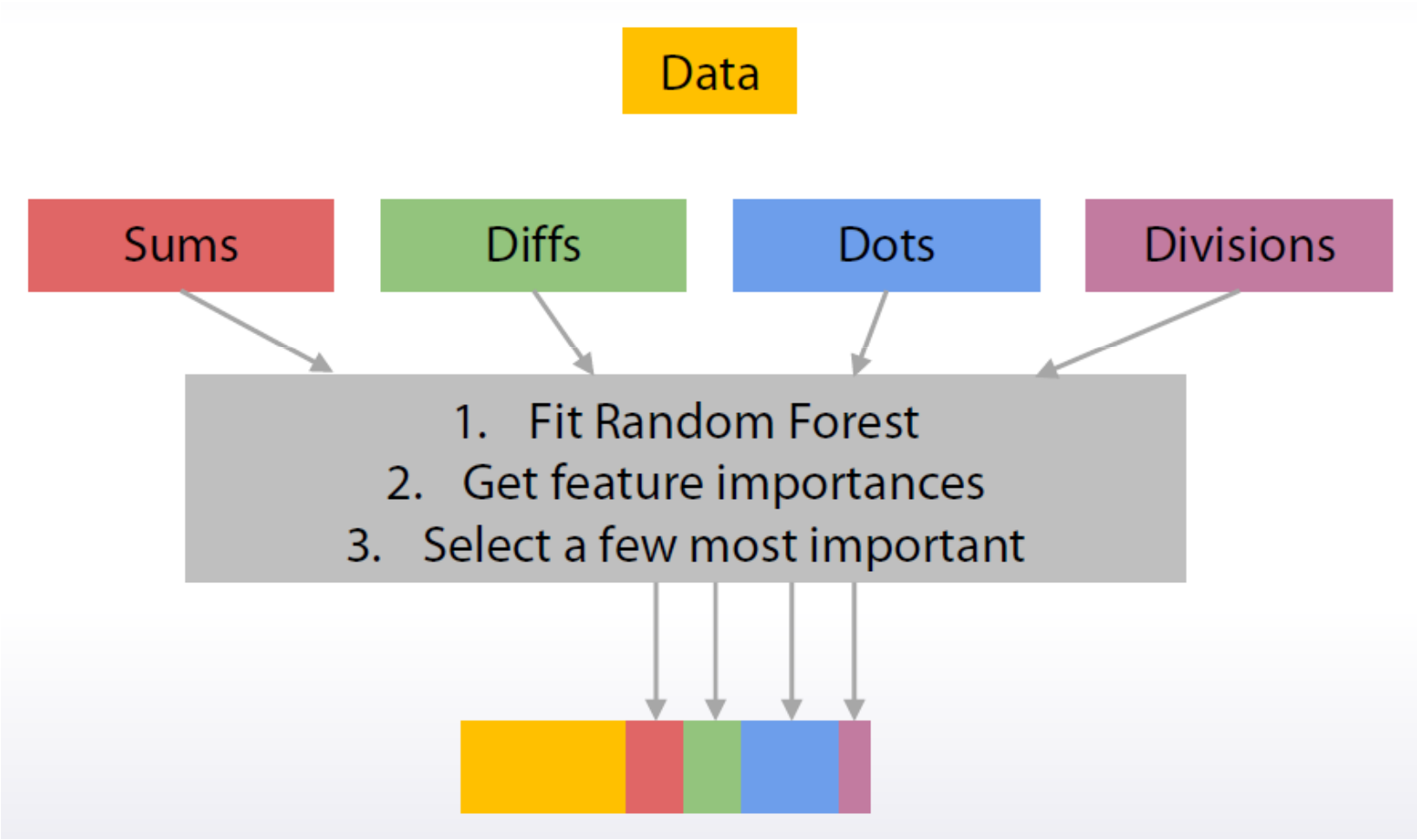
- **Practical Notes**
  - We have a lot of possible interactions – N*N for N features.
    - → Even more if use several types in interactions
  - **Need to reduce its' number**
    - → **Dimensionality reduction**
    - → **Feature selection**

