Dimension Reduction and Iterative Consensus Clustering

Shaina Race, Carl Meyer

Southeastern Clustering and Ranking Workshop

August 24, 2009

Document Clustering

- For document clustering, we create a **term-document matrix**, **A**, as follows:

$$A_{m\times n} = \begin{array}{c} \\ \text{Term } 1 \\ \\ \text{Term } i \\ \\ \text{Term } m \end{array} \begin{array}{c} \begin{array}{cccc} \text{Doc } 1 & & \text{Doc } j & \text{Doc } n \end{array} \\ \begin{pmatrix} & & & | \\ & & & | \\ & & & | \\ - & - & - & f_{ij} \\ & & & \end{pmatrix} \end{array}$$

  Where $f_{i,j}$ is the frequency of term i in document j.
- Various types of term-weighting can be used in place of raw frequencies. For our experiments, we simply normalized the columns.
- Each column of **A** represents the coordinates of a document in the m-dimensional “term-space", where each standard basis vector represents one term from the dictionary.

Singular Value Decomposition (SVD)

- Decomposes $\mathbf{A} = \mathbf{U}\,\Sigma\mathbf{V}^T$ where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and Σ is a diagonal matrix of singular values.

Singular Value Decomposition (SVD)

- Decomposes $\mathbf{A} = \mathbf{U}\,\Sigma\mathbf{V}^T$ where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and Σ is a diagonal matrix of singular values.
- The truncated SVD yields the closest rank r approximation to $\mathbf{A}$ in the 2-norm.

$$\mathbf{a}_j \approx \sum_{i=1}^{r} [V^T]_{i,j}\sigma_i\mathbf{u}_i$$

Singular Value Decomposition (SVD)

- Decomposes $\mathbf{A} = \mathbf{U}\,\Sigma\mathbf{V}^T$ where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and Σ is a diagonal matrix of singular values.
- The truncated SVD yields the closest rank r approximation to $\mathbf{A}$ in the 2-norm.

$$\mathbf{a}_j \approx \sum_{i=1}^{r} [V^T]_{i,j}\sigma_i\mathbf{u}_i$$

- Thus, a column $\mathbf{v}_j$ of the truncated $\mathbf{V}^T$ is the coordinates of $\mathbf{a}_j$ once projected into the lower dimensional space spanned by the orthogonal basis

$$(\sigma_1\mathbf{u}_1, \sigma_2\mathbf{u}_2, \ldots \sigma_r\mathbf{u}_r)$$

.

Singular Value Decomposition (SVD)

- Decomposes $\mathbf{A} = \mathbf{U}\,\Sigma\mathbf{V}^T$ where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and Σ is a diagonal matrix of singular values.
- The truncated SVD yields the closest rank r approximation to $\mathbf{A}$ in the 2-norm.

$$\mathbf{a}_j \approx \sum_{i=1}^{r} [V^T]_{i,j}\sigma_i\mathbf{u}_i$$

- Thus, a column $\mathbf{v}_j$ of the truncated $\mathbf{V}^T$ is the coordinates of $\mathbf{a}_j$ once projected into the lower dimensional space spanned by the orthogonal basis

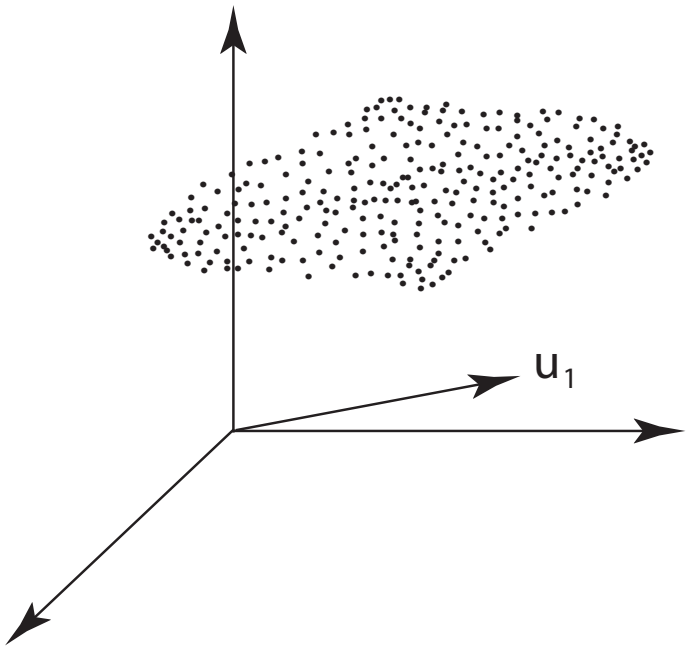
$$(\sigma_1\mathbf{u}_1, \sigma_2\mathbf{u}_2, \ldots \sigma_r\mathbf{u}_r)$$

.

