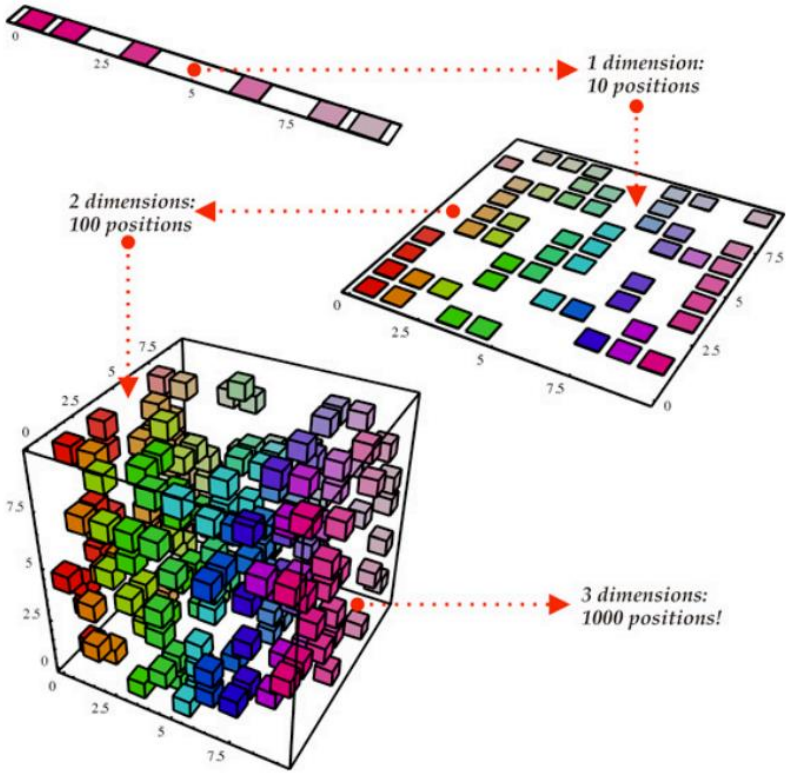


Lecture 6: Dimensionality Reduction

Pilsung Kang

School of Industrial Management Engineering

Korea University

AGENDA

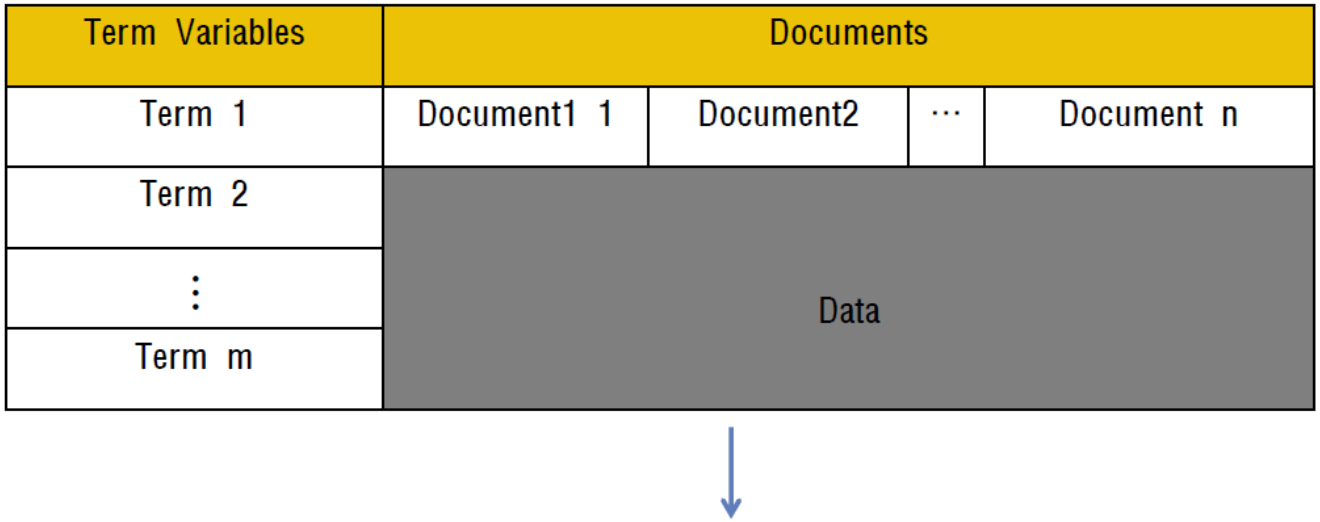
01 Dimensionality Reduction

02 Feature Selection

03 Feature Extraction: LSA & t-SNE

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)



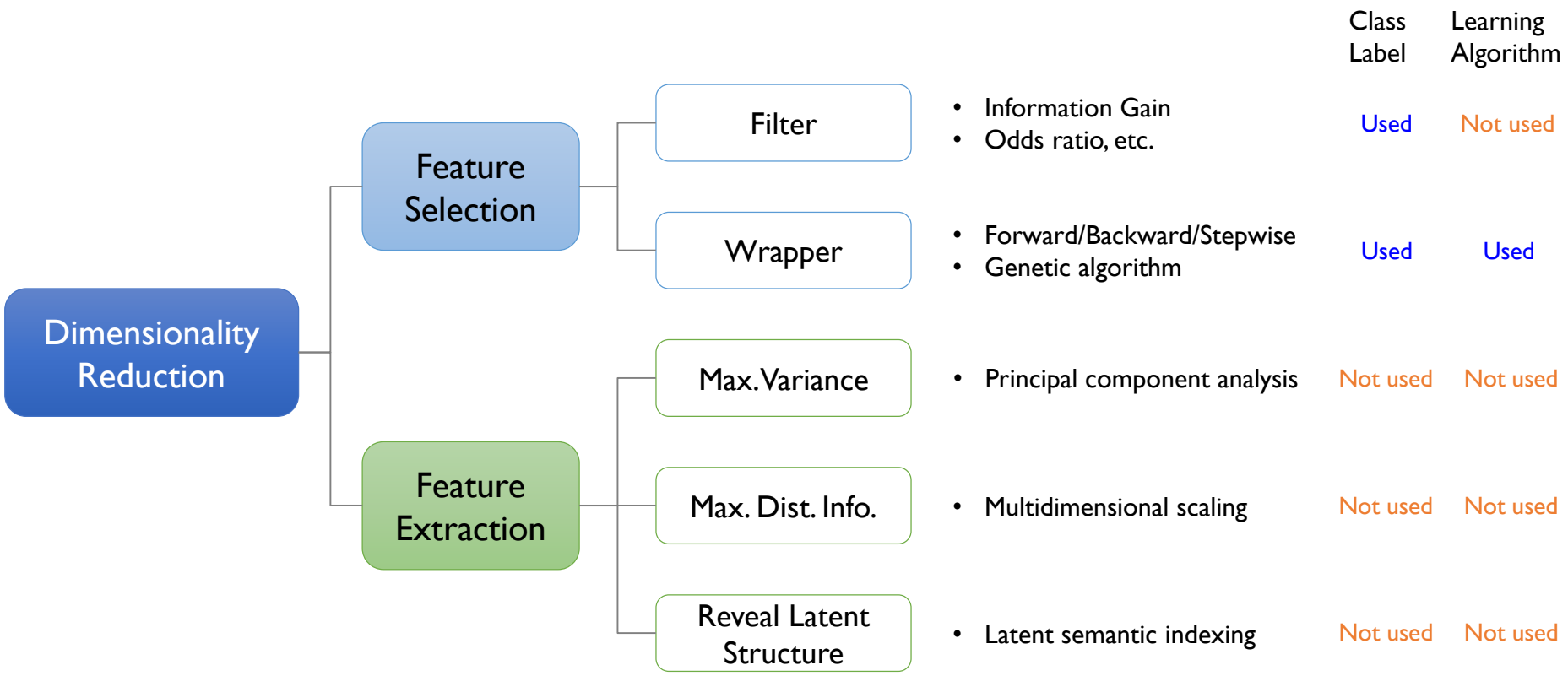
- **Problem 1**: High dimensionality (N. terms >> N. documents)
- **Problem 2**: Sparseness (Most elements in a term-document matrix are zero)

