Clustering and Non-negative Matrix Factorization

Presented by Mohammad Sajjad Ghaemi

DAMAS LAB,
Computer Science and Software Engineering Department,
Laval University

12 April 2013

Outline

- **What is clustering ?**
- **NMF overview**
- **Cost functions and multiplicative algorithms**
- **A geometric interpretation of NMF**
- **r-separable NMF**

What is clustering ?

According to the label information we have 3 categories of learning

- **Supervised learning**
  Learning from labeled data.

What is clustering ?

According to the label information we have 3 categories of learning

- **Supervised learning**
  Learning from labeled data.

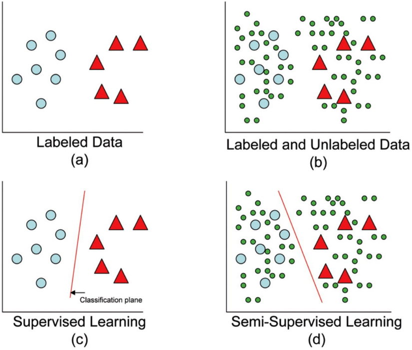
- **Semi-supervised learning**
  Learning from both labeled and unlabeled data.

What is clustering ?

According to the label information we have 3 categories of learning

- **Supervised learning**
  Learning from labeled data.
- **Semi-supervised learning**
  Learning from both labeled and unlabeled data.
- **Unsupervised learning**
  Learning from unlabeled data.

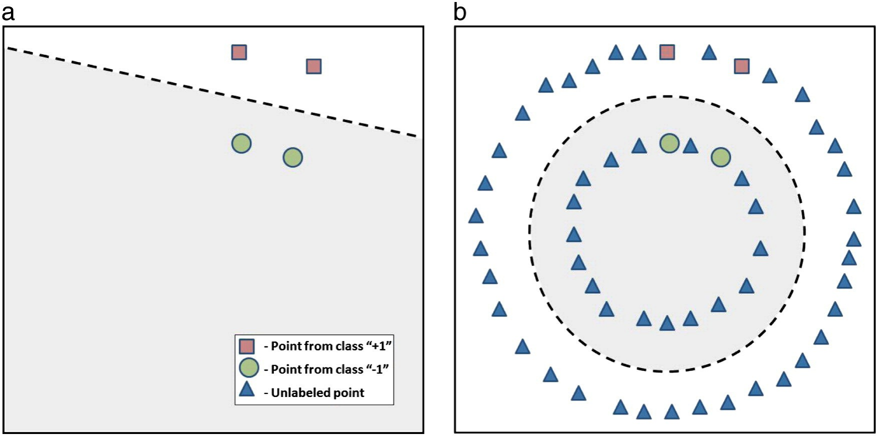
Label information



Taken from [Jain,2010]

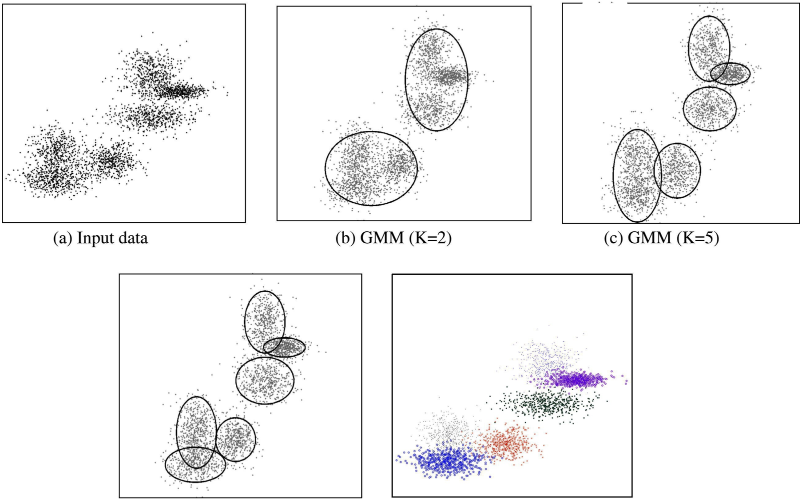
[Jain,2010] Jain, A.K., 2010. Data clustering : 50 years beyond k-means. Pattern Recognition Letters 31, 651666.

