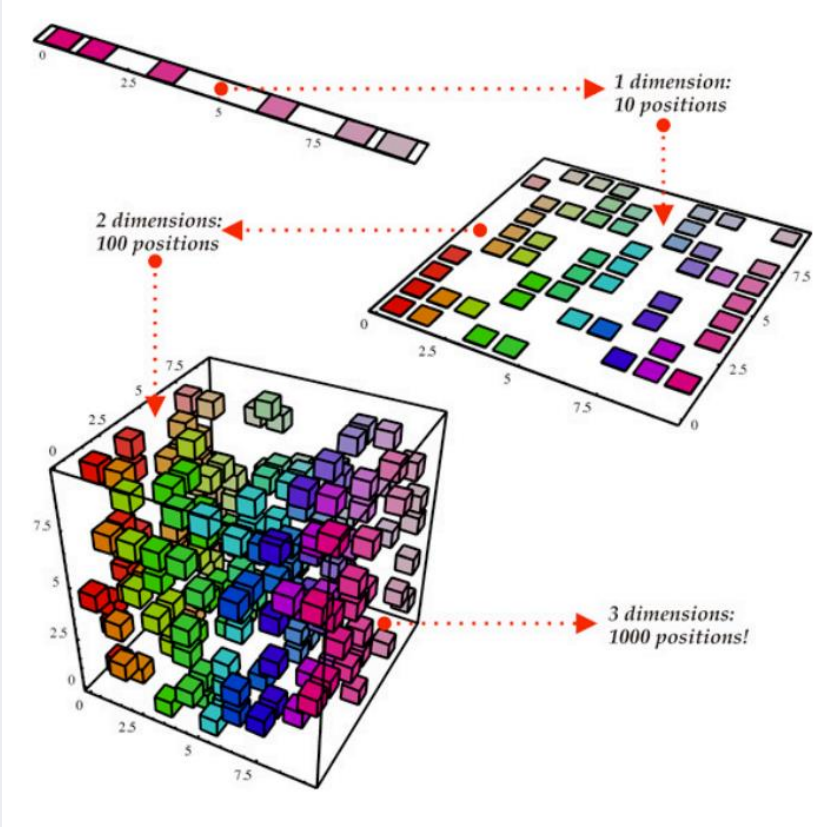


Lecture 6: Dimensionality Reduction

Pilsung Kang

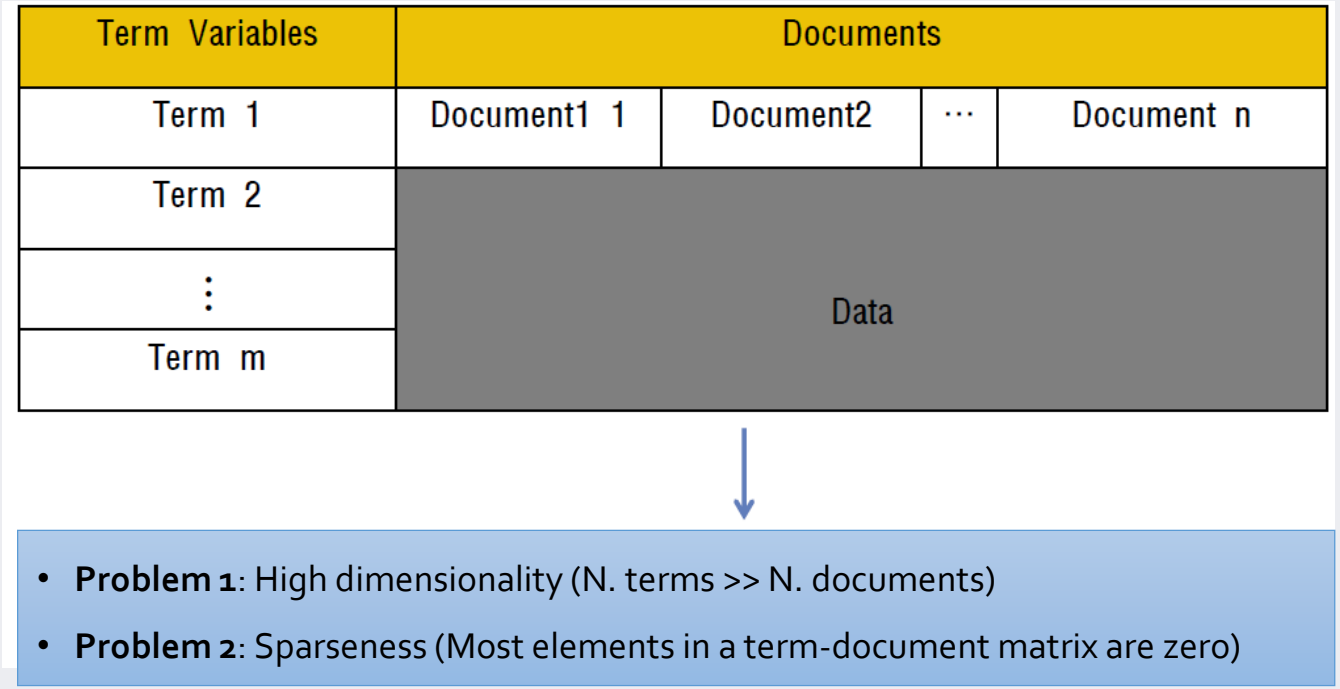
School of Industrial Management Engineering

Korea University

AGENDA

Dimensionality Reduction

- Common features of text data
  - ✓ In general, a document consists of a large number of terms (words)
  - ✓ Only a few of them are actually relevant to text mining tasks even after some preprocessing (stop-words removal, stemming, lemmatization, etc)

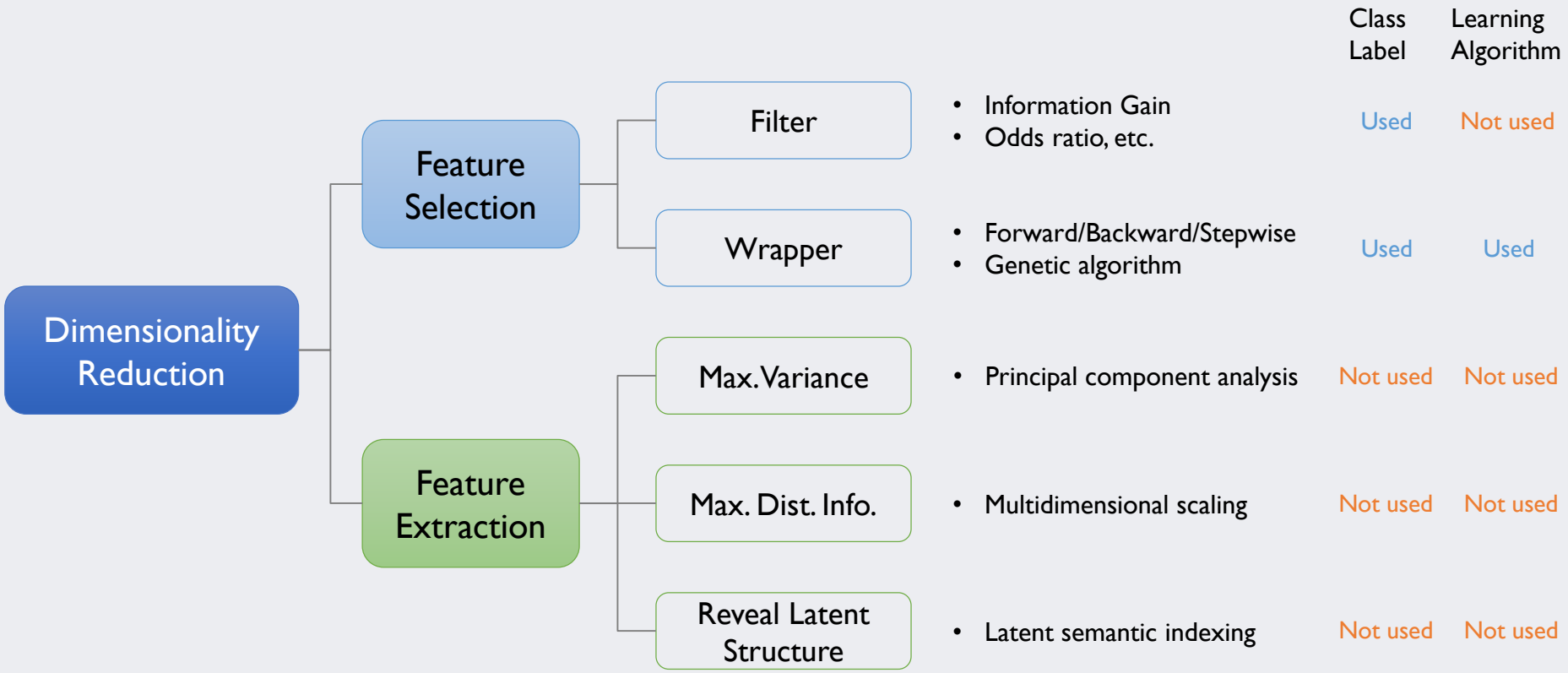


Dimensionality Reduction

- Why is dimensionality reduction necessary?
  - ✓ To make large problems computationally efficient (conserving computation, storage and network resources)
  - ✓ To improve the quality of text mining results
    - Improve classification accuracy or clustering modularity
    - Reduce the amount of training data needed to obtain a desired level of performance

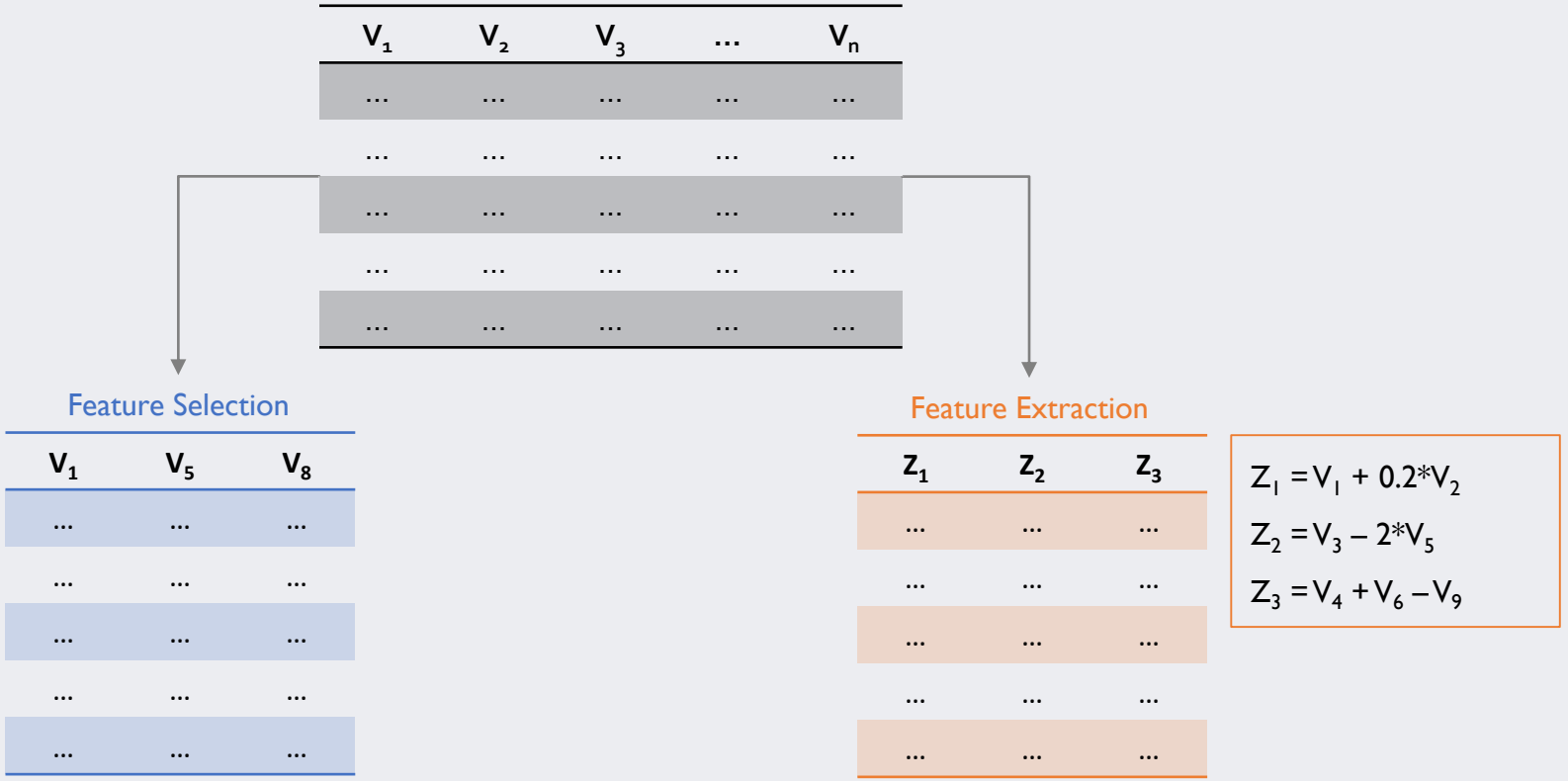
Dimensionality Reduction

- A simplified taxonomy of dimensionality reduction techniques



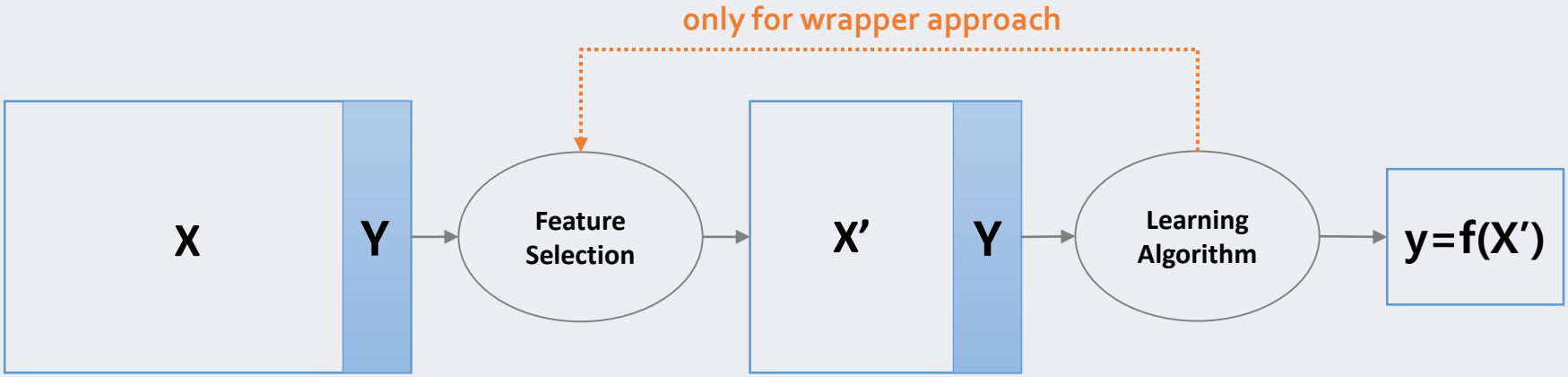
Dimensionality Reduction

- Feature selection vs. feature extraction
  - ✓ Feature selection: select a small subset of original variables
  - ✓ Feature extraction: construct/extract a new set of features based on the original variables



Dimensionality Reduction

- Filter approach vs. Wrapper approach
  - ✓ **Filter**: select a set of features based on pre-defined criteria
    - no feedback loop, independent of the learning algorithm
  - ✓ **Wrapper**: evaluate a subset with a learning algorithm and repeat the process until a certain level of performance is achieved
    - Feedback loop exists, dependent on the learning algorithm



AGENDA

01 Dimensionality Reduction

02 Feature Selection

03 Feature Extraction: LSA & t-SNE

04 R Exercise

Artificial Data Set

- 10 Documents with 10 Terms
  - ✓ Binary classification/categorization problem
  - ✓ 6 positive documents & 4 negative documents
  - ✓ Binary Term-Document matrix

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
Term 1	1	1	1	1	1	1	0	0	0	0
Term 2	0	0	0	0	0	0	1	1	1	1
Term 3	1	1	1	1	1	1	1	1	1	1
Term 4	1	1	1	1	1	1	1	1	0	0
Term 5	0	0	0	1	1	1	1	1	1	1
Term 6	1	1	1	0	0	0	0	0	0	0
Term 7	0	0	0	0	0	0	1	1	0	0
Term 8	1	0	1	0	1	0	1	0	1	0
Term 9	1	1	1	0	0	0	1	0	0	0
Term 10	1	0	0	0	0	0	0	0	1	1
Class	Pos	Pos	Pos	Pos	Pos	Pos	Neg	Neg	Neg	Neg

Feature Selection Metric 1-4

- Document frequency (DF)
  - ✓ Simply count the number of total documents in which a word *w* is presented.

$$DF(w) = N_D(w)$$

- Accuracy (Acc)/Accuracy ratio (AccR)
  - ✓ Expected accuracy of a simple classifier built from the single feature

$$Acc(w) = N(Pos, w) - N(Neg, w) \qquad AccR(w) = \left| \frac{N(Pos, w)}{N(Pos)} - \frac{N(Neg, w)}{N(Neg)} \right|$$