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

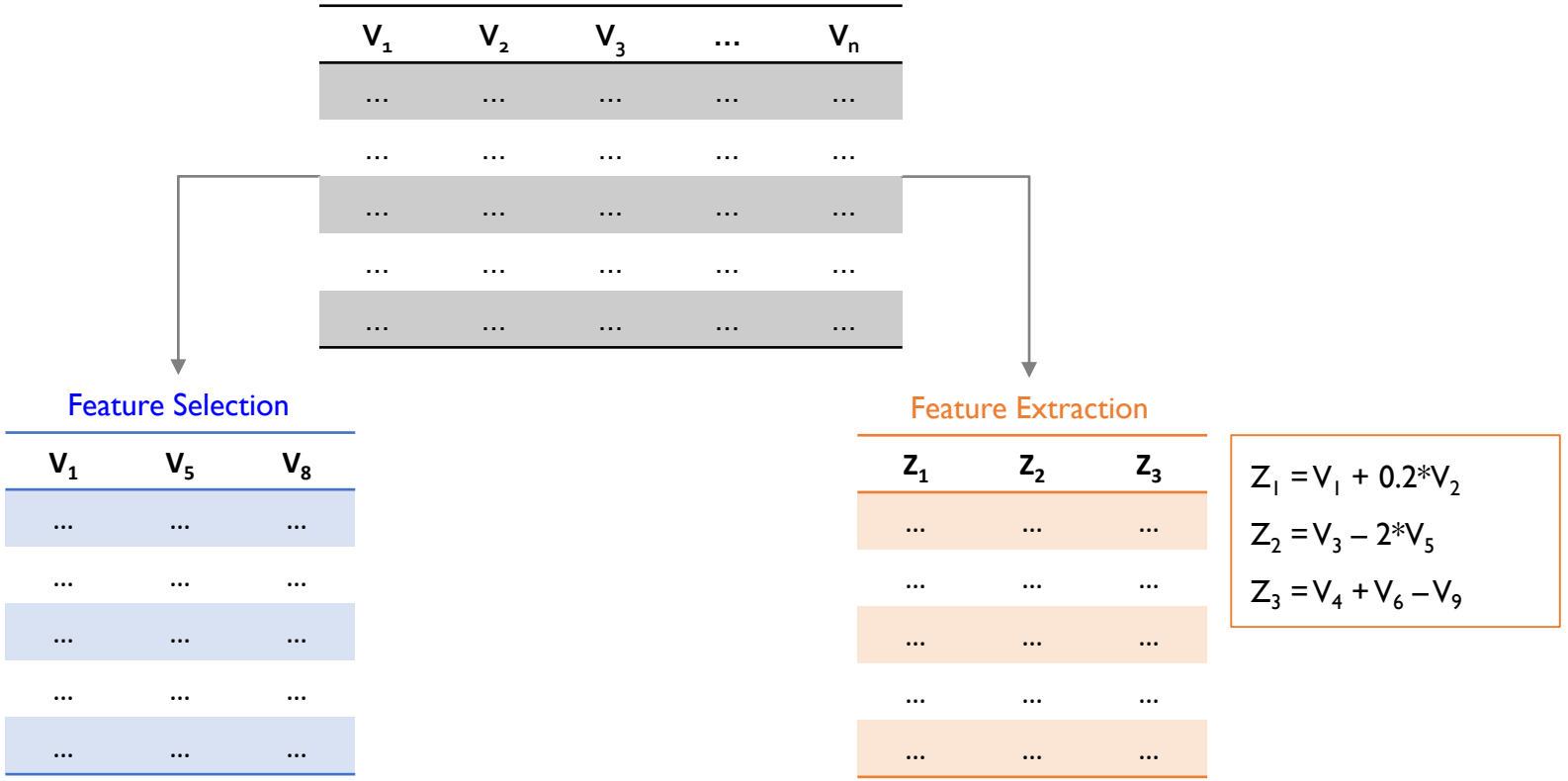
Dimensionality Reduction

- A simplified taxonomy of dimensionality reduction techniques



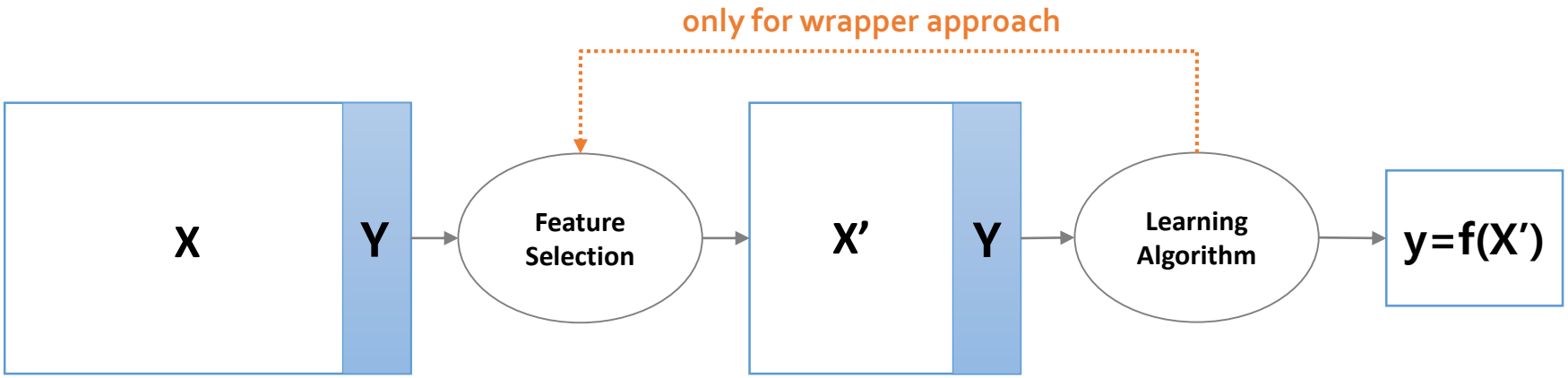
Dimensionality Reduction

- 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

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

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

	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

- For Term 1: DF(w) = 6
- For Term 2: DF(w) = 4
- For Term 3: DF(w) = 10

Feature Selection Metric 1-4

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

$$Acc(w) = N(Pos, w) - N(Neg, w)$$

	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

- For Term 1: $N(Pos, w) = 6, N(Neg, w) = 0,\ Acc(w) = 6$
- For Term 2: $N(Pos, w) = 0, N(Neg, w) = 4,\ Acc(w) = -4$
- For Term 3: $N(Pos, w) = 6, N(Neg, w) = 4,\ Acc(w) = 2$

Feature Selection Metric 1-4

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

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

	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

- For Term 1: $\frac{N(Pos,w)}{N(Pos)} = \frac{6}{6} = 1, \frac{N(Neg,w)}{N(Neg)} = \frac{0}{4} = 0, AccR(w) = 1$
- For Term 2: $\frac{N(Pos,w)}{N(Pos)} = \frac{0}{6} = 0, \frac{N(Neg,w)}{N(Neg)} = \frac{4}{4} = 1, AccR(w) = 1$
- For Term 3: $\frac{N(Pos,w)}{N(Pos)} = \frac{6}{6} = 1, \frac{N(Neg,w)}{N(Neg)} = \frac{4}{4} = 1, AccR(w) = 0$

Feature Selection Metric 1-4

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

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

	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

- For Term 1: $\frac{N(Pos,w)}{N(Pos)} = \frac{6}{6} = 1, \frac{N(Neg,w)}{N(Neg)} = \frac{0}{4} = 0, PR(w) = \infty$
- For Term 2: $\frac{N(Pos,w)}{N(Pos)} = \frac{0}{6} = 0, \frac{N(Neg,w)}{N(Neg)} = \frac{4}{4} = 1, AccR(w) = 0$
- For Term 3: $\frac{N(Pos,w)}{N(Pos)} = \frac{6}{6} = 1, \frac{N(Neg,w)}{N(Neg)} = \frac{4}{4} = 1, AccR(w) = 1$

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

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

    - Add 1 to any zero count in the denominator to avoid division by zero

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
Term 8	1	0	1	0	1	0	1	0	1	0
Term 9	1	1	1	0	0	0	1	0	0	0

    - For Term 8: $\frac{N(Pos,w)}{N(Neg,w)} = \frac{3}{2}, \frac{N(Neg,\bar{w})}{N(Pos,\bar{w})} = \frac{2}{3}, OddR(w) = 1$
    - For Term 9: $\frac{N(Pos,w)}{N(Neg,w)} = \frac{3}{1}, \frac{N(Neg,\bar{w})}{N(Pos,\bar{w})} = \frac{3}{3}, OddR(w) = 3$

Feature Selection Metric 5-6

- Odds ratio Numerator (OddN)

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

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
Term 8	1	0	1	0	1	0	1	0	1	0
Term 9	1	1	1	0	0	0	1	0	0	0

- For Term 8: $N(Pos, w) = 3, N(Neg, \overline{w}) = 2, OddN(w) = 6$
- For Term 9: $N(Pos, w) = 3, N(Neg, \overline{w}) = 3, OddN(w) = 9$

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 7

- F1-Measure

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

	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

- For Term 1: $F1(w) = \frac{2 \times 6}{6+6} = 1$
- For Term 2: $F1(w) = \frac{2 \times 0}{6+4} = 0$
- For Term 3: $F1(w) = \frac{2 \times 6}{6+10} = 0.75$

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 I

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

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

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

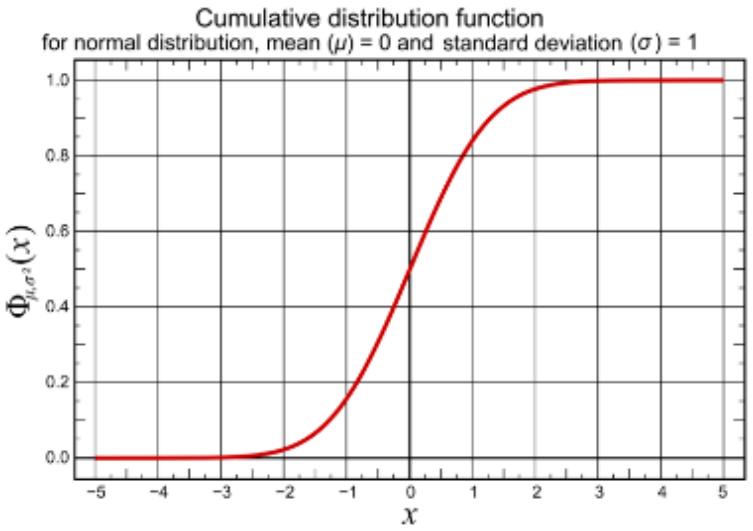
$$\chi^2(T1) = \frac{10 \times [0.6 \times 0.4 - 0 \times 0]^2}{0.6 \times 0.4 \times 0.6 \times 0.4} = 10.00 \qquad \chi^2(T4) = \frac{10 \times [0.6 \times 0.2 - 0.2 \times 0]^2}{0.8 \times 0.2 \times 0.6 \times 0.4} = 3.75$$

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 (2003)
  - ✓ 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

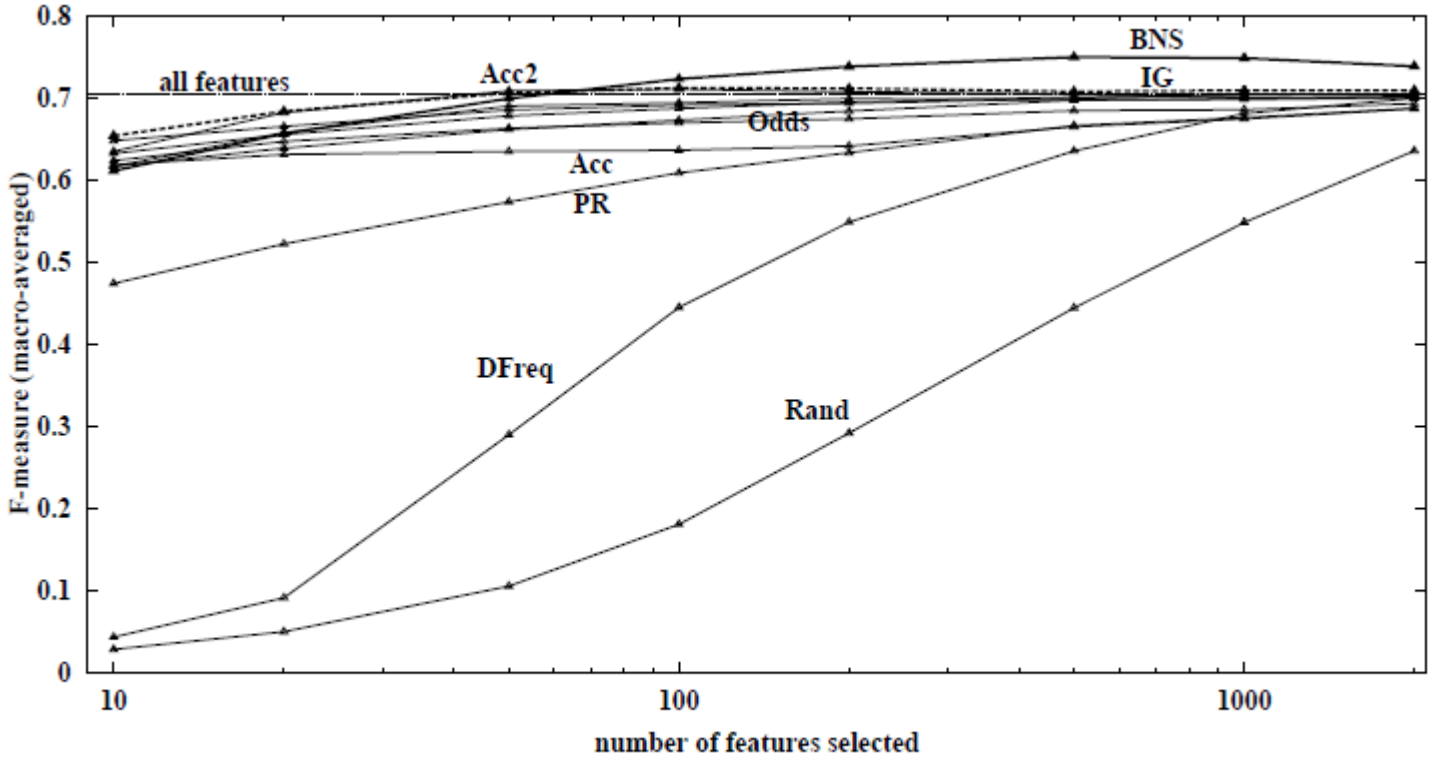


Figure 4. F-measure averaged over 229 problems for each metric, varying the number of features.