Semi-supervised learning and manifold assumption



Taken from [Jain,2010]

Unsupervised learning or clustering



(a) Input data (b) GMM (K=2) (c) GMM (K=5)

(d) GMM (K=6) (e) True labels, K = 6

Taken from [Jain,2010]

What is clustering ?

- Clustering is grouping similar objects together.

What is clustering ?

- Clustering is grouping similar objects together.
- Clusterings are usually not "right" or "wrong", different clusterings/clustering criteria can reveal different things about the data.

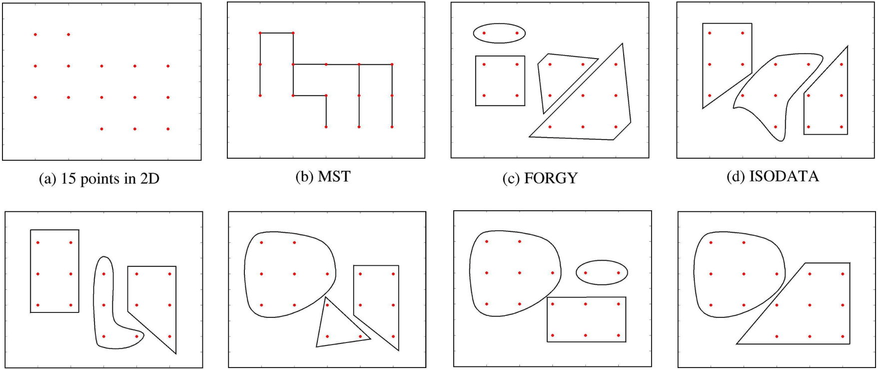
What is clustering ?

- Clustering is grouping similar objects together.
- Clusterings are usually not "right" or "wrong", different clusterings/clustering criteria can reveal different things about the data.
- There is no objectively "correct" clustering algorithm, but "clustering is in the eye of the beholder".

What is clustering ?

- Clustering is grouping similar objects together.
- Clusterings are usually not "right" or "wrong", different clusterings/clustering criteria can reveal different things about the data.
- There is no objectively "correct" clustering algorithm, but "clustering is in the eye of the beholder".
- Clustering algorithms :
  - Employ some notion of distance between objects.
  - Have an explicit or implicit criterion defining what a good cluster is.
  - Heuristically optimize that criterion to determine the clustering.

Comparing various clustering algorithms



(a) 15 points in 2D (b) MST (c) FORGY (d) ISODATA (e) WISH (f) CLUSTER (g) Complete-link (h) JP

Taken from [Jain,2010]

Outline

- **What is clustering ?**
- **NMF overview**
- **Cost functions and multiplicative algorithms**
- **A geometric interpretation of NMF**
- **r-separable NMF**

Matrix factorization

- **NMF (Non-negative Matrix Factorization)**

  Question :
  Given a non-negative matrix V, find non-negative matrix factors W and H,

$$V \approx WH$$

Matrix factorization

- **NMF (Non-negative Matrix Factorization)**

  Question :
  Given a non-negative matrix V, find non-negative matrix factors W and H,