- Probability Ratio (PR)
  - ✓ The probability of the word given the positive class divided by the probability of the word given the negative class

$$PR(w) = \frac{N(Pos, w)}{N(Pos)} / \frac{N(Neg, w)}{N(Neg)}$$

Feature Selection Metric 1-4

- Compute the metric 1-4 for the data set

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	DF	Acc	AccR	PR
Term 1	1	1	1	1	1	1	0	0	0	0	6	6	1.00	Inf
Term 2	0	0	0	0	0	0	1	1	1	1	4	-4	1.00	0.00
Term 3	1	1	1	1	1	1	1	1	1	1	10	2	0.00	1.00
Term 4	1	1	1	1	1	1	1	1	0	0	8	4	0.50	2.00
Term 5	0	0	0	1	1	1	1	1	1	1	7	-1	0.50	0.50
Term 6	1	1	1	0	0	0	0	0	0	0	3	3	0.50	Inf
Term 7	0	0	0	0	0	0	1	1	0	0	2	-2	0.50	0.00
Term 8	1	0	1	0	1	0	1	0	1	0	5	1	0.00	1.00
Term 9	1	1	1	0	0	0	1	0	0	0	4	2	0.25	2.00
Term 10	1	0	0	0	0	0	0	0	1	1	3	-1	0.33	0.33
Class	Pos	Pos	Pos	Pos	Pos	Pos	Neg	Neg	Neg	Neg				

Feature Selection Metric 5-6

- Odds ratio (OddR)
  - ✓ Reflect the odds of the word occurring in the positive class normalized by that of the negative class
    - It has been used for relevance ranking in information retrieval

$$OddR(w) = \frac{N(Pos, w)}{N(Neg, w)} \times \frac{N(Neg, \overline{w})}{N(Pos, \overline{w})}$$

    - Add 1 to any zero count in the denominator to avoid division by zero
- Odds ratio Numerator (OddN)

$$OddN(w) = N(Pos, w) \times N(Neg, \overline{w})$$

Feature Selection Metric 7

- F1-Measure
  - ✓ Expected accuracy of a simple classifier built from the single feature

$$F1(w) = \frac{2 \times Recall(w) \times Precision(w)}{Recall(w) + Precision(w)}$$

$$Recall(w) = \frac{N(Pos, w)}{N(Pos, w) + N(Pos, \overline{w})}, \qquad Precision(w) = \frac{N(Pos, w)}{N(Pos, w) + N(Neg, w)}$$

  - ✓ By doing some arithmetic operations, we can derive

$$F1(w) = \frac{2 \times N(Pos, w)}{N(Pos) + N(w)}$$

  - ✓ In F1 measure, negative features are devalued compared to positive features

Feature Selection Metric 5-7

- Compute the metric 5-7 for the data set

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	OddR	OddN	F1
Term 1	1	1	1	1	1	1	0	0	0	0	24.00	24	1.00
Term 2	0	0	0	0	0	0	1	1	1	1	0.00	0	0.00
Term 3	1	1	1	1	1	1	1	1	1	1	0.00	0	0.75
Term 4	1	1	1	1	1	1	1	1	0	0	4.00	12	0.86
Term 5	0	0	0	1	1	1	1	1	1	1	0.00	0	0.46
Term 6	1	1	1	0	0	0	0	0	0	0	4.00	12	0.67
Term 7	0	0	0	0	0	0	1	1	0	0	0.00	0	0.00
Term 8	1	0	1	0	1	0	1	0	1	0	1.00	6	0.55
Term 9	1	1	1	0	0	0	1	0	0	0	3.00	9	0.60
Term 10	1	0	0	0	0	0	0	0	1	1	0.20	2	0.22
Class	Pos	Pos	Pos	Pos	Pos	Pos	Neg	Neg	Neg	Neg			

Feature Selection Metric 8

- Information Gain: IG
  - ✓ Measures the decrease in entropy when the feature is given vs. absent.
  - ✓ Entropy without the information provided by the term w

$$Entropy(absent\ w) = \sum_{C \in \{Pos,Neg\}} -P(C) \times \log(P(C))$$

$$Entropy(given\ w) = P(w)\left[\sum_{C \in \{Pos,Neg\}} -P(C|w) \times \log(P(C|w))\right] + P(\overline{w})\left[\sum_{C \in \{Pos,Neg\}} -P(C|\overline{w}) \times \log(P(C|(\overline{w}))\right]$$

$$IG(w) = Entropy(absent\ w) - Entropy(given\ w)$$

Feature Selection Metric 8

- Information Gain: IG
  - ✓ For Term 1

$$Entropy(absent\ w) = -P(Pos) \times \log(P(Pos)) - P(Neg) \times \log(P(Neg))$$
$$= -0.6 \times \log(0.6) - 0.4 \times \log(0.4)$$
$$= 0.29$$

$$Entropy(given\ w) = -P(w)[-P(Pos|w) \times \log(P(Pos|w)) - P(Neg|w) \times \log(P(Neg|w))]$$
$$= P(\bar{w})\left[-P(Pos|\bar{w}) \times \log(P(Pos|\bar{w})) - P(Neg|\bar{w}) \times \log\left(P(Neg|\bar{w})\right)\right]$$
$$= 0.6[-1 \times \log(1) - 0 \times \boxed{\log(0)}] + 0.4[-0 \times \boxed{\log(0)} - 1 \times \log(1)]$$
$$= 0$$

Convert log(o) to zero

$$IG(w) = 0.29 - 0 = 0.29$$

Feature Selection Metric 9

- Chi-squared statistic (χ^2)
  - ✓ Measures divergence from the distribution expected if one assumes the feature occurrence is independent of the class label

$$\chi^2(w) = \frac{N \times [P(Pos,w) \times P(Neg,\overline{w}) - P(Neg,w) \times P(Pos,\overline{w})]^2}{P(w) \times P(\overline{w}) \times P(Pos) \times P(Neg)}$$

Term 1	Pos	Neg	Total
w	6	0	6
$\overline{w}$	0	4	4
total	6	4	10

Term 4	Pos	Neg	Total
w	6	2	8
$\overline{w}$	0	2	2
total	6	4	10

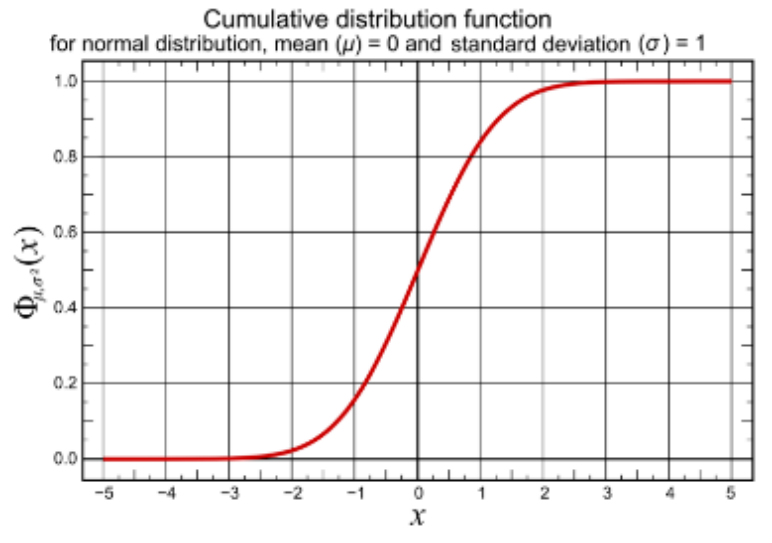
$$\chi^2(T1) = \frac{10 \times [0.6 \times 0.4 - 0 \times 0]^2}{0.6 \times 0.4 \times 0.6 \times 0.4} = 10.00 \qquad \chi^2(T4) = \frac{10 \times [0.6 \times 0.2 - 0.2 \times 0]^2}{0.8 \times 0.2 \times 0.6 \times 0.4} = 3.75$$

Feature Selection Metric 10

- Bi-Normal Separation (BNS)
  - ✓ Measures the degree of separation assuming that the occurrence of a feature in a document is a random process following a normal distribution

$$BNS(w) = \left| F^{-1}\left(\frac{N(Pos,w)}{N(Pos)}\right) - F^{-1}\left(\frac{N(Neg,w)}{N(Neg)}\right) \right|$$

F: c.d.f of the standard normal distribution



Term 4	Pos	Neg	Total
w	6	2	8
$\overline{w}$	0	2	2
total	6	4	10

$$BNS(w) = |F^{-1}(1) - F^{-1}(0.5)|$$
$$\approx |F^{-1}(0.9995) - F^{-1}(0.5)|$$
$$= |3.29 - 0| = 3.29$$

Feature Selection Metric 8-10

- Compute the metric 8-10 for the data set

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	IG	χ^2	BNS
Term 1	1	1	1	1	1	1	0	0	0	0	0.29	10.00	6.58
Term 2	0	0	0	0	0	0	1	1	1	1	0.29	10.00	6.58
Term 3	1	1	1	1	1	1	1	1	1	1	0.00	0.00	0.00
Term 4	1	1	1	1	1	1	1	1	0	0	0.10	3.75	3.29
Term 5	0	0	0	1	1	1	1	1	1	1	0.08	2.86	3.29
Term 6	1	1	1	0	0	0	0	0	0	0	0.08	2.86	3.29
Term 7	0	0	0	0	0	0	1	1	0	0	0.10	3.75	3.29
Term 8	1	0	1	0	1	0	1	0	1	0	0.00	0.00	0.00
Term 9	1	1	1	0	0	0	1	0	0	0	0.01	0.63	0.67
Term 10	1	0	0	0	0	0	0	0	1	1	0.03	1.27	0.97
Class	Pos	Pos	Pos	Pos	Pos	Pos	Neg	Neg	Neg	Neg			

Feature Selection Metric: Summary

- Comparison of the 10 feature selection metrics
  - ✓ For the positive class, the Term 4 is included for the top 3 variables by all metrics, followed by Term 1 and Term 6

	DF	Acc	AccR	PR	OddR	OddN	F1	IG	χ^2	BNS	Top3
Term 1	6	6	1.00	Inf	24.00	24	1.00	0.29	10.00	6.58	9
Term 2	4	-4	1.00	0.00	0.00	0	0.00	0.29	10.00	6.58	4
Term 3	10	2	0.00	1.00	0.00	0	0.75	0.00	0.00	0.00	2
Term 4	8	4	0.50	2.00	4.00	12	0.86	0.10	3.75	3.29	10
Term 5	7	-1	0.50	0.50	0.00	0	0.46	0.08	2.86	3.29	3
Term 6	3	3	0.50	Inf	4.00	12	0.67	0.08	2.86	3.29	6
Term 7	2	-2	0.50	0.00	0.00	0	0.00	0.10	3.75	3.29	4
Term 8	5	1	0.00	1.00	1.00	6	0.55	0.00	0.00	0.00	0
Term 9	4	2	0.25	2.00	3.00	9	0.60	0.01	0.63	0.67	1
Term 10	3	-1	0.33	0.33	0.20	2	0.22	0.03	1.27	0.97	0

Empirical Study

Forman (2003)

- Empirical study conducted by Forman (2010)
  - ✓ Data sets: 229 text classification tasks (from Reuters, TREC, OHSUMED, etc.)
  - ✓ SVM as a base classifier, one-against-all method for multiclass problems
  - ✓ Performances are evaluated in terms of accuracy, precision, recall, and F-1 measure

- Analysis purpose
  - ✓ To obtain the best overall classification performance regardless of the number of features
  - ✓ To find the best metric when only a very small number of features is selected
    - For limited resources, fast classification, and large scalability
  - ✓ Contract the performance under high-skew and low-skew class distribution situations

Empirical Study

Forman (2003)

- Metrics considered

Name	Description	Formula
Acc	Accuracy	$tp - fp$
Acc2	Accuracy balanced†	$\lvert tpr - fpr \rvert$
BNS	Bi-Normal Separation†	$\lvert F^{-1}(tpr) - F^{-1}(fpr) \rvert$ where F is the Normal c.d.f.
Chi	Chi-Squared‡	$t(tp, (tp+fp)P_{pos}) + t(fn, (fn+tn)P_{pos}) + t(fp, (tp+fp)P_{neg}) + t(tn, (fn+tn)P_{neg})$ where $t(count,expect) = (count - expect)^2 / expect$
DFreq	Document Frequency†‡o	$tp + fp$
F1	F_1-Measure	$\frac{2\,recall\ precision}{(recall + precision)} = \frac{2tp}{(pos + tp + fp)}$
IG	Information Gain†‡	$e(pos, neg) - [\, P_{word}\, e(tp, fp) + P_{\overline{word}}\, e(fn, tn)\,]$ where $e(x,y) = -\frac{x}{x+y}\log_2\frac{x}{x+y} - \frac{y}{x+y}\log_2\frac{y}{x+y}$
OddN	Odds Ratio Numerator	$tpr\,(1 - fpr)$
Odds	Odds Ratio†	$\frac{tpr\,(1-fpr)}{(1-tpr)\,fpr} = \frac{tp\ tn}{fp\ fn}$
Pow	Power	$(1-fpr)^k - (1-tpr)^k$ where k=5
PR	Probability Ratio	tpr / fpr
Rand	Random‡o	random()

† Acc2, BNS, DFreq, IG, and Odds select a substantial number of negative features.
‡ Chi, IG, DFreq, and Rand also generalize for multi-class problems.
o DFreq and Rand do not require the class labels.

Empirical Study

Forman (2003)

- Experimental result (1/5)
  - ✓ BNS performed best by a wide margin when using 500 to 1,000 features
  - ✓ IG can achieve slightly better performance then the model with all features

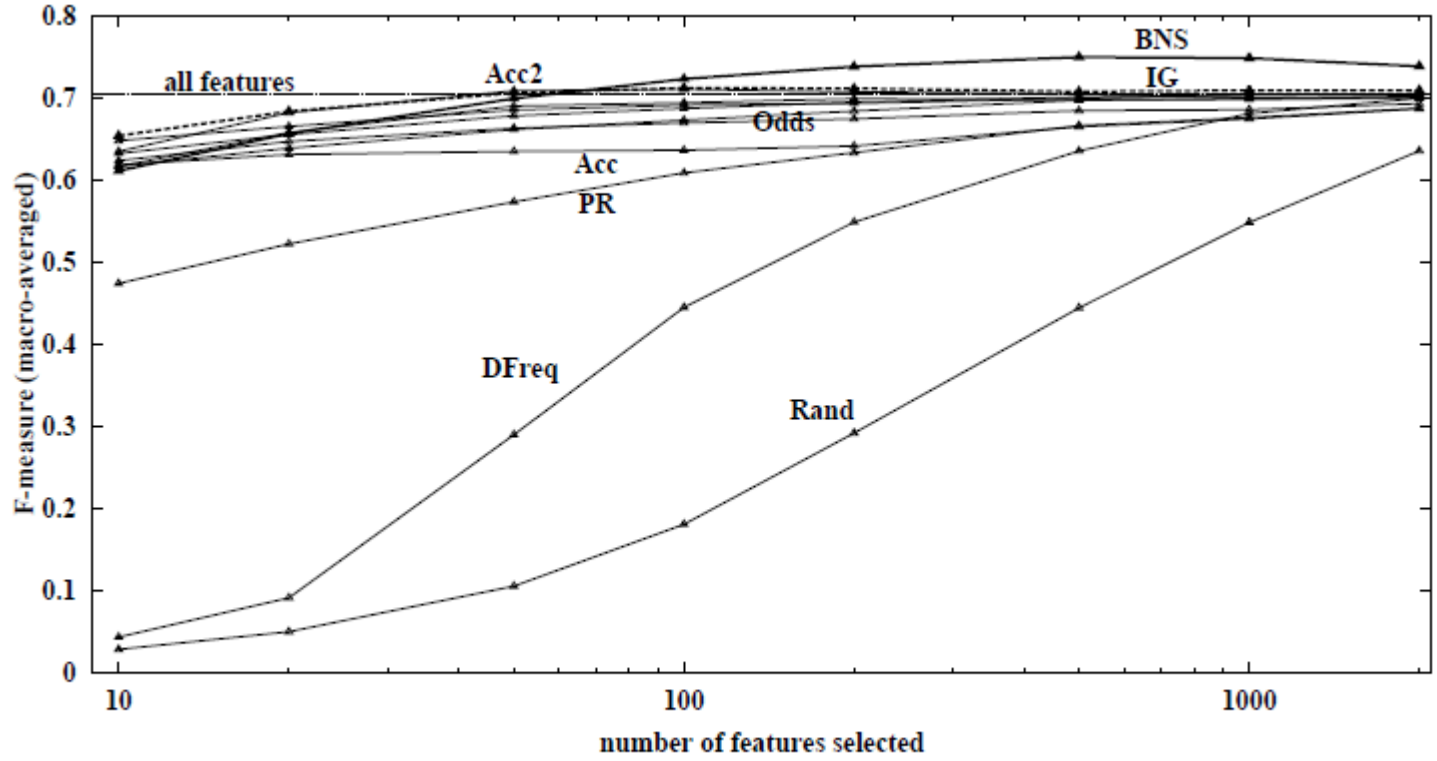


Figure 4. F-measure averaged over 229 problems for each metric, varying the number of features.