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

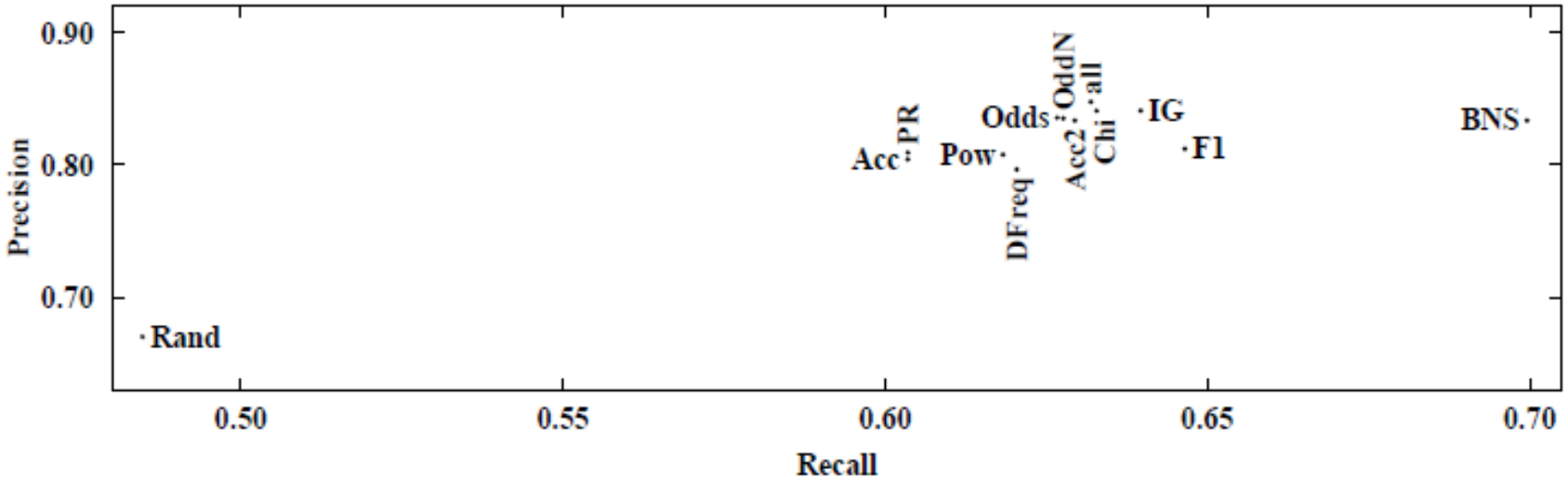


Figure 5. Precision-Recall tradeoffs from Figure 4 at 1000 features selected.

Empirical Study

Forman (2003)

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

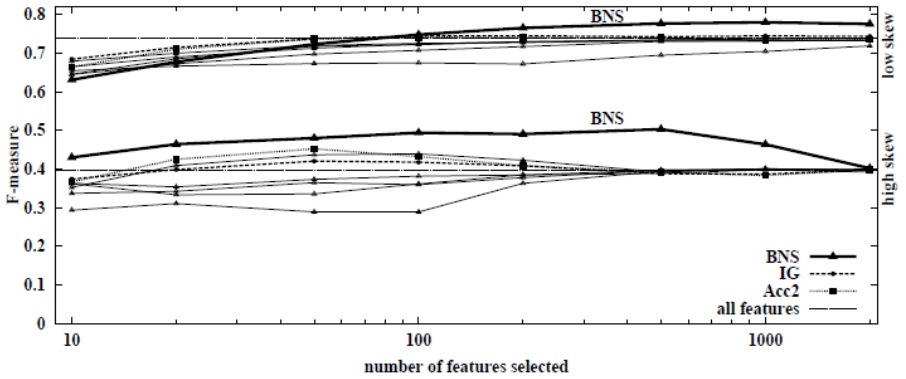


Figure 7. Average F-measure for each metric in low-skew and high-skew situations (threshold 1:67, the 90th percentile), as we vary the number of features. (To improve readability, we omitted Rand, DFreq, and PR.)

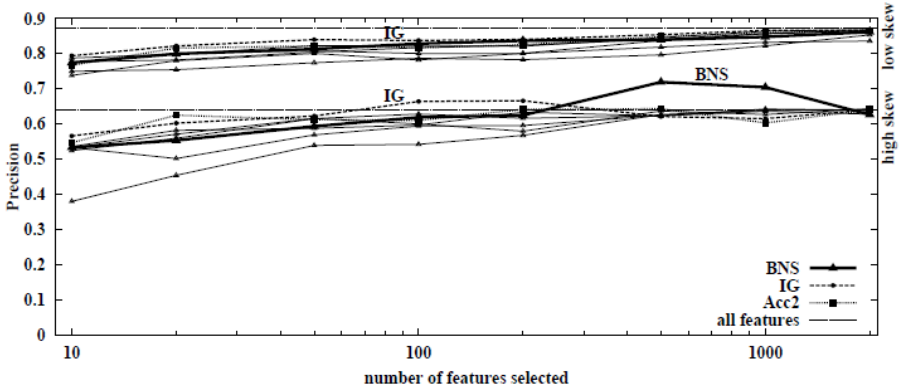


Figure 8. As Figure 7, but for precision.

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

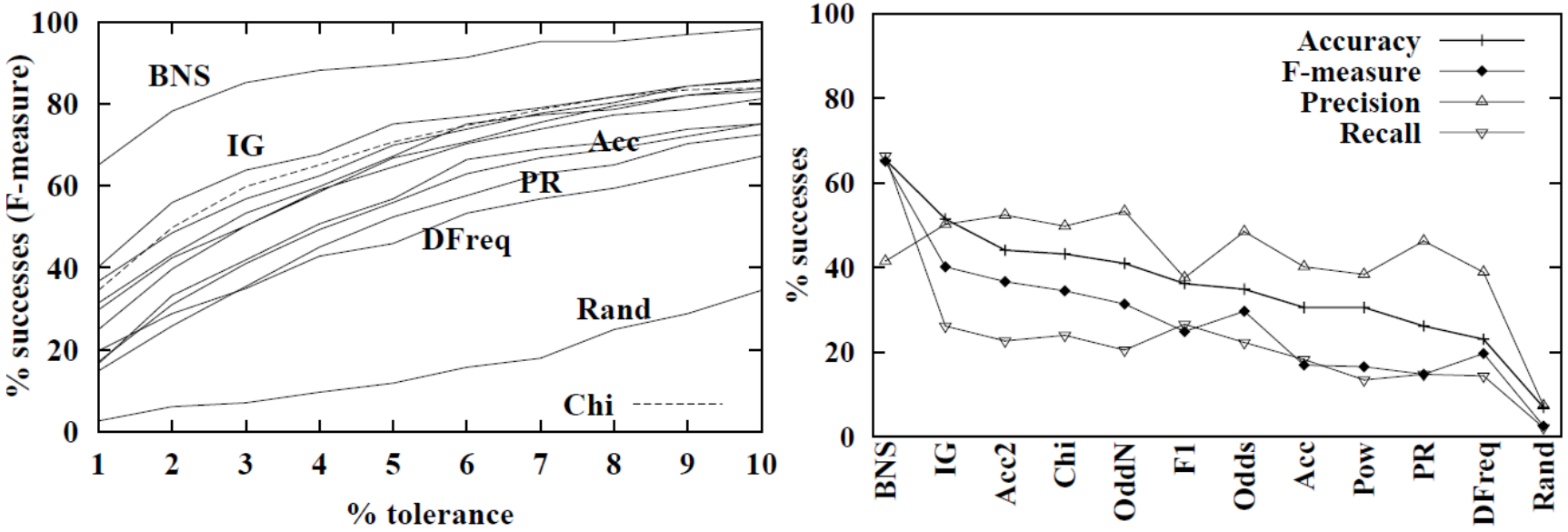


Figure 9. (a) Percentage of problems on which each metric scored within x% tolerance of the best F-measure of any metric. (b) Same, for F-measure, recall, and precision at a fixed tolerance of 1%, and for accuracy at a tolerance of 0.1%.

Empirical Study

Forman (2003)

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

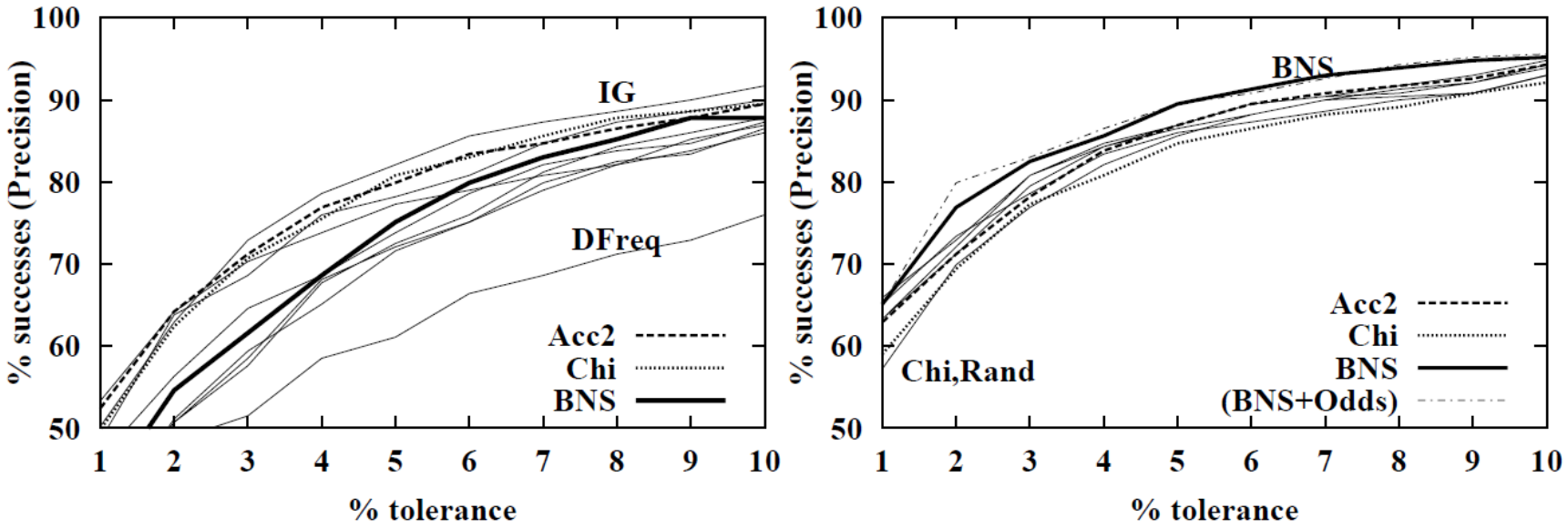


Figure 10. (a) As Figure 9a, but for precision. (b) Same axes and scale, but for each metric combined with IG. (Except the BNS+Odds curve is not combined with IG.)