shkim.hi@gmail.com 김수현

Feature Interactions

➢ Example of interaction generation pipeline

- interaction type 별 모든 피처 쌍의 interaction을 구성한 후 아래의 1,2,3을 수행함



shkim.hi@gmail.com 김수현

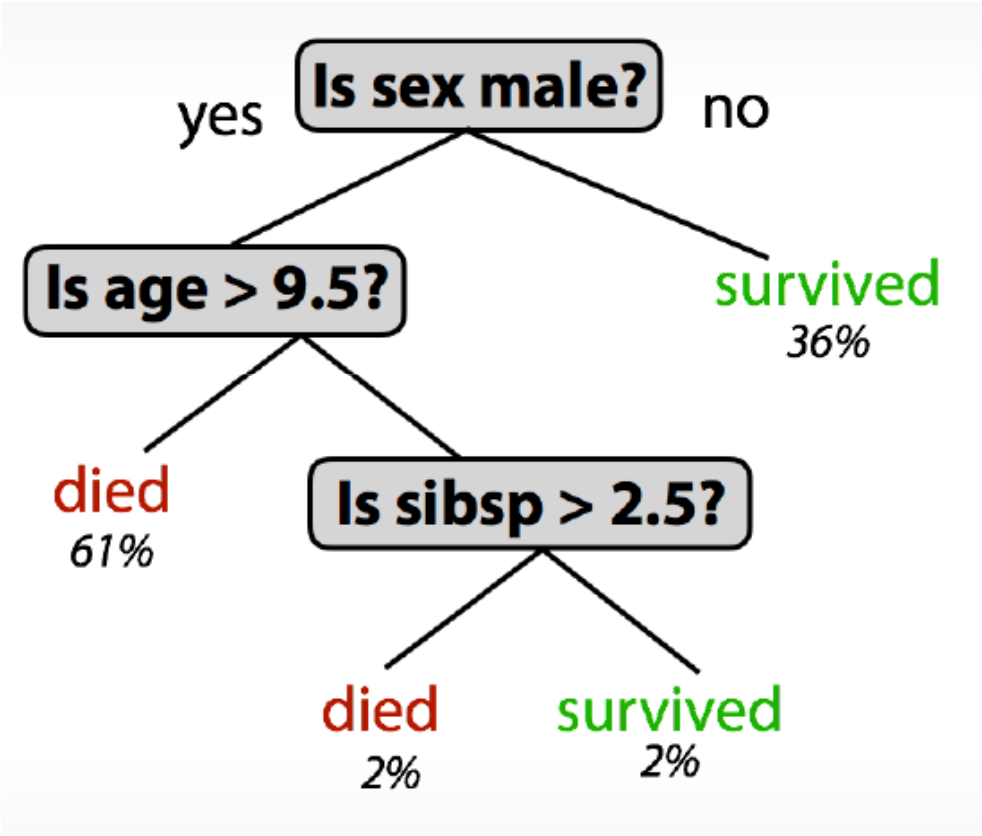
Feature Interactions

- **Interactions' order**
  - We looked at 2nd order interactions.
  - Such approach can be generalized for higher orders.
    - → It is hard to do generation and selection automatically.
    - → Manual building of high-order interactions is some kind of art.

shkim.hi@gmail.com 김수현

Feature Interactions

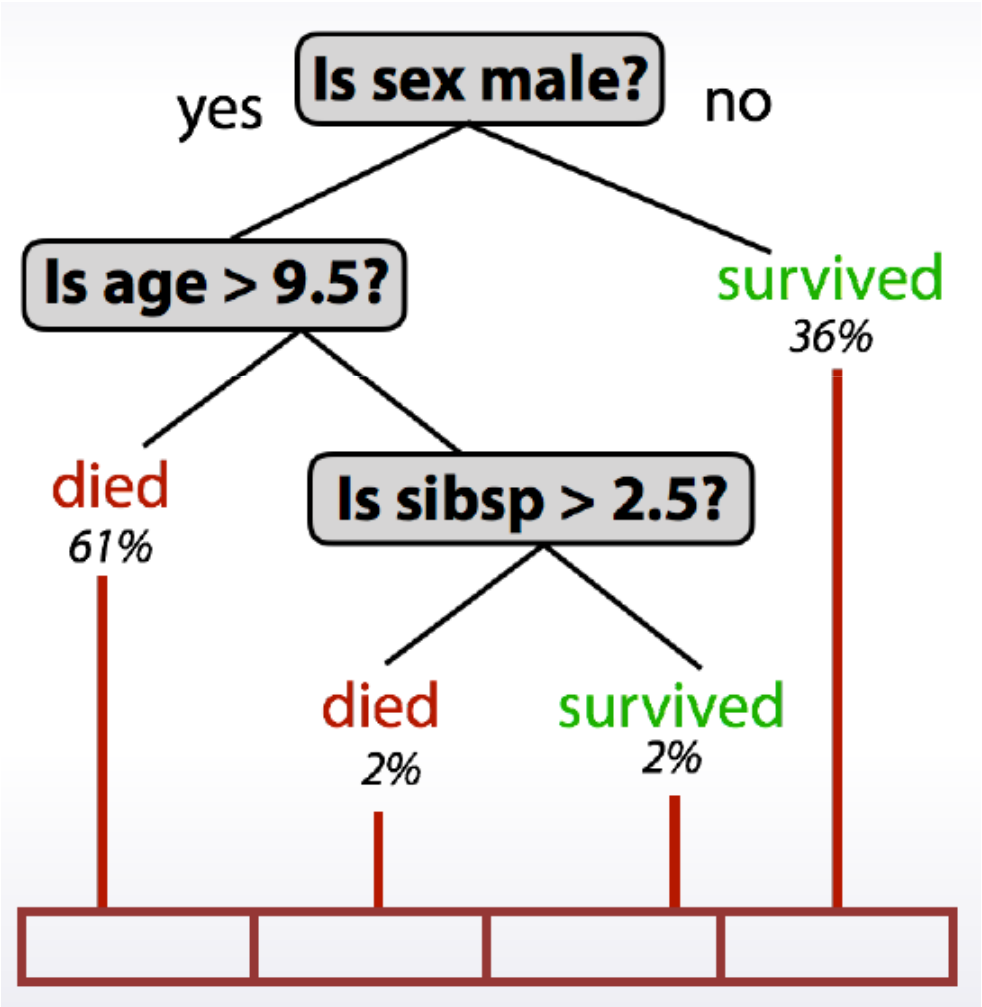
- **Extract features from DT**
  - DT에서 categirical feature 생성 방법



