Lifting the Curse: Exploring Dimensionality Reduction on Text Clustering Applications

L. Akritidis[1], P. Bozanis[1]

[1]School of Science and Technology, International Hellenic University

13th International Conference on Information, Intelligence, Systems and Applications (IISA 2022)

Ionian University, Corfu, Greece

18 – 20 July, 2022

Text clustering

- Nowadays, huge amounts of text are being generated on the Web.
- Vast number of relevant applications:
  - instant messengers, social networks, e-mail clients, news portals, blog communities, commercial platforms, etc.
- Plus: There is a constantly growing requirement for effectively identifying documents of similar content.
- Text clustering: the **unsupervised** problem of identifying and grouping together **semantically similar** documents in **previously unexplored** text collections.
  - One of the most emerging problems of the machine learning discipline.

Challenges

- **The text is diverse**: two or more documents may express similar or identical meanings, despite the fact that they consist of completely different words.
- Text diversity has three side-effects:
- It **blurs** the semantic similarities between two documents, making it hard to identify their thematic affinity.
- It leads to **sparse** vector representations: most elements in the generated text vectors are zero.
- **High-dimensionality**: the generated text vectors are very long.

Dimensionality Reduction

- A well-known technique for **limiting the feature space size** and **discovering latent meaningful variables** in the input data.
- Particularly valuable when the raw data is sparse and its processing by machine learning algorithms becomes computationally very expensive.
- State-of-the-art dimensionality reduction algorithms for text data:
  - Singular Value Decomposition (SVD).
  - Principal Component Analysis (PCA), Independent Component Analysis.
  - Truncated Singular Value Decomposition (TSVD).
  - Non-Negative Matrix Factorization (NMF).

An experimental study

- In this paper, we conduct a study with the aim of evaluating the impact of dimensionality reduction techniques in text clustering applications.
- 8 state-of-the-art data clustering algorithms were evaluated in terms of effectiveness (i.e. clustering quality) and efficiency (i.e. running times).
- We used 6 datasets and we applied several dimensionality reduction methods with different degrees of reduction.
  - Progressive reduction of the input space size by 1, 2, 3, … orders of magnitude.

Datasets

- 6 datasets were used in this study (see Table I).
- Tweet: 2472 Tweets relevant to 89 queries of TREC microblog tracks 2011 and 2012.
- PriceRunner: 3564 titles of 1280 different TV models.
- TitleSet/SnippetSet: 11109 headlines from 152 stories published on Google News.
- Newsgroups, a traditional benchmark - 20 thousand news stories in 20 classes.
- Wines: a collection of 11258 descriptions for 88 wine varieties produced by 995 wineries.

TABLE I
TEXT CLUSTERING DATASETS

Dataset	Samples	Dimensions	Clusters
Tweet	2472	5076	89
PriceRunner TVs	3564	2720	1280
TitleSet	11109	8079	152
SnippetSet	11109	18436	152
20 Newsgroups	20000	9887	20
Wines	11258	7173	88

Clustering algorithms

- 8 algorithms from 4 categories (see Table II):
  - Space Partitioning: k-Means, Mini-Batch k-Means.
  - Hierarchical: Agglomerative (Single Linkage), Agglomerative (Ward), BIRCH.
  - Spectral: Traditional Spectral Clustering algorithm.
  - Density-Based: DBSCAN, OPTICS.

TABLE II
CLUSTERING ALGORITHMS AND HYPER-PARAMETER SETTING

Clustering Algorithm	**Hyper-parameters**
k-means	Number of clusters: Actual. Max iterations: 200. Centroid initialization: k-means++.
MiniBatch k-means	Number of clusters: Actual. Max iterations: 200. Centroid initialization: k-means++. Batch size: 1024.
BIRCH	Number of clusters: Actual. Cluster radius threshold: 0.5. Max number of clusters in a node: 50.
Agglomerative Clustering	Number of clusters: Actual. Linkage: Complete. Distance measure: Euclidean.
Agglomerative (Ward)	Number of clusters: Actual. Linkage: Ward. Distance measure: Euclidean.
Spectral Clustering	Number of clusters: Actual. Affinity: RBF. γ: 1.0.
DBSCAN	ϵ: 0.5. Nearest Neighbors: 5. Distance measure: Euclidean.
OPTICS	ϵ: 0.5. Nearest Neighbors: 5. Distance measure: Euclidean.

Text preprocessing

- The input raw text was converted to lowercase.
- A word-level tokenizer converted each document into a bag of words.
- A simple regex was applied to remove punctuation.
- The WordNet lemmatizer was subsequently employed to convert each word to its meaningful base form.
- The two sets were individually vectorized by applying the well-known tf-idf transformation.
- L2 normalization of the generated vectors.

Truncated Singular Value Decomposition (TSVD)

- TSVD is a variant of Principal Component Analysis (PCA).
- Recall that PCA identifies the principal components by maximizing the variance of the projected data.
  - PCA is not feasible to sparse matrices.
- In contrast, TSVD does not center the data before computing the singular value decomposition.
  - It operates efficiently on sparse matrices.
- TSVD is often known as Latent Semantic Analysis (LSA).