- We'll use the columns of $\mathbf{V}^T$ as a lower dimensional representation of the columns of $\mathbf{A}$ for the purposes of clustering.

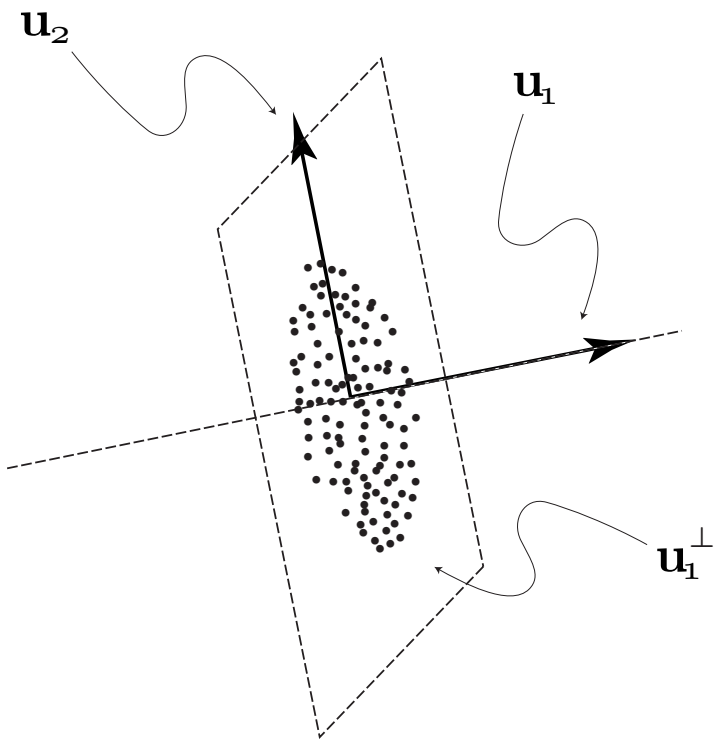
Geometry of Singular Vectors when A is centered

- The first left-hand singular vector, $\mathbf{u}_1$, of the centered matrix $\mathbf{C} = \mathbf{A} - \mu\mathbf{e}^T$ is the direction along which the variance of the data is maximal.



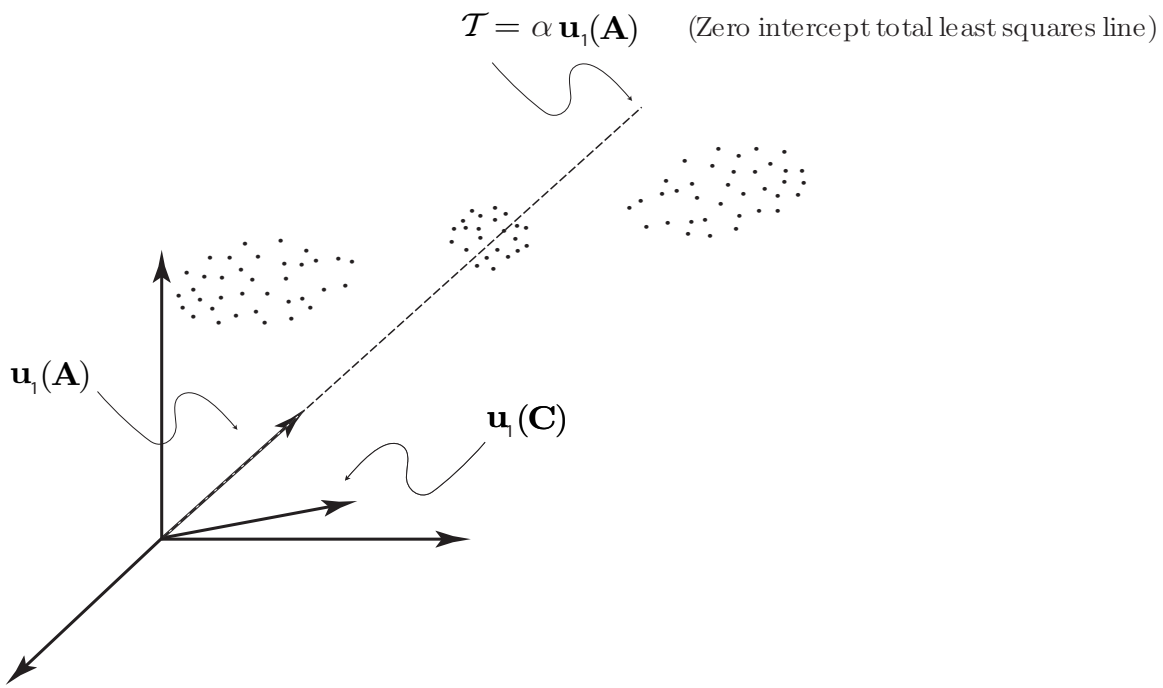
Geometry of Singular Vectors when A is centered

- The second left singular vector of $\mathbf{C}$, $\mathbf{u}_2$, is the direction orthogonal to $\mathbf{u}_1$ along which the variance is maximal.