shkim.hi@gmail.com 김수현

Feature Interactions

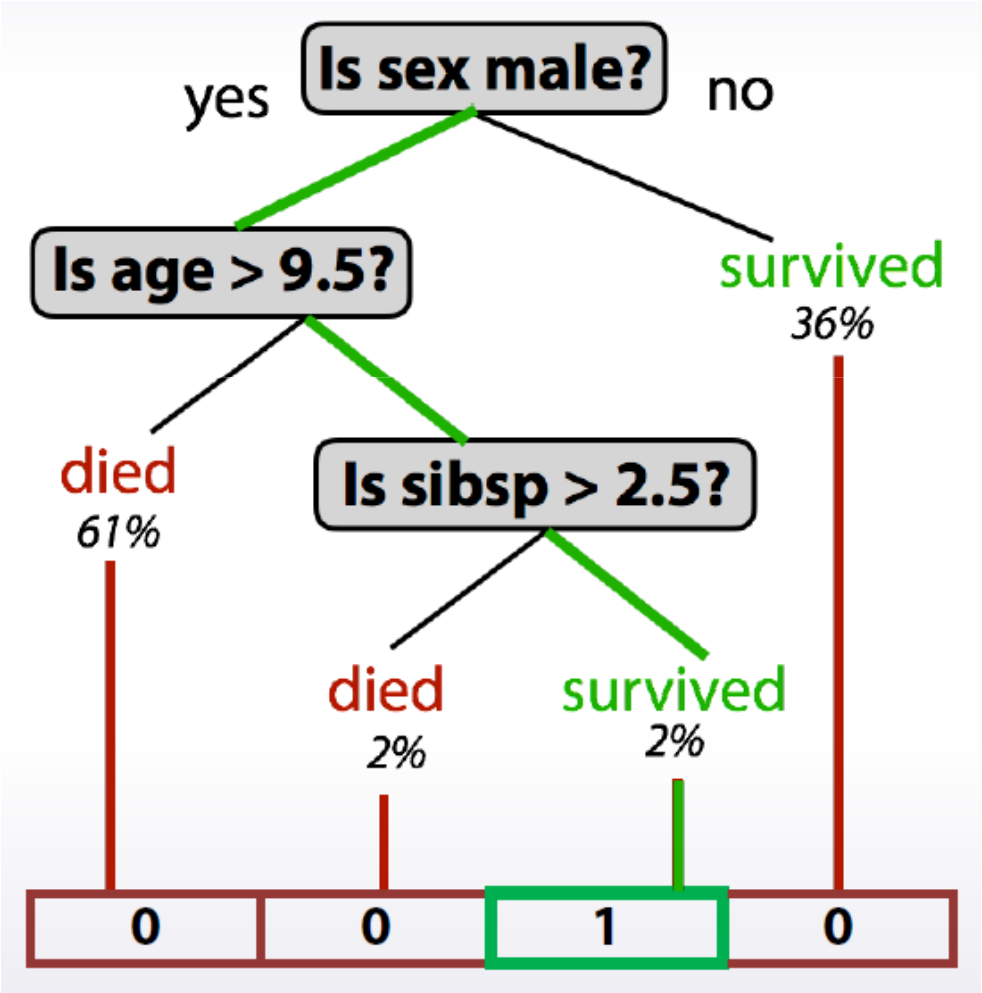
- **Extract features from DT**
  - 각 리프를 이진 피처로 매핑함



shkim.hi@gmail.com 김수현

Feature Interactions

- **Extract features from DT**
  - 리프의 객체의 인덱스는 새로운 범주 형 피쳐의 값으로 사용될 수 있음



shkim.hi@gmail.com 김수현

Feature Interactions

How to use it

- sklearn.tree.DecisionTreeClassifier
  - → apply() : 각 샘플이 예측 된 리프의 인덱스를 반환함
- Python xgboost 패키지
  - → import xgboost as xgb
  - → leafindex = bst.predict(dtest, pred_leaf=True)

In sklearn:

```
tree_model.apply()
```

In xgboost:

```
booster.predict(pred_leaf=True)
```

predict (*data, output_margin=False, ntree_limit=0, pred_leaf=False, pred_contribs=False, approx_contribs=False, pred_interactions=False, validate_features=True*) ¶

shkim.hi@gmail.com 김수현

Feature Interactions

- **Conclusion**
  - We looked at ways to build an interaction of categorical attributes
  - Extended this approach to real-valued features
  - Learn how to extract features via decision trees

shkim.hi@gmail.com 김수현

Advanced Feature Engineering II

- **Contents**
  - Statistics and distance based features
  - Matrix Factorizations
  - Feature Interactions
  - **t-SNE (t-distributed Stochastic Neighbor Embedding)**

shkim.hi@gmail.com 김수현

t−SNE (t−distributed Stochastic Neighbor Embedding)

- **Manifold learning methods : 비선형 차원 축소**
  - LLE (Locally Linear Embedding, 지역 선형 임베딩)
  - LTSA (Local Tangent Space Alignment)
  - Isomap
  - MDS (Multi-Dimensional Scaling)
  - SpectralEmbedding
  - t-SNE

