VERİ MADENCİLİĞİ
Farklı Demetleme Yöntemleri

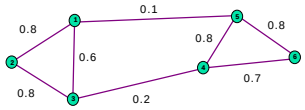
Yrd. Doç. Dr. Şule Gündüz Öğüdücü
www.cs.itu.edu.tr/~gunduz/courses/verimaden/

Çizge

- Veri kümesi $D=\{x_1,x_2,...,x_K\}$
- Veri kümesi ağırlıklı, yönsüz, bağlı bir çizge ile temsil edilir: G(V,E)

  $V=\{x_i\}$ veri kümesindeki nesnelerden oluşan düğümler kümesi

  $E=\{w_{ij}\}$ x_i ve x_j düğümleri arasındaki ağırlık



Çizge Tabanlı Demetleme

- $S=\{V,N,W,C\}$
  - V: veri kümesindeki nesnelerden oluşan düğümler kümesi
  - $N \subseteq V \times V$
  - W: N kümesinin elemanları için simetrik benzerlik matrisi
  - P: Demetleme kriteri
- Çizge Bölme: P demetleme kriterini enbüyütecek şekilde V kümesini k demede bölmek $(C=\{C_1,...,C_k\})$.
  - Her demet bir altçizge $G_i(V_i,E_i)$

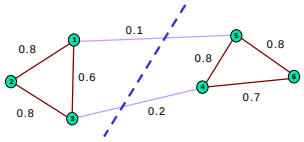
$$\bigcup_{i=1}^{k} V_i = V$$

$$E_i = \{\{u,v\} \in E \wedge u,v \in V_i\}$$

- Problem: Çizge tabanlı demetleme yöntemleri için P demetleme kriteri nedir?

Çizge Tabanlı Demetleme Problemi

- Demetlemenin sağlaması gereken koşullar:
  1. Aynı demetlerdeki nesnelerin birbirine daha çok banzemesi
  2. Farklı demetlerdeki nesneler birbirine daha az benzemesi
- Aynı koşullar çizge tabanlı demetlemeye uygulanırsa



1. Aynı grup içindeki ağırlıkları enbüyütme
2. Farklı gruplar arasındaki ağırlıkları enküçültme

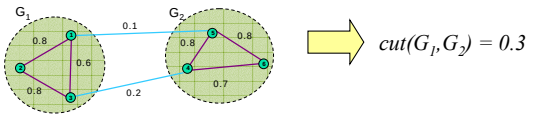
Çizge Tabanlı Demetleme için Tanımlar

- Tanımlar:
  - uzaklık d, benzerlik s
    - $d=1-s$
  - C_i ve C_j demetleri arasındaki uzaklık: $d(C_i,C_j)$
    - tek bağ, tam bağ ya da ortalama
  - C_i demedinin çapı: $diam(C_i)$
    - C_i demedinde bulunan en uzak iki nesne arasındaki uzaklık
    - C_i demedinden bulunan tüm nesneler arasındaki uzaklıkların ortalaması

Çizge Kesmesi

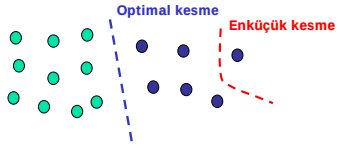
- Çizge Kesmesi: Demetleri biribirine bağlayan ayrıtların ağırlıklarının toplamı

$$cut(G_1,G_2) = \sum_{x_i \in G_1, x_j \in G_2} w_{ij}$$



Çizge Kesmesi Kriteri

- Kriter: En küçük kesme (Minimum-cut)
  - Demetler arasındaki ağırlıkları en küçültüyor
    - min *cut* (G_1, G_2)



- Problem:
  - Sadece demetlerearsı ağırlıkları enküçültüyor
  - Demet içi ağırlıkları göz önüne almıyor

Çizge Tabanlı Demetleme Kriteri

- Demet içindeki ağırlıkların enbüyük, demetler arasındaki ağırlıkların enküçük olması (min-max cut)

$$minimize \sum_{m=1}^{k} \frac{cut(G_m, G \setminus G_m)}{\sum_{v_i, v_j \in Gm} w_{ij}}$$