Geometry of SVD when A is uncentered

- The first left singular vector of **A** is the direction of the least-squares line through the origin.



Principal Direction Divisive Partitioning (PDDP)

- Algorithm proposed by Daniel Boley at Univ. of MN in 2002

Principal Direction Divisive Partitioning (PDDP)

- Algorithm proposed by Daniel Boley at Univ. of MN in 2002
- Iterative process partitions data into 2 clusters with each iteration, based upon their projection onto the direction of maximal variance.

Principal Direction Divisive Partitioning (PDDP)

- Algorithm proposed by Daniel Boley at Univ. of MN in 2002
- Iterative process partitions data into 2 clusters with each iteration, based upon their projection onto the direction of maximal variance.
- PDDP can be adapted to use more than just the principal singular vector.

Principal Direction Divisive Partitioning (PDDP)

- Algorithm proposed by Daniel Boley at Univ. of MN in 2002
- Iterative process partitions data into 2 clusters with each iteration, based upon their projection onto the direction of maximal variance.
- PDDP can be adapted to use more than just the principal singular vector.
- We will often use the results from PDDP to seed the k-means algorithm with an initial guess

Illustration of one iteration of PDDP

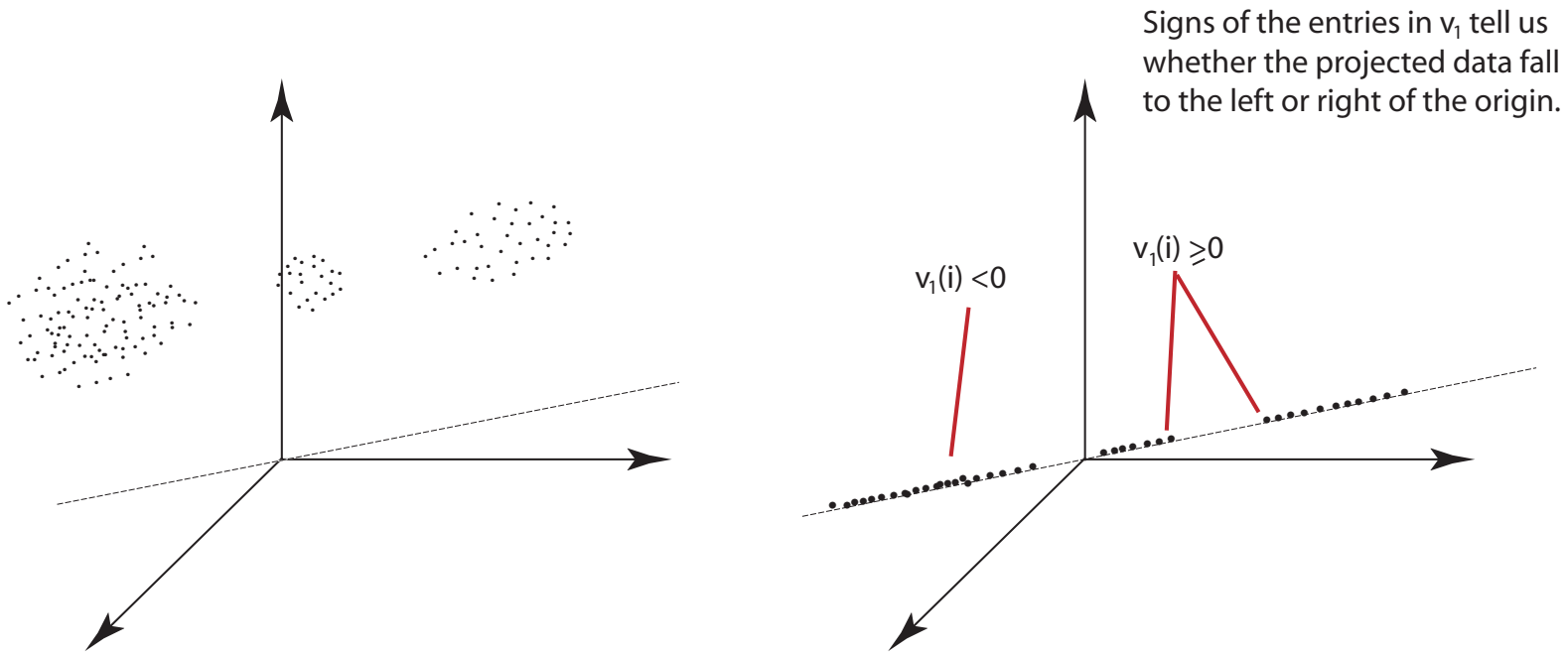


Figure: Data Cloud Projected onto the span of $\mathbf{u}_1(\mathbf{C})$

Nonnegative Matrix Factorization

- The **nonnegative matrix factorization** (NMF) seeks to decompose a nonnegative matrix into the product of two nonnegative matrices: $\mathbf{A}_{m\times n} = \mathbf{W}_{m\times r}\mathbf{H}_{r\times n}$.