    → **비슷한 데이터는 가까이, 비슷하지 않은 데이터는 멀리 떨어지도록 차원을 축소함**

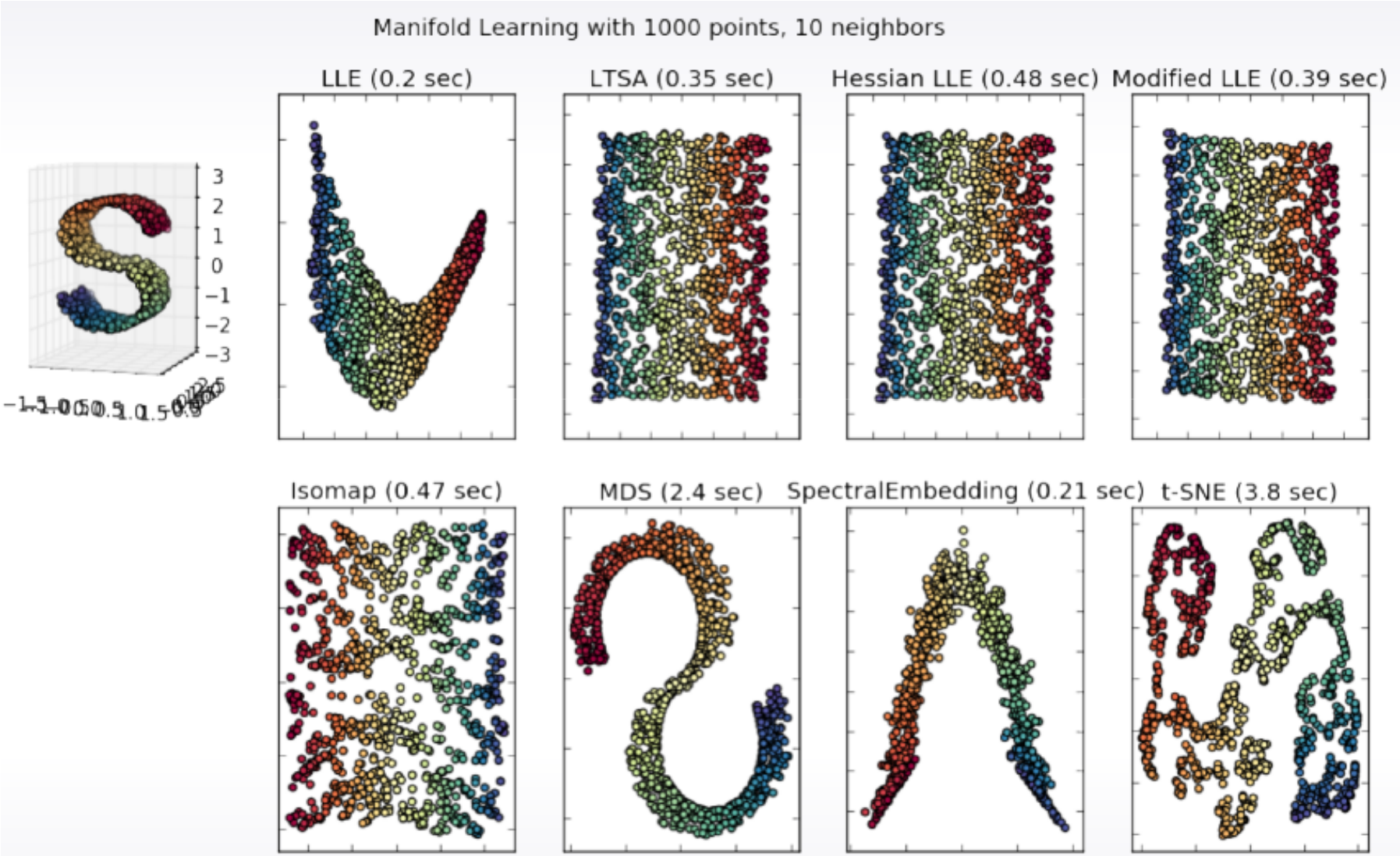
    → 주로 시각화에 많이 사용되며, 특히 고차원 공간에 있는 데이터의 군집을 시각화할 때 사용함

  - https://scikit-learn.org/stable/modules/manifold.html
  - **from sklearn.manifold import LocallyLinearEmbedding, LTSA, Isomap**
  - **from sklearn.manifold import MDS, SpectralEmbedding, TSNE**
  - https://scikit-learn.org/stable/modules/classes.html#module-sklearn.manifold

shkim.hi@gmail.com 김수현

t−SNE (t−distributed Stochastic Neighbor Embedding)