AGENDA

01 Dimensionality Reduction

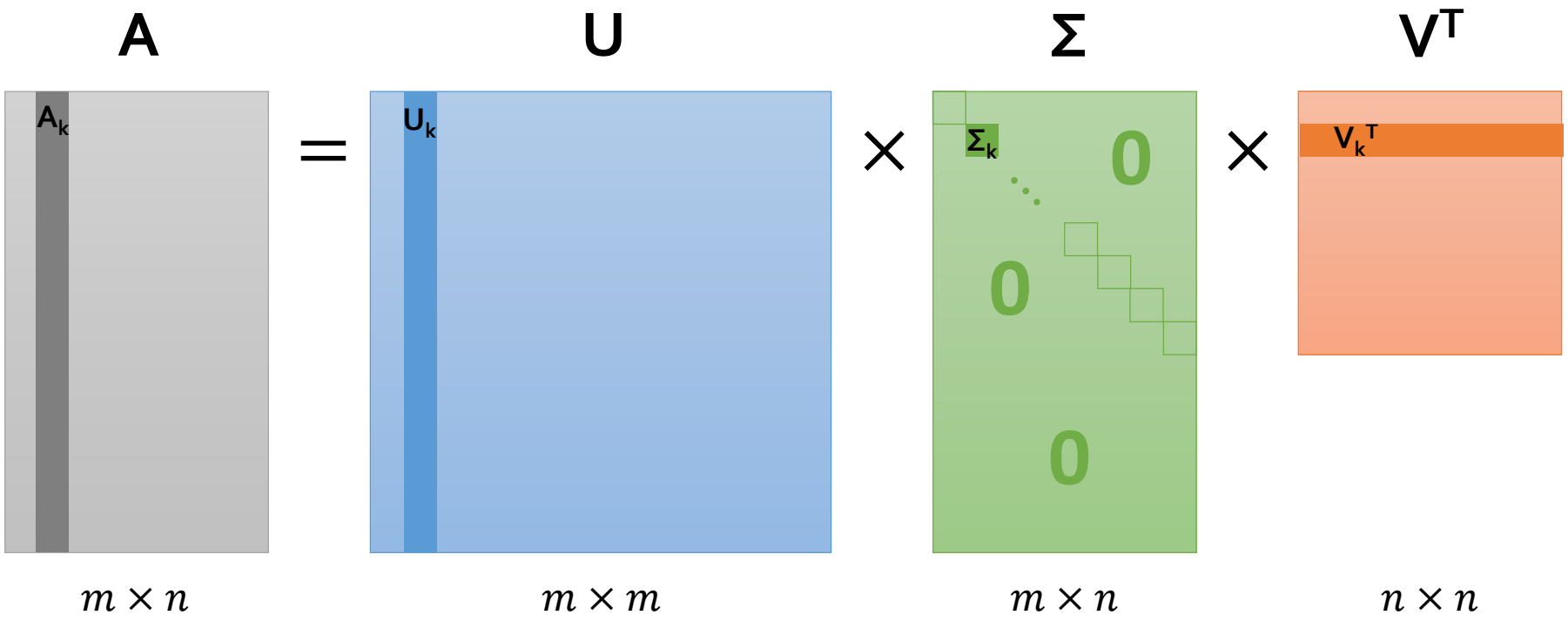
02 Feature Selection

03 Feature Extraction: LSA & t-SNE

Latent Semantic Analysis

Deerwester et al. (1990)

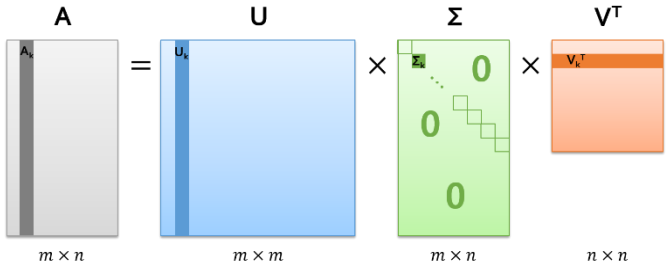
- Singular Value Decomposition: SVD
  - ✓ A factorization of a real or complex matrix
  - ✓ For a rectangular m by n matrix A (m > n) (Term-Document Matrix in Text Mining)
  - ✓ Full SVD becomes



Latent Semantic Analysis

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

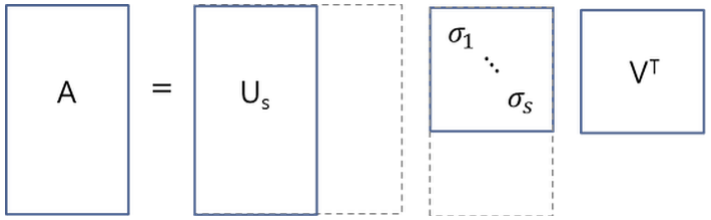
$$\mathbf{U}^{\mathbf{T}}\mathbf{U} = \mathbf{V}^{\mathbf{T}}\mathbf{V} = \mathbf{I}$$



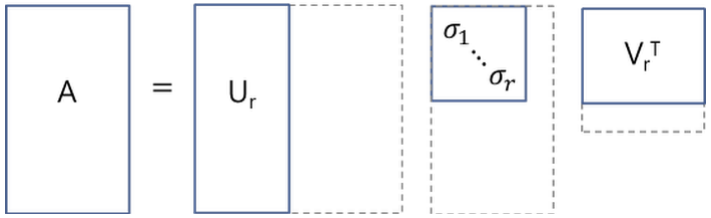
  - ✓ The number of positive singular values in **Σ** = Rank(A)
  - ✓ Running time: O(mnc)
    - c: the average number of words per document

Latent Semantic Analysis

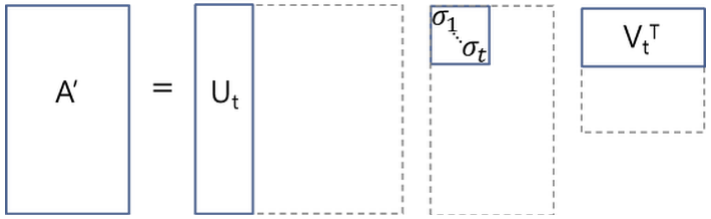
- Reduces SVDs
  - ✓ Thin SVD: Transform **Σ** to a square matrix & remove the corresponding columns of U



  - ✓ 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

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

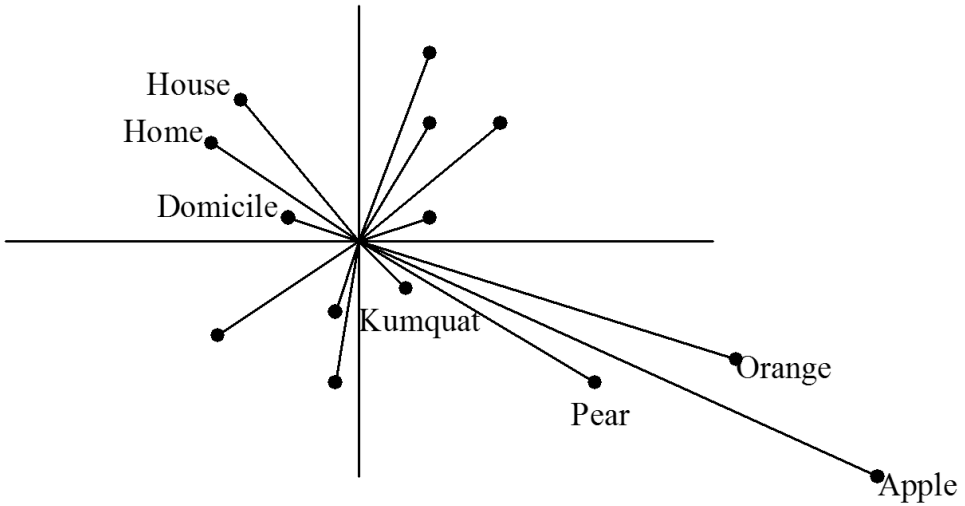
  - ✓ Truncated SVD

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

Latent Semantic Analysis

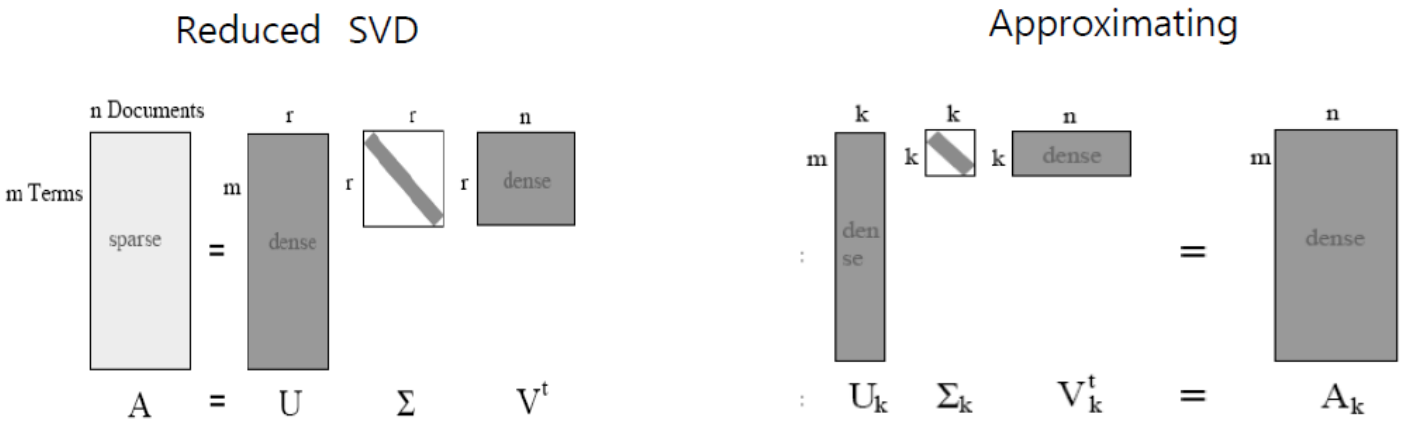
Papadimitriou et al.