Empirical Study

Forman (2003)

- Experimental result (2/5)
  - ✓ A high recall of BNS contributes to a high F1-measure compared to others
  - ✓ If precision is the central goal, IG and χ^2 can be good choices

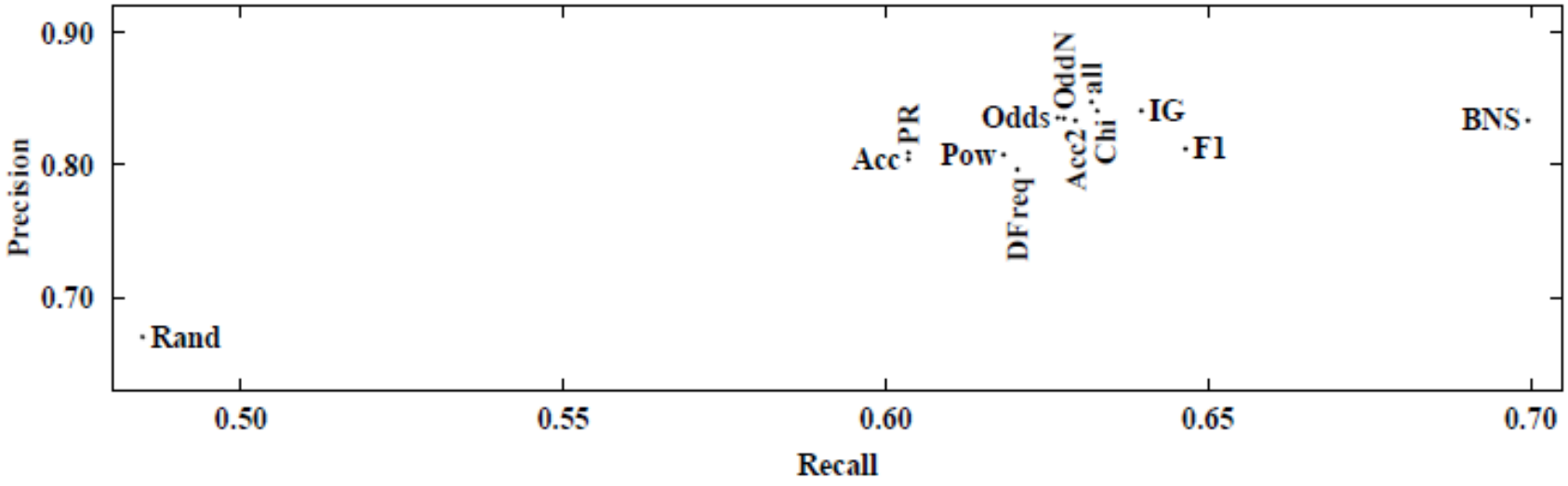


Figure 5. Precision-Recall tradeoffs from Figure 4 at 1000 features selected.

Empirical Study

Forman (2003)

- Experimental result (3/5)
  - ✓ Performances are degraded when the degree of class imbalance increases

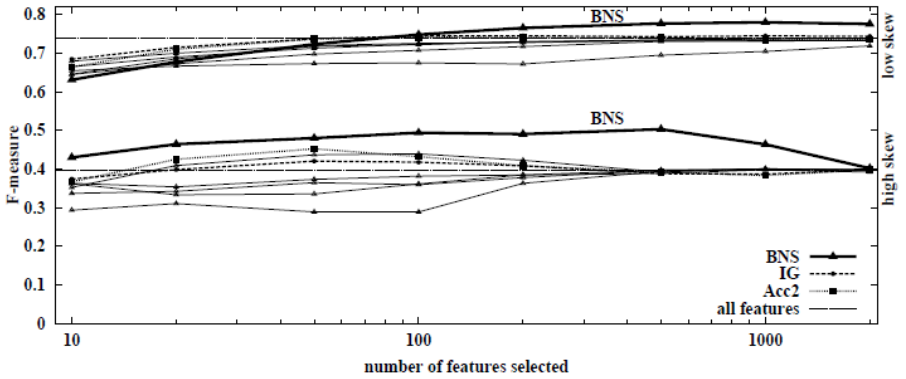


Figure 7. Average F-measure for each metric in low-skew and high-skew situations (threshold 1:67, the 90th percentile), as we vary the number of features. (To improve readability, we omitted Rand, DFreq, and PR.)

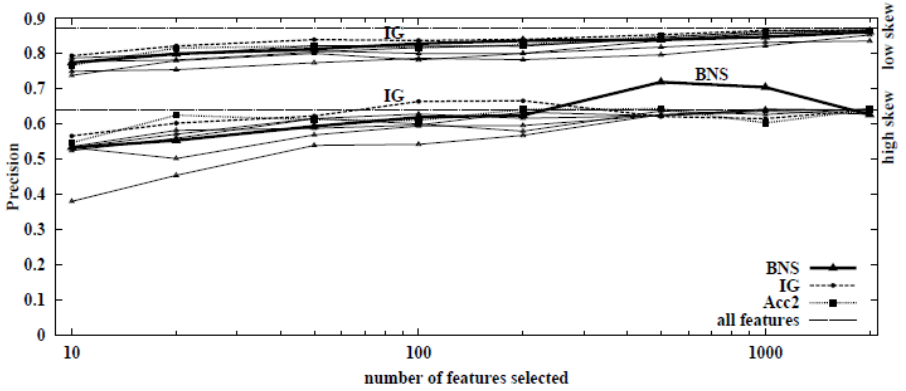


Figure 8. As Figure 7, but for precision.

Empirical Study

Forman (2003)

- Experimental result (4/5)
  - ✓ In terms of F-measure, BNS performed better than other feature selection metrics, followed by IG, and χ^2

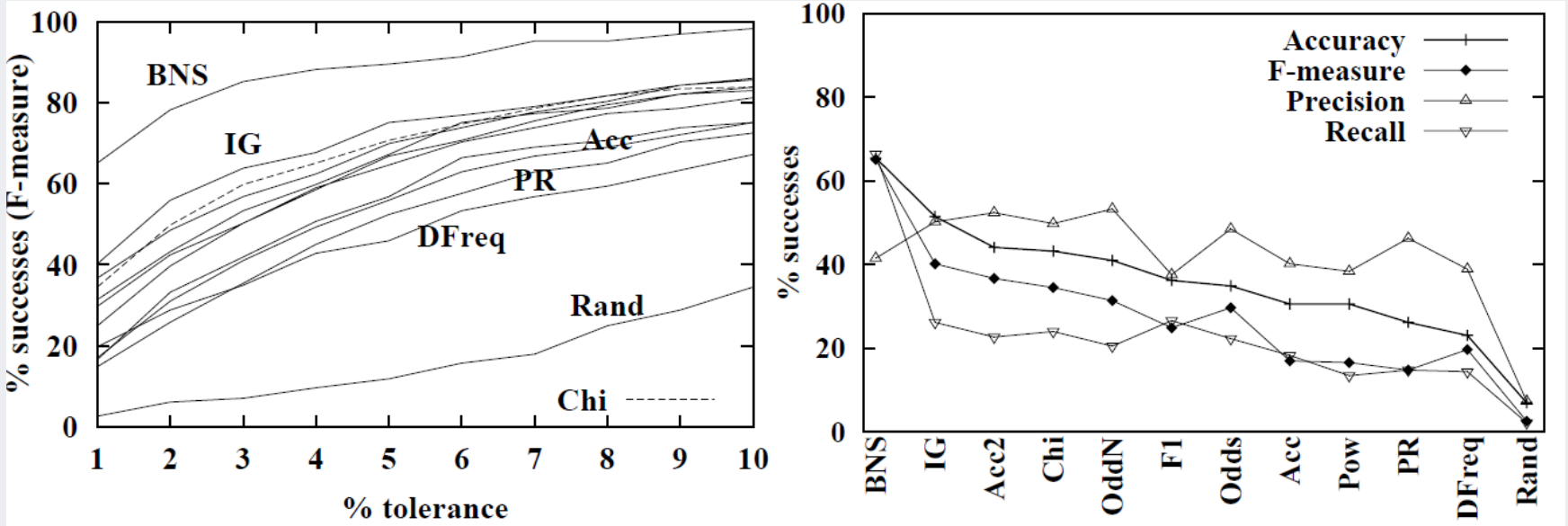


Figure 9. (a) Percentage of problems on which each metric scored within x% tolerance of the best F-measure of any metric. (b) Same, for F-measure, recall, and precision at a fixed tolerance of 1%, and for accuracy at a tolerance of 0.1%.

Empirical Study

Forman (2003)

- Experimental result (5/5)
  - ✓ In terms of precision, IG performed better than other metrics

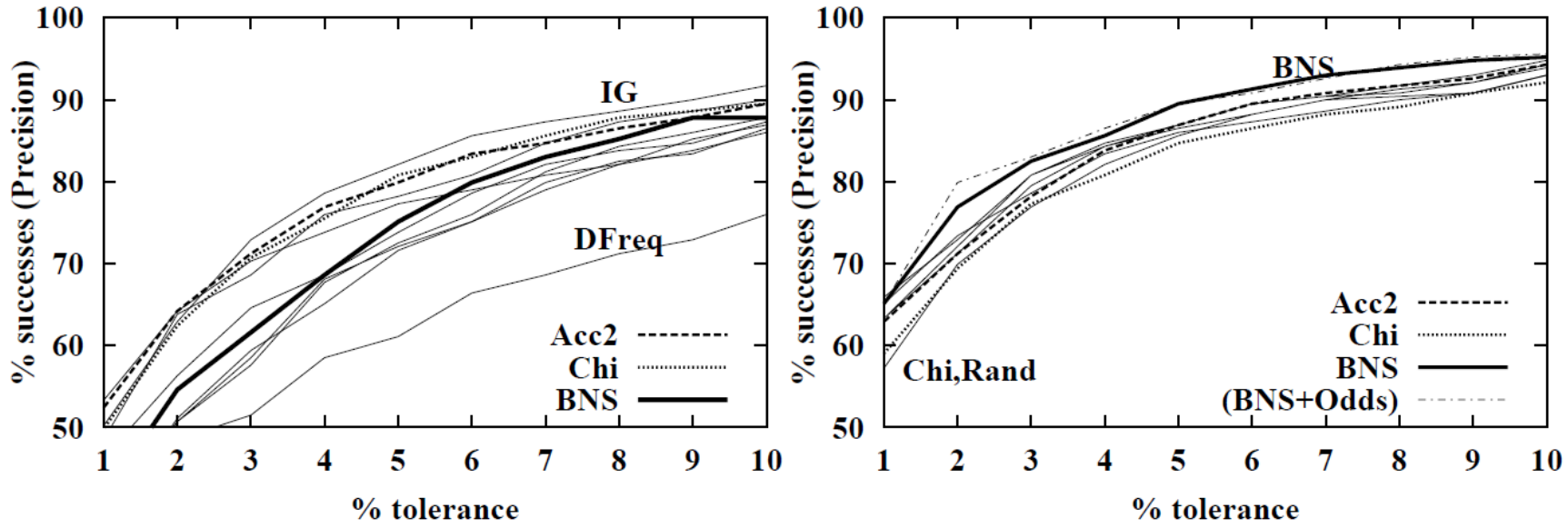


Figure 10. (a) As Figure 9a, but for precision. (b) Same axes and scale, but for each metric combined with IG. (Except the BNS+Odds curve is not combined with IG.)

AGENDA

01 Dimensionality Reduction

02 Feature Selection

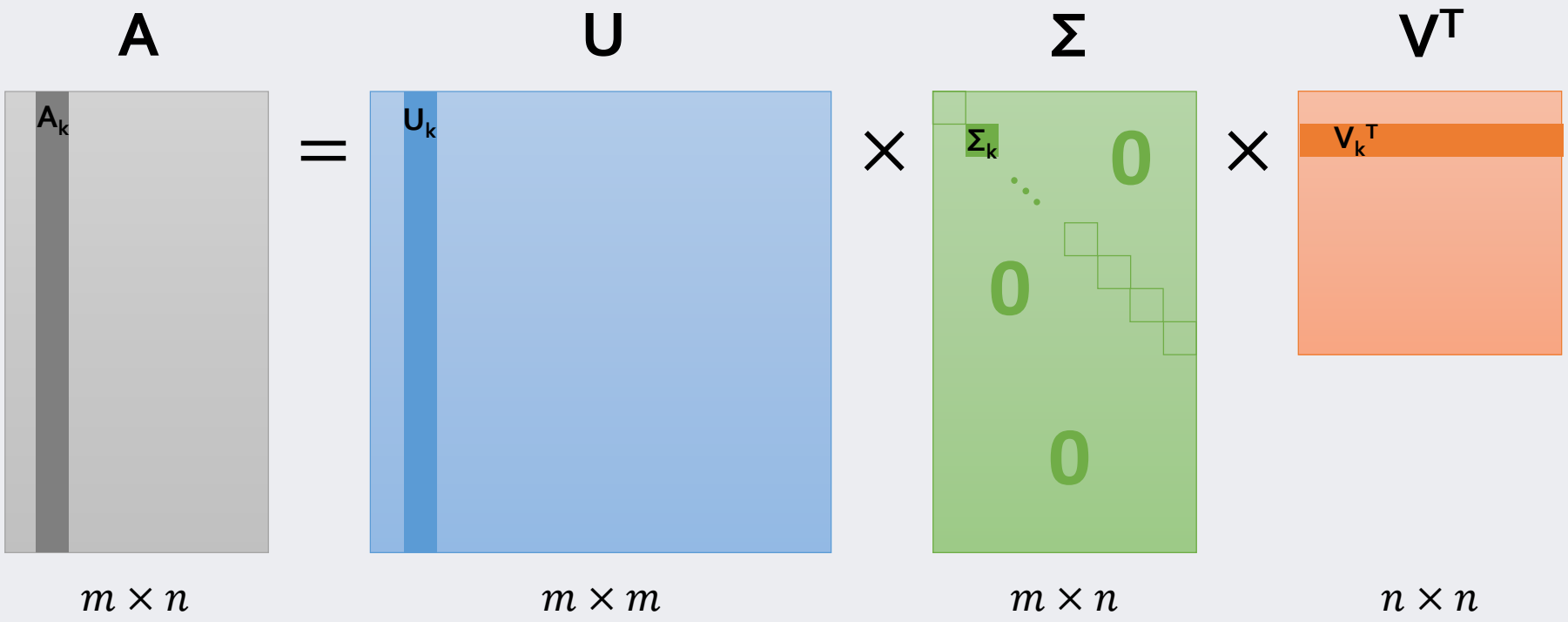
03 Feature Extraction: LSA & t-SNE

04 R Exercise

Latent Semantic Analysis

Deerwester et al. (1990)

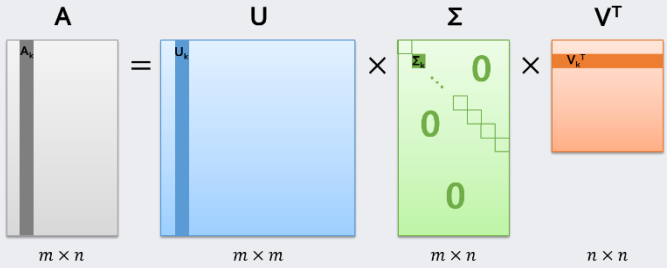
- Singular Value Decomposition: SVD
  - ✓ A factorization of a real or complex matrix
  - ✓ For a rectangular m by n matrix A (m > n) (Term-Document Matrix in Text Mining)
  - ✓ Full SVD becomes



Latent Semantic Analysis

- Properties of SVD
  - ✓ Singular vectors of the matrix U and V are orthogonal

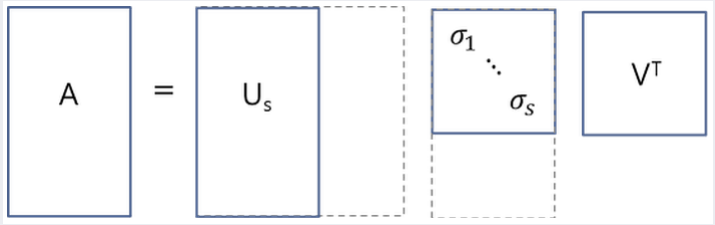
$$\mathbf{U}^{\mathbf{T}}\mathbf{U} = \mathbf{V}^{\mathbf{T}}\mathbf{V} = \mathbf{I}$$



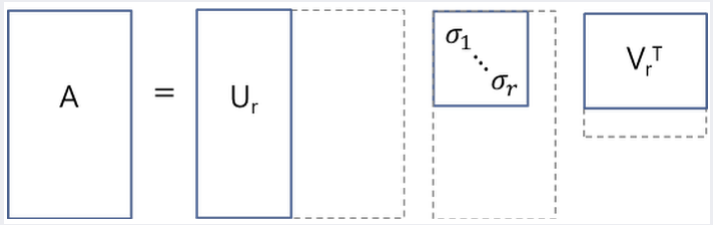
  - ✓ The number of positive singular values in **Σ** = Rank(A)
  - ✓ Running time: O(mnc)
    - c: the average number of words per document
- A simple example

Latent Semantic Analysis

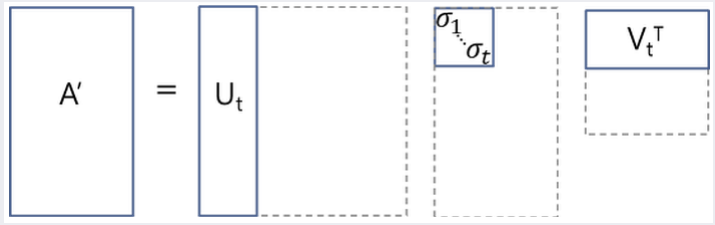
- Reduces SVDs
  - ✓ Thin SVD: Transform **Σ** to a square matrix & remove the corresponding columns of U



  - ✓ Compact SVD: Reduce **Σ** by removing zero-singular values and corresponding vectors in U and V



  - ✓ Truncated(Approximated) SVD: Preserve top t largest singular values in Σ and their corresponding vectors in U and V



Latent Semantic Analysis

- A simple example
  - ✓ SVD decomposition

$$A = \begin{bmatrix} 2 & 3 \\ 1 & 4 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad U = \begin{bmatrix} 0.82 & -0.58 & 0 & 0 \\ 0.58 & 0.82 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad S = \begin{bmatrix} 5.47 & 0 & 0 & 0 \\ 0 & 0.37 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad V = \begin{bmatrix} 0.40 & -0.91 \\ 0.91 & 0.40 \end{bmatrix}$$