- Demetler daha dengeli
- Problemin optimal çözümü NP karmaşık
  - sezgisel yöntemler

İzgesel Demetleme

- Yaklaşım:
  - Çizge düğümlerin ikili benzerliklerinden oluşan *W* simetrik matrisi ile temsil edilir
  - Lineer cebir yöntemleri kullanılır
- Bir matrisin özdeğerleri ve özvektörleri yapısı hakkında bilgi verir

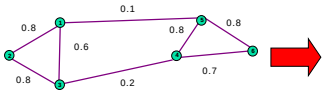
$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

  - *İzgesel Çizge Kuramı*
    - Bir matrisin izgesi incelenir.
    - *İzge (Spectrum)*: Çizgenin öz değerlerinin büyüklüklerine göre sıralanmış özvektörü

$$\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_n\}$$

Matris Yapısı

- Benzerlik matrisi (*W*)
  - *n*×*n* matris (*n*: düğüm sayısı)
  - *W*=[w_{ij}]: x_i ve x_j düğümleri arasındaki ayrıtın ağırlığı

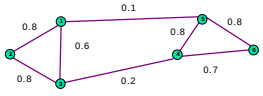


	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	0.8	0.6	0	0.1	0
x_2	0.8	0	0.8	0	0	0
x_3	0.6	0.8	0	0.2	0	0
x_4	0	0	0.2	0	0.8	0.7
x_5	0.1	0	0	0.8	0	0.8
x_6	0	0	0	0.7	0.8	0

- Özellikleri:
  - Simetrik matris

Matris Yapısı

- Derece matrisi (*D*)
  - *n*×*n* diagonal matris
  - $D(i,i) = \sum w_{ij}$ x_i düğümünden diğer düğümlere olan ayrıtların ağırlıklarının toplamı

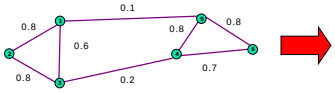


	x_1	x_2	x_3	x_4	x_5	x_6
x_1	1.5	0	0	0	0	0
x_2	0	1.6	0	0	0	0
x_3	0	0	1.6	0	0	0
x_4	0	0	0	1.7	0	0
x_5	0	0	0	0	1.7	0
x_6	0	0	0	0	0	1.5

Matris Yapısı

- Laplacian matrisi (*L*)
  - *n*×*n* simetrik matris

$$L = D - A$$



	x_1	x_2	x_3	x_4	x_5	x_6
x_1	1.5	-0.8	-0.6	0	-0.1	0
x_2	-0.8	1.6	-0.8	0	0	0
x_3	-0.6	-0.8	1.6	-0.2	0	0
x_4	0	0	-0.2	1.7	-0.8	-0.7
x_5	-0.1	0	0	-0.8	1.7	-0.8
x_6	0	0	0	-0.7	-0.8	1.5

- Özellikleri:
  - Özdeğerler pozitif gerçel sayılar
  - Özvektörler gerçel ve dikey
  - Özdeğerler ve özvektörler çizge yapısı hakkında bilgi veriyor.

Optimal Enküçük Kesme Bulma (Hall'70, Fiedler'73)

- İki altçizgeye (G_1,G_2) bölünen çizge bir vektörle temsil edilir. $p_i = \begin{cases} +1 \ if \ x_i \in G_1 \\ -1 \ if \ x_i \in G_2 \end{cases}$
- Bölmenin kesmesini enküçültmek için *f(p)* fonksiyonunu enküçültecek *p* vektörü bulunur:

$$f(p) = \sum_{i,j \in V} w_{ij}(p_i - p_j)^2 = p^T L\, p$$

Laplacian matrisi

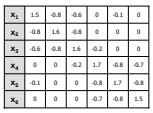
- *Rayleigh Kuramına* göre:
  - *f(p)*'nin enküçük değeri *L* matrisinin ikinci enküçük özdeğeri ile elde edilir.
  - *p* için optimal çözüm Fiedler vektörü olarak bilinen λ_2 vektörüdür.