Nonnegative Matrix Factorization

- The **nonnegative matrix factorization** (NMF) seeks to decompose a nonnegative matrix into the product of two nonnegative matrices: $\mathbf{A}_{m\times n} = \mathbf{W}_{m\times r}\mathbf{H}_{r\times n}$.
- The decomposition is created by solving the following nonlinear optimization problem:

$$\min \|\mathbf{A} - \mathbf{WH}\|_F^2 \text{ such that } \mathbf{W} \geq 0 \text{ and } \mathbf{H} \geq 0$$

Nonnegative Matrix Factorization

- The **nonnegative matrix factorization** (NMF) seeks to decompose a nonnegative matrix into the product of two nonnegative matrices: $\mathbf{A}_{m\times n} = \mathbf{W}_{m\times r}\mathbf{H}_{r\times n}$.
- The decomposition is created by solving the following nonlinear optimization problem:

$$\min \|\mathbf{A} - \mathbf{WH}\|_F^2 \text{ such that } \mathbf{W} \geq 0 \text{ and } \mathbf{H} \geq 0$$

- The inner dimension of the factorization, r, must be input by the user.

Nonnegative Matrix Factorization

- The **nonnegative matrix factorization** (NMF) seeks to decompose a nonnegative matrix into the product of two nonnegative matrices: $\mathbf{A}_{m\times n} = \mathbf{W}_{m\times r}\mathbf{H}_{r\times n}$.
- The decomposition is created by solving the following nonlinear optimization problem:

$$\min \|\mathbf{A} - \mathbf{WH}\|_F^2 \text{ such that } \mathbf{W} \geq 0 \text{ and } \mathbf{H} \geq 0$$

- The inner dimension of the factorization, r, must be input by the user.
- The result is an additive, parts-based approximation to each data column $\mathbf{a}_j$ in the form of a linear combination of “feature" vectors, $\mathbf{w}_i$, as follows:

Nonnegative Matrix Factorization

- The **nonnegative matrix factorization** (NMF) seeks to decompose a nonnegative matrix into the product of two nonnegative matrices: $\mathbf{A}_{m\times n} = \mathbf{W}_{m\times r}\mathbf{H}_{r\times n}$.
- The decomposition is created by solving the following nonlinear optimization problem:

$$\min \|\mathbf{A} - \mathbf{WH}\|_F^2 \text{ such that } \mathbf{W} \geq 0 \text{ and } \mathbf{H} \geq 0$$

- The inner dimension of the factorization, r, must be input by the user.
- The result is an additive, parts-based approximation to each data column $\mathbf{a}_j$ in the form of a linear combination of "feature" vectors, $\mathbf{w}_i$, as follows:

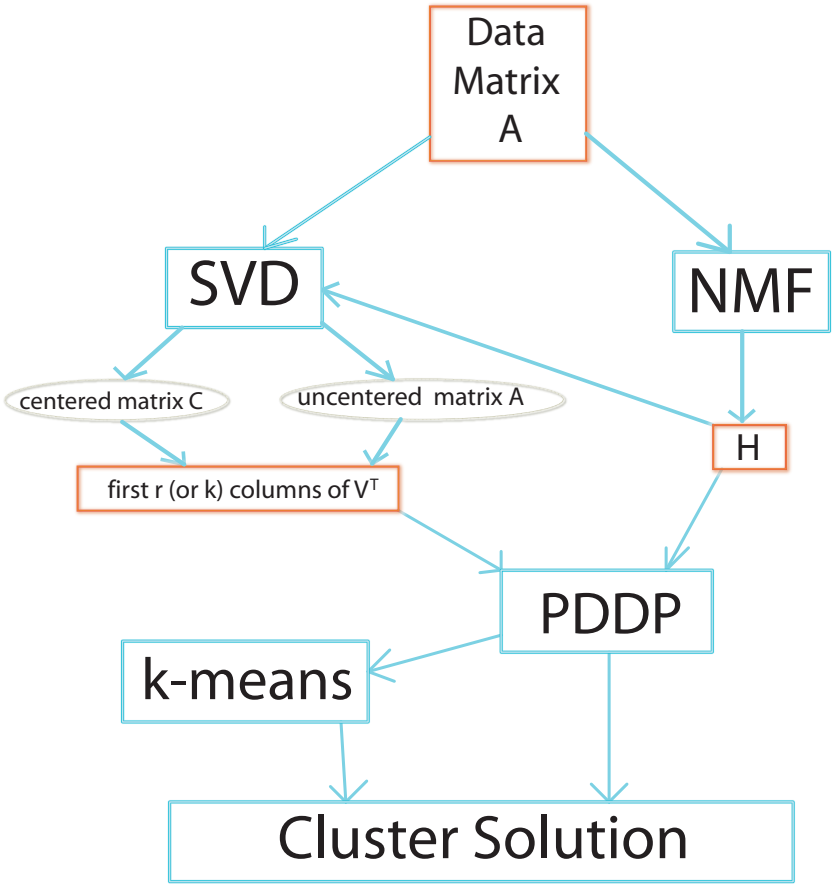
$$\mathbf{a}_j \approx \sum_{i=1}^{r} \mathbf{h}_{i,j}\mathbf{w}_i$$