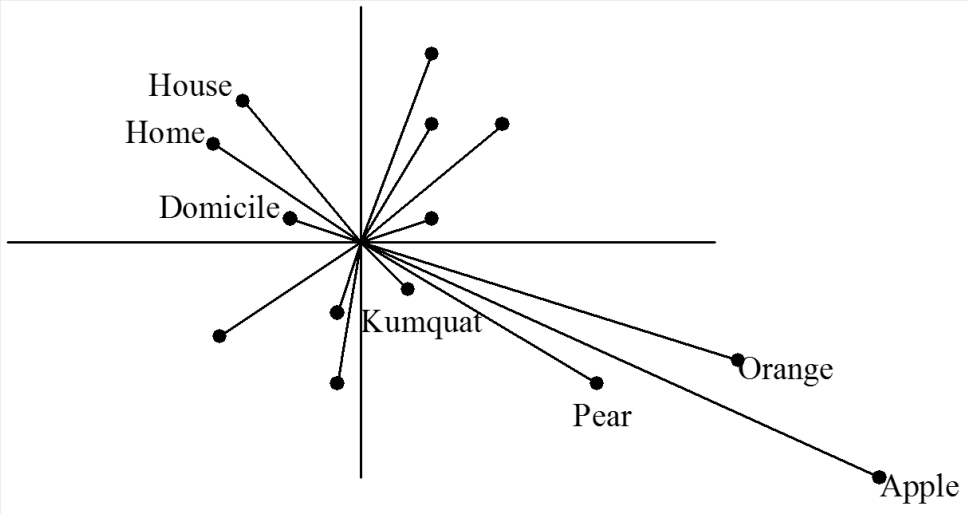
  - ✓ Truncated SVD

$$A' = U_1 \times S_1 \times V_1^T = \begin{bmatrix} 0.82 \\ 0.58 \\ 0 \\ 0 \end{bmatrix} \times [5.47] \times \begin{bmatrix} 0.40 & 0.91 \end{bmatrix} = \begin{bmatrix} 1.79 & 4.08 \\ 1.27 & 2.89 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Latent Semantic Analysis

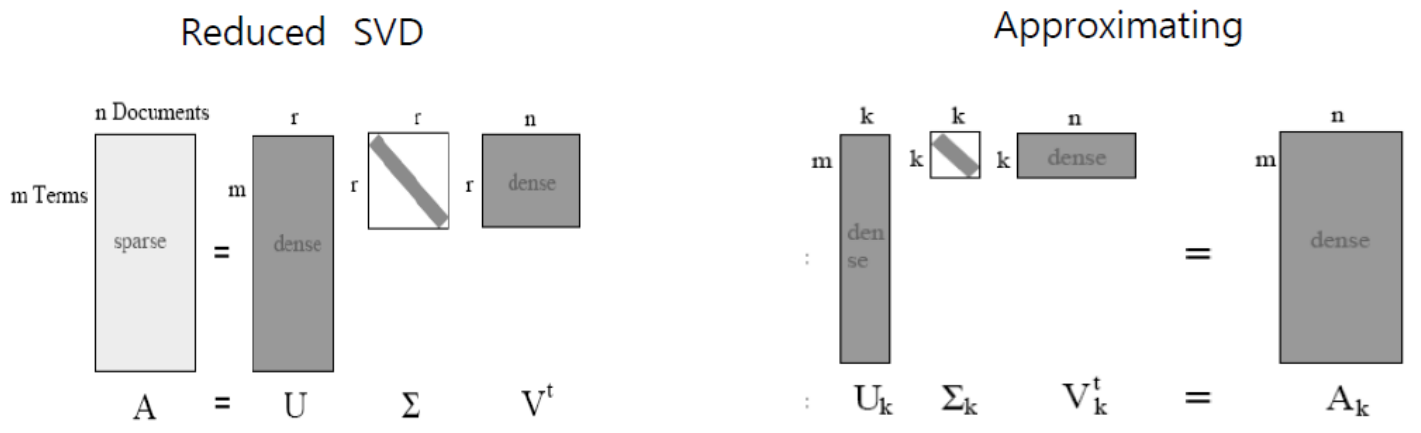
Papadimitriou et al.

- Latent Semantic Indexing
  - ✓ A process of term-document matrix to expose statistical structure by converting a high dimensional space to lower dimensional space
    - Can be used to not only to reduce the number of features, but also to reduce the number of documents
  - ✓ Latent: Captures associations which are not explicit
  - ✓ Semantic: Represent meaning as a function of similarity to other entities



Latent Semantic Analysis

- Reduce the dimensions using SVD



✓ Step 1) Construct the approximated matrix A_k from the original term-document matrix A using SVD

$$\mathbf{A} \approx \mathbf{A}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T$$

✓ Step 2) Multiply the transpose of U_k to obtain k (<<m) by n term-document matrix

$$\mathbf{U}_k^T \mathbf{A}_k = \mathbf{U}_k^T \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T = \mathbf{I} \mathbf{\Sigma}_k \mathbf{V}_k^T = \mathbf{\Sigma}_k \mathbf{V}_k^T$$

✓ Step 3: Apply data mining algorithms

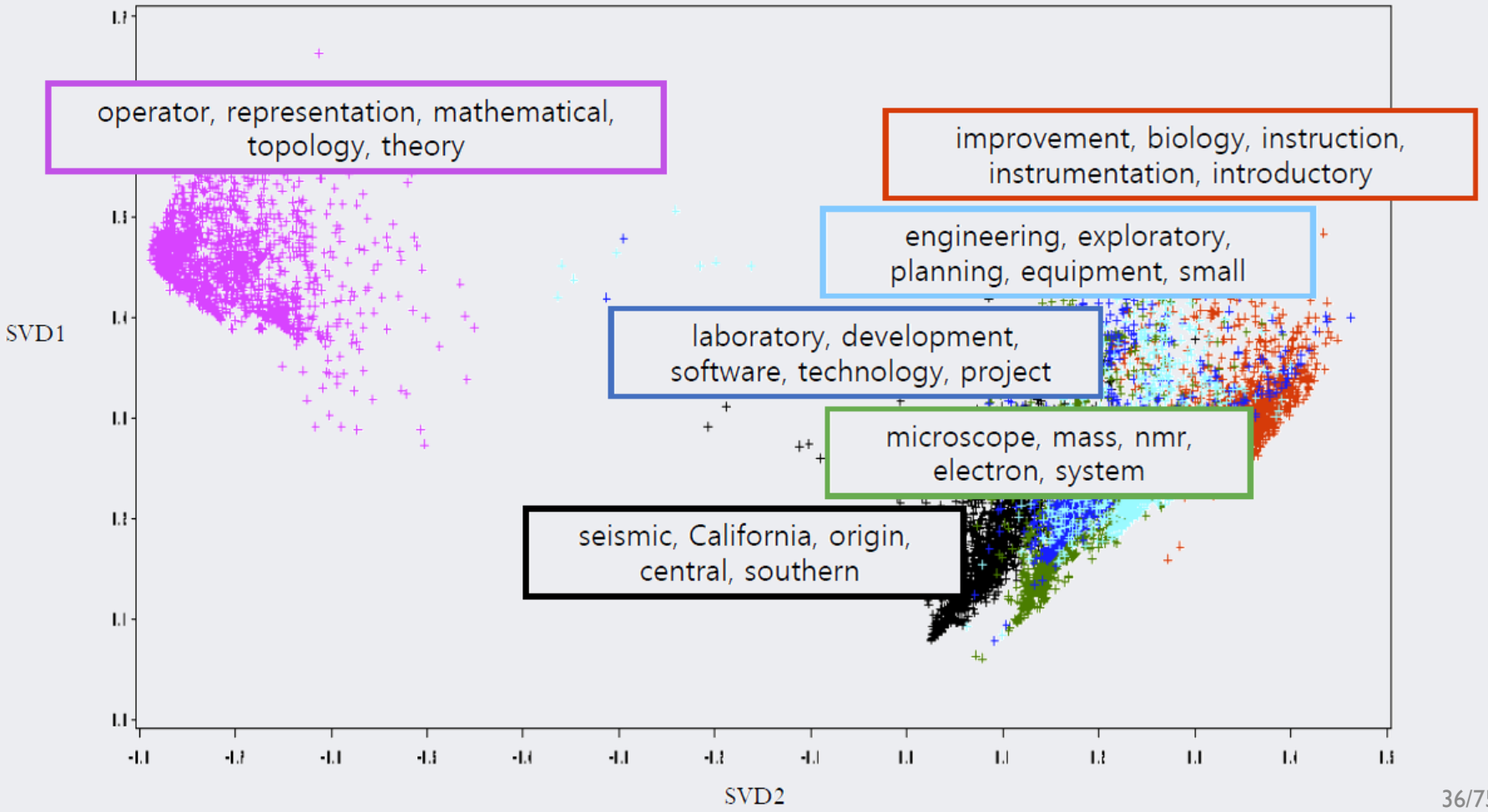
Latent Semantic Analysis

- SVD in text mining
  - ✓ Data: 41,717 abstracts of the research projects that were funded by National Science Foundation (NSF) between 1990 and 2003
  - ✓ Top 10 positive and negative keywords for each SVD

(−) SVD1	(+)	(−) SVD2	(+)	(−) SVD3	(+)
sister−chromatid	real	sum	electronics	revised	evaporation
plastids	other	diophantine	enhanced	major	agglomerate
segregation	reduction	mathematical	video	introductory	mobility
pneumoniae	dimensional	yang−mills	computer−based	miniority	transient
males	complex	noetherian	improved	computer−based	multicomponent
candida	multiple	differential equations	major	micro computer	lateral
trait	expansion	equations	support	laboratory	distribution
decline	domain	invariant	instructional	unix−based	copolymer
factorial	vector	asymtotic	theme	ample	compacted
gmp	stability	non−commutative	capability	inquiry	long

Latent Semantic Analysis

- SVD in text mining
  - ✓ Visualize the project in the reduced 2-D space



Stochastic Neighbor Embedding

Hinton and Roweis (2002)

- Stochastic Neighbor Embedding (SNE)
  - ✓ It is more important to get local distances right than non-local ones
  - ✓ SNE has a probabilistic way of deciding if a pairwise distance is local
  - ✓ Convert each high-dimensional similarity into the probability that one data point will pick the other data point as its neighbor
    - Probability of picking j given in high D
    - Probability of picking j given in low D

$$p_{j|i} = \frac{e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma_i^2}}}{\sum_k e^{-\frac{||\mathbf{x}_i - \mathbf{x}_k||^2}{2\sigma_i^2}}} \qquad q_{j|i} = \frac{e^{-||\mathbf{y}_i - \mathbf{y}_j||^2}}{\sum_k e^{-||\mathbf{y}_i - \mathbf{y}_k||^2}}$$

Stochastic Neighbor Embedding

- Picking the Radius of the Gaussian in p
  - ✓ We need to use different radii in different parts of the space so that we keep the effective number of neighbors about constant
  - ✓ A big radius leads to a high entropy for the distribution over neighbors of i, whereas a small radius leads to a low entropy
  - ✓ Decide what entropy you want and then find the radius that produces that entropy

$$Perplexity(P_i) = 2^{H(P_i)}$$

$$H(P_i) = \sum_j p_{j|i} log_2 p_{j|i}$$

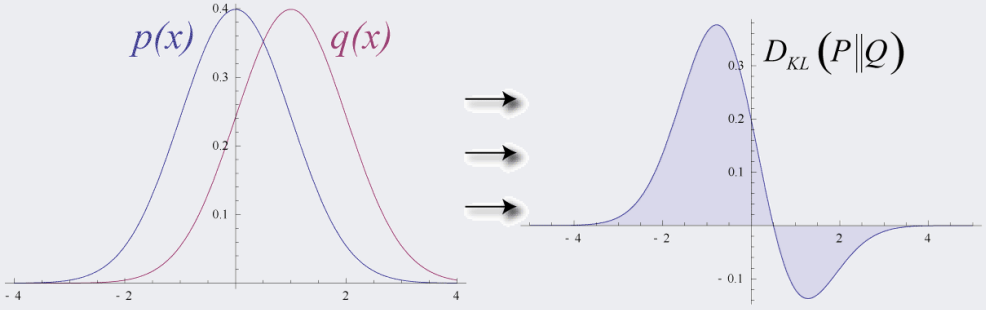
$$p_{j|i} = \frac{e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma_i^2}}}{\sum_k e^{-\frac{||\mathbf{x}_i - \mathbf{x}_k||^2}{2\sigma_i^2}}}$$

  - ✓ The performance of SNE is fairly robust to changes in the perplexity (5~50)

Stochastic Neighbor Embedding

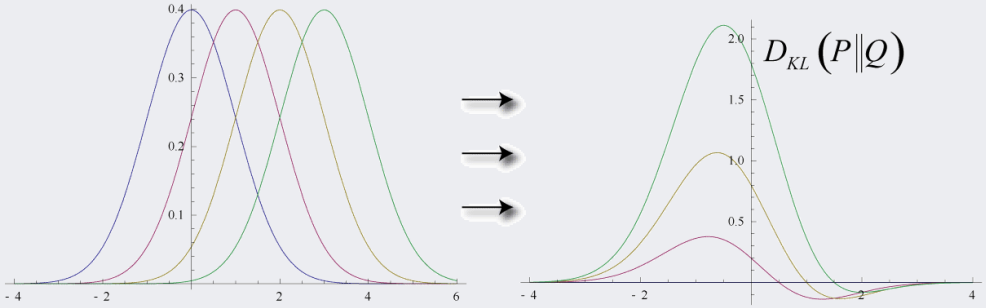
- Cost Function for a Low-dimensional Representation
  - ✓ Kullback-Leibler divergence
    - A non-symmetric measure of the difference between two probability distribution P and Q

$$Cost = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} log \frac{p_{j|i}}{q_{j|i}}$$



Original Gaussian PDF's

KL Area to be Integrated



Stochastic Neighbor Embedding

- Cost Function for a Low-dimensional Representation
  - ✓ Kullback-Leibler divergence
    - A non-symmetric measure of the difference between two probability distribution P and Q

$$Cost = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} log \frac{p_{j|i}}{q_{j|i}}$$

  - ✓ Gradient

$$\frac{\partial C}{\partial \mathbf{y}_i} = 2 \sum_j (\mathbf{y}_j - \mathbf{y}_i)(p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})$$

  - ✓ Update the coordinate in the lower dimension to minimize the Cost function

Symmetric SNE

- Turning conditional probabilities into pairwise probabilities

$$p_{ij} = \frac{e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma_i^2}}}{\sum_{k \neq l} e^{-\frac{||\mathbf{x}_k - \mathbf{x}_l||^2}{2\sigma_i^2}}} \quad \blacktriangleright \quad p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n} \qquad \sum_j p_{ij} > \frac{1}{2n}$$

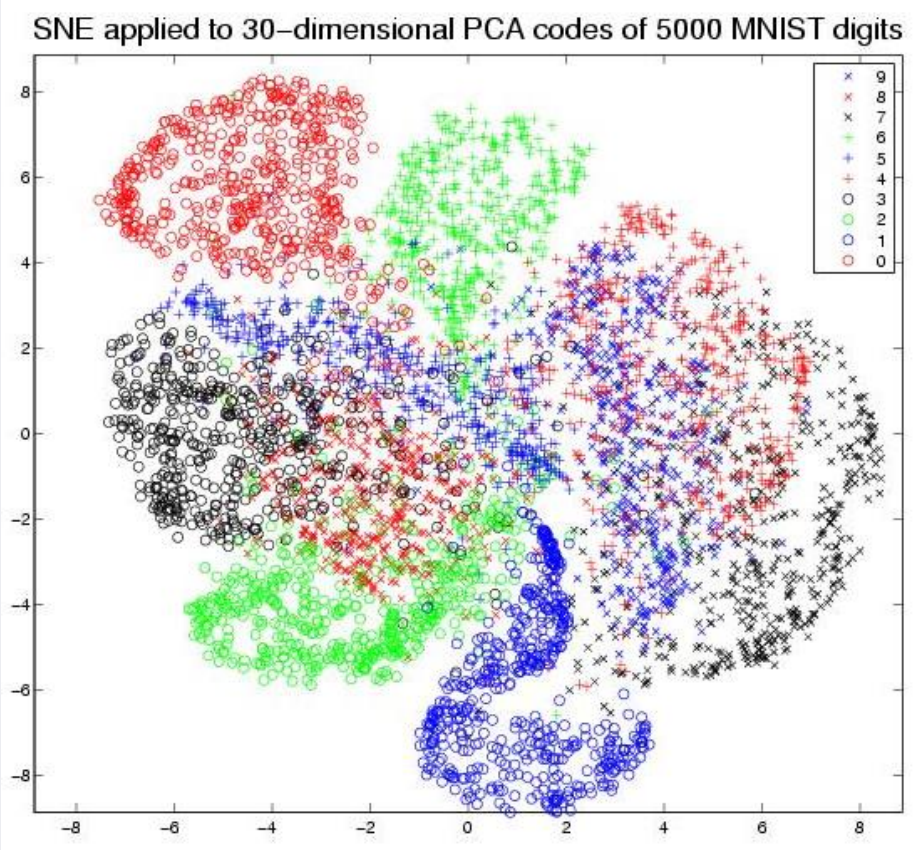
✓ Cost function and gradient

$$Cost = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}}$$

$$\frac{\partial C}{\partial \mathbf{y}_i} = 4 \sum_j (\mathbf{y}_j - \mathbf{y}_i)(p_{ij} - q_{ij})$$