- 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

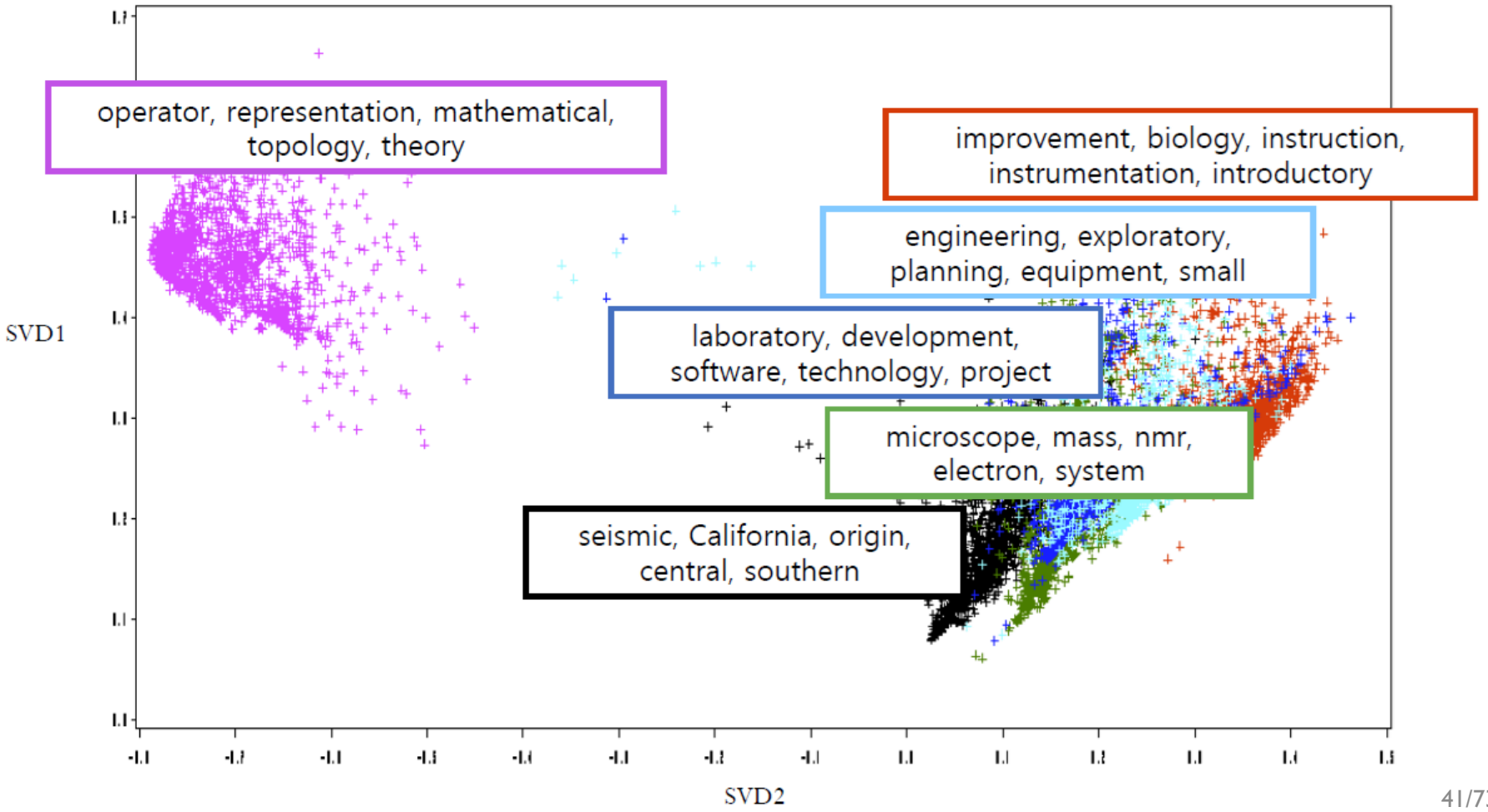
Latent Semantic Analysis

- 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

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

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

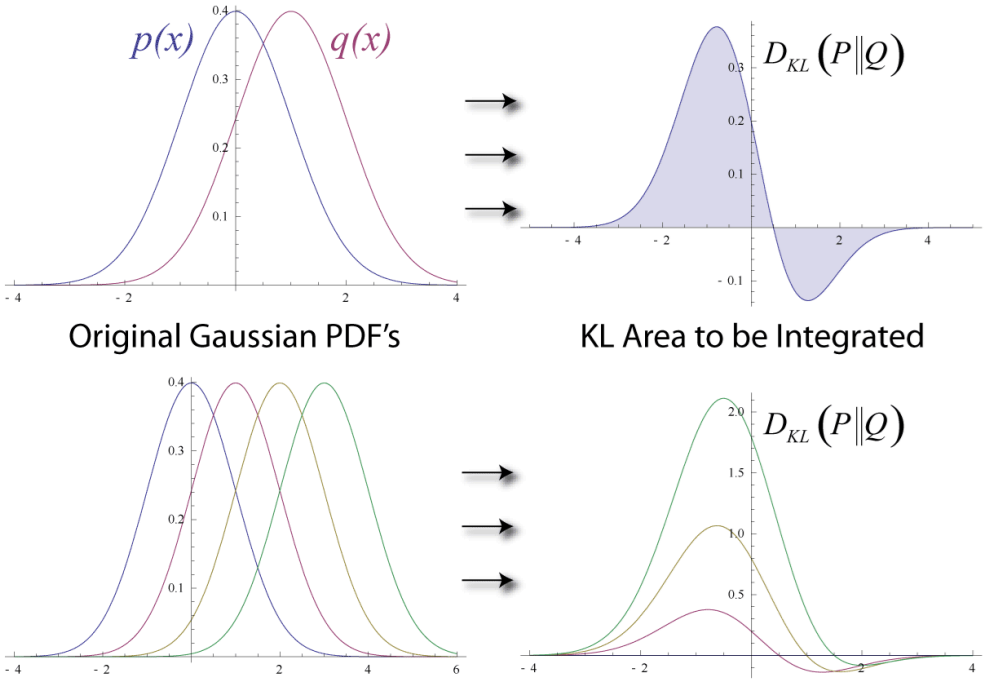
$$p_{j|i} = \frac{e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma_i^2}}}{\sum_k e^{-\frac{||\mathbf{x}_i - \mathbf{x}_k||^2}{2\sigma_i^2}}}$$

  - ✓ 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

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



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

Stochastic Neighbor Embedding

- Cost Function for a Low-dimensional Representation
  - ✓ Gradient
    - Differencing Cost is tedious because $\mathbf{y}_k$ affect q_{ij} via the normalized term in Eq. (3), but the result is simple Hinton and Roweis (2002)
    - The gradient has a surprisingly simple form Maaten and Hinton (2008)

Stochastic Neighbor Embedding

- Gradient of the cost function (Optional)

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

$$C = \sum_i \sum_j p_{j|i} log p_{j|i} - \sum_i \sum_j p_{j|i} log q_{j|i}$$

$$C' = -\sum_i \sum_j p_{j|i} log q_{j|i} \qquad \left(\frac{\partial C}{\partial y_t} = \frac{\partial C'}{\partial y_t}\right)$$

$$C' = \underbrace{-\sum_i p_{t|i} log q_{t|i}}_{①} \underbrace{-\sum_j p_{j|t} log q_{j|t}}_{②} \underbrace{-\sum_{i \neq t} \sum_{j \neq t} p_{i|j} log q_{i|j}}_{③}$$

Stochastic Neighbor Embedding

- Gradient of the cost function (Optional)

$$d_{ti} = exp(-||\mathbf{y}_t - \mathbf{y}_i||^2) = d_{it}$$

$$\frac{\partial d_{ti}}{\partial \mathbf{y}_t} = d'_{ti} = -2(\mathbf{y}_t - \mathbf{y}_i)exp(-||\mathbf{y}_t - \mathbf{y}_i||^2) = -2(\mathbf{y}_t - \mathbf{y}_i)d_{ti}$$

$$q_{t|i} = \frac{exp(-||\mathbf{y}_i - \mathbf{y}_t||^2)}{\sum_{k \neq i} exp(-||\mathbf{y}_i - \mathbf{y}_k||^2)} = \frac{d_{it}}{\sum_{k \neq i} d_{ik}}$$

$$q_{j|t} = \frac{exp(-||\mathbf{y}_t - \mathbf{y}_j||^2)}{\sum_{k \neq t} exp(-||\mathbf{y}_t - \mathbf{y}_k||^2)} = \frac{d_{tj}}{\sum_{k \neq t} d_{tk}}$$

$$q_{i|j} = \frac{exp(-||\mathbf{y}_j - \mathbf{y}_i||^2)}{\sum_{k \neq j} exp(-||\mathbf{y}_j - \mathbf{y}_k||^2)} = \frac{d_{ji}}{\sum_{k \neq j} d_{jk}}$$

Stochastic Neighbor Embedding

- Gradient of the cost function ① (Optional)

$$\frac{\partial}{\partial y_t}\Big(-\sum_i p_{t|i} log q_{t|i}\Big) = -\sum_i p_{t|i} \cdot \frac{1}{q_{t|i}} \cdot \frac{\partial q_{t|i}}{\partial y_t}$$

$$= -\sum_i p_{t|i} \cdot \frac{1}{q_{t|i}} \cdot \frac{d'_{it} \cdot (\sum_{k \neq i} d_{ik}) - d_{it} \cdot d'_{it}}{(\sum_{k \neq i} d_{ik})^2}$$

$$= -\sum_i p_{t|i} \cdot \frac{1}{q_{t|i}} \cdot \frac{-2(\mathbf{y}_t - \mathbf{y}_i) \cdot d_{it} \cdot (\sum_{k \neq i} d_{ik}) + 2(\mathbf{y}_t - \mathbf{y}_i) \cdot d_{it}^2}{(\sum_{k \neq i} d_{ik})^2}$$

$$= -\sum_i p_{t|i} \cdot \frac{1}{q_{t|i}} \cdot \Big(-2(\mathbf{y}_t - \mathbf{y}_i) \cdot q_{t|i} + 2(\mathbf{y}_t - \mathbf{y}_i) \cdot q_{t|i}^2\Big)$$

$$= \sum_i p_{t|i} \cdot 2(\mathbf{y}_t - \mathbf{y}_i)(1 - q_{t|i})$$

Stochastic Neighbor Embedding

- Gradient of the cost function ② (Optional)

$$\frac{\partial}{\partial y_t}\Big(-\sum_j p_{j|t} log q_{j|t}\Big) = -\sum_j p_{j|t} \cdot \frac{1}{q_{j|t}} \cdot \frac{\partial q_{j|t}}{\partial y_t}$$

$$= -\sum_j p_{j|t} \cdot \frac{1}{q_{j|t}} \cdot \frac{d'_{tj} \cdot (\sum_{k \neq t} d_{tk}) - d_{tj} \cdot (\sum_{k \neq t} d'_{tk})}{(\sum_{k \neq t} d_{tk})^2}$$

$$= -\sum_j p_{j|t} \cdot \frac{1}{q_{j|t}} \cdot \frac{-2(\mathbf{y}_t - \mathbf{y}_j) \cdot d_{tj} \cdot (\sum_{k \neq t} d_{tk}) - d_{tj} \cdot (\sum_{k \neq t} d'_{tk})}{(\sum_{k \neq t} d_{tk})^2}$$

$$= 2\sum_j p_{j|t} \cdot (\mathbf{y}_t - \mathbf{y}_j) + \sum_j p_{j|t} \cdot \frac{\sum_{k \neq t} d'_{tk}}{\sum_{k \neq t} d_{tk}} \qquad (d'_{tt} = 0, \sum_j p_{j|t} = 1)$$

$$= 2\sum_j p_{j|t} \cdot (\mathbf{y}_t - \mathbf{y}_j) + \sum_j \cdot \frac{d'_{tj}}{\sum_{k \neq t} d_{tk}}$$

Stochastic Neighbor Embedding

- Gradient of the cost function ② (Optional)

$$= 2\sum_j p_{j|t} \cdot (\mathbf{y}_t - \mathbf{y}_j) + \sum_j \cdot \frac{d'_{tj}}{\sum_{k \neq t} d_{tk}}$$

$$= 2\sum_j p_{j|t} \cdot (\mathbf{y}_t - \mathbf{y}_j) - 2\sum_j (\mathbf{y}_t - \mathbf{y}_j) \cdot \frac{d_{tj}}{\sum_{k \neq t} d_{tk}}$$

$$= 2\sum_j p_{j|t} \cdot (\mathbf{y}_t - \mathbf{y}_j) - 2\sum_j (\mathbf{y}_t - \mathbf{y}_j) \cdot q_{j|t}$$

$$= 2\sum_j (\mathbf{y}_t - \mathbf{y}_j)(p_{j|t} - q_{j|t})$$

Stochastic Neighbor Embedding

- Gradient of the cost function ③ (Optional)

$$\frac{\partial}{\partial y_t}\Big(-\sum_{i\neq t}\sum_{j\neq t} p_{i|j} log q_{i|j}\Big) = -\sum_{i\neq t}\sum_{j\neq t} p_{i|j}\cdot\frac{1}{q_{i|j}}\cdot\frac{\partial q_{i|j}}{\partial y_t}$$

$$= -\sum_{i\neq t}\sum_{j\neq t} p_{i|j}\cdot\frac{1}{q_{i|j}}\cdot\frac{d'_{ji}\cdot\sum_{k\neq j} d_{jk} - d_{ji}\cdot d'_{jt}}{(\sum_{k\neq j} d_{jk})^2} \qquad (d'_{ji}=0)$$

$$= -\sum_{i\neq t}\sum_{j\neq t} p_{i|j}\cdot\frac{1}{q_{i|j}}\cdot\frac{2(\mathbf{y}_t-\mathbf{y}_j)\cdot d_{ji}\cdot d_{jt}}{(\sum_{k\neq j} d_{jk})^2}$$

$$= -\sum_{i\neq t}\sum_{j\neq t} p_{i|j}\cdot\frac{1}{q_{i|j}}\cdot 2(\mathbf{y}_t-\mathbf{y}_j)\cdot q_{i|j}\cdot q_{t|j}$$

$$= -\sum_{i\neq t}\sum_{j\neq t} 2(\mathbf{y}_t-\mathbf{y}_j)\cdot p_{i|j}\cdot q_{t|j}$$