İzgesel Çizge Demetleme

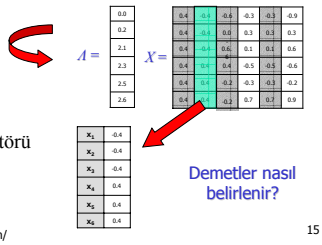
- A. Pothen, H.D. Simon and K. Paul Liou. *Partitioning Sparse Matrices with Eigenvectors of Graphs*, SIAM J. Mat. Theory and Appl., Vol. 11, No. 3, pp. 430 - 452, 1990.
  - Önişleme
    - veri kümesinin matris olarak temsil edilmesi
  - Ayrıştırma
    - Matrisin özvektörlerinin ve özdeğerlerinin bulunması
    - Veri kümesindeki her nesnenin bir veya daha çok özvektörü kullanılarak daha küçük bir boyuta taşınması
  - Gruplama
    - Yeni boyutlardan yararlanarak nesnelerin iki veya daha fazla demede ayrılması

Spectral Bi-partitioning Algoritması

1. Önişleme
   - Çizgenin *L* Laplacian matrisini oluşturma



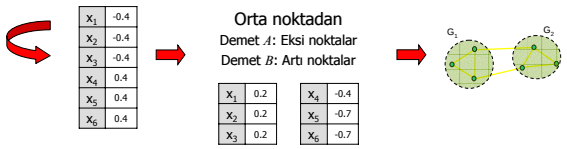
2. Ayrıştırma
   - *L* matrisinin özdeğerlerinin Λ ve özvektörlerinin X bulunması
   - Düğümlerin λ_2 özvektörü ile temsil edilmesi



Demetler nasıl belirlenir?

Spectral Bi-partitioning Algoritması

- Gruplama
  - Tek boyutlu vektörde bulunan elemanlar sıralanır
  - Vektör ikiye bölünür
- Bölme noktası nasıl belirlenir?
  - Ortalamadan ya da orta noktadan bölünür



K-Yönlü İzgesel Demetleme

- Biz çizge *k* adet altçizgeye bölünmek isteniyor.
- İki yaklaşım
  - Yinelemeli ikiye demetleme (L. Hagen, A.B. Kahng, *New spectral methods for ratio cut partitioning and clustering*, IEEE Trans. Comput. Aided Des. 11,1992)
    - Yinelemeli olarak ikiye demetleme algoritmasını hiyerarşiik olarak uygulanması
  - Daha fazla sayıda özvektörü kullanarak demetleme (J. Shi and J. Malik. *Normalized cuts and image segmentation*. IEEE Trans. *on Pattern Analysis and Machine Intelligence*, 22(8):888-905, 2000.=
    - Özvektörleri kullanarak veriyi daha az boyutlu bir uzaya taşır

K-Yönlü İzgesel Demetleme

- Çizgeler arasındaki optimal kesmeyi yaklaşık olarak bulabilir (Shi & Malik, 2002).
- Veri içindeki grupları belirgin hale getirir (M. Brand and K. Huang. *A unifying theorem for spectral embedding and clustering,* Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, January 2003.)
  - Benzer nesneler arasındaki ilişki kuvvetleniyor, daha az benzer nesneler arasındaki ilişki zayıflıyor.

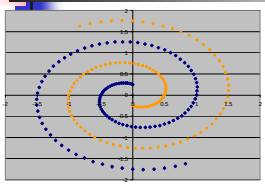
K-Özvektör Demetleme

- *k* özvektör kullanarak demetleme yapılıyor (A. Ng, M. Jordan, and Y. Weiss. *On spectral clustering: Analysis and an algorithm,* In Advances in Neural Information Processing Systems 14: Proceedings of the 2001.)
  1. Önişleme: ölçeklendirilmiş ağırlık matrisi oluşturulur

$$A' = D^{-1/2} A D^{-1/2}$$

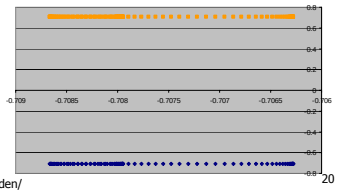
  2. Ayrıştırma: A' matrisinin özvektörleri bulunur. Veri kümesi en büyük *k* özdeğer ile temsil edilir
  3. Demetleme: k-means algoritması kullanılarak $n \times k$ boyutundaki veri *k* demede ayrılır.