Symmetric SNE

- Crowding problem
  - ✓ The area accommodating moderately distant data points is not large enough compared with the area accommodating nearby data points



t-SNE

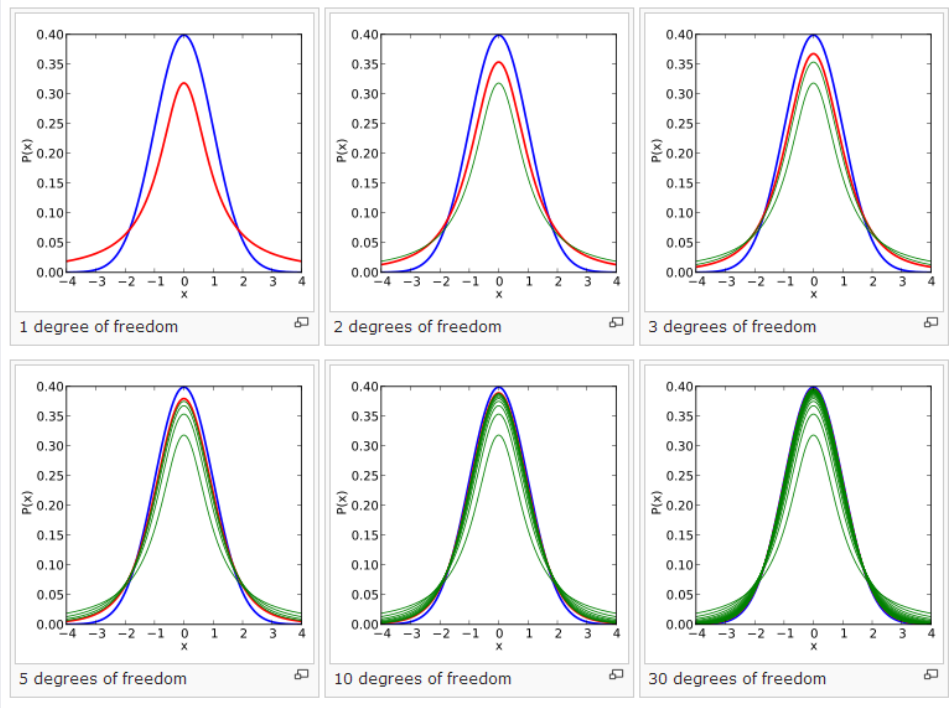
Maaten and Hinton (2008)

- Resolution to the Crowding Problem
  - ✓ Use a probability distribution that has much heavier tails than a Gaussian to convert distances into probabilities in the low-dimensional map
  - ✓ Student's t-distribution with one degree of freedom

$$f(t) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

$$\Gamma(n) = (n-1)!$$

$$q_{j|i} = \frac{(1+||\mathbf{y}_i - \mathbf{y}_j||^2)^{-1}}{\sum_{k\neq l}(1+||\mathbf{y}_k - \mathbf{y}_l||^2)^{-1}}$$



t-SNE

- Optimization of t-SNE

$$p_{j|i} = \frac{e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma_i^2}}}{\sum_k e^{-\frac{||\mathbf{x}_i - \mathbf{x}_k||^2}{2\sigma_i^2}}} \qquad q_{j|i} = \frac{(1 + ||\mathbf{y}_i - \mathbf{y}_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||\mathbf{y}_k - \mathbf{y}_l||^2)^{-1}}$$

✓ Gradient:

$$\frac{\partial C}{\partial \mathbf{y}_i} = 4 \sum_j (\mathbf{y}_j - \mathbf{y}_i)(p_{ij} - q_{ij})(1 + ||\mathbf{y}_i - \mathbf{y}_j||^2)^{-1}$$

t-SNE

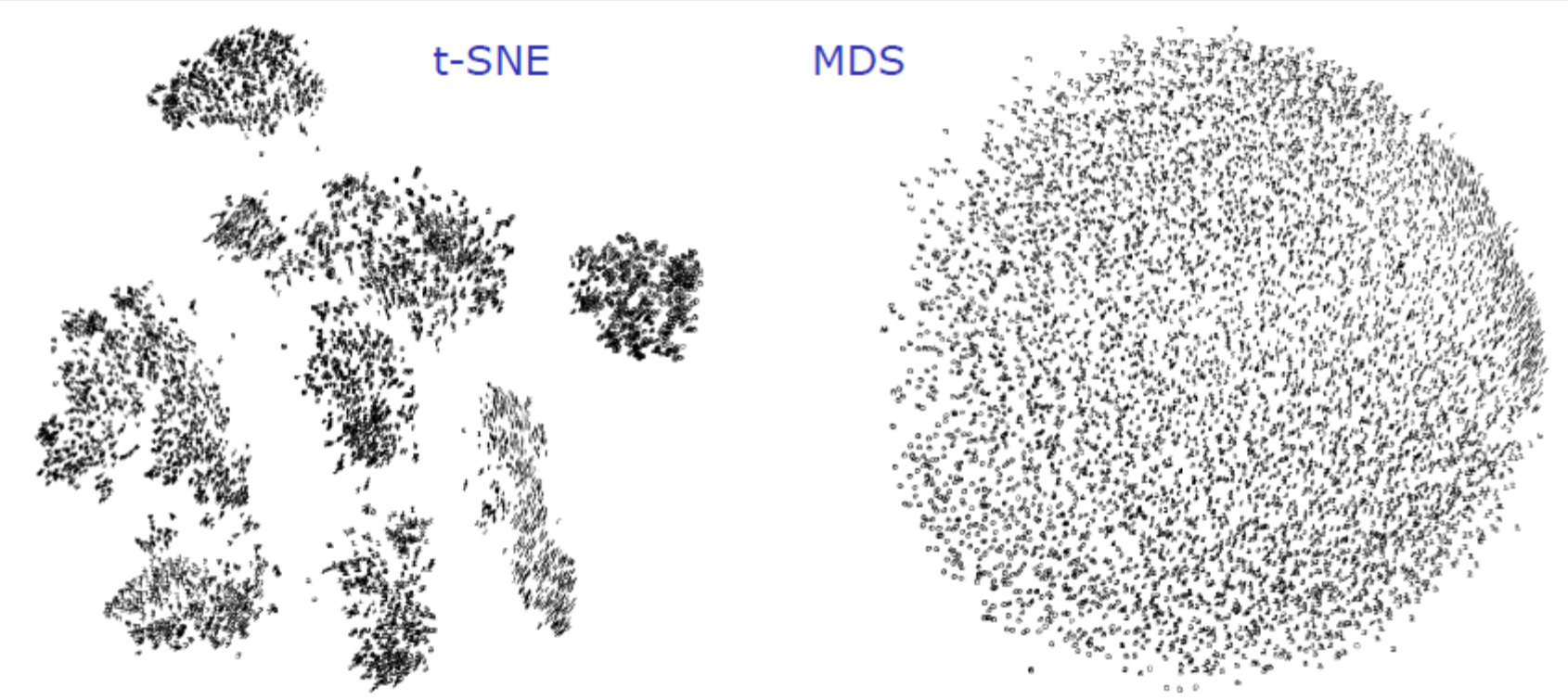
- t-SNE algorithm

Algorithm 1: Simple version of t-Distributed Stochastic Neighbor Embedding.

Data: data set $\mathcal{X} = \{x_1, x_2, ..., x_n\}$,
cost function parameters: perplexity *Perp*,
optimization parameters: number of iterations T, learning rate η, momentum $\alpha(t)$.
Result: low-dimensional data representation $\mathcal{Y}^{(T)} = \{y_1, y_2, ..., y_n\}$.
begin
  compute pairwise affinities $p_{j|i}$ with perplexity *Perp* (using Equation 1)
  set $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$
  sample initial solution $\mathcal{Y}^{(0)} = \{y_1, y_2, ..., y_n\}$ from $\mathcal{N}(0, 10^{-4}I)$
  for *t=1* **to** T **do**
    compute low-dimensional affinities q_{ij} (using Equation 4)
    compute gradient $\frac{\delta C}{\delta \mathcal{Y}}$ (using Equation 5)
    set $\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta \frac{\delta C}{\delta \mathcal{Y}} + \alpha(t)\left(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)}\right)$
  end
end

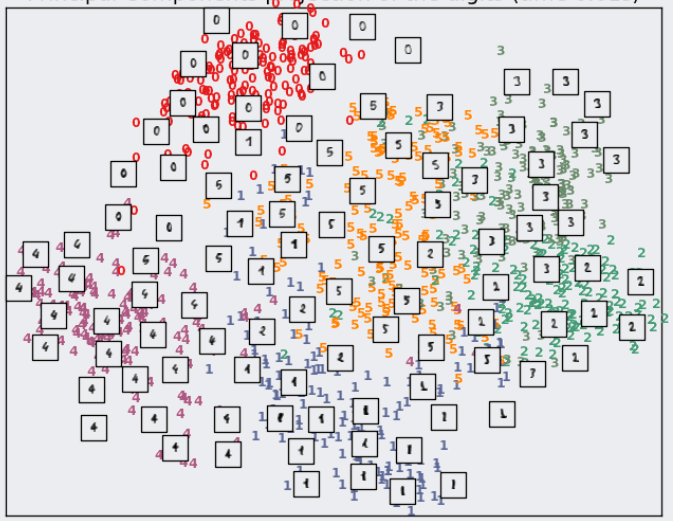
t-SNE vs. MDS

- MNIST dataset

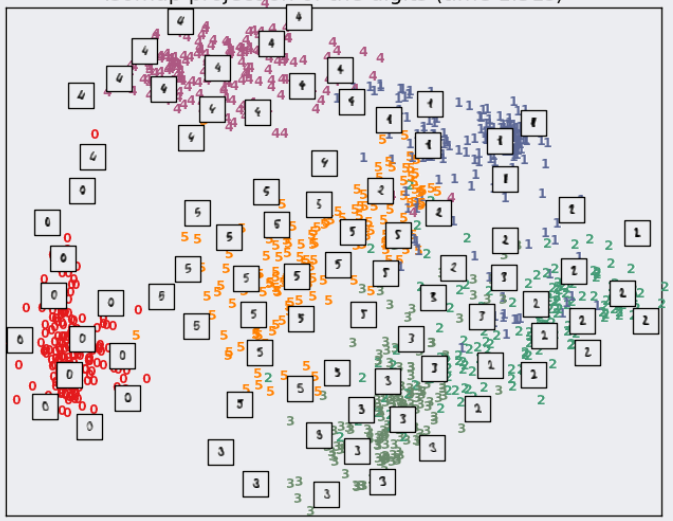


t-SNE Examples

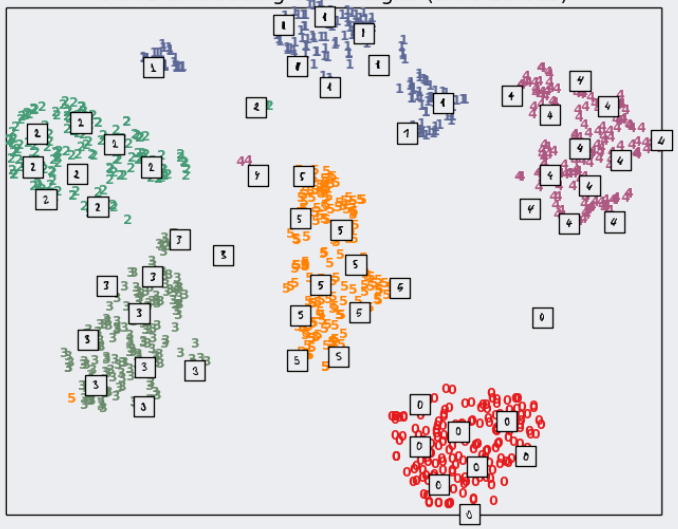
Principal Components projection of the digits (time 0.01s)



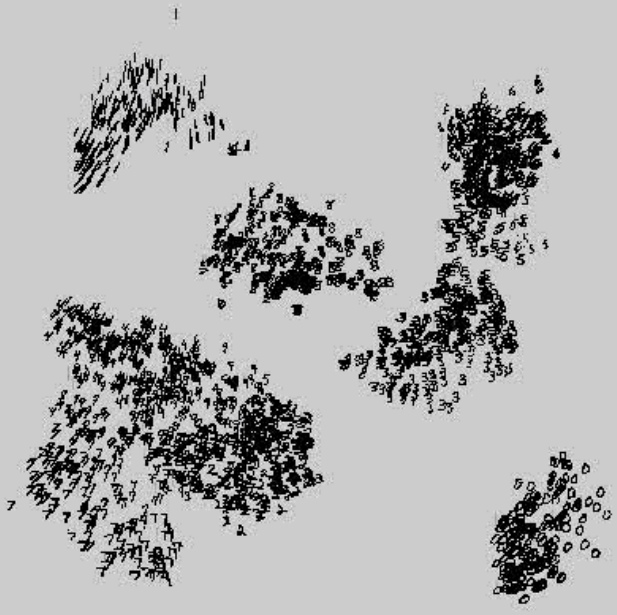
Isomap projection of the digits (time 1.51s)



t-SNE embedding of the digits (time 15.61s)