Stochastic Neighbor Embedding

- Gradient of the cost function ① + ③ (Optional)

$$\sum_i p_{t|i} \cdot 2(\mathbf{y}_t - \mathbf{y}_i)(1 - q_{t|i}) - \sum_{i \neq t} \sum_{j \neq t} 2(\mathbf{y}_t - \mathbf{y}_j) \cdot p_{i|j} \cdot q_{t|j}$$

Replace the subscript i with j

$$= 2\sum_j (\mathbf{y}_t - \mathbf{y}_j) \cdot p_{t|j} - 2\sum_j (\mathbf{y}_t - \mathbf{y}_j) \cdot p_{t|j} \cdot q_{t|j} - 2\sum_{i \neq t} \sum_{j \neq t} (\mathbf{y}_t - \mathbf{y}_j) \cdot p_{i|j} \cdot q_{t|j}$$

$$= 2\sum_j (\mathbf{y}_t - \mathbf{y}_j) \cdot p_{t|j} - 2\sum_i \sum_j (\mathbf{y}_t - \mathbf{y}_j) \cdot p_{i|j} \cdot q_{t|j}$$

$$= 2\sum_j (\mathbf{y}_t - \mathbf{y}_j) \cdot p_{t|j} - 2\sum_j \sum_i p_{i|j} \cdot (\mathbf{y}_t - \mathbf{y}_j) \cdot q_{t|j} \qquad (\sum_i p_{i|j} = 1)$$

$$= 2\sum_j (\mathbf{y}_t - \mathbf{y}_j) \cdot p_{t|j} - 2\sum_j (\mathbf{y}_t - \mathbf{y}_j) \cdot q_{t|j} = 2\sum_j (\mathbf{y}_t - \mathbf{y}_j)(p_{t|j} - q_{t|j})$$

Stochastic Neighbor Embedding

- Gradient of the cost function ① + ② + ③ (Optional)

$$2\sum_j (\mathbf{y}_t - \mathbf{y}_j)(p_{j|t} - q_{j|t}) + 2\sum_j (\mathbf{y}_t - \mathbf{y}_j)(p_{t|j} - q_{t|j})$$

$$= 2\sum_j (\mathbf{y}_t - \mathbf{y}_j)(p_{t|j} - q_{t|j} + p_{j|t} - q_{j|t})$$

- Update the coordinate in the lower dimension to minimize the cost function
  - ✓ Gradient update with a momentum term

$$\mathcal{Y}^{(t+1)} = \mathcal{Y}^{(t)} + \eta\frac{\partial C}{\partial \mathcal{Y}} + \alpha(t)\left(\mathcal{Y}^{(t)} - \mathcal{Y}^{(t-1)}\right)$$

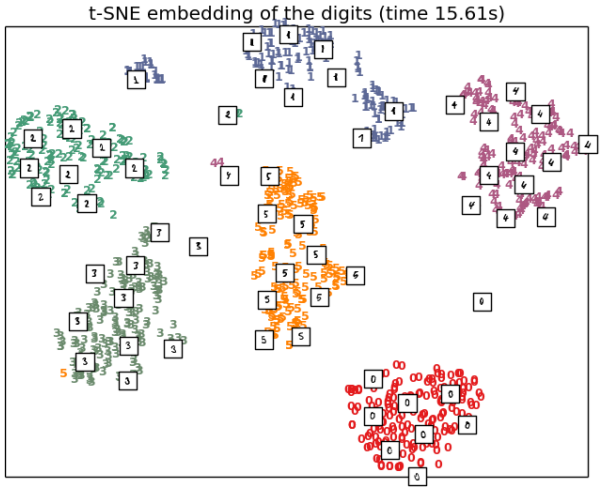
Stochastic Neighbor Embedding

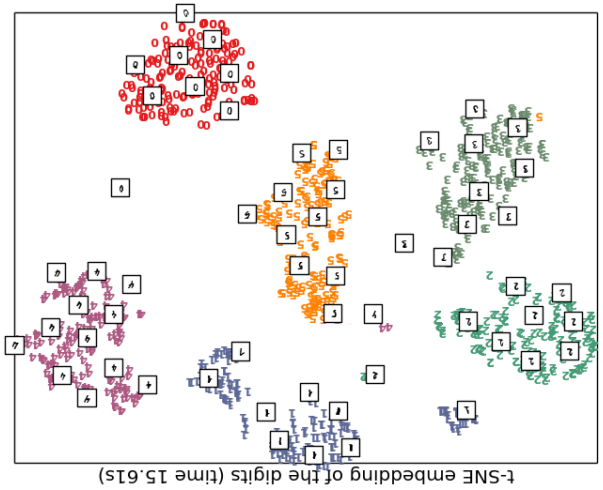
- From the original paper

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

- In this lecture note

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





Symmetric SNE

- Turning conditional probabilities into pairwise probabilities

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

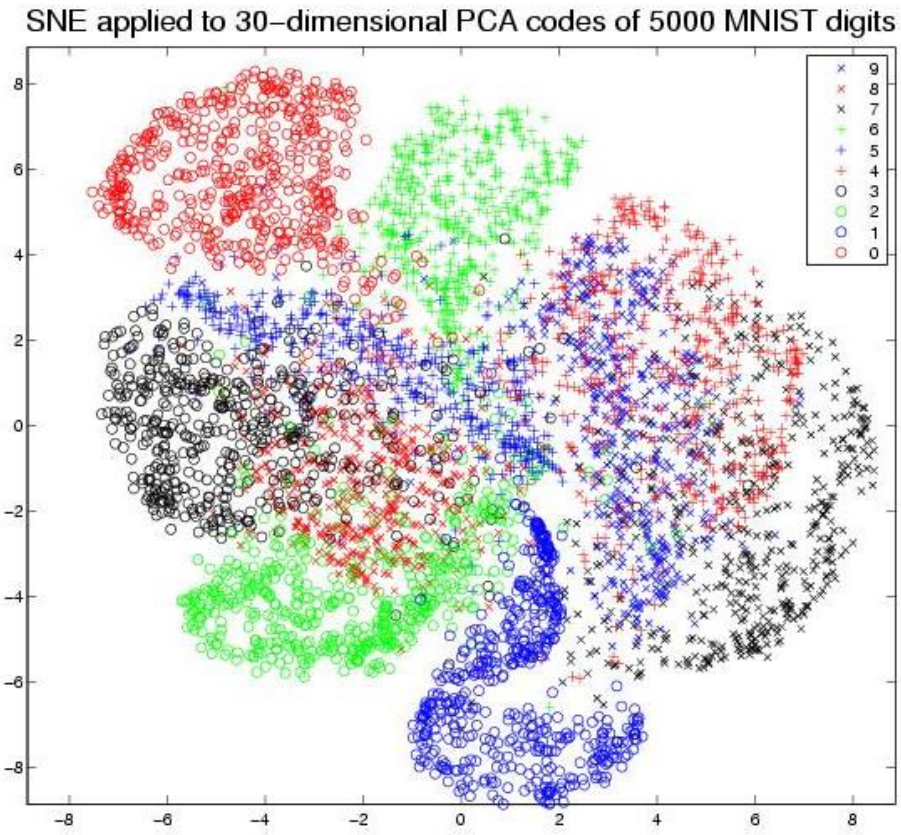
✓ Cost function and gradient

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

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

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

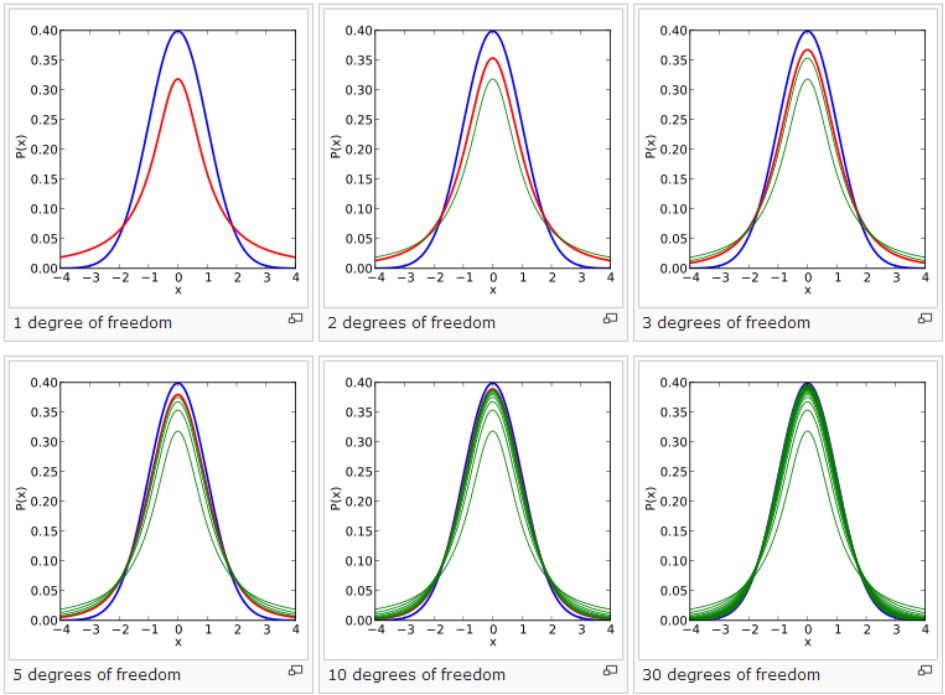
Maaten and Hinton (2008)

- 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

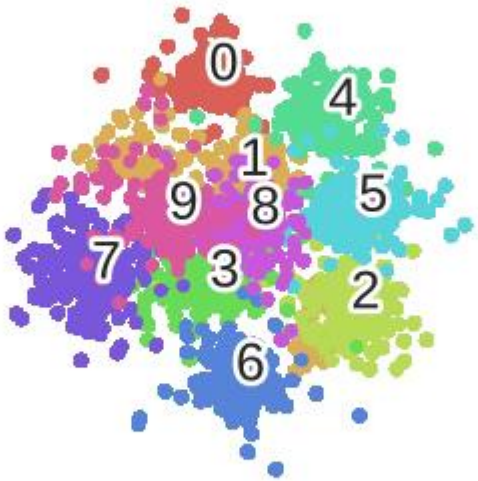
- t-SNE algorithm

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

Data: data set $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$,
cost function parameters: perplexity *Perp*,
optimization parameters: number of iterations T, learning rate η, momentum $\alpha(t)$.
Result: low-dimensional data representation $\mathcal{Y}^{(T)} = \{y_1, y_2, \ldots, y_n\}$.
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 $\mathcal{Y}^{(0)} = \{y_1, y_2, \ldots, y_n\}$ 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