➢ Manifold learning methods : 비선형 차원 축소

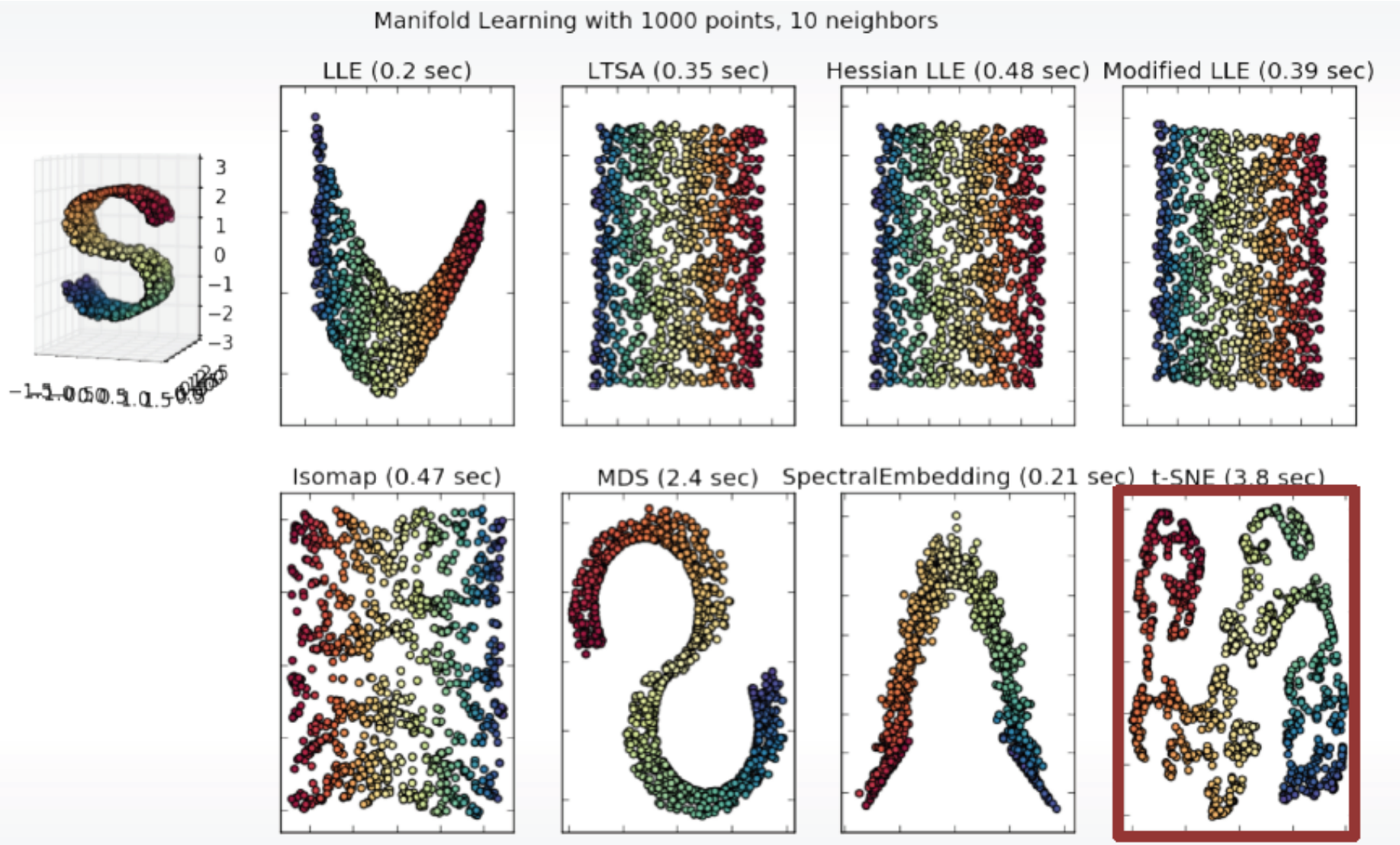


Comparison of Manifold Learning methods, http://scikitlearn.org/stable/auto_examples/manifold/plot_compare_methods.html#sphx-glr-auto-examples-manifold-plot-compare-methods-py

shkim.hi@gmail.com 김수현

t−SNE (t−distributed Stochastic Neighbor Embedding)

➢ Manifold learning methods : 비선형 차원 축소



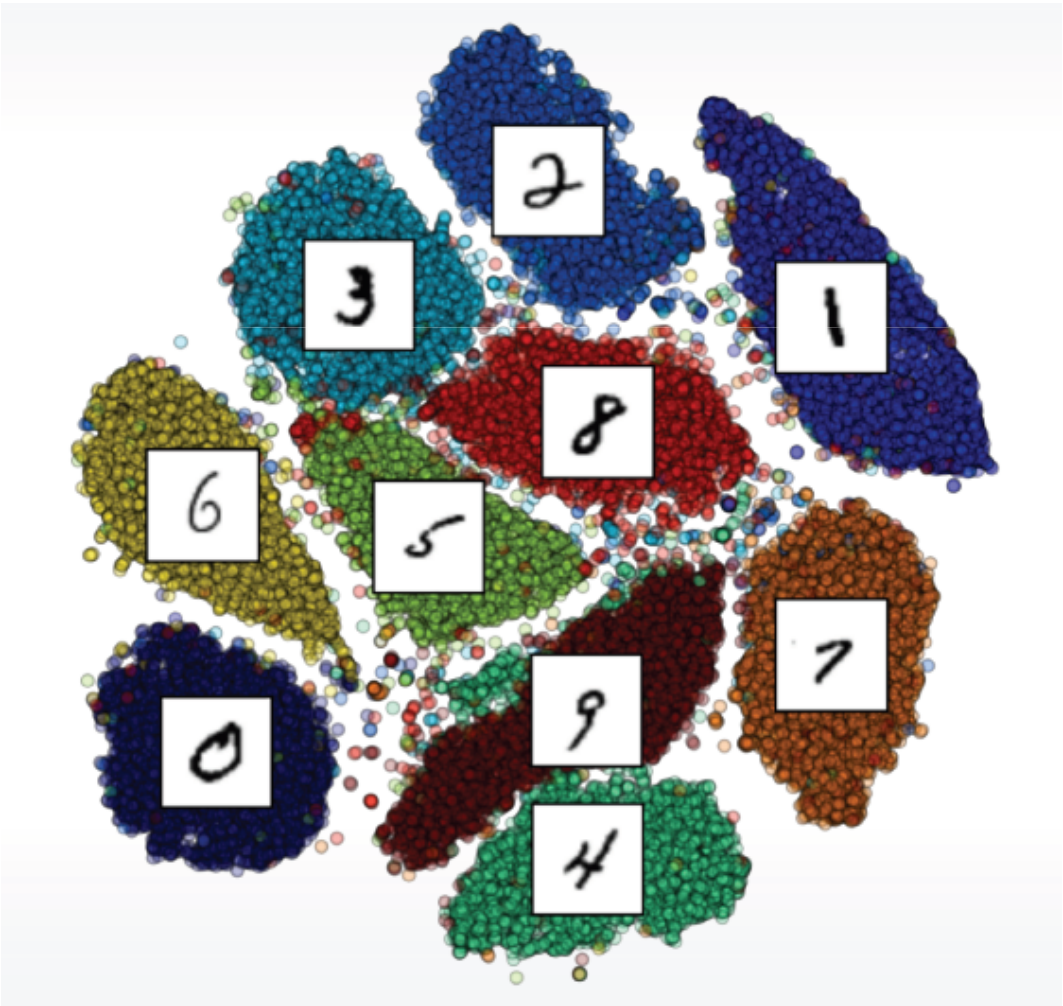
Comparison of Manifold Learning methods, http://scikitlearn.org/stable/auto_examples/manifold/plot_compare_methods.html#sphx-glr-auto-examples-manifold-plot-compare-methods-py

shkim.hi@gmail.com 김수현

t−SNE (t−distributed Stochastic Neighbor Embedding)