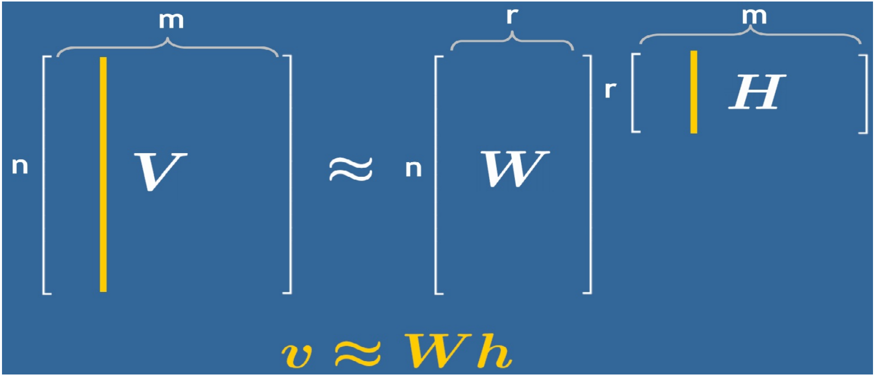
  $$V \approx WH$$

  Answer :
  Non-negative Matrix Factorization (NMF)

  Advantage of non-negativity Interpretability

- NMF is **NP-hard**

Matrix factorization



Matrix factorization

- Generally, factorization of matrices is not unique
  - Principal Component Analysis
  - Singular Value Decomposition
  - Nyström Method

Matrix factorization

- Generally, factorization of matrices is not unique
  - Principal Component Analysis
  - Singular Value Decomposition
  - Nyström Method

- Non-negative Matrix Factorization differs from the above methods.

Matrix factorization

- Generally, factorization of matrices is not unique
  - Principal Component Analysis
  - Singular Value Decomposition
  - Nyström Method
- Non-negative Matrix Factorization differs from the above methods.
- NMF enforces the constraint that the factors must be non-negative.

Matrix factorization

- Generally, factorization of matrices is not unique
  - Principal Component Analysis
  - Singular Value Decomposition
  - Nyström Method
- Non-negative Matrix Factorization differs from the above methods.
- NMF enforces the constraint that the factors must be non-negative.
- All elements must be equal to or greater than zero.

Matrix factorization

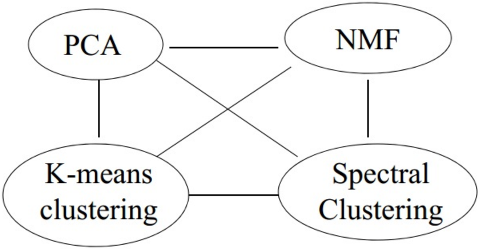
- Is there any unique solution to the NMF problem ?

Matrix factorization

- Is there any unique solution to the NMF problem ?
-
$$V \approx WD^{-1}DH$$
- NMF has the drawback of being highly ill-posed.

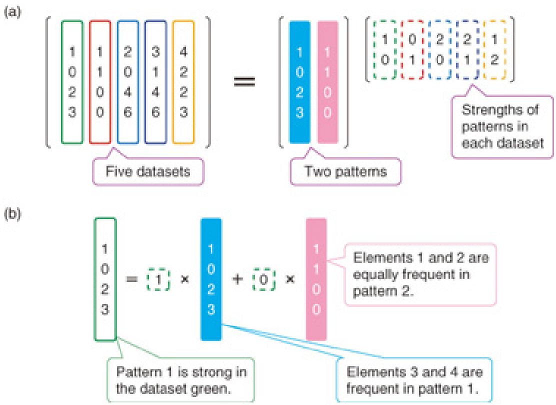
NMF is interesting because it does data clustering

Data Clustering = Matrix Factorizations



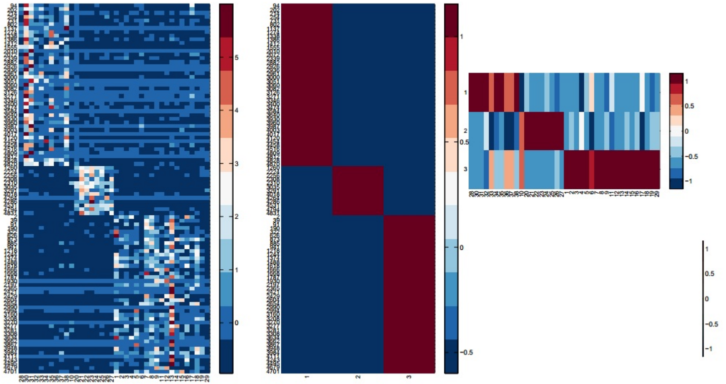
Many unsupervised learning methods are closely related in a simple way (Ding, He, Simon, SDM 2005).