NMF for Dimension Reduction

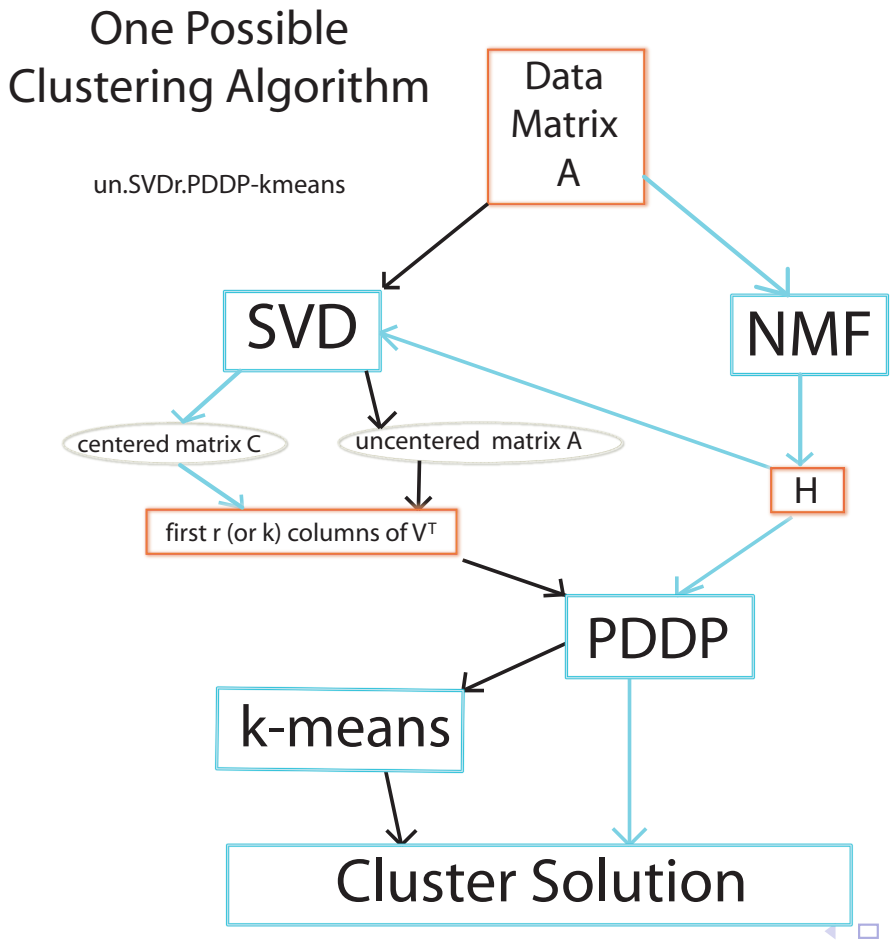
- Columns of H represent the coordinates of each document after projection into the lower dimensional “feature-space" spanned by the columns of W.
- We’ll use the columns of **H** as a lower dimensional representation of the columns of **A** for the purposes of clustering.