Evaluation Measures (1)

- **Mutual Information (MI)** determines the similarity between two clusterings U and V:

$$MI(U,V) = \sum_{i=1}^{|V|} \sum_{j=1}^{|U|} \frac{|U_i \cap V_j|}{N} \log \frac{N|U_i \cap V_j|}{|U_i||V_j|}$$

- MI increases when the number of clusters is large, regardless of whether there is actually more information shared. **Adjusted Mutual Information (AMI)** confronts this problem:

$$AMI(U,V) = \frac{MI(U,V) - \mathbb{E}[MI(U,V)]}{\big(H(U) + H(V)\big)/2 - \mathbb{E}[MI(U,V)]}$$

- **Random Index (RI)**: the ratio between the correctly grouped pairs divided by the total number of pairs, ignoring permutations.
- **Adjusted Random Index (ARI)** corrects RI for chance:

$$ARI(U,V) = \frac{RI(U,V) - \mathbb{E}[RI(U,V)]}{\max RI(U,V) - \mathbb{E}[RI(U,V)]}$$

Evaluation Measures (2)

- **V-measure**, or **Normalized Mutual Information (NMI)** combines the metrics of cluster completeness C and homogeneity G:

$$NMI(U,V) = \mathcal{V} = \frac{2CG}{1 + C + G}$$

- **Completeness C**: the ability of an algorithm to place all the members of a class into same cluster.
- The **homogeneity G** of a cluster U indicates the purity of U; or, the ability of algorithm to avoid placing elements from different classes into the same cluster.

Results (Effectiveness, Tweet dataset)

Dataset	Samples	Dimensions	Clusters
Tweet	2472	5076	89
PriceRunner TVs	3564	2720	1280

- Tweet: The hierarchical methods BIRCH and Ward were the most accurate.
- Most methods exhibited a performance decrease of 1-9% for 10x dimensionality reduction.
- **The density-based methods, DBSCAN and OPTICS did not perform well.**
- **BIRCH does not tolerate excessive reductions of the feature space by more than one orders of magnitude.**

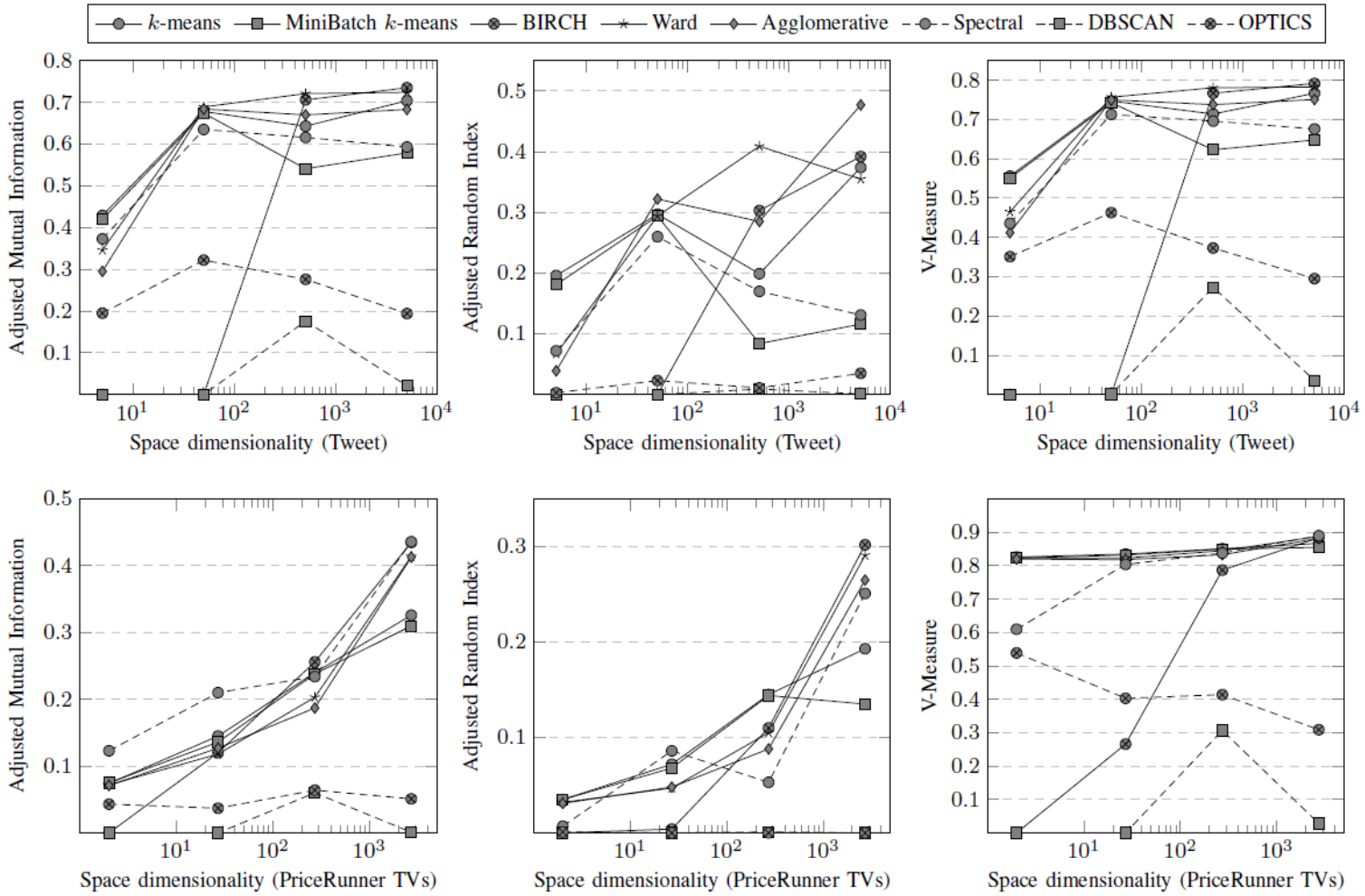

Fig. 1. Adjusted Mutual Info (left column), Adjusted Random Index (central column), and V-measure (right column) of the 8 clustering methods of Table II against input spaces of variable dimensionality (logarithmic scaling). The 3 diagrams at the top concern the Tweet dataset, whereas the 3 diagrams at the bottom concern PriceRunner TVs. The rightmost markers represent the original input spaces with all features included, i.e., without dimensionality reduction.