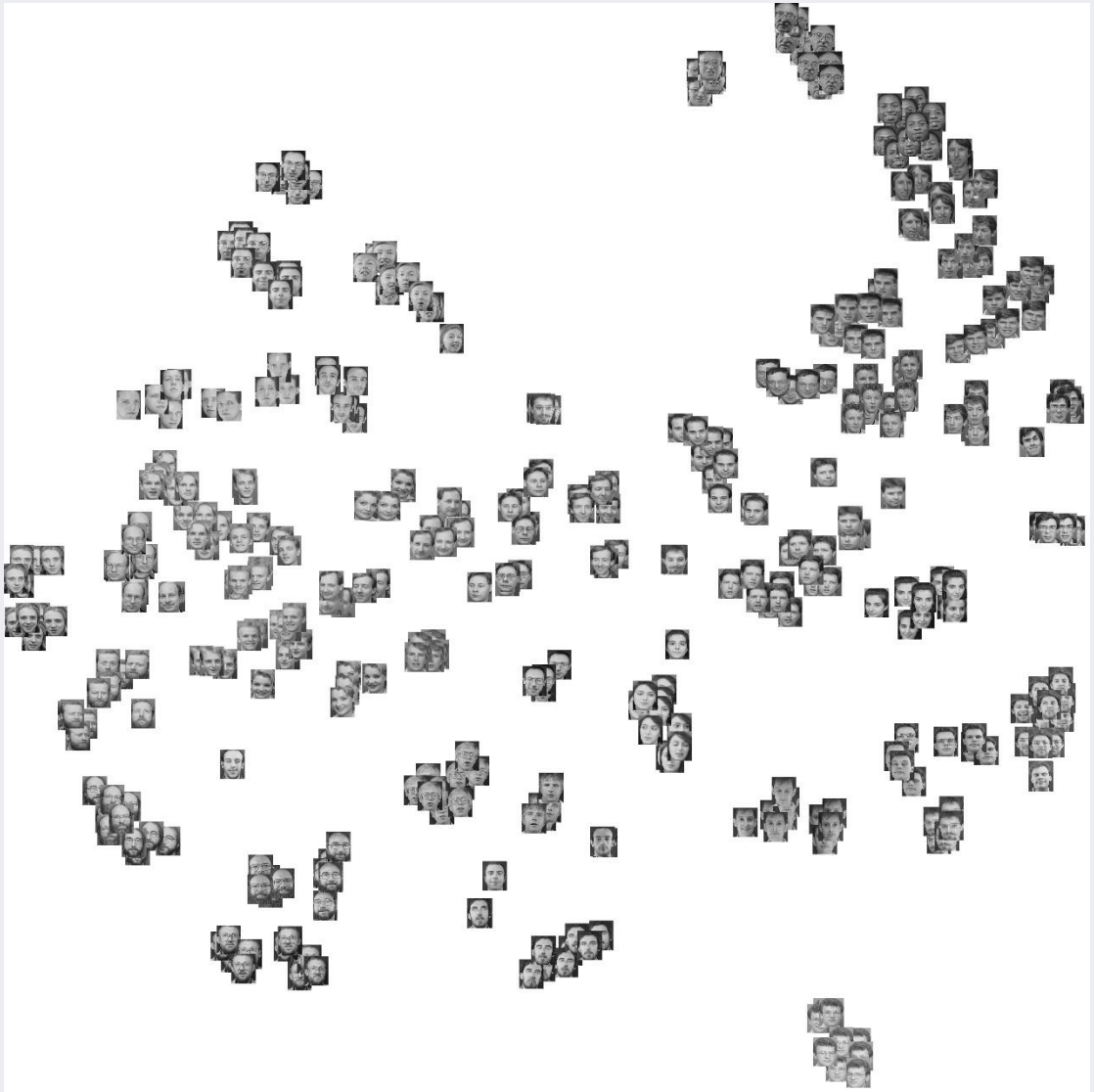


t-SNE Examples



t-SNE Examples

- Olivetti faces datasets



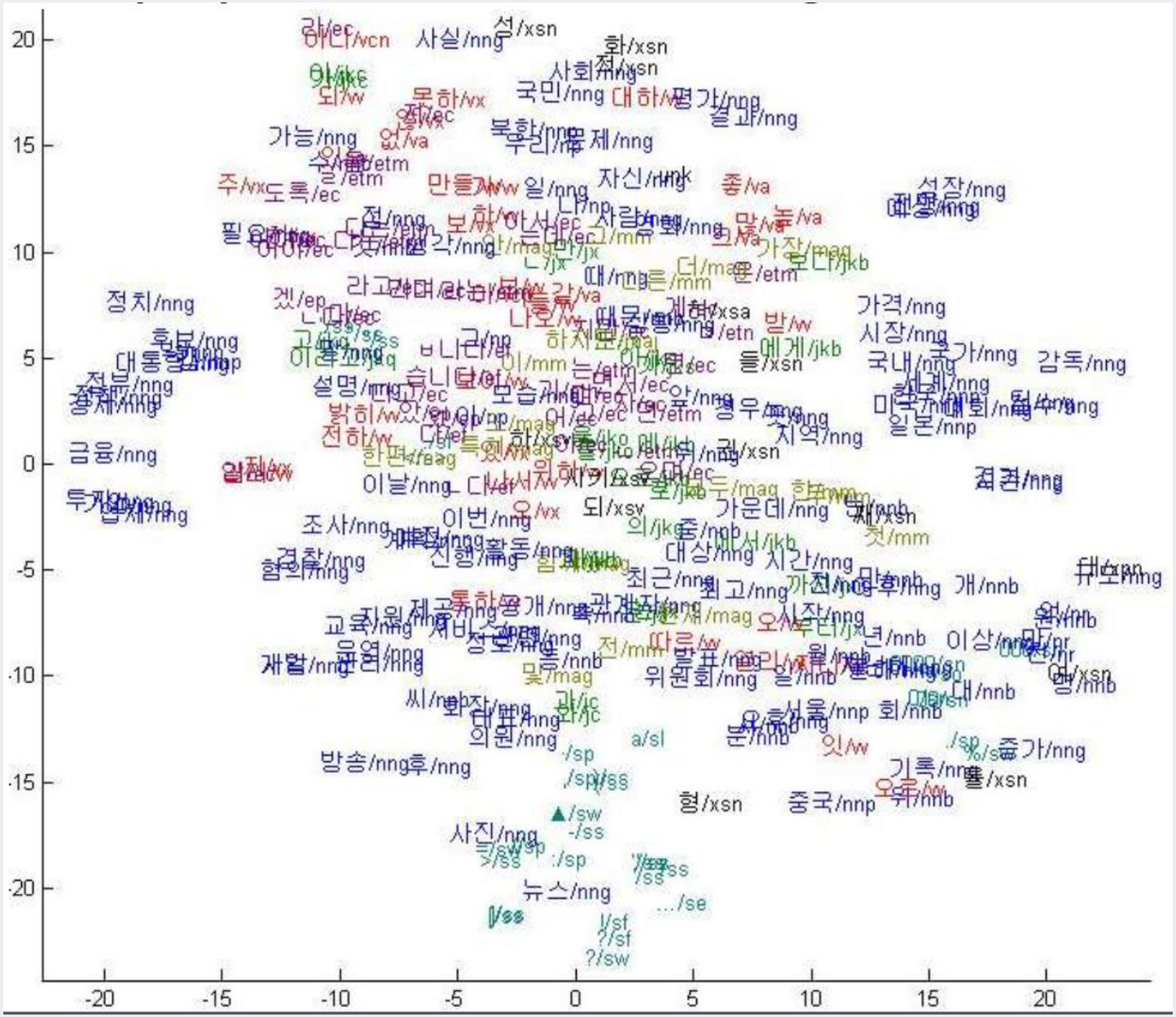
t-SNE Examples

- Netflix dataset

t-SNE Examples

- CalTech-101

t-SNE Examples



AGENDA

01 Dimensionality Reduction

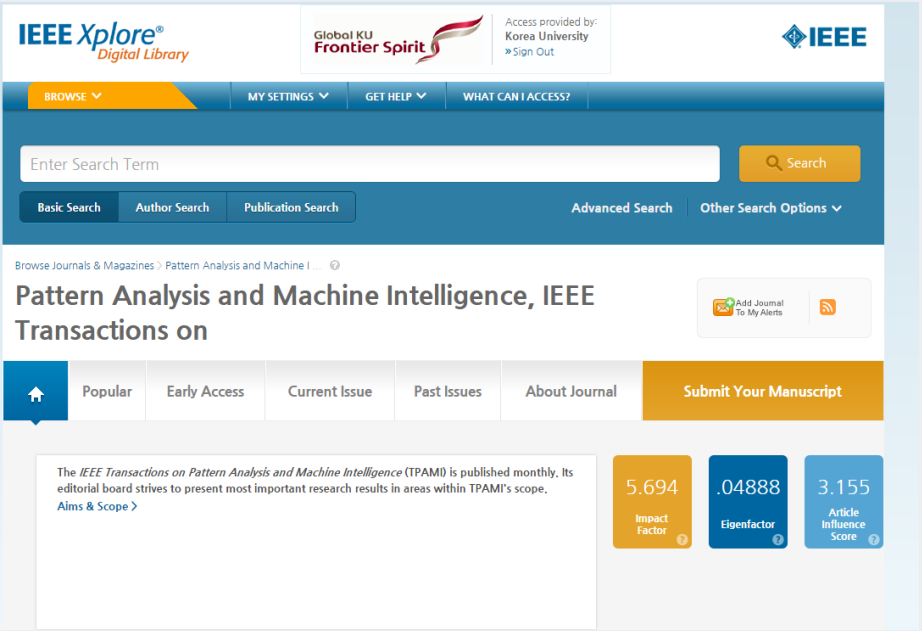
02 Feature Selection

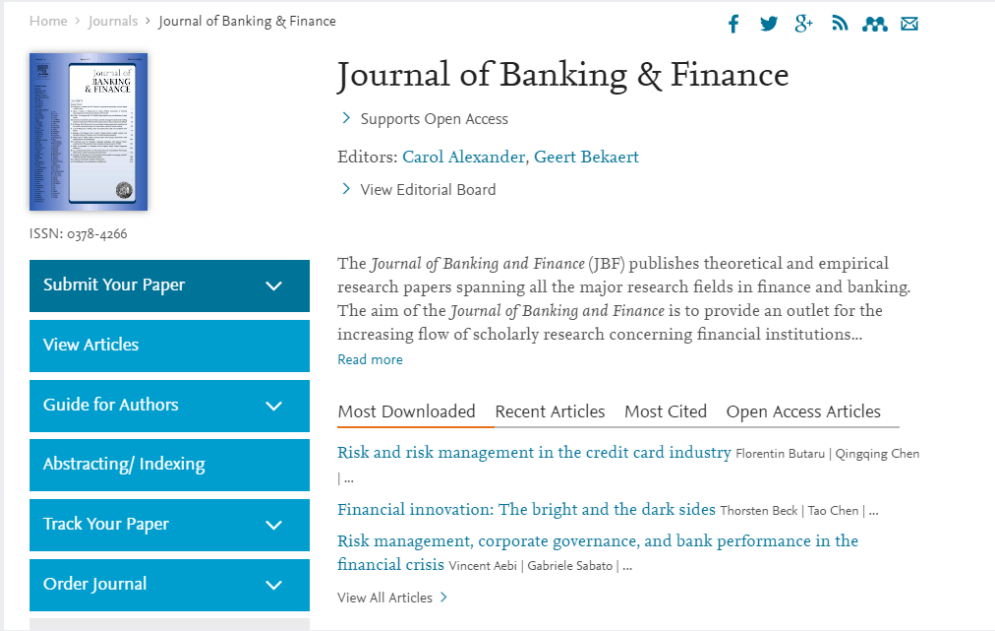
03 Feature Extraction: LSA & t-SNE

04 R Exercise

Feature Selection Metrics

- Dataset: Abstracts of the papers published since 2015
  - ✓ Journal 1: IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)
  - ✓ Journal 2: Journal of Banking and Finance (JoF)





Feature Selection Metrics

- Construct Corpus

```
install.packages("tm")
install.packages("FSelector")
install.packages("stringi")

library(tm)
library(FSelector)
library(stringi)

# Load the data
TPAMI <- read.csv("IEEE_TPAMI_2015_2017.csv", encoding = "UTF-8", stringsAsFactors = FALSE)
JoF <- read.csv("Journal of Banking and Finance_2015_2017.csv", encoding = "UTF-8", stringsAsFactors = FALSE)

Journal.Data <- rbind(TPAMI[,c(4,12)], JoF[,c(4,12)])
names(Journal.Data) <- c("Journal", "Abstract")

# Construct the corpus for each journal with the abstracts
Journal.Corpus <- Corpus(VectorSource(Journal.Data$Abstract))
```

Data	
JoF	521 obs. of 13 variables
Journal.Data	954 obs. of 2 variables
TPAMI	433 obs. of 13 variables

Feature Selection Metrics

- Preprocessing
  - ✓ To lower case, remove punctuations/numbers/stopwords, stemming

```
# Preprocessing
# 1: to lower case
Journal.Corpus <- tm_map(Journal.Corpus, content_transformer(stri_trans_tolower))

# 2: remove puntuations
Journal.Corpus <- tm_map(Journal.Corpus, content_transformer(removePunctuation))

# 3. remove numbers
Journal.Corpus <- tm_map(Journal.Corpus, content_transformer(removeNumbers))

# 4. remove stopwords (SMART stopwords list)
myStopwords <- c(stopwords("SMART"))

Journal.Corpus <- tm_map(Journal.Corpus, removeWords, myStopwords)

# 5. Stemming
Journal.Corpus <- tm_map(Journal.Corpus, stemDocument)
```

Feature Selection Metrics

- Construct Term-Document Matrices
  - ✓ 6,131 terms and 954 documents

```
# Term-Document Matrix
Journal.TDM <- TermDocumentMatrix(Journal.Corpus, control = list(minWordLength = 1))

Journal.Frequency.TDM <- as.matrix(Journal.TDM)

pos_idx <- which(Journal.Frequency.TDM >= 1)
Journal.Binary.TDM <- Journal.Frequency.TDM
Journal.Binary.TDM[pos_idx] <- 1
Terms <- rownames(Journal.Binary.TDM)

# Remove multibyte strings
Terms <- iconv(Terms, "latin1", "ASCII", sub="")
rownames(Journal.Frequency.TDM) <- Terms
rownames(Journal.Binary.TDM) <- Terms
```

Feature Selection Metrics

- Acc

```r
# Dimensionality Reduction 1: Feature Selection --------------------------
Journal.Binary.DTM <- data.frame(t(as.matrix(Journal.Binary.TDM)))
Journal.Binary.DTM <- data.frame(Journal = Journal.Data$Journal, Journal.Binary.DTM)

# Metric 1: Acc
pos_idx <- which(Journal.Binary.DTM$Journal == "IEEE Transactions on Pattern Analysis and Machine Intelligence")
Acc <- sapply(Journal.Binary.DTM[,-1], function(x) sum(x[pos_idx])-sum(x[-pos_idx]))
FS_Acc <- data.frame(Terms, Acc)
# Sort the terms w.r.t. Acc
FS_Acc <- FS_Acc[order(FS_Acc$Acc, decreasing = TRUE),]
FS_Acc[1:30,]
FS_Acc[(length(Terms)-30):length(Terms),]
```

```
> FS_Acc[1:30,]
                      Terms Acc
ieee                   ieee 433
propos               propos 227
imag                   imag 220
method               method 218
algorithm         algorithm 176
problem             problem 153
approach           approach 148
stateoftheart stateoftheart 148
demonstr           demonstr 140
learn                 learn 139
```

```
> FS_Acc[(length(Terms)-30):length(Terms),]
              Terms  Acc
lower         lower  -71
econom       econom  -72
signific   signific  -73
portfolio portfolio  -73
trade         trade  -76
volatil     volatil  -80
effect       effect  -81
asset         asset  -81
investor   investor  -82
relat         relat  -83
```

Feature Selection Metrics

- F1-Measures

```r
# Metric 2: F1-Measure
F1 <- sapply(Journal.Binary.DTM[,-1], function(x) 2*sum(x[pos_idx])/(length(pos_idx)+sum(x)))
FS_F1 <- data.frame(Terms, F1)
# Sort the terms w.r.t. F1
FS_F1 <- FS_F1[order(FS_F1$F1, decreasing = TRUE),]
FS_F1[1:30,]
FS_F1[(length(Terms)-30):length(Terms),]
```

```
> FS_F1[1:30,]
                      Terms        F1
ieee                   ieee 1.0000000
propos               propos 0.7352185
method               method 0.6984572
imag                   imag 0.6748092
approach           approach 0.5827233
algorithm         algorithm 0.5821138
problem             problem 0.5692308
model                 model 0.5383707
demonstr           demonstr 0.5281804
stateoftheart stateoftheart 0.5094664
```

```
> FS_F1[(length(Terms)-30):length(Terms),]
                      Terms F1
bertrandtyp     bertrandtyp  0
dissip               dissip  0
leas                   leas  0
domicil             domicil  0
abovement         abovement  0
cartel               cartel  0
deduc                 deduc  0
fsa                     fsa  0
malfunct           malfunct  0
threeregim       threeregim  0
```

Feature Selection Metrics

- χ^2

```
# Metric 3: Chi-squared
CS <- chi.squared(Journal ~ ., Journal.Binary.DTM)
FS_CS <- data.frame(Terms, CS)
FS_CS <- FS_CS[order(FS_CS$attr_importance, decreasing = TRUE),]
FS_CS[1:30,]
FS_CS[(length(Terms)-30):length(Terms),]
```

```
> FS_CS[1:30,]
                      Terms attr_importance
ieee                   ieee       1.0000000
elsevi                elsevi       0.9173953
imag                    imag       0.5990994
propos                propos       0.5670766
method                method       0.5636770
algorithm          algorithm       0.5165255
market                market       0.5035224
find                    find       0.4947054
stateoftheart stateoftheart       0.4700437
financi              financi       0.4635734
```

```
> FS_CS[(length(Terms)-30):length(Terms),]
                      Terms attr_importance
bertrandtyp     bertrandtyp                0
dissip               dissip                0
leas                   leas                0
domicil             domicil                0
abovement         abovement                0
cartel               cartel                0
deduc                 deduc                0
fsa                     fsa                0
malfunct           malfunct                0
threeregim       threeregim                0
```

Feature Selection Metrics

- Information Gain

```
# Metric 4: Information Gain
IG <- information.gain(Journal ~ ., Journal.Binary.DTM)
FS_IG <- data.frame(Terms, IG)
FS_IG <- FS_IG[order(FS_IG$attr_importance, decreasing = TRUE),]
FS_IG[1:30,]
FS_IG[(length(Terms)-30):length(Terms),]
```

```
> FS_IG[1:30,]
                     Terms attr_importance
ieee                  ieee      0.68888672
elsevi              elsevi      0.54263266
imag                  imag      0.22041242
method              method      0.17258485
market              market      0.17090670
propos              propos      0.17064017
algorithm        algorithm      0.16023126
financi            financi      0.14623957
stateoftheart stateoftheart      0.14002477
find                  find      0.13363626
```

```
> FS_IG[(length(Terms)-30):length(Terms),]
                     Terms attr_importance
bertrandtyp     bertrandtyp              0
dissip               dissip              0
leas                   leas              0
domicil             domicil              0
abovement         abovement              0
cartel               cartel              0
deduc                 deduc              0
fsa                     fsa              0
malfunct           malfunct              0
threeregim       threeregim              0
```

Feature Selection Metrics

- Top 30 significant words identified by each selection method

Terms_Acc	Acc_Score	Terms_F1	F1_Score	Terms_Chi-Squared	Chi-Squared Score	Terms_Information Gain	Information Gain_Score
ieee	433	ieee	1	ieee	1	ieee	0.688886724
propos	227	propos	0.735218509	elsevi	0.91739535	elsevi	0.542632657
imag	220	method	0.698457223	imag	0.59909938	imag	0.22041242
method	218	imag	0.67480916	propos	0.567076632	method	0.17258485
algorithm	176	approach	0.58272328	method	0.563677041	market	0.170906699
problem	153	algorithm	0.582113821	algorithm	0.51652555	propos	0.170640171
approach	148	problem	0.569230769	market	0.503522439	algorithm	0.160231258
stateoftheart	148	model	0.53837072	find	0.494705401	financi	0.146239569
demonstr	140	demonstr	0.528180354	stateoftheart	0.470043715	stateoftheart	0.140024765
learn	139	stateoftheart	0.509466437	financi	0.463573412	find	0.133636257
comput	132	paper	0.507109005	risk	0.452834266	risk	0.128948573
dataset	123	show	0.504975124	learn	0.442924111	learn	0.115063533
object	120	data	0.503419973	problem	0.434490377	bank	0.105610361
present	110	result	0.500568828	comput	0.420012101	problem	0.100333224
applic	107	perform	0.496493689	demonstr	0.416395618	comput	0.099562072
base	101	learn	0.494845361	approach	0.409496058	demonstr	0.093185842
featur	101	base	0.493902439	bank	0.397382326	price	0.092881688
task	100	comput	0.485470085	object	0.393646731	firm	0.088201631
challeng	99	dataset	0.479865772	dataset	0.386068845	approach	0.08708702
experiment	97	present	0.458961474	task	0.370180389	object	0.086585265
work	90	object	0.455652174	price	0.365322266	task	0.082917404
recognit	88	featur	0.433447099	experiment	0.363765318	dataset	0.080424016
optim	86	applic	0.43006993	challeng	0.358639387	experiment	0.080022259
local	84	set	0.427406199	applic	0.357524012	stock	0.077431402
experi	84	optim	0.404802744	firm	0.35562738	return	0.076959178
set	82	experi	0.395147314	return	0.352433844	challeng	0.073844599
formul	80	framework	0.393939394	present	0.349119571	recognit	0.071450907
train	80	challeng	0.386029412	recognit	0.344091548	applic	0.069399643
detect	78	task	0.379888268	stock	0.339457031	period	0.066847821
represent	77	effici	0.37883959	featur	0.330231693	formul	0.066727585