➢ Example: tSNE on MNIST

- MNIST 데이터 세트와 같이 데이터가 명시적인 구조를 갖는 경우 tSNE 플롯에 반영 될가능성이 높음 → **t-SNE가 EDA에서 많이 사용됨**



shkim.hi@gmail.com 김수현

t−SNE (t−distributed Stochastic Neighbor Embedding)

- **sklearn t-SNE 라이브러리 사용 방법**
  - 고차원 원공간 →저차원의 임베딩 공간

$$p_{j|i} = \frac{exp(-|x_i - x_j|^2/2\sigma_i^2)}{\sum_{k \neq i} exp(-|x_i - x_k|^2/2\sigma_i^2)}$$

```
from sklearn.manifold import TSNE

TSNE(n_components=2, perplexity=30.0, early_exaggeration=12.0,
     learning_rate=200.0, n_iter=1000, n_iter_without_progress=300,
     min_grad_norm=1e-07, metric='euclidean', init='random', verbose=0,
     random_state=None, method='barnes_hut', angle=0.5)
```

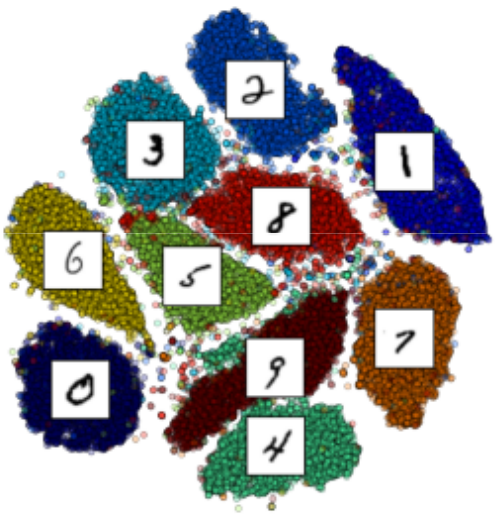
parameter	note
n_components	임베딩 공간의 차원
perplexity	σ_i 의 기준, 학습에 영향을 주는 점들의 개수를 조절
metric	원 공간에서의 두 점간의 거리 척도
method	원 공간 데이터의 인덱싱 방식

shkim.hi@gmail.com 김수현

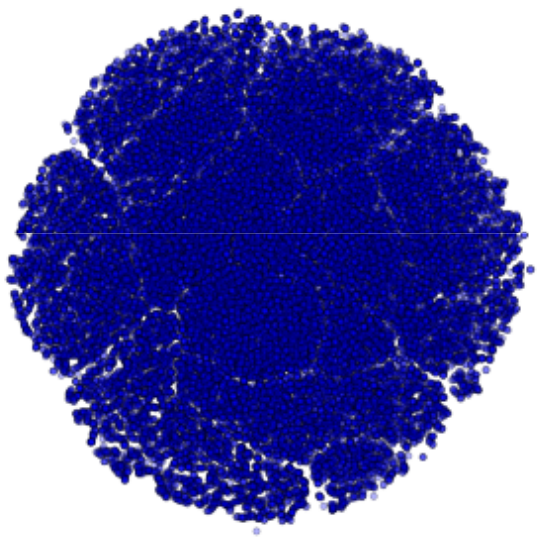
t−SNE (t−distributed Stochastic Neighbor Embedding)

➢ Interpretation of tSNE

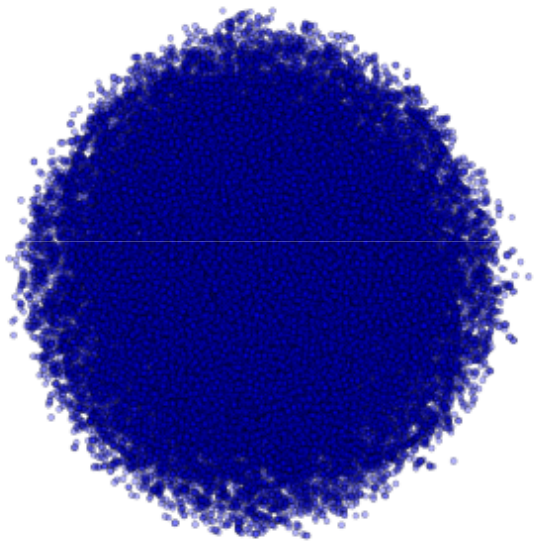
- ◆ parameter 조정에 따라 결과가 매우 달라짐
  - → perplexity parameter 조정 예