Numerical example



taken from Katsuhiko Ishiguro, et al. Extracting Essential Structure from Data.

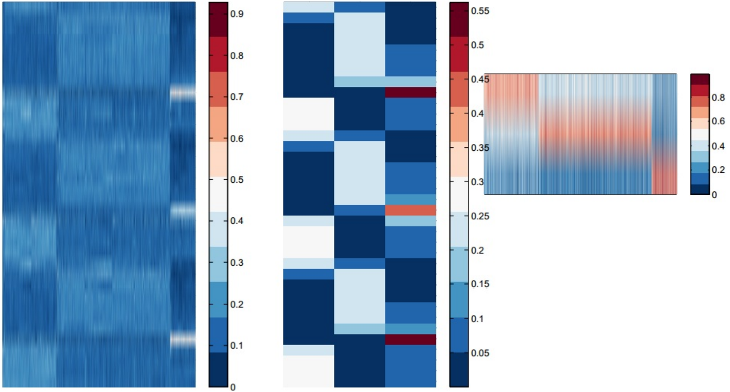
Heat map of NMF on the gene expression data



The left is the gene expression data where each column corresponds to a sample, the middle is the basis matrix, and the right is the coefficient matrix.

taken from Yifeng Li, et al. The Non-Negative Matrix Factorization Toolbox for Biological Data Mining

Heat map of NMF clustering on a yeast metabolic



The left is the gene expression data where each column corresponds to a gene, the middle is the basis matrix, and the right is the coefficient matrix.

taken from Yifeng Li, et al. The Non-Negative Matrix Factorization Toolbox for Biological Data Mining

Outline

- **What is clustering ?**
- **NMF overview**
- **Cost functions and multiplicative algorithms**
- **A geometric interpretation of NMF**
- **r-separable NMF**

How to solve it

Two conventional and convergent algorithms

- Square of the Euclidean distance

$$||A - B||^2 = \sum_{ij} (A_{ij} - B_{ij})^2$$

How to solve it

Two conventional and convergent algorithms

- Square of the Euclidean distance

$$||A - B||^2 = \sum_{ij} (A_{ij} - B_{ij})^2$$

- Generalized Kullback-Leibler divergence

$$D(A||B) = \sum_{ij} (A_{ij} log \frac{A_{ij}}{B_{ij}} - A_{ij} + B_{ij})$$

How to minimize it

- Minimize $||V - WH||^2$ or $D(V||WH)$
- Convex in W only or H only (not convex in both variables)
- Goal-finding local minima (tough to get global minima)
- Gradient descent ?
  - Slow convergence
  - Sensitive to the step size
  - inconvenient for large data

Cost functions and gradient based algorithm for square Euclidean distance

- Minimize $||V - WH||^2$
- $W_{ik}^{new} = W_{ik} - \mu_{ik} \bigtriangledown_W$
  where $\bigtriangledown_W$ is the gradient of the approximation objective function with respect to W.
- Without loss of generality, we can assume that $\bigtriangledown_W$ consists of $\bigtriangledown^+$ and $\bigtriangledown^-$, positive and unsigned negative terms, respectively. That is,

$$\bigtriangledown_W = \bigtriangledown^+ - \bigtriangledown^-$$

- According to the steepest gradient descent method $W_{ik}^{new} = W_{ik} - \mu_{ik}(\bigtriangledown_{ik}^+ - \bigtriangledown_{ik}^-)$ can minimize the NMF objectives.
- By assuming that each matrix element has its own learning rate $\mu_{ik} = \frac{W_{ik}}{\bigtriangledown_{ik}^+}$ we have,