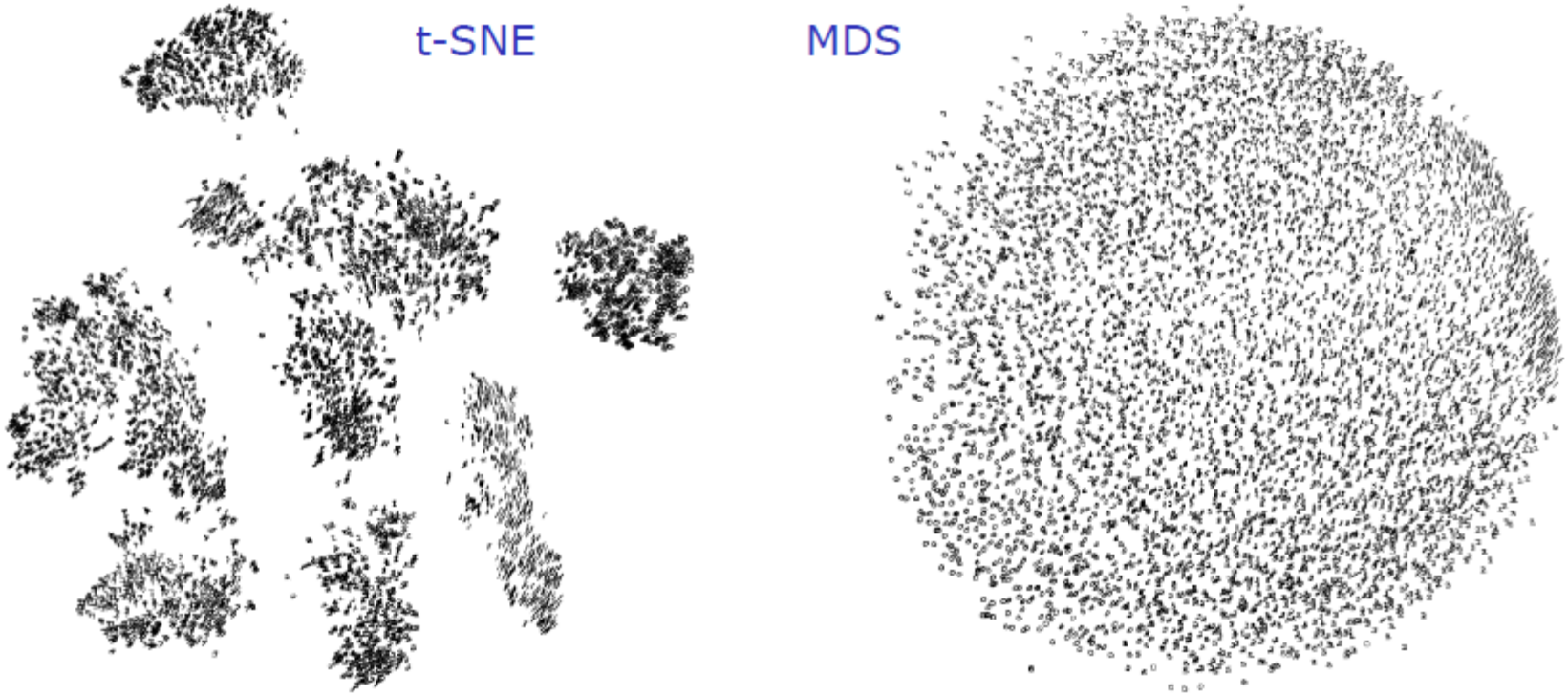
t-SNE

- Training t-SNE



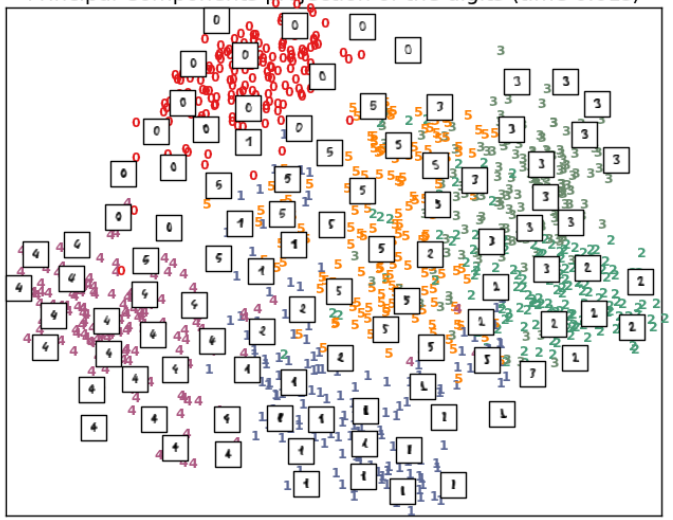
t-SNE vs. MDS

- MNIST dataset

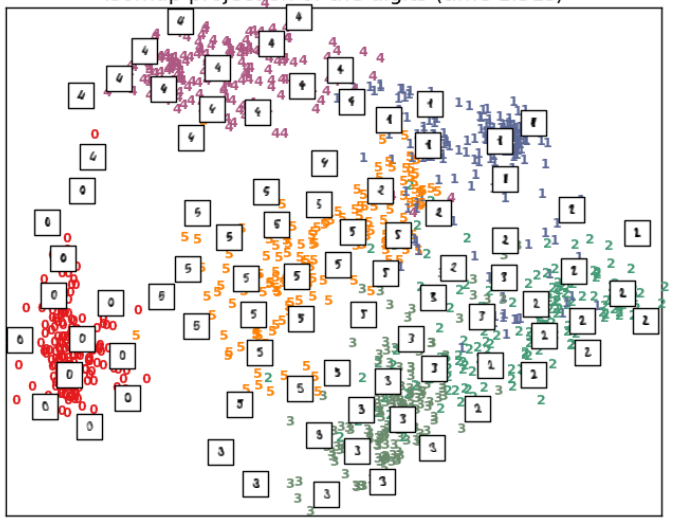


t-SNE Examples

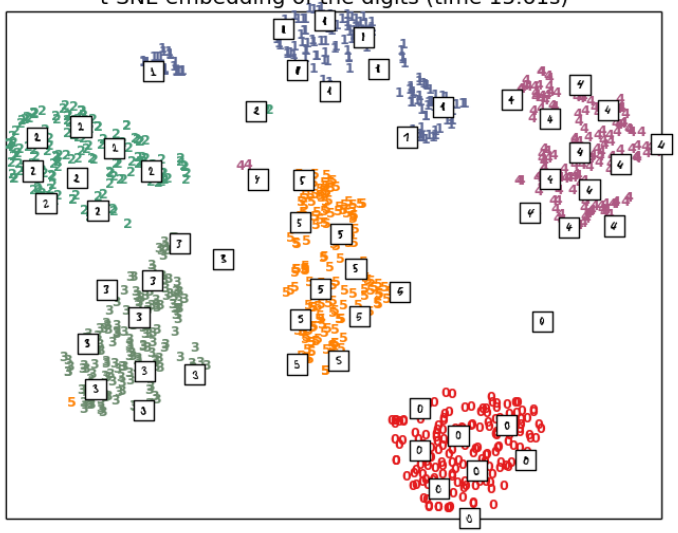
Principal Components projection of the digits (time 0.01s)



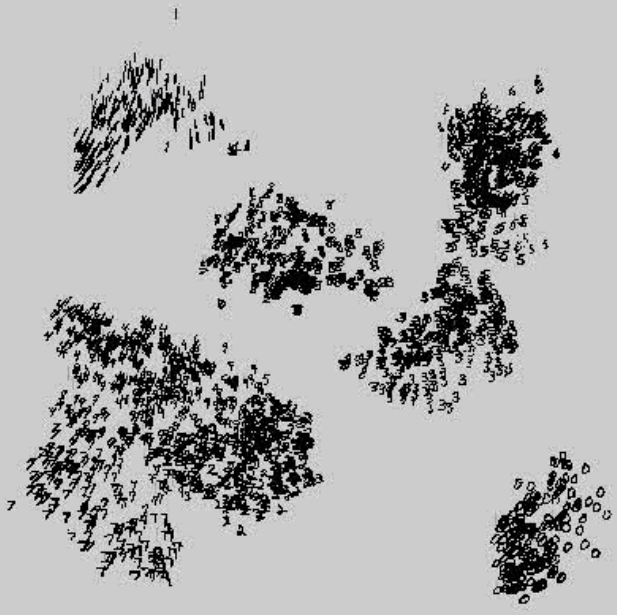
Isomap projection of the digits (time 1.51s)