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_m \end{pmatrix} \longrightarrow \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ \vdots \\ h_r \end{pmatrix}$$

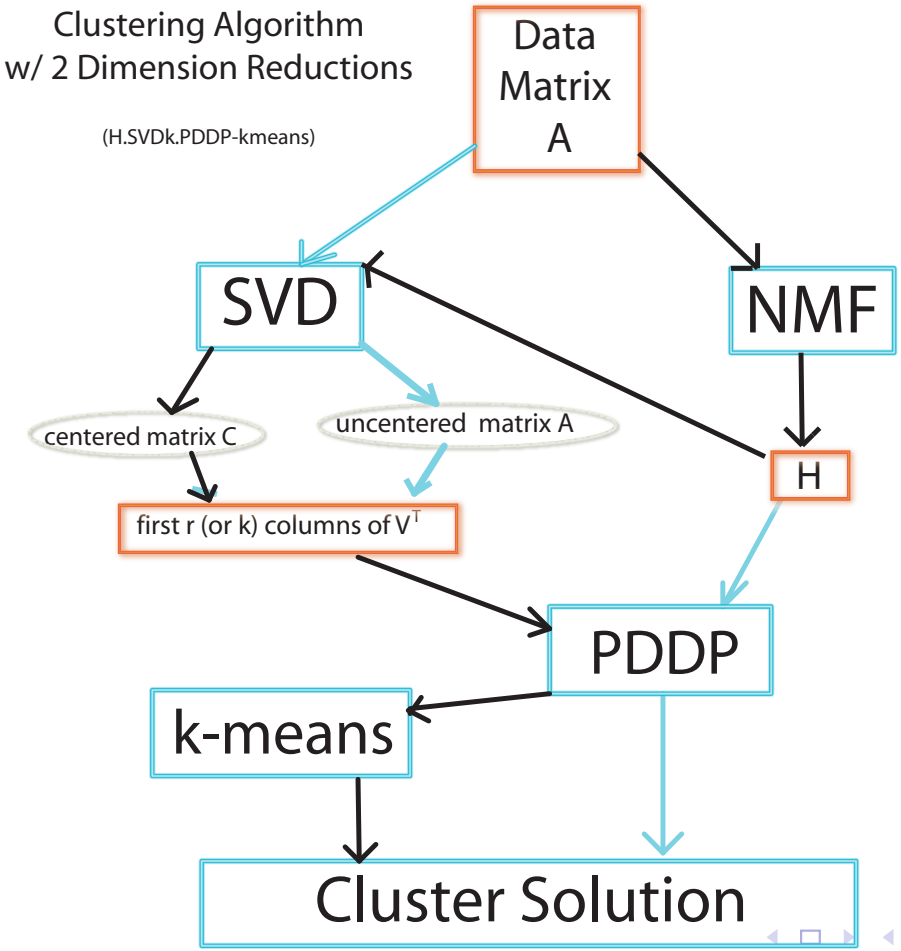
New Algorithms out of Old



New Algorithms out of Old



New Algorithms out of Old



Consensus Clustering

- Since no single algorithm will perform better than all others on a given class of data, we propose using several algorithms to find agreement upon clusters.

Consensus Clustering

- Since no single algorithm will perform better than all others on a given class of data, we propose using several algorithms to find agreement upon clusters.
- We create an adjacency matrix for each clustering, whose $(i, j)^{th}$ entry is 1 if $\mathbf{a}_i$ and $\mathbf{a}_j$ were clustered together and 0 otherwise.

Consensus Clustering

- Since no single algorithm will perform better than all others on a given class of data, we propose using several algorithms to find agreement upon clusters.
- We create an adjacency matrix for each clustering, whose $(i,j)^{th}$ entry is 1 if $\mathbf{a}_i$ and $\mathbf{a}_j$ were clustered together and 0 otherwise.
- We sum the adjacency matrices from various algorithms to create a **consensus matrix**, **M** whose $(i,j)^{th}$ entry reveals the number of times $\mathbf{a}_i$ and $\mathbf{a}_j$ were clustered together.