Results (Effectiveness, Tweet & PriceRunner)

Dataset	Samples	Dimensions	Clusters
Tweet	2472	5076	89
PriceRunner TVs	3564	2720	1280

- 2 orders of magnitude reduction → the performance degraded further.
- 3 orders of magnitude reduction (5 features) → the accuracy drops below acceptable levels.
- PriceRunner: The hierarchical and spectral methods performed well in the original feature space.
- In this dataset, even the smallest reduction in the number of features led to significant losses.
- In the reduced spaces, the Ward and Agglomerative methods were heavily affected.

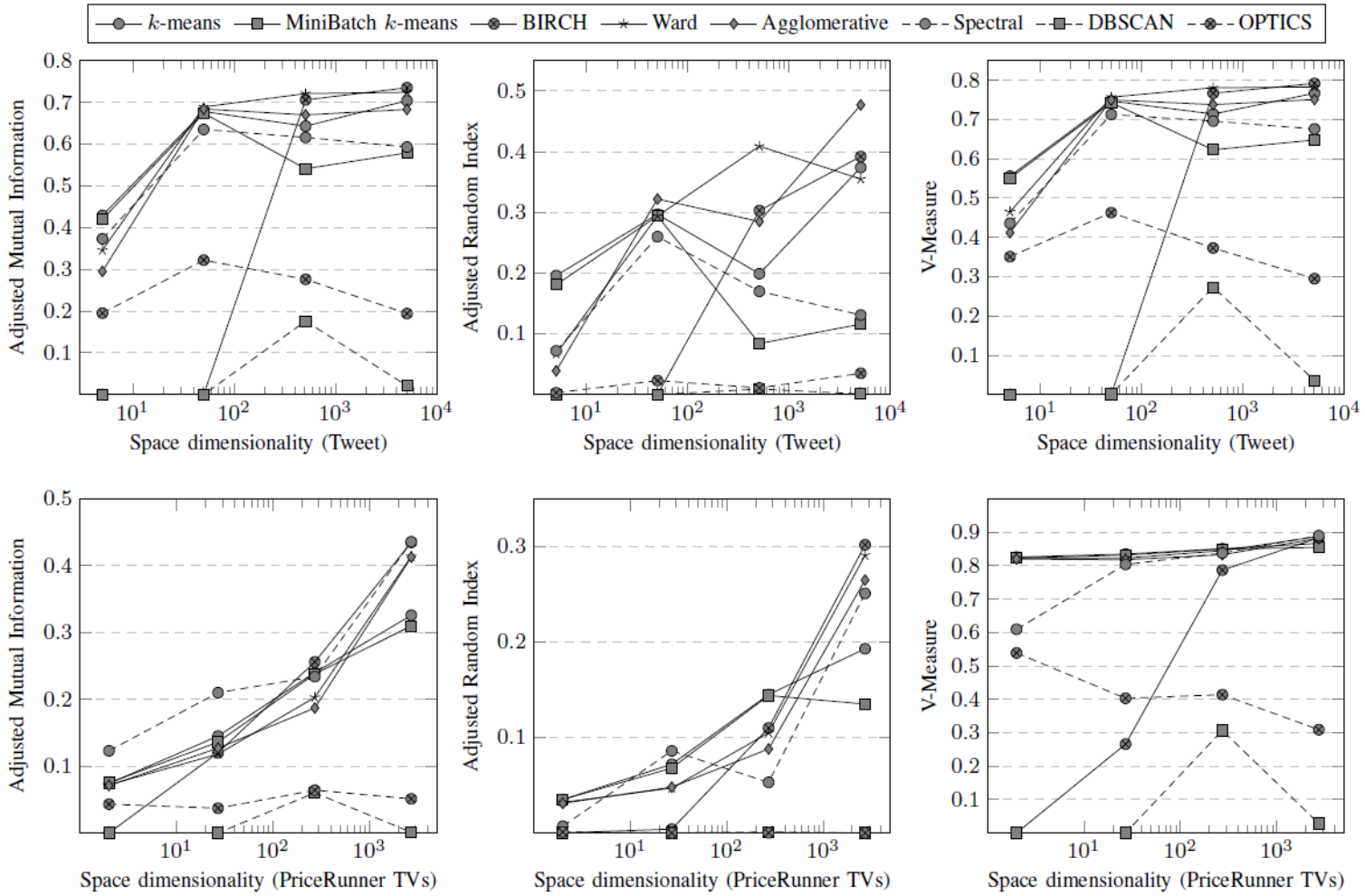

Fig. 1. Adjusted Mutual Info (left column), Adjusted Random Index (central column), and V-measure (right column) of the 8 clustering methods of Table II against input spaces of variable dimensionality (logarithmic scaling). The 3 diagrams at the top concern the Tweet dataset, whereas the 3 diagrams at the bottom concern PriceRunner TVs. The rightmost markers represent the original input spaces with all features included, i.e., without dimensionality reduction.

Results (Effectiveness, TitleSet)

Dataset	Samples	Dimensions	Clusters
TitleSet	11109	8079	152
20 Newsgroups	20000	9887	20

- Similar behavior to the Tweets dataset.
- BIRCH and Ward (**not** Agglomerative) were again the most accurate methods in the original feature space.
- For most methods, space reductions by 1 or 2 orders of magnitude led to no, or slight degradations of AMI and NMI.
- BIRCH is the most significant exception.
- Similarly to the Tweets dataset, the ARI measure provides a diverse impression.
- It seems that MiniBatch and Spectral clustering work better with fewer features.