Multiplicative vs. Additive rules

By taking the gradient of the cost function with respect to W we have,

$$\begin{aligned}
&\nabla_W = WHH^T - VH^T \\
&W_{ik}^{new} = W_{ik} - \mu_{ik}((WHH^T)_{ik} - (VH^T)_{ik}) \\
&\mu_{ik} = \frac{W_{ik}}{(WHH^T)_{ik}} \\
&W_{ik}^{new} = W_{ik}\frac{(VH^T)_{ik}}{(WHH^T)_{ik}}
\end{aligned}$$

Similar for H,

$$H_{ik}^{new} = H_{ik}\frac{(W^TV)_{ik}}{(W^TWH)_{ik}} \tag{1}$$

Cost functions and gradient based algorithm for square Euclidean distance

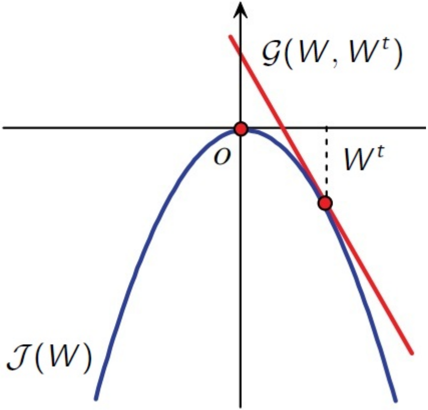
- The provided justification for multiplicative update rule does not have any theoretical guarantee that the resulting updates will monotonically decrease the objective function !

- Currently, the auxiliary function technique is the most widely accepted approach for monotonicity proof of multiplicative updates.

- Given an objective function $\mathcal{J}(W)$ to be minimized, $\mathcal{G}(W, W^t)$ is called an auxiliary function if it is a tight upper bound of $\mathcal{J}(W)$, that is,
  - $\mathcal{G}(W, W^t) \geq \mathcal{J}(W)$
  - $\mathcal{G}(W, W) = \mathcal{J}(W)$

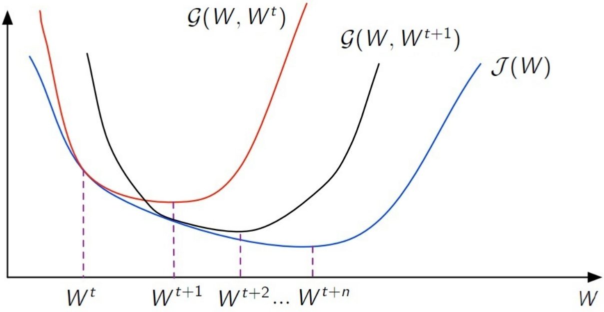
  for any W and W^t.
- Then iteratively applying the rule $W^{new} = arg \min_{W^t} G(W^t, W)$, results in a monotonically decrease of $\mathcal{J}(W)$.

Auxiliary function



Paraboloid function $\mathcal{J}(W)$ and its corresponding auxiliary function $\mathcal{G}(W, W^t)$, where $\mathcal{G}(W, W^t) \geq \mathcal{J}(W)$ and $\mathcal{G}(W^t, W^t) = \mathcal{J}(W^t)$ taken from Zhaoshui He, et al. IEEE TRANSACTIONS ON NEURAL NETWORKS 2011

Auxiliary function



Using an auxiliary function $\mathcal{G}(W, W^t)$ to minimize an objective function $\mathcal{J}(W)$. The auxiliary function is constructed around the current estimate of the minimizer ; the next estimate is found by minimizing the auxiliary function, which provides an upper bound on the objective function. The procedure is iterated until it converges to a stationary point (generally, a local minimum) of the objective function.