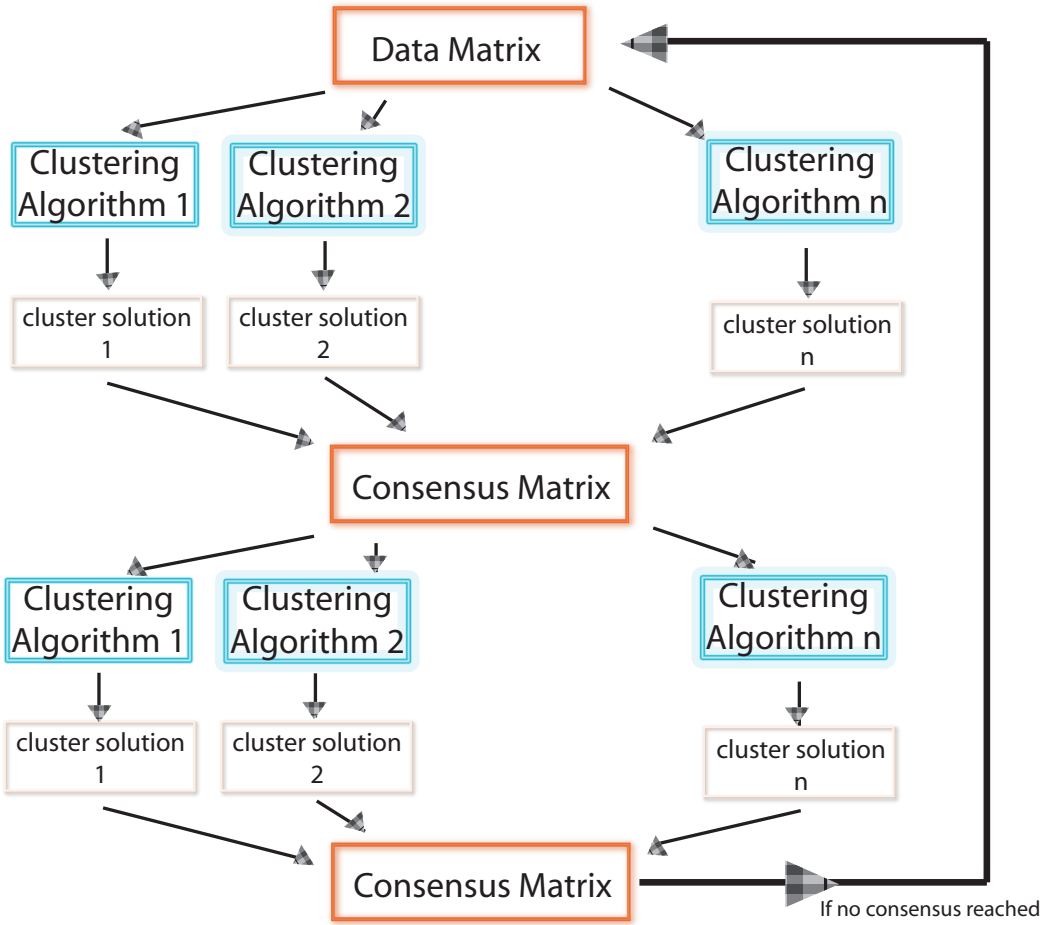
Consensus Clustering

- Since no single algorithm will perform better than all others on a given class of data, we propose using several algorithms to find agreement upon clusters.
- We create an adjacency matrix for each clustering, whose $(i,j)^{th}$ entry is 1 if $\mathbf{a}_i$ and $\mathbf{a}_j$ were clustered together and 0 otherwise.
- We sum the adjacency matrices from various algorithms to create a **consensus matrix**, **M** whose $(i,j)^{th}$ entry reveals the number of times $\mathbf{a}_i$ and $\mathbf{a}_j$ were clustered together.
- Entries in the consensus matrix that are below a certain tolerance may be changed to zero.

Consensus Clustering

- Since no single algorithm will perform better than all others on a given class of data, we propose using several algorithms to find agreement upon clusters.
- We create an adjacency matrix for each clustering, whose $(i,j)^{th}$ entry is 1 if $\mathbf{a}_i$ and $\mathbf{a}_j$ were clustered together and 0 otherwise.
- We sum the adjacency matrices from various algorithms to create a **consensus matrix**, **M** whose $(i,j)^{th}$ entry reveals the number of times $\mathbf{a}_i$ and $\mathbf{a}_j$ were clustered together.
- Entries in the consensus matrix that are below a certain tolerance may be changed to zero.
- This consensus matrix is then clustered using the same algorithms to see if the algorithms will agree upon a solution.

Iterating the Consensus Process



Medlars/Cranfield/CISI - Medical and Scientific Abstracts - 4000 docs/11000 terms -$k = 3$ clusters

Accuracies for Med/Cran/CISI after Dimension Reduction to $r = 3$

Algorithm	r=k=3	Consensus 1	Consensus 2	Consensus 3
NMF Basic	0.70084811	0.89771267	0.9306091	0.89719866
PDDP	0.83012079	0.89437163	0.89437163	0.89719866
PDDP-kmeans	0.96376253	0.89282961	0.89282961	0.89719866
SVDr-PDDP-kmeans	0.81624261	0.72834747	0.769211	0.89719866
un.SVDr-PDDP-kmeans	0.75970188	0.89719866	0.89719866	0.89719866
H-PDDP	0.59650475	0.89437163	0.89874068	0.89719866
H-PDDP-kmeans	0.71626831	0.89719866	0.89719866	0.89719866
H-SVDk-PDDP-kmeans	0.8234387	0.89334361	0.89334361	0.9308661
H-un.SVDk-PDDP-kmeans	0.71446929	0.9308661	0.9308661	0.89719866

Medlars/Cranfield/CISI - Medical and Scientific Abstracts - 4000 doc, 11000 terms - $k = 3$ clusters