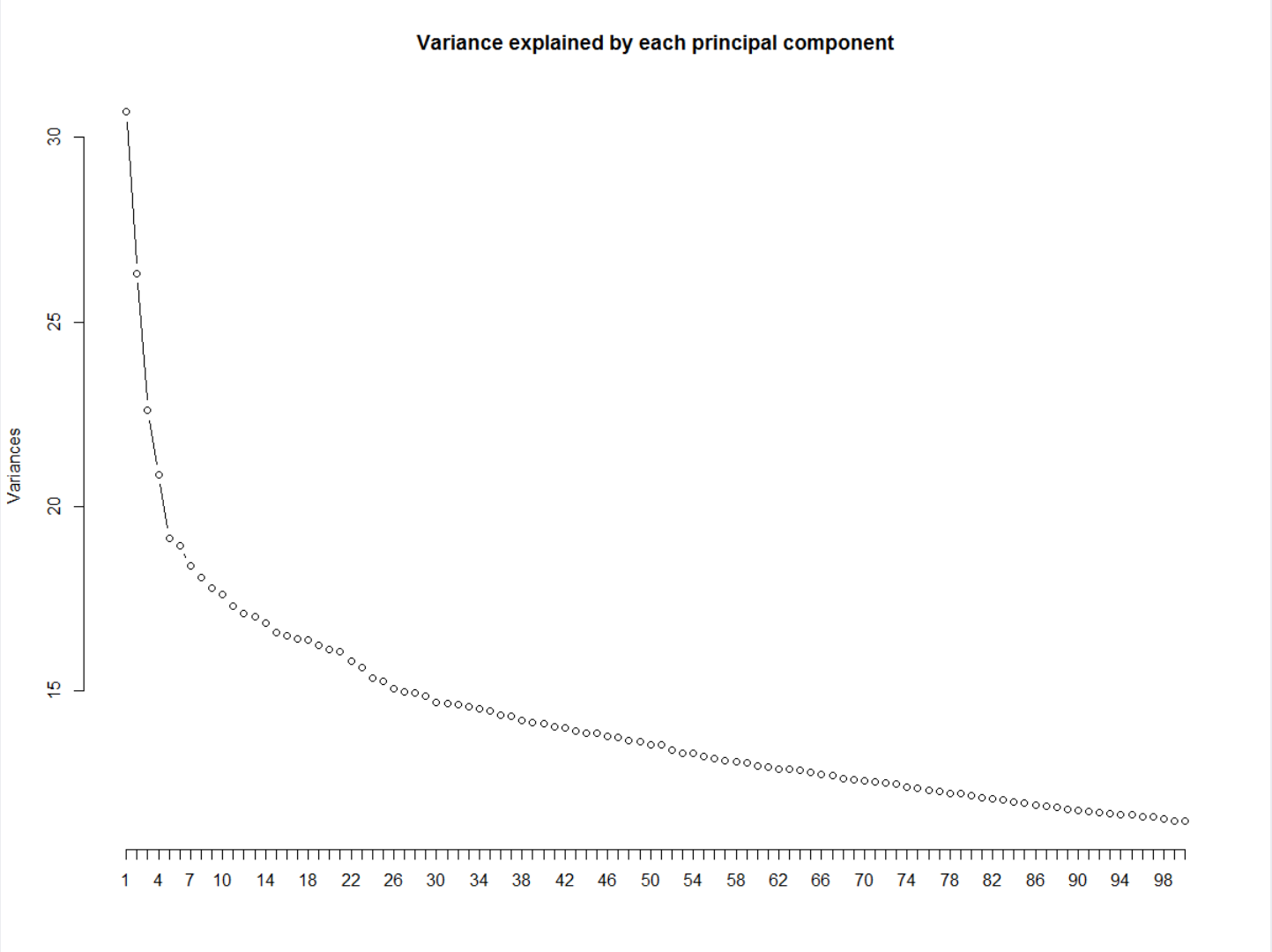
Feature Extraction: PCA

- Feature extraction by PCA

```
# Dimensionality Reduction 2-1: PCA ------------------------------------
Journal.Frequency.DTM <- data.frame(t(as.matrix(Journal.Frequency.TDM)))
PCs <- prcomp(Journal.Frequency.DTM, scale = TRUE)
summary(PCs)
screeplot(PCs, npcs = 100, type = "lines", main = "Variance explained by each principal component")

Journal.PCs <- predict(PCs)

# Plot the articles in 2-dim
par(mfrow = c(1,2))
plot(Journal.PCs[,1:2], type = "n", xlim = c(-10,10), ylim = c(-5,5))
text(Journal.PCs[1:30,1], Journal.PCs[1:10,2], label = paste("TPAMI_", 1:30, sep = ""), col=4)
text(Journal.PCs[434:463,1], Journal.PCs[11:20,2], label = paste("JoF_", 1:30, sep = ""), col=10)

plot(Journal.PCs[,3:4], type = "n", xlim = c(-5,5), ylim = c(-2,5))
text(Journal.PCs[1:30,3], Journal.PCs[1:10,4], label = paste("TPAMI_", 1:30, sep = ""), col=4)
text(Journal.PCs[434:463,3], Journal.PCs[11:20,4], label = paste("JoF_", 1:30, sep = ""), col=10)
```

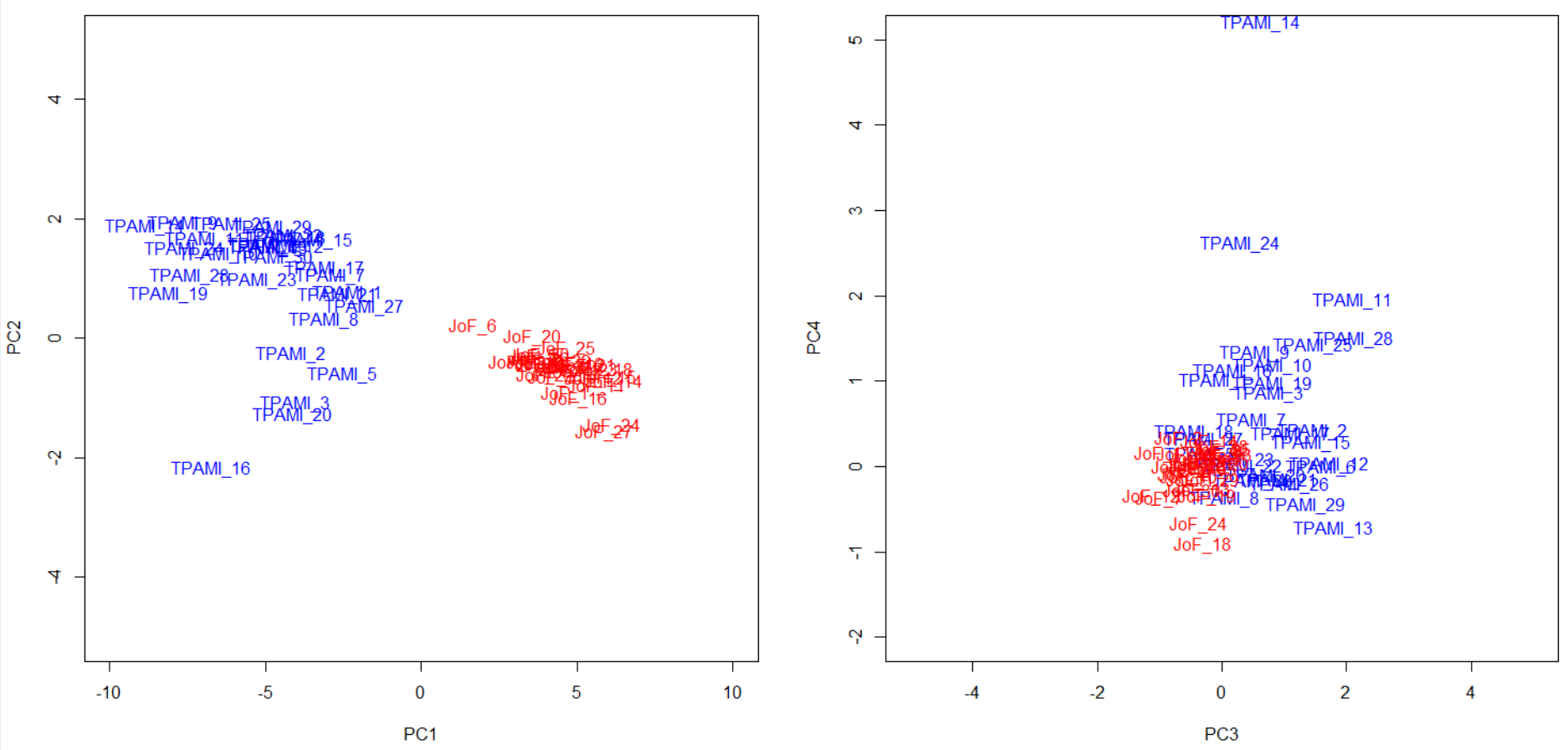
Feature Extraction: PCA

- Feature extraction by PCA



Feature Extraction: PCA

- Scatter plots in PC1 vs. PC2 and PC3 vs. PC4



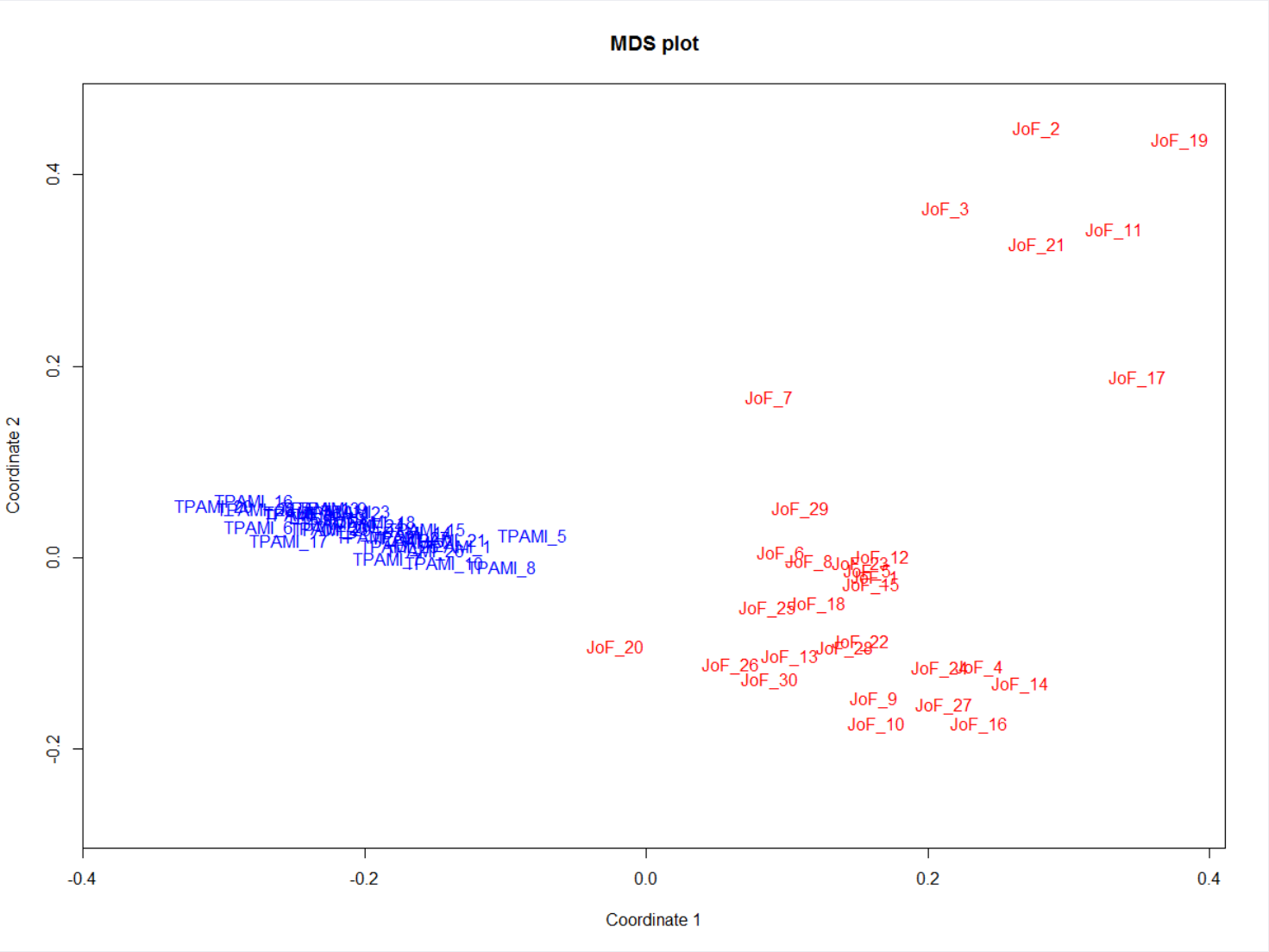
Feature Extraction: MDS

- Feature extraction by MDS

```
# Dimensionality Reduction 2-2: MDS ---------------------------------------
Journal.Cor <- cor(Journal.Frequency.TDM)
Journal.Dist <- 1-Journal.Cor
Journal.Dist <- as.data.frame(Journal.Dist)

MDS <- cmdscale(Journal.Dist, eig = TRUE, k = 10)
Journal.MDS <- MDS$points

par(mfrow = c(1,1))
plot(Journal.MDS[,1], Journal.MDS[,2], xlab = "Coordinate 1", ylab = "Coordinate 2",
     main = "MDS plot", type = "n")
text(Journal.MDS[1:30,1], Journal.MDS[1:30,2], label = paste("TPAMI_", 1:30, sep = ""), col=4)
text(Journal.MDS[434:463,1], Journal.MDS[434:463,2], label = paste("JoF_", 1:30, sep = ""), col=10)
```

Feature Extraction: MDS

- Scatter plot with the first and second axes



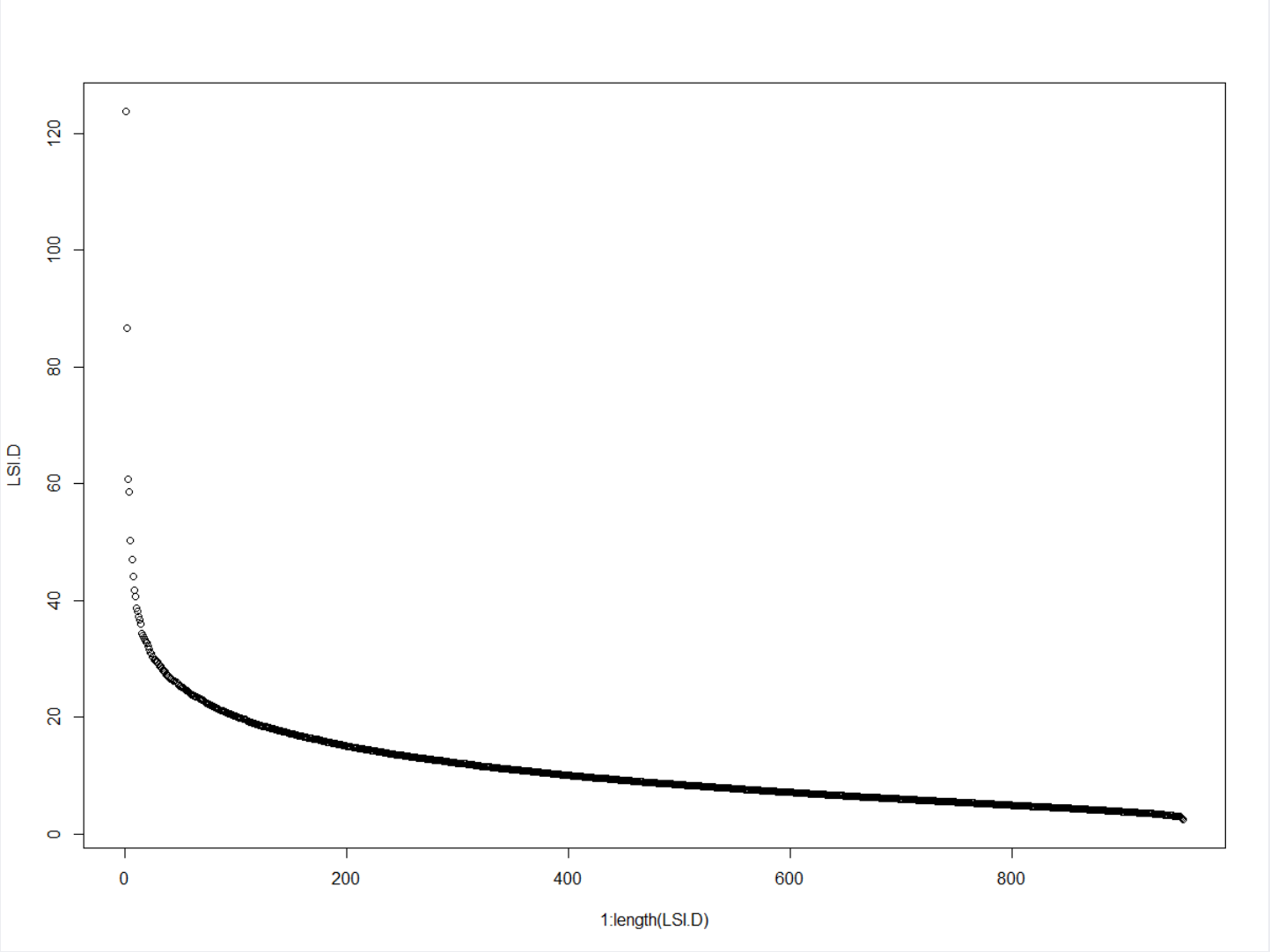
Feature Extraction: LSA

- Feature extraction by LSA

```
# Dimensionality Reduction 2-3: LSI ------------------------------------
SVD.Mat <- svd(Journal.Frequency.TDM)
LSI.D <- SVD.Mat$d
LSI.U <- SVD.Mat$u
LSI.V <- SVD.Mat$v

# Plot the singluar vectors
plot(1:length(LSI.D), LSI.D)

# Select 2 features for documents
Document.Mat <- t((diag(2)*LSI.D[1:2]) %*% t(LSI.V[,1:2]))

plot(Document.Mat[,1], Document.Mat[,2], xlab = "SVD 1", ylab = "SVD 2",
     main = "LSI plot for Documents", type = "n")
text(Document.Mat[1:30,1], Document.Mat[1:30,2], label = paste("TPAMI_", 1:30, sep = ""), col=4)
text(Document.Mat[434:463,1], Document.Mat[434:463,2], label = paste("JoF_", 1:30, sep = ""), col=10)

# Select 2 features for Terms
Term.Mat <- LSI.U[,1:2]  %*% (diag(2)*LSI.D[1:2])

plot(Term.Mat[,1], Term.Mat[,2], xlab = "SVD 1", ylab = "SVD 2", type = "n",
     main = "LSI plot for Terms")
text(Term.Mat[,1], Term.Mat[,2], label = Terms)

plot(Term.Mat[,1], Term.Mat[,2], xlab = "SVD 1", ylab = "SVD 2", type = "n",
     xlim = c(0.04,0.06), ylim = c(0.04,0.06), main = "LSI plot for Terms")
text(Term.Mat[,1], Term.Mat[,2], label = Terms)
```