Updates for H of Euclidean distance

If $K(h^t)$ is the diagonal matrix

$$K_{ii}(h^t) = \frac{(W^t W h^t)_i}{h_i^t}$$

then

$$G(h, h^t) = J(h^t) + (h - h^t)^T \bigtriangledown J(h^t) + \frac{1}{2}(h - h^t)^T K(h^t)(h - h^t)$$

is an auxiliary function for

$$J(h) = \frac{1}{2} \sum_i (v_i - \sum_k W_{ik} h_i)$$

Proof

- The update rule can be obtained by taking the derivative of the $G(h, h^t)$ w.r.t h and then set it to zero,

$$\bigtriangledown J(h^t) + (h - h^t)K(h^t) = 0$$

$$h = h^t - K(h^t)^{-1} \bigtriangledown J(h^t)$$

- Since $J(h^t)$ is non-increasing under this auxiliary function, by writing the components of this equation explicitly, we obtain,

$$h_i^{t+1} = h_i^t \frac{(W^t V)_i}{(W^t W h^t)_i}$$

- Can be shown similarly for W of Euclidean distance.

Outline

- **What is clustering ?**
- **NMF overview**
- **Cost functions and multiplicative algorithms**
- **A geometric interpretation of NMF**
- **r-separable NMF**

A geometric interpretation of NMF

- Given M and its NMF $M \approx UV$ one can scale M and U such that they become column stochastic implying that V is column stochastic :

$$M \approx UV \iff M' = MD_m = (UD_u)(D_u^{-1}VD_m) = U'V'$$

$$M(:,j) = \sum_{i=1}^{k} U(:,i)V(i,j) \quad \textit{with} \quad \sum_{i=1}^{k} V(i,j) = 1$$

A geometric interpretation of NMF

- Given M and its NMF $M \approx UV$ one can scale M and U such that they become column stochastic implying that V is column stochastic :

$$M \approx UV \iff M' = MD_m = (UD_u)(D_u^{-1}VD_m) = U'V'$$

$$M(:,j) = \sum_{i=1}^{k} U(:,i)V(i,j) \quad \text{with} \quad \sum_{i=1}^{k} V(i,j) = 1$$

- Therefore, the columns of M are convex combination of the columns of U.

A geometric interpretation of NMF

- Given M and its NMF $M \approx UV$ one can scale M and U such that they become column stochastic implying that V is column stochastic :

$$M \approx UV \iff M' = MD_m = (UD_u)(D_u^{-1}VD_m) = U'V'$$

$$M(:,j) = \sum_{i=1}^{k} U(:,i)V(i,j) \quad \quad with \quad \quad \sum_{i=1}^{k} V(i,j) = 1$$

- Therefore, the columns of M are convex combination of the columns of U.

- In other terms, $conv(M) \subset conv(U) \subset \Delta^m$ where Δ^m is the unit simplex.

A geometric interpretation of NMF

- Given M and its NMF $M \approx UV$ one can scale M and U such that they become column stochastic implying that V is column stochastic :

$$M \approx UV \iff M' = MD_m = (UD_u)(D_u^{-1}VD_m) = U'V'$$

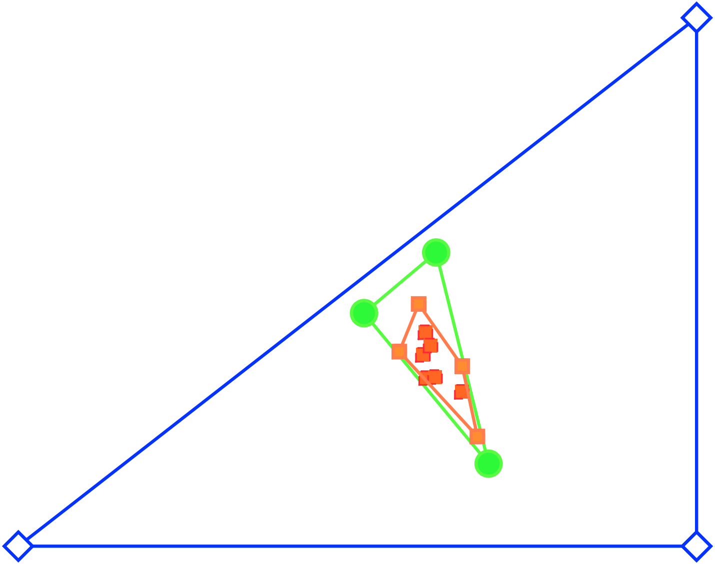
$$M(:,j) = \sum_{i=1}^{k} U(:,i)V(i,j) \quad \textit{with} \quad \sum_{i=1}^{k} V(i,j) = 1$$

- Therefore, the columns of M are convex combination of the columns of U.