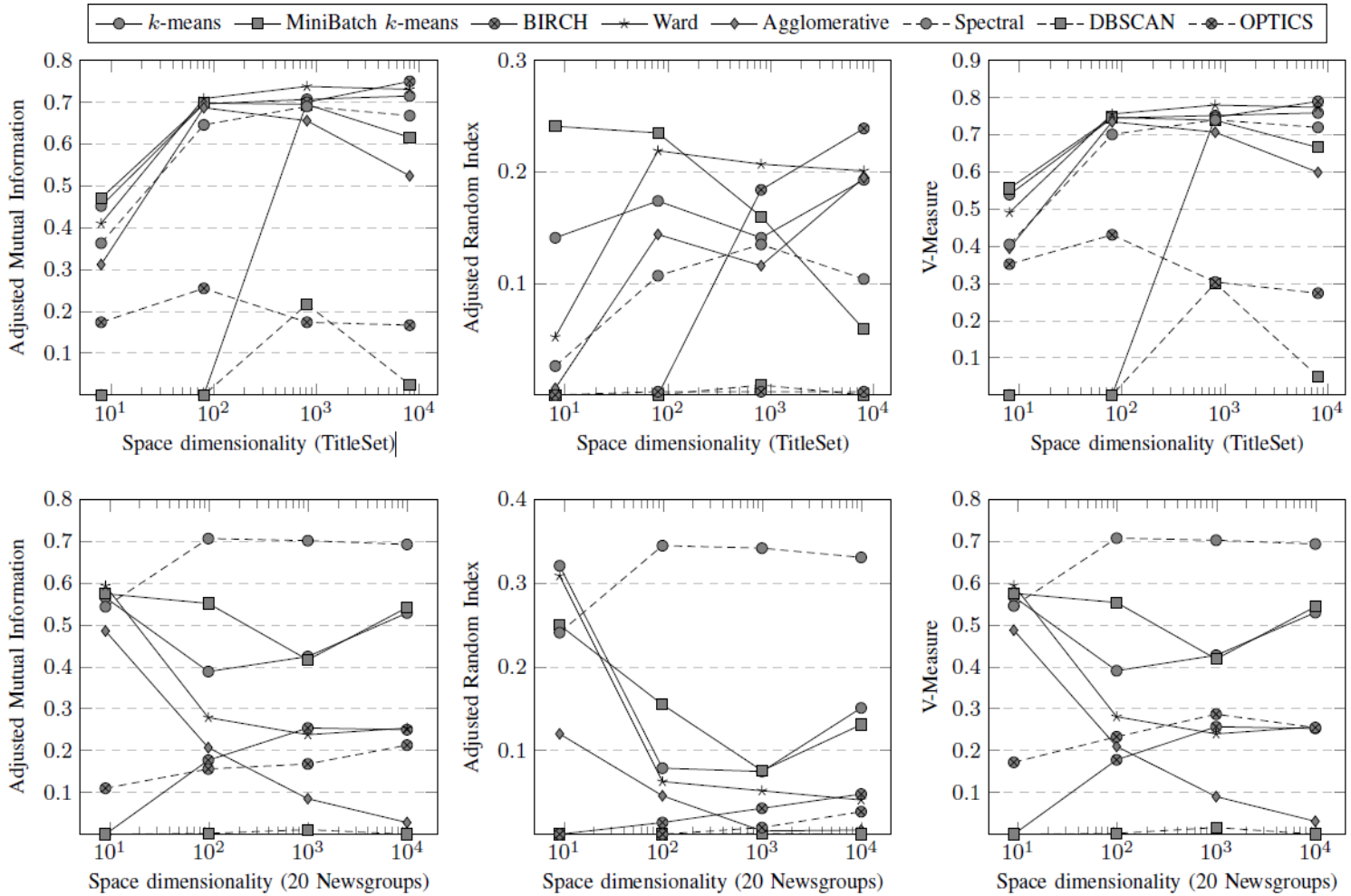

Fig. 2. Adjusted Mutual Info (left column), Adjusted Random Index (central column) and V-measure (right column) of the 8 clustering methods of Table II against input spaces of variable dimensionality (logarithmic scaling). The 3 diagrams at the top concern the TitleSet dataset, and the 3 diagrams at the bottom concern 20 Newsgroups. The rightmost markers represent the original input spaces with all features included, i.e., without dimensionality reduction.

Results (Effectiveness, 20 Newsgroups)

Dataset	Samples	Dimensions	Clusters
TitleSet	11109	8079	152
20 Newsgroups	20000	9887	20

- Best method: Spectral clustering in all dimensional spaces.
- The accuracy in this test was not affected by dimensionality reduction.
- With only a few exceptions, **Spectral clustering is robust to reductions of the feature space by one or two orders of magnitude.**