Accuracies for Med/Cran/CISI after dimension reduction to $r = 15$

Algorithm	r=15	Consensus 1	Consensus 2	Consensus 3
NMF Basic	[]	0.96530455	0.95334439	0.96453354
PDDP	0.83012079	0.94962735	0.94962735	0.96376253
PDDP-kmeans	0.96376253	0.96530455	0.96530455	0.96453354
SVDr-PDDP-kmeans	0.94500129	0.71652531	0.61783603	0.73477255
un.SVDr-PDDP-kmeans	0.78026214	0.61578001	0.92572603	0.83320483
H-PDDP	0.86096119	0.94962735	0.97044462	0.96376253
H-PDDP-kmeans	0.97584169	0.96530455	0.96556155	0.96453354
H-SVDk-PDDP-kmeans	0.86533025	0.96067849	0.95451041	0.96530455
H-un.SVDk-PDDP-kmeans	0.60164482	0.96530455	0.96453354	0.96453354

Benchmark Data Set Collected by Sinka and Corne

- 11000 documents proposed as a benchmark collection for document clustering.

Benchmark Data Set Collected by Sinka and Corne

- 11000 documents proposed as a benchmark collection for document clustering.
- Documents pertain to 4 broad topics (banking/finance, programming, science, and sport)

Benchmark Data Set Collected by Sinka and Corne

- 11000 documents proposed as a benchmark collection for document clustering.
- Documents pertain to 4 broad topics (banking/finance, programming, science, and sport)
- Each topic contains 2 or 3 subtopics (commercial banks, insurance agencies, java, astronomy, biology, etc).

Benchmark Data Set Collected by Sinka and Corne

- 11000 documents proposed as a benchmark collection for document clustering.
- Documents pertain to 4 broad topics (banking/finance, programming, science, and sport)
- Each topic contains 2 or 3 subtopics (commercial banks, insurance agencies, java, astronomy, biology, etc).
- Documents were extracted automatically from the web.

Benchmark Data Set Collected by Sinka and Corne

- 11000 documents proposed as a benchmark collection for document clustering.
- Documents pertain to 4 broad topics (banking/finance, programming, science, and sport)
- Each topic contains 2 or 3 subtopics (commercial banks, insurance agencies, java, astronomy, biology, etc).
- Documents were extracted automatically from the web.
  - Some long detailed articles
  - Some just list of words, addresses, or links.
  - $\rightarrow$ Noisy Data!

Benchmark Data - subset BCFG - $k = 4$ clusters

Cluster Accuracies for BenchmarkBCFG after Dimension Reduction to $r = 4$

Algorithm	r=4	Consensus 1	Consensus 2	Consensus 3
NMF Basic	0.62725	0.69275	0.57725	0.69425
PDDP	0.4505	0.67775	0.69325	0.69425
PDDP-kmeans	0.34025	0.69325	0.69375	0.69425
SVDr-PDDP-kmeans	0.69825	0.549	0.51575	0.69425
un.SVDr-PDDP-kmeans	0.74725	0.69475	0.6945	0.69425
H-PDDP	0.582	0.67775	0.5665	0.69425
H-PDDP-kmeans	0.65775	0.69275	0.5745	0.69425
H-SVDk-PDDP-kmeans	0.66125	0.67075	0.695	0.69425
H-un.SVDk-PDDP-kmeans	0.6825	0.69475	0.57825	0.69425

Benchmark Data - subset BCFG-$k = 4$ clusters

Experiment 3: Cluster Accuracies for BenchmarkBCFG after Dimension Reduction to $r = 10$

Algorithm	r=10	Consensus 1	Consensus 2	Consensus 3	Consensus 5
NMF Basic		0.74725	0.75225	0.58025	0.74775
PDDP	0.4505	0.74275	0.75125	0.74775	0.74775
PDDP-kmeans	0.34025	0.75275	0.748	0.75225	0.74775
SVDr-PDDP-kmeans	0.71975	0.5945	0.49725	0.6395	0.72225
un.SVDr-PDDP-kmeans	0.67325	0.703	0.51575	0.52225	0.63225
H-PDDP	0.71	0.74275	0.752	0.74775	0.74775
H-PDDP-kmeans	0.7635	0.74725	0.75225	0.58025	0.74775
H-SVDk-PDDP-kmeans	0.7255	0.736	0.748	0.7455	0.76525
H-un.SVDk-PDDP-kmeans	0.788	0.74975	0.748	0.58	0.74775

- The choices for the size of dimension reduction, r, and the various combinations of algorithms produce hundreds of clusterings for the consensus approach.
- Consensus Clustering shows potential as a technique to determine a final clustering solution through many different algorithms.
- Although the final clustering solution determined through Consensus Clustering is not guaranteed to be optimal, experiments suggest that the technique provides a solution that is well above the average of the algorithms used.