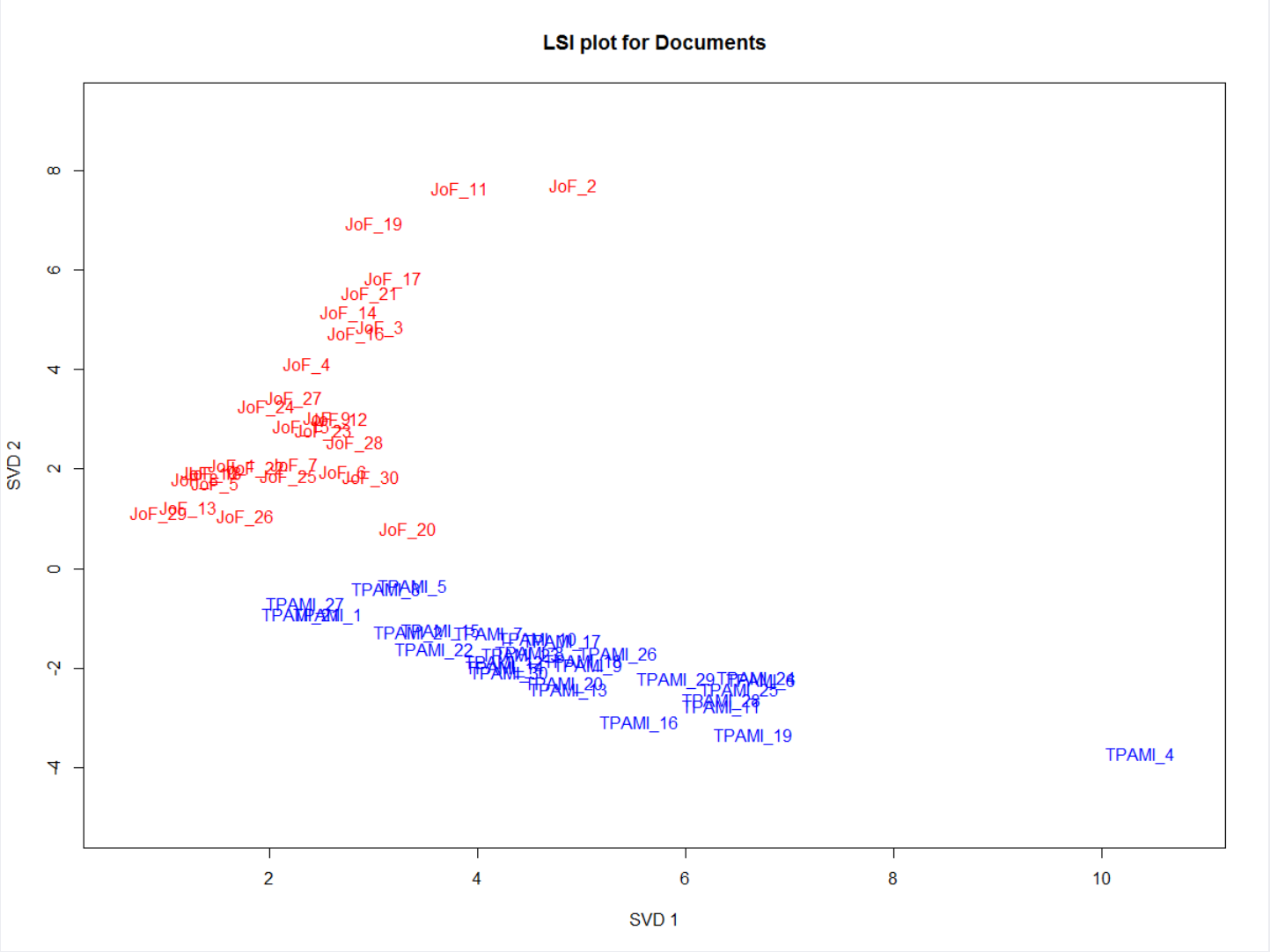
Feature Extraction: LSA

- Scatter plot for singular values



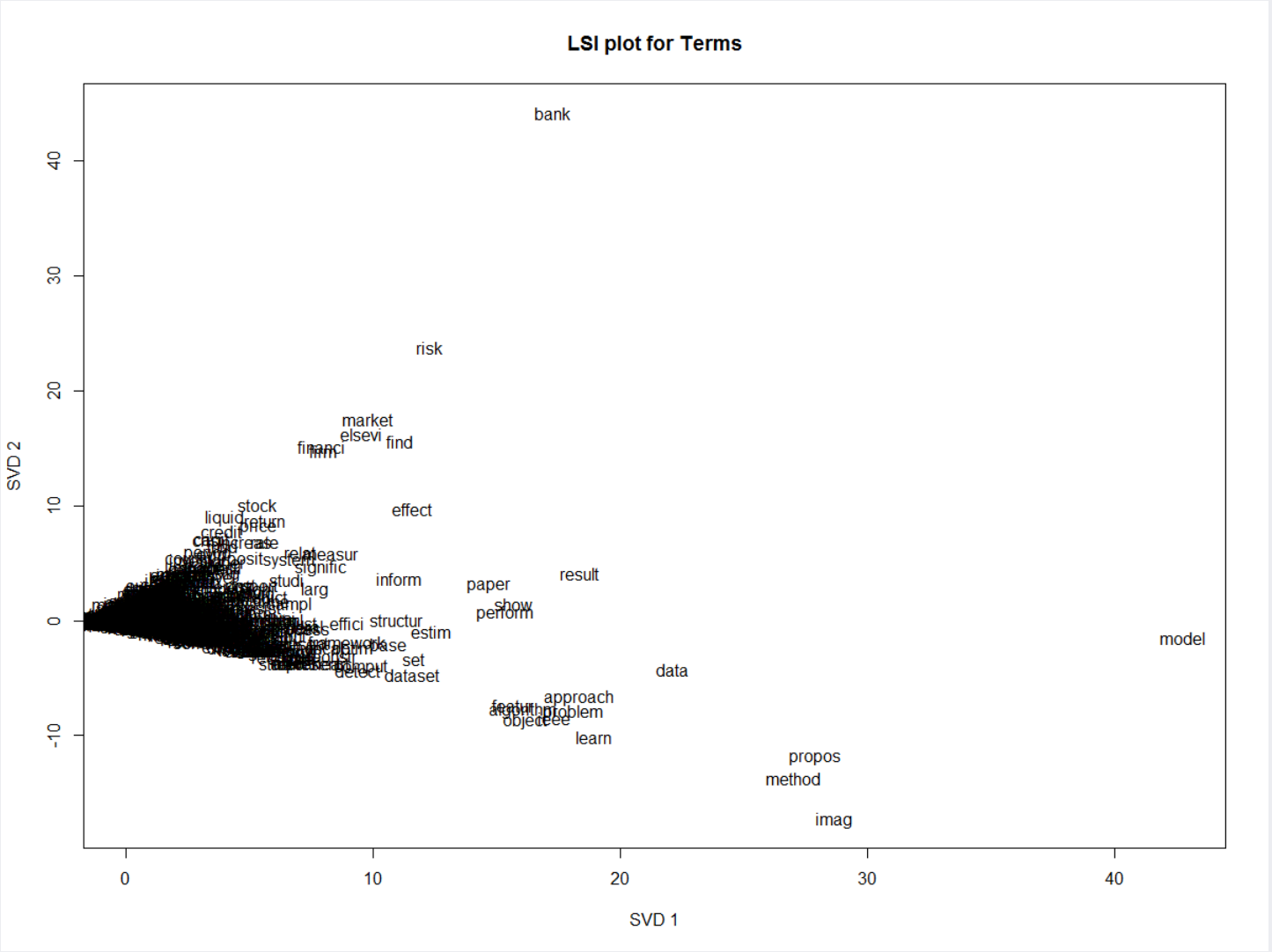
Feature Extraction: LSA

- Select 2 features for the documents



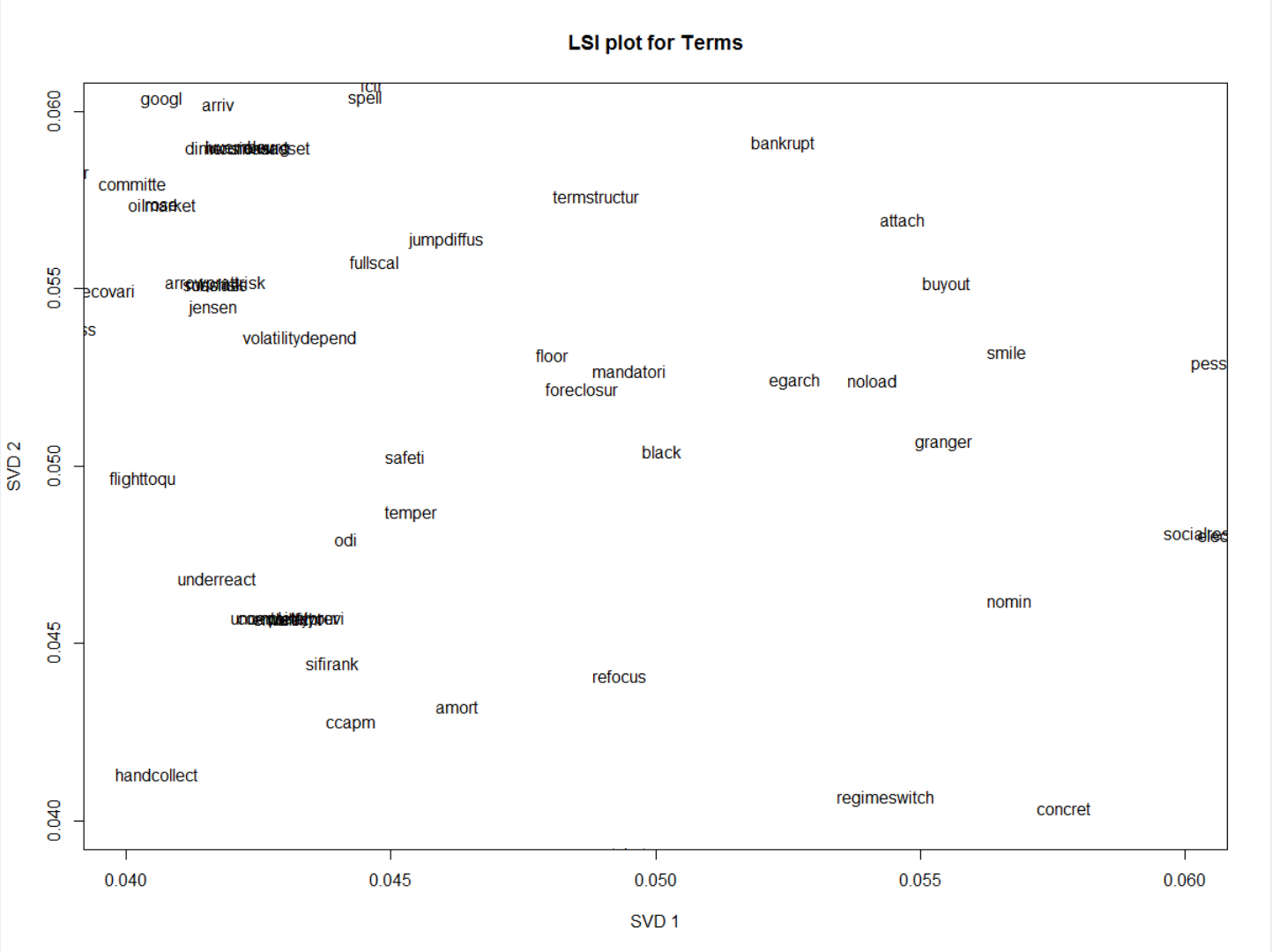
Feature Extraction: LSA

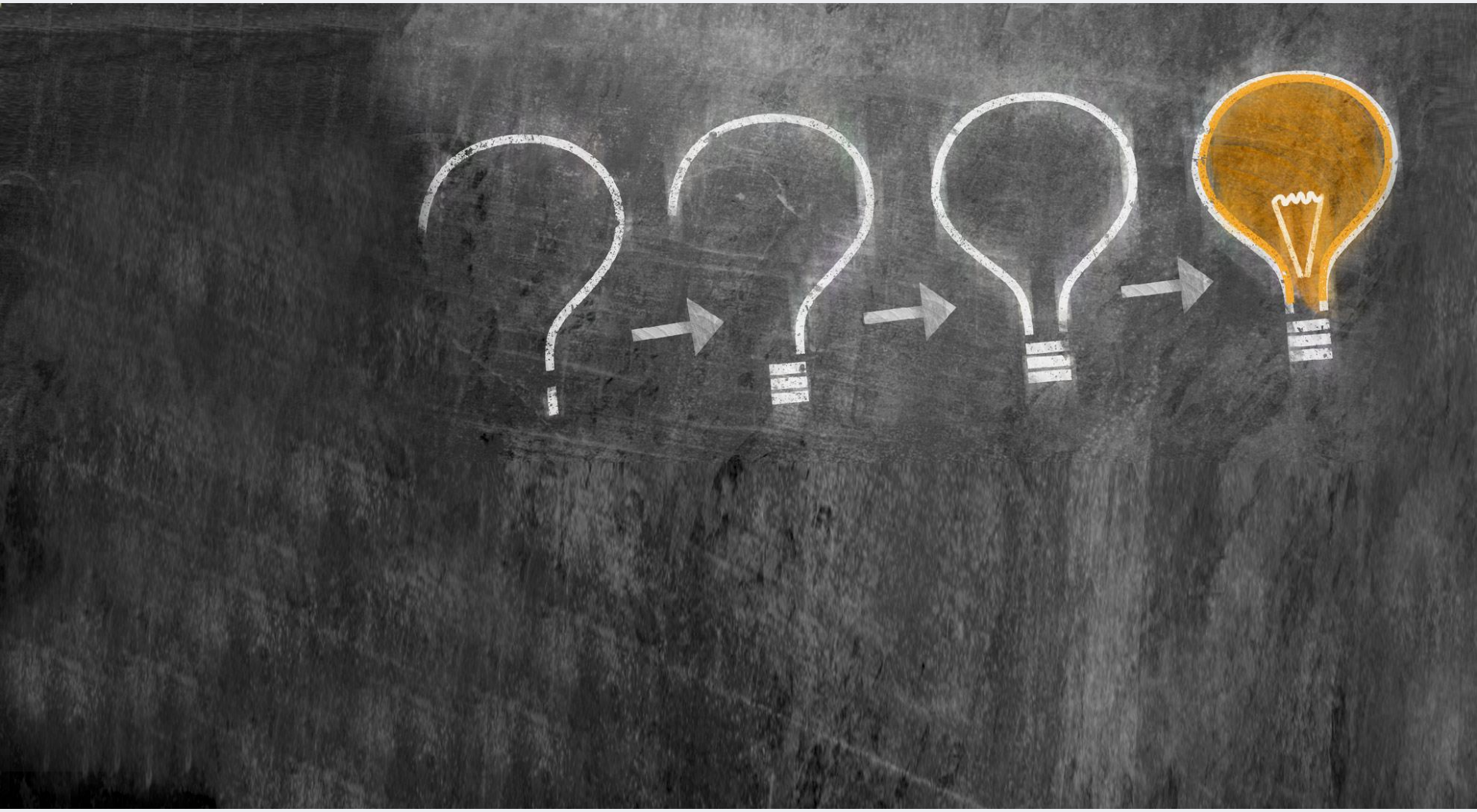
- Select 2 features for the terms



Feature Extraction: LSA

- Select 2 features for the terms





References

Research Papers

- Collobert et al. (2011). Natural Language Processing (Almost) from Scratch. Journal of Machine Learning Research 12: 2493-2537.
- Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. Journal of the American society for information science, 41(6), 391.
- Forman, G. (2003). An Extensive Empirical Study of Feature Selection Metrics for Text Classification, Journal of Machine Learning Research 3: 1289-1305.
- Hinton, G., & Roweis, S. (2002, January). Stochastic neighbor embedding. In NIPS (Vol. 15, pp. 833-840).
- Lee, C. (2015). Word and Phrase Embedding in Deep Learning. 2015 Tutorial on Natural Language Processing.
- Maaten, L. V. D., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9(Nov), 2579-2605.
- Papadimitriou et al. Latent Semantic Indexing: A Probabilistic Analysis.

References

Other Materials

- Figure in the first page: http://www.turingfinance.com/artificial-intelligence-and-statistics-principal-component-analysis-and-self-organizing-maps/
- t-SNE homepage: https://lvdmaaten.github.io/tsne/
- 이영섭. (2010). 텍스트마이닝 기법의 이해와 활용사례
- 이창기. (2015). Word and Phrase Embedding in Deep Learning. Workshop on "Deep Learning for Natural Language Processing".