- In other terms, $conv(M) \subset conv(U) \subset \Delta^m$ where Δ^m is the unit simplex.

- Solving exact NMF is equivalent to finding a poltyope $conv(U)$ between $conv(M)$ and Δ^m with minimum number of vertices.

A geometric interpretation of NMF



Outline

- **What is clustering ?**
- **NMF overview**
- **Cost functions and multiplicative algorithms**
- **A geometric interpretation of NMF**
- **r-separable NMF**

Separability Assumption

- If matrix V satisfies separability condition, it is possible to compute optimal solutions in **polynomial time**.

Separability Assumption

- If matrix V satisfies separability condition, it is possible to compute optimal solutions in **polynomial time**.

- r-separable NMF

  There exists an NMF (W, H) of rank r with $V = WH$ where each column of W is equal to a column of V.

Separability Assumption

- If matrix V satisfies separability condition, it is possible to compute optimal solutions in **polynomial time**.

- r-separable NMF

  There exists an NMF (W, H) of rank r with $V = WH$ where each column of W is equal to a column of V.

- V is r-separable $\iff V \approx WH = W[I_r, H']\Pi = [W, WH']\Pi$

  For some $H' \geq 0$ with columns sum to one, some permutation matrix Π, and I_r is the r-by-r identity matrix.

A geometric interpretation of separability

$$\mathrm{conv}(V) = \mathrm{conv}(W) = \mathrm{conv}(V(:,K)), \quad K \subset \{1, 2, \ldots, n\}, |K| = r.$$

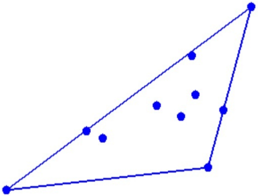


Figure: r columns of V are equal to the columns of W, and the remaining ones belong to the convex hull of the columns of W (that is, $\mathrm{conv}(W)$).

Separability Assumption

- Under separability, NMF reduces to the following problem :

  Given a set of points (the normalized columns of V), identify the vertices of its convex hull.

Separability Assumption

- Under separability, NMF reduces to the following problem :

  Given a set of points (the normalized columns of V), identify the vertices of its convex hull.

- We are still very far from knowing the best ways to compute the convex hull for general dimensions, despite the variety methods proposed for convex hull problem.

Separability Assumption

- Under separability, NMF reduces to the following problem :

  Given a set of points (the normalized columns of V), identify the vertices of its convex hull.

- We are still very far from knowing the best ways to compute the convex hull for general dimensions, despite the variety methods proposed for convex hull problem.

- However, we want to design algorithms which are
  - **Fast :** they should be able to deal with large-scale real-world problems where n is $10^6 - 10^9$.
  - **Robust :** if noise is added to the separable matrix, they should be able to identifying the right set of columns.

Separability Assumption

- For a separable matrix V, we have,

$$\begin{aligned} V &\approx WH \\ &= W[I_r, H']\Pi \\ &= [W, WH']\Pi \\ &= [W, WH']\begin{pmatrix} I_r & H' \\ 0_{(n-r)\times r} & 0_{(n-r)\times(n-r)} \end{pmatrix}\Pi \\ &= VX \end{aligned}$$

where Π is a permutation matrix, the columns of $H' \geq 0$ sum to one.

- Therefore for r-separable NMF, we need to solve the following optimization problem according to some constraints,

$$||V(:,i) - VX(:,i)||_2 \leq \epsilon \text{ for all } i.$$

Question

Question ?