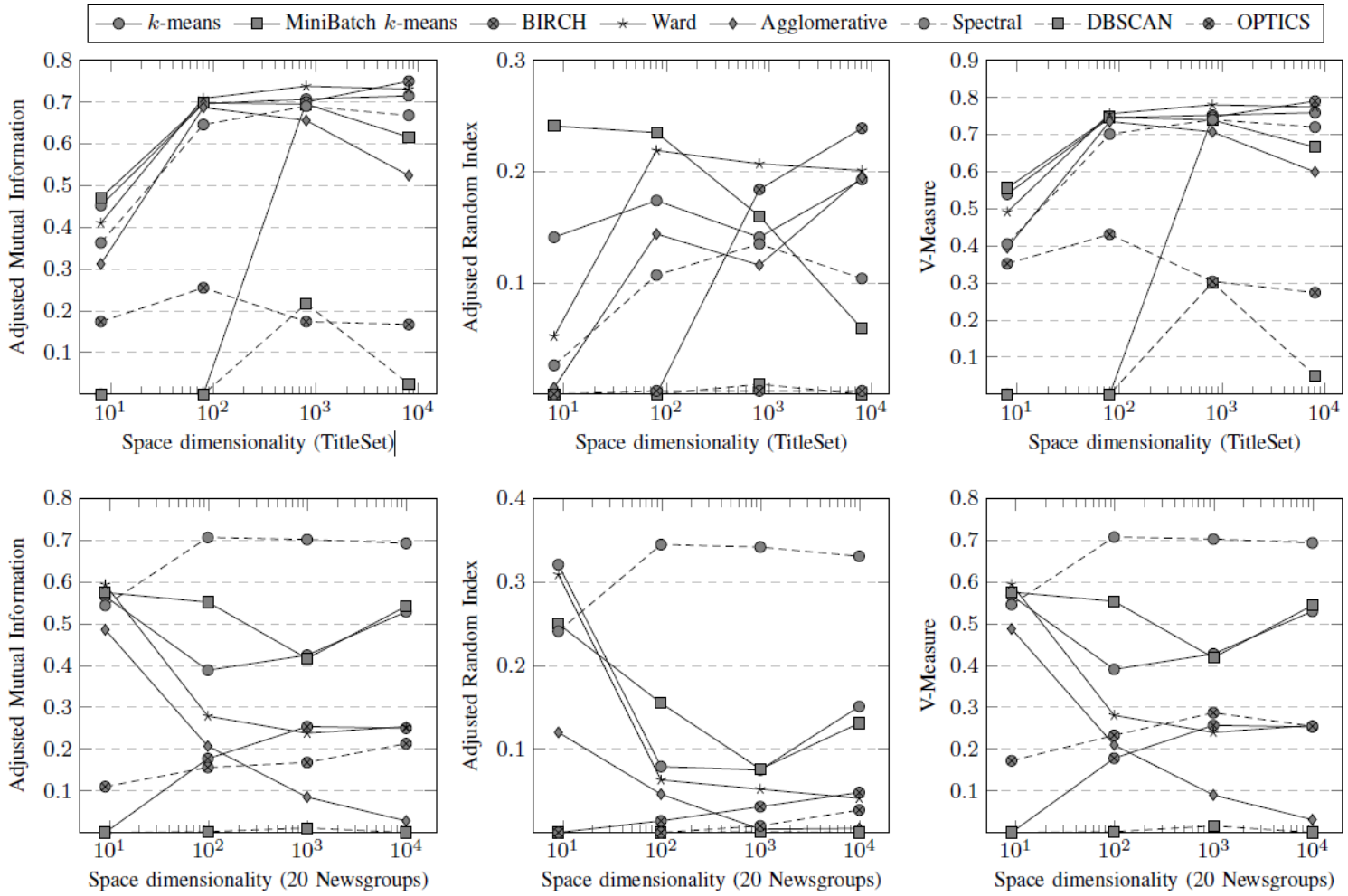

Fig. 2. Adjusted Mutual Info (left column), Adjusted Random Index (central column) and V-measure (right column) of the 8 clustering methods of Table II against input spaces of variable dimensionality (logarithmic scaling). The 3 diagrams at the top concern the TitleSet dataset, and the 3 diagrams at the bottom concern 20 Newsgroups. The rightmost markers represent the original input spaces with all features included, i.e., without dimensionality reduction.

Results (Effectiveness, SnippetSet)

Dataset	Samples	Dimensions	Clusters
SnippetSet	11109	18436	152
Wines	11258	7173	88

- The hierarchical methods (Agglomerative, BIRCH, Ward) and Spectral clustering were particularly effective in all spaces.
- The dimensionality reduction by two orders of magnitude led to small losses in performance (except from BIRCH).