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

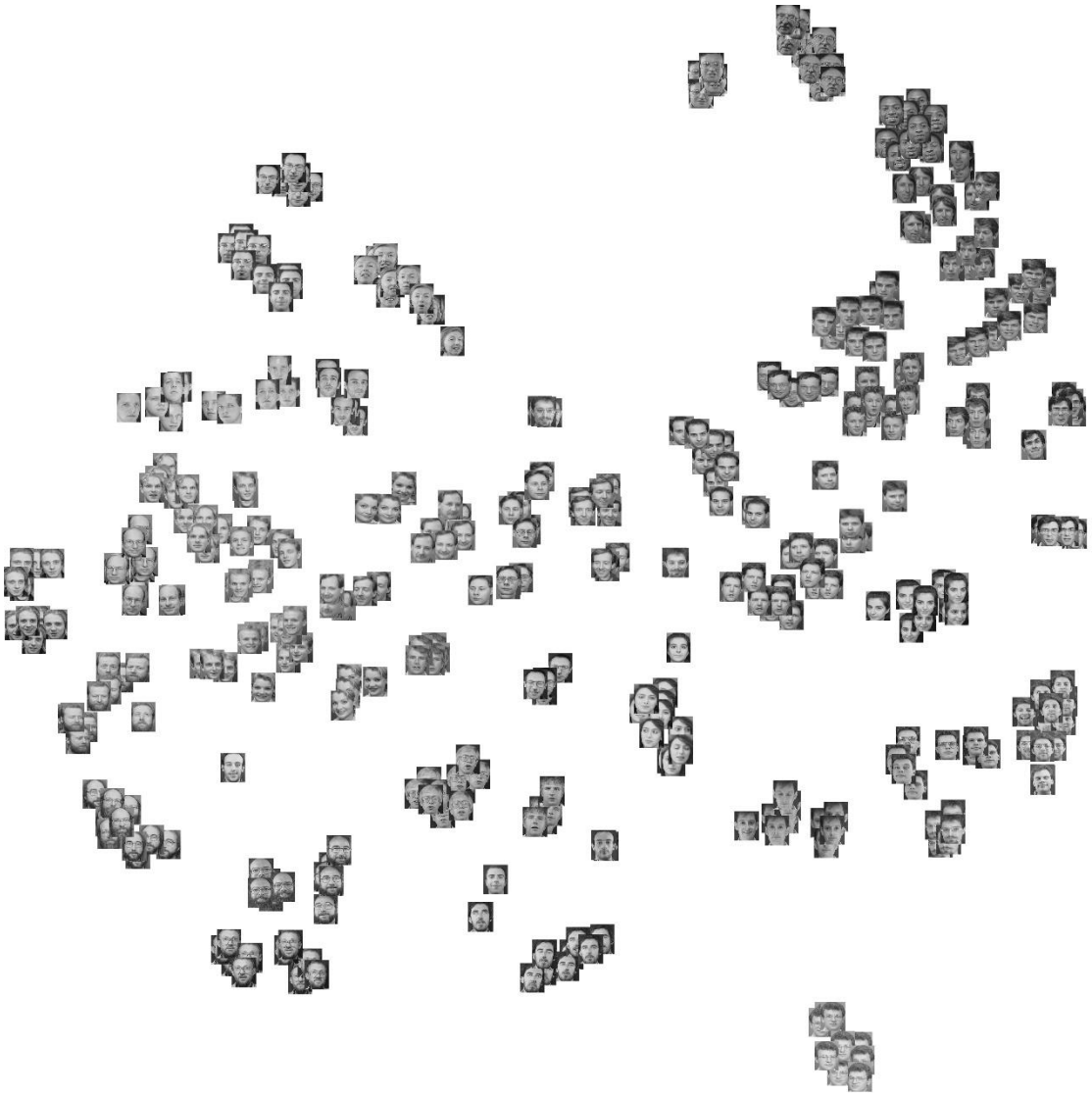


t-SNE Examples



t-SNE Examples

- Olivetti faces datasets



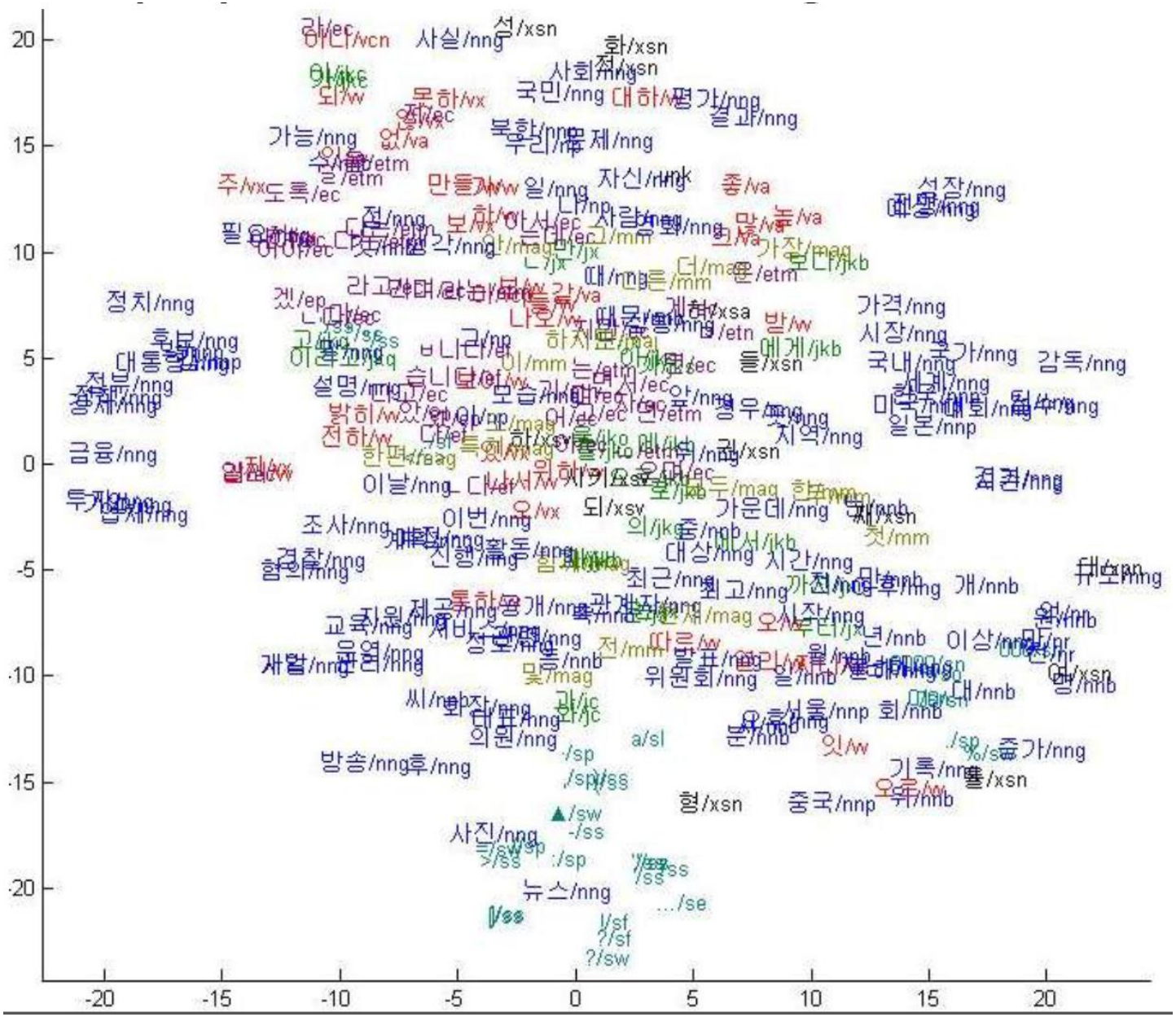
t-SNE Examples

- Netflix dataset

t-SNE Examples

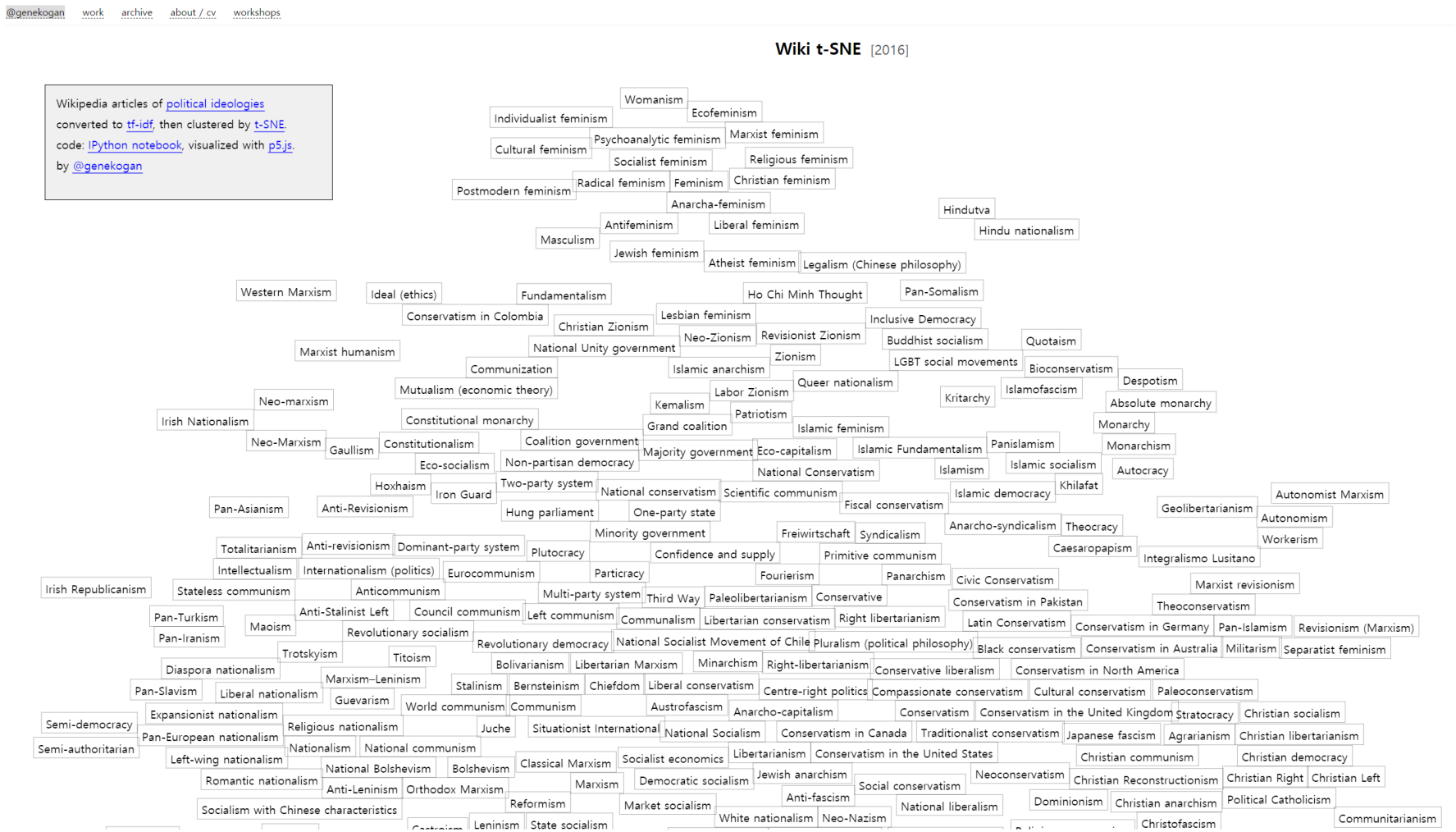
- CalTech-101

t-SNE Examples



t-SNE Examples

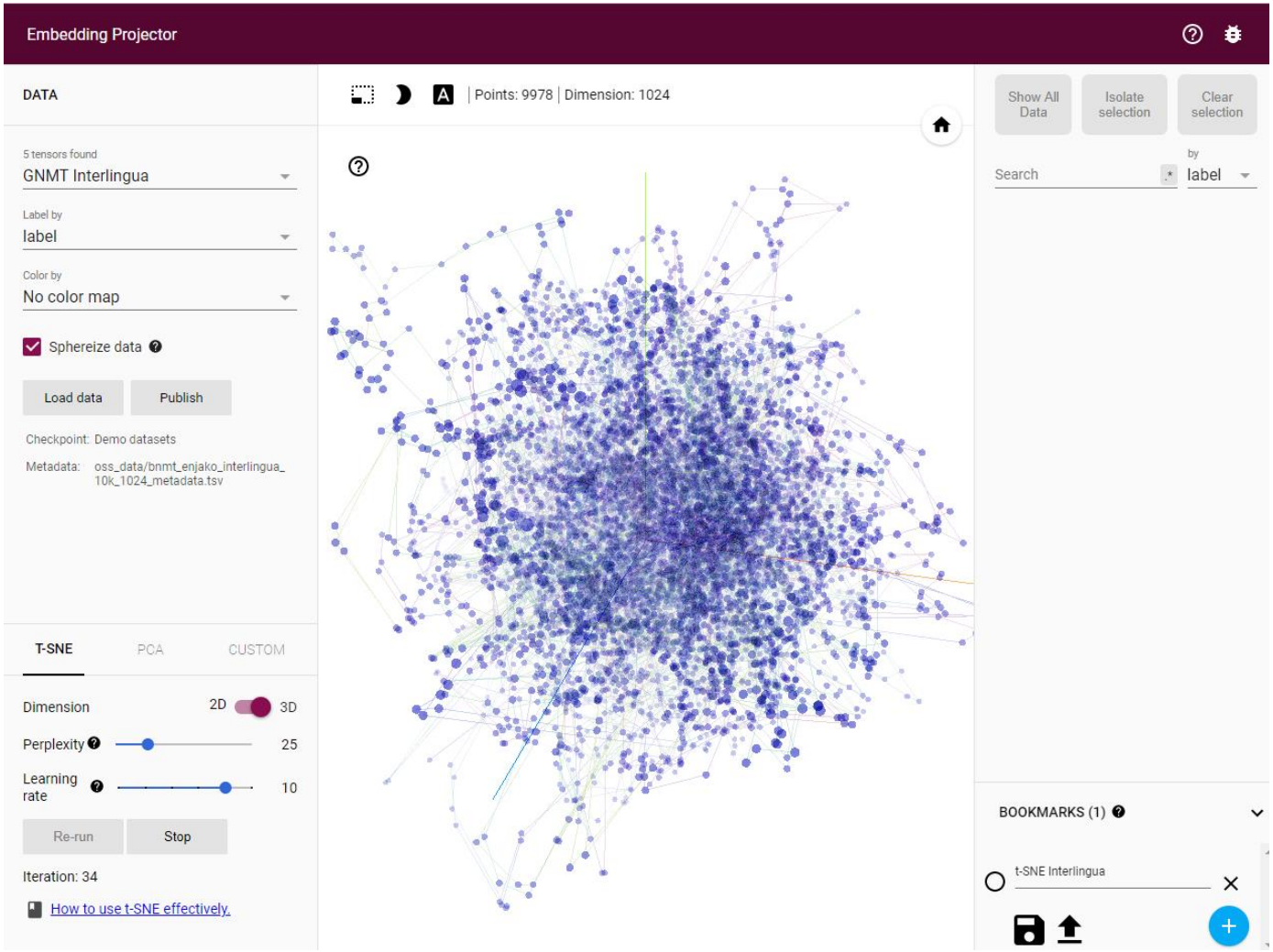
- Wiki t-SNE



http://genekogan.com/works/wiki-tSNE/

t-SNE Examples

- Embedding Projector by Google



http://projector.tensorflow.org/


ANY
questions?

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".