MNIST



MNIST



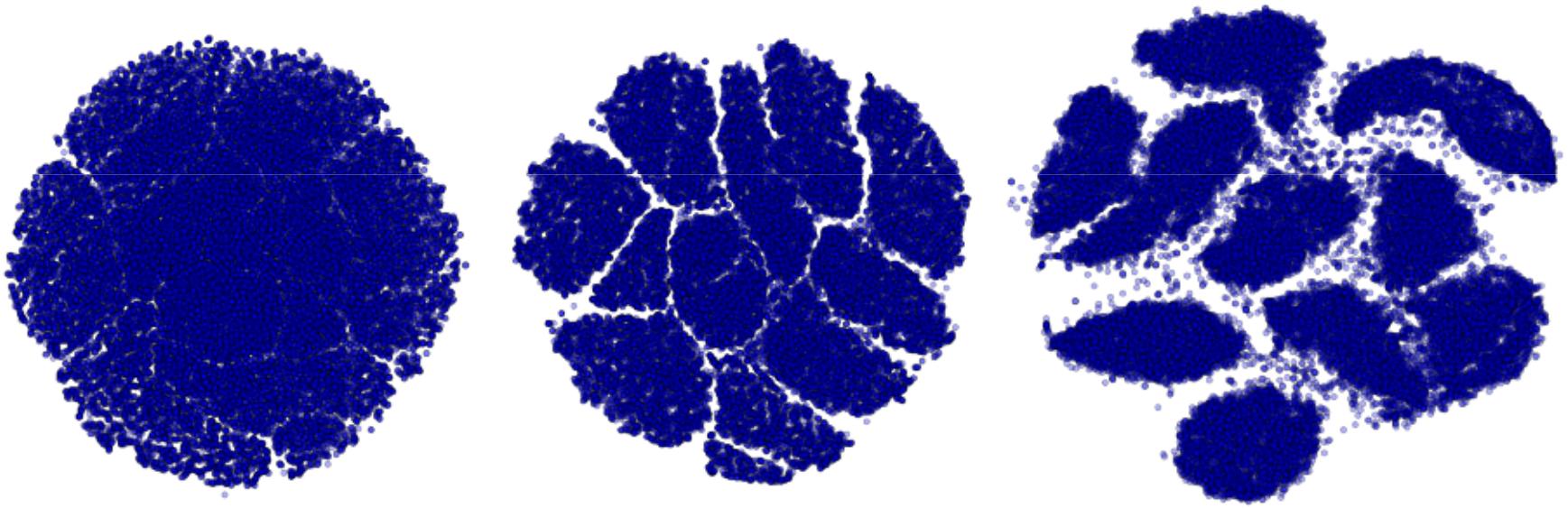
Random Values

shkim.hi@gmail.com 김수현

t−SNE (t−distributed Stochastic Neighbor Embedding)

➢ MNIST: different perplexities

- perplexity 파라미터의 값에 따라 투영 결과가 달라짐
  - → 투영 결과를 보고 원래 클러스터의 크기를 유추하기는 어려움
- 결과를 해석하는 것은 쉬운 일이아님

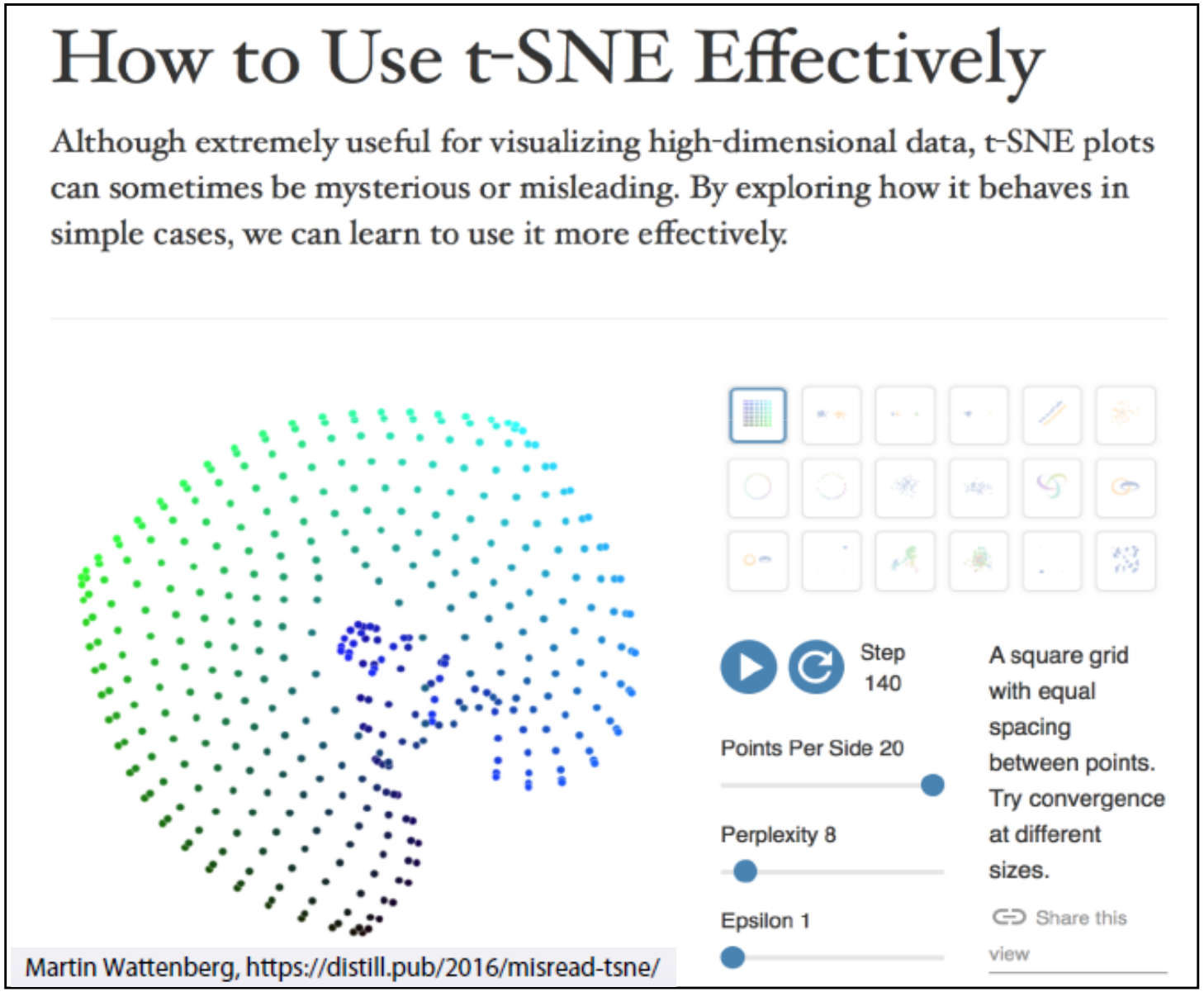


Perplexity=3 Perplexity=10 Perplexity=150

shkim.hi@gmail.com 김수현

t−SNE (t−distributed Stochastic Neighbor Embedding)