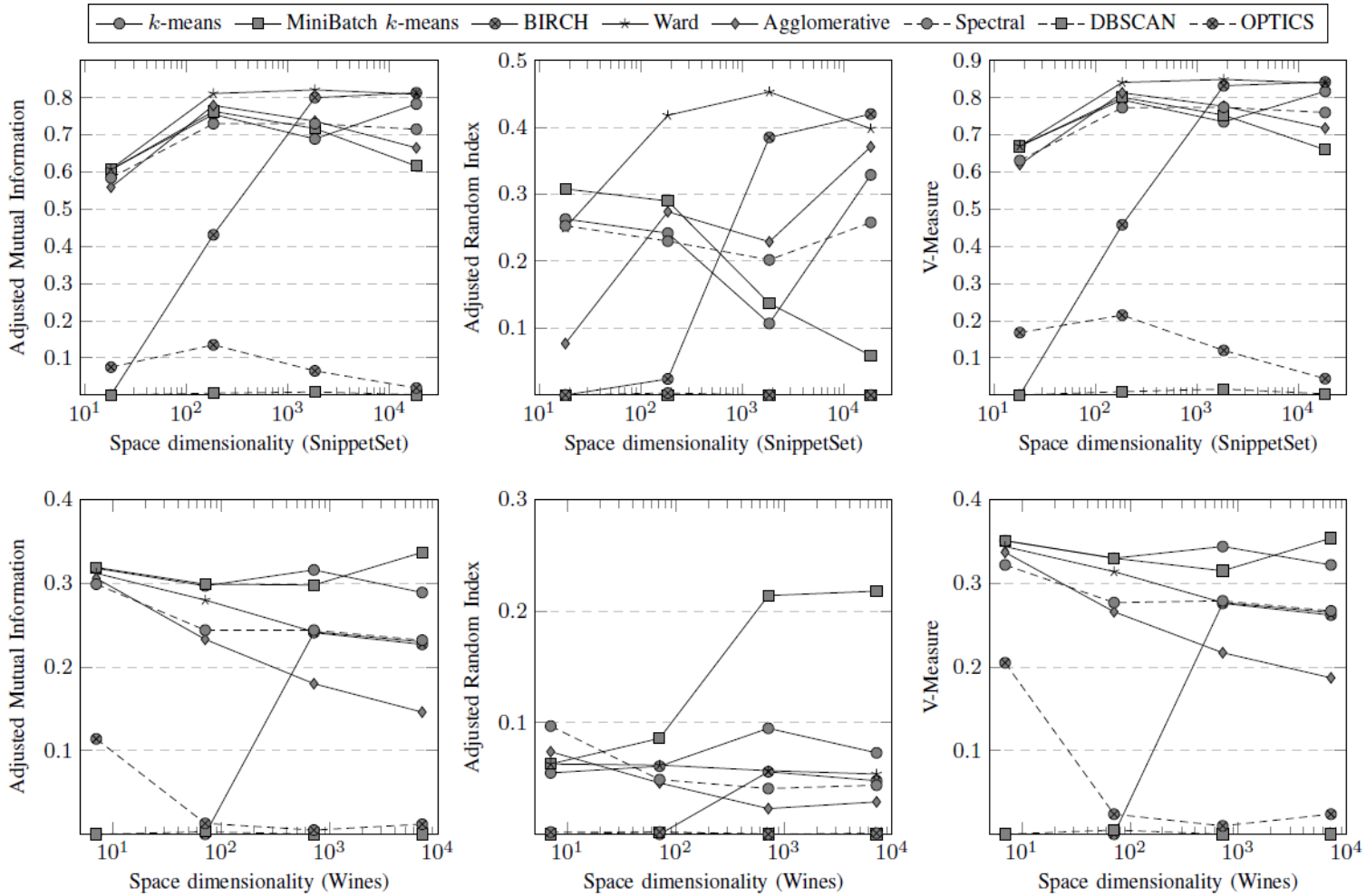

Fig. 3. Adjusted Mutual Info (left column), Adjusted Random Index (central column) and V-measure (right column) of the 8 clustering methods of Table II against input spaces of variable dimensionality (logarithmic scaling). The 3 diagrams at the top concern the SnippetSet dataset, and the 3 diagrams at the bottom concern Wines. The rightmost markers represent the original input spaces with all features included, i.e., without dimensionality reduction.

Results (Effectiveness, Wines)

Dataset	Samples	Dimensions	Clusters
SnippetSet	11109	18436	152
Wines	11258	7173	88

- The partitioning methods (k-Means, MiniBatch k-Means) were the most powerful.
- On the other hand, hierarchical methods were rather ineffective in this dataset.
- For aggressive reductions (2 or more orders of magnitude), the rates at which the performance degraded were mixed.
- k-Means and Spectral clustering were particularly stable, in contrast to BIRCH.

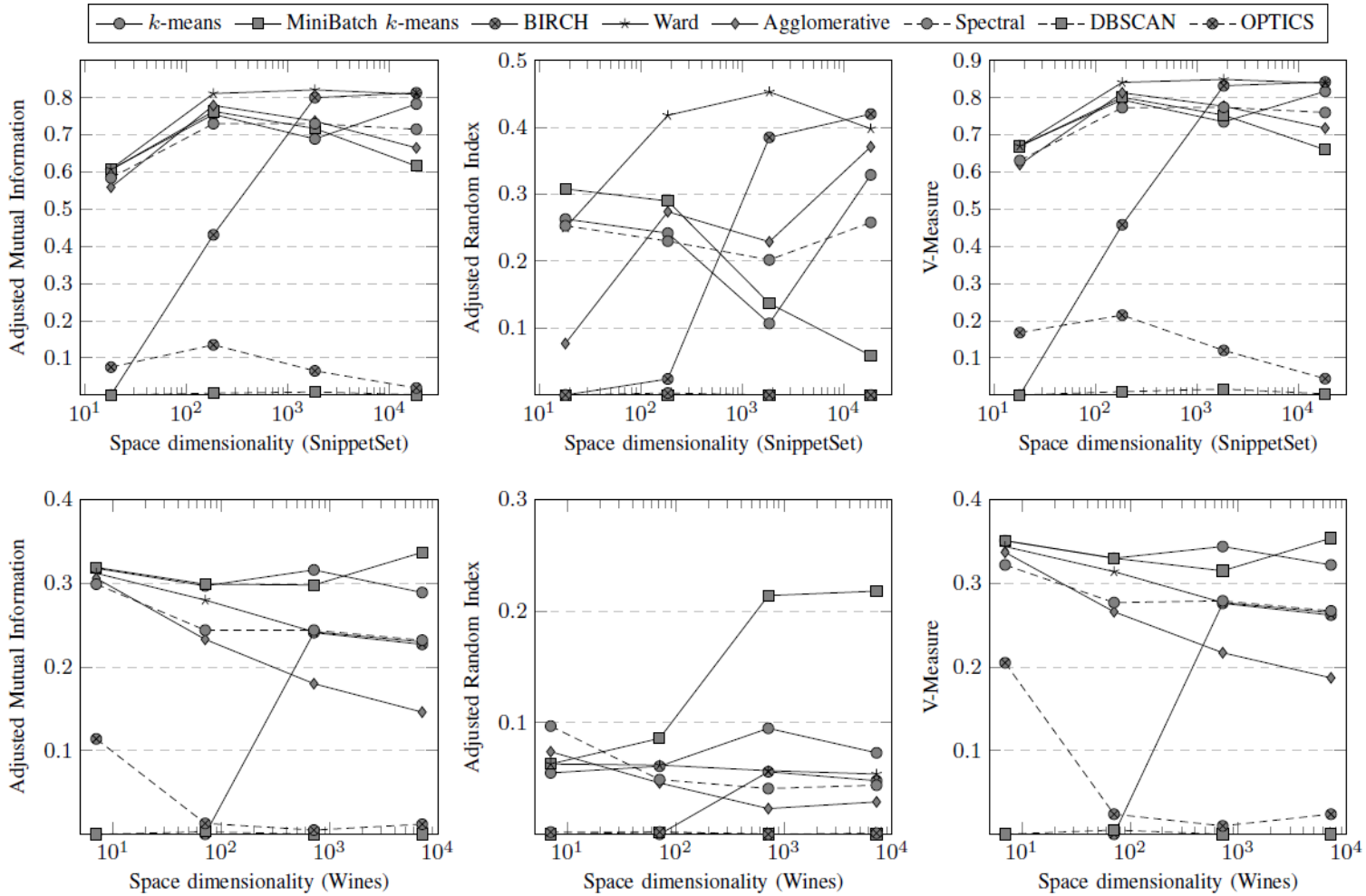

Fig. 3. Adjusted Mutual Info (left column), Adjusted Random Index (central column) and V-measure (right column) of the 8 clustering methods of Table II against input spaces of variable dimensionality (logarithmic scaling). The 3 diagrams at the top concern the SnippetSet dataset, and the 3 diagrams at the bottom concern Wines. The rightmost markers represent the original input spaces with all features included, i.e., without dimensionality reduction.

Results (Clustering Times, 1)

- The partitioning k-Means and MiniBatch k-Means were the fastest methods.
- Acceleration rate → almost linear to the size of the input space.
- The hierarchical methods were slower.
- Ward and Agglomerative had almost equal running times, but slightly slower than BIRCH.
- Their speed-up was almost linear to the reduction in the size of the input space.
- In contrast, Spectral clustering was not benefited from dimensionality reduction.

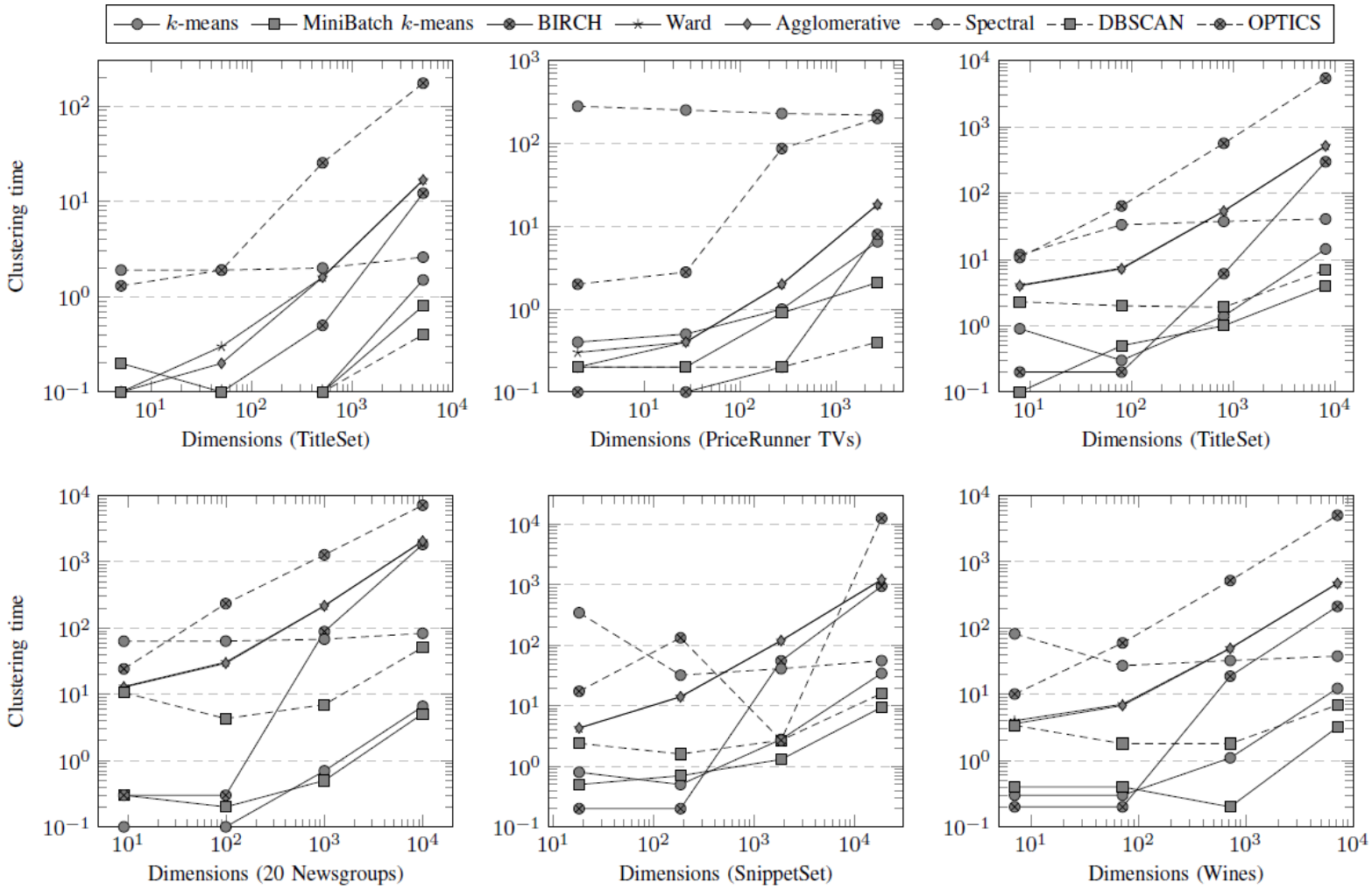

Fig. 4. Execution times of the 8 clustering methods of Table II against input spaces of variable dimensionality (logarithmic scaling). The rightmost markers represent the original input spaces with all features included, i.e., without dimensionality reduction.