➢ Interactive Example



-t-SNE 결과를 이해하고 해석하는 방법 설명됨
-훌륭한 Interactive 데모도 있음

-EDA에서 t-SNE는 데이터로부터 새로운 피처를 생성하는 기능으로도 사용

shkim.hi@gmail.com 김수현

t−SNE (t−distributed Stochastic Neighbor Embedding)

➢ Practical Notes

- Result heavily depends on hyperparameters (perplexity)
  - → Good practice is to **use several projections with different perplexities (5-100)**
- Due to stochastic nature, tSNE provides different projections even for the same data\hyperparams
  - → **Train and test should be projected together**
- tSNE runs for a long time with a big number of features
  - → **it is common to do dimensionality reduction before projection.**
  - → 500개 이상의 피처인 경우에는 100개 이하로 축소해야 함

shkim.hi@gmail.com 김수현

t−SNE (t−distributed Stochastic Neighbor Embedding)

Practical Notes

- Implementation of tSNE can be found in sklearn library.
- But personally I prefer you use stand-alone implementation python package tsne due to its' faster speed.

shkim.hi@gmail.com 김수현

t−SNE (t−distributed Stochastic Neighbor Embedding)

Conclusion

- tSNE is a great tool for visualization
- It can be used as feature as well
- Be careful with interpretation of results
- Try different perplexities

shkim.hi@gmail.com 김수현

Graded Advanced Features Engineering II Quiz

1. Imagine that we apply X = PCA(n_components=5).fit_transform(data) and data has shape (5000, 53). What is the shape of X?

   - ○ (5, 53)
   - ○ (5000, 5)
   - ○ (5, 5000)
   - ○ (53, 5)

shkim.hi@gmail.com 김수현

Graded Advanced Features Engineering II Quiz

2. **To which data NMF is NOT applicable?**

- ○ Bag-of-words matrix
- ○ Standartized matrix
- ○ One-Hot encoded feature

shkim.hi@gmail.com 김수현

Graded Advanced Features Engineering II Quiz

3. **Suppose we have 2 categorical features: f1 with A possible values and f2 with B possible values. How many values will their interaction have?**

- ○ Exactly A + B
- ○ Exactly A * B
- ○ Less or equal to A * B
- ○ max(A, B)

shkim.hi@gmail.com 김수현

Graded Advanced Features Engineering II Quiz

4. **Imagine we have 2 categorical features represented as integers: f1 with all values in range [0, 1000] and f2 with values in range [0, 100]. What is the correct way to build their interaction?**

- ○ f1 + f2
- ○ f1.astype(str) + f2.astype(str)
- ○ f1.astype(str) + "_" + f2.astype(str)
- ○ (f1 + f2).astype(str)

shkim.hi@gmail.com 김수현

Graded Advanced Features Engineering II Quiz

5. **What is a correct way to get t-SNE projection of train and test data?**

- ○ Apply t-SNE to the train and after that to the test.
- ○ Apply t-SNE to the test first and after to train.
- ○ Apply t-SNE to concatenation of train and test and split projection back.
- ○ Doesn't matter, all variants will produce the same result.

shkim.hi@gmail.com 김수현

Graded Advanced Features Engineering II Quiz

6. **Is it possible to do t-SNE projection into 20-dimensional space?**

- ○ Yes, why not.
- ○ No, only 2-dim or 3-dim projections are possible.

shkim.hi@gmail.com 김수현

Advanced Features Engineering II – 참고 자료

- **Matrix Factorization:**
  - Overview of Matrix Decomposition methods (sklearn)
    - → https://scikit-learn.org/stable/modules/decomposition.html

- **Interactions:**
  - Facebook Research's paper about extracting categorical features from trees
    - → https://research.fb.com/publications/practical-lessons-from-predicting-clicks-on-ads-at-facebook/
  - Example: Feature transformations with ensembles of trees (sklearn)
    - → https://scikit-learn.org/stable/auto_examples/ensemble/plot_feature_transformation.html

shkim.hi@gmail.com 김수현

Advanced Features Engineering II – 참고 자료

- **t-SNE:**
  - Multicore t-SNE implementation
    - → https://github.com/DmitryUlyanov/Multicore-TSNE
  - Comparison of Manifold Learning methods (sklearn)
    - → https://scikit-learn.org/stable/auto_examples/manifold/plot_compare_methods.html
  - How to Use t-SNE Effectively (distill.pub blog)
    - → https://distill.pub/2016/misread-tsne/
  - tSNE homepage (Laurens van der Maaten)
    - → https://lvdmaaten.github.io/tsne/
  - Example: tSNE with different perplexities (sklearn)
    - → https://scikit-learn.org/stable/auto_examples/manifold/plot_t_sne_perplexity.html#sphx-glr-auto-examples-manifold-plot-t-sne-perplexity-py

shkim.hi@gmail.com 김수현