Results (Clustering Times, 2)

- Dimensionality reduction has a positive impact on the running times of clustering algorithms.
- However, the execution acceleration is occasionally sublinear to the size of the input vector space.
- A dimensionality reduction by one, two, etc. orders of magnitude is not always translated to a speed-up by one, two, etc. orders of magnitude.

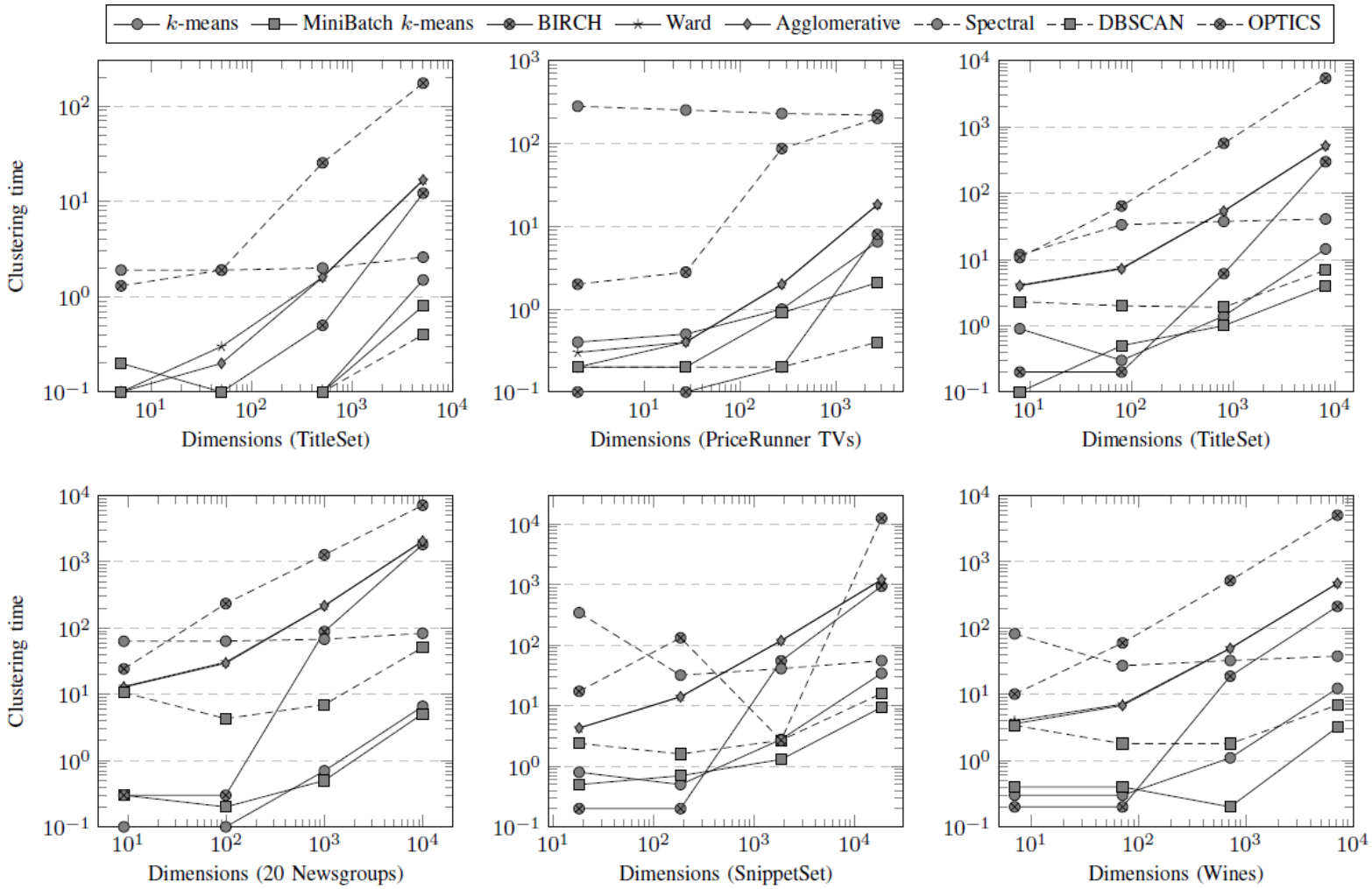

Fig. 4. Execution times of the 8 clustering methods of Table II against input spaces of variable dimensionality (logarithmic scaling). The rightmost markers represent the original input spaces with all features included, i.e., without dimensionality reduction.

Conclusions

- What did we learn from this experimental study?
- Regardless of the input vector space size, the density based methods (i.e. DBSCAN and OPTICS) did not perform well on text clustering tasks.
- BIRCH does not tolerate excessive dimensionality reductions, that is, by more than one orders of magnitude.
- Spectral clustering is robust to reductions of the feature space by one or two orders of magnitude.
- The impact of dimensionality reduction is sometimes unpredictable; there are cases where some algorithms perform better on reduced dimensional spaces.
- Dimensionality reduction benefits the running times of text clustering algorithms. In most cases, the smaller the dimensionality, the faster the clustering procedure is.

Thank you for watching

I would be happy to answer your questions.

Please send them to lakritidis@ihu.gr