Örnek



Veri kümesi iki spiral şeklinde gruptan oluşuyor

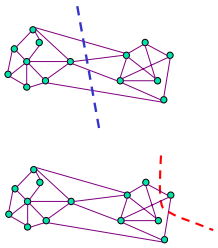
⇒ k-means algoritmasının performansı bu durumda çok düşük



En büyük iki özdeğer kullanılarak veri k-means algoritması kullanılarak demetlenebiliyor

Çizge Tabanlı Demetleme Yöntemi (Kannan'00)

- En küçük kesme bulunarak demetlenirse altçizgeler arasındaki nesneler dengeli dağılmayabilir.
- Demetlerin kalitesi önemli (Ravi Kannan and Santosh Vempala and Adrian Vetta, *On Clusterings: Good, Bad, and Spectral,* Proceedings of the 41st Annual Symposium on the Foundation of Computer Science, 2000.)



Çizge Tabanlı Demetleme Yöntemi (Kannan'00)

- Bir kesme $(S, \bar{S})$ için genişlik

$$\psi(S) = \frac{\sum_{x_i \in S, x_j \in \bar{S}} w_{ij}}{\min\{|S|, |\bar{S}|\}}$$

- Bir kesme $(S, \bar{S})$ için iletkenlik

$$\phi(S) = \frac{\sum_{x_i \in S, x_j \in \bar{S}} w_{ij}}{\min\{c(S), c(\bar{S})\}}$$

c(S) şu şekilde tanımlanmıştır $c(S) = c(S,V) = \sum_{x_i \in S} \sum_{x_j \in V} w_{ij}$

- Bir demedin genişliği (iletkenliği) demet içindeki kesmelerin genişliklerinin (iletkenliklerinin) en küçüğü
- Demetlemenin genişliği (iletkenliği) demetlerin genişliklerinin (iletkenliklerinin) en küçüğü
- Genişliğin (iletkenliğin) büyük olması iyi bir demetleme olduğunu gösteriyor

Çizge Tabanlı Demetleme Yöntemi (Kannan'00)

- Çizgeyi demetlemek için iki kriter beraber kullanılıyor:
  - Her demedin iletkenliği (genişliği) en az α değerinde olmalı
  - Demetler arası ayrıtların ağırlıklarının toplamının bütün ayrıtların ağırlıklarının toplamına oranı ε değerinden büyük olmamalı
- Problemin çözümü NP-karmaşık olduğu için yaklaşık bir çözüm önerilyor.

Evrimsel Algoritma Tabanlı Çizge Demetleme

- Çizge demetleme problemi NP-karmaşık bir problem olduğundan doğa esinli algoritmalar kullanılarak problem çözülebilir (Ş.Uyar and Ş.Oguducu, *A New Graph-Based Evolutionary Approach to Sequence Clustering*, The Fourth International Conference on Machine Learning and Applications, 2005)
- Amaç:
  - Aynı demetteki nesneler arasındaki ayrıtların ağırlıklarının toplamının, demetler arasındaki ayrıtların ağırlıklarının toplamına oranını enbüyütmek
  - Demet sayısını adaptif olarak belirlemek.

Evrimsel Algoritma Tabanlı Çizge Demetleme

Algorithm 1 Flow of the steady-state EA

```
1: randomly initialize population
2: while max no of fitness evaluations not reached do
3:    select parents
4:    create child through cross-over
5:    mutate child
6:    if child is not duplicate then
7:       apply heuristic disband to child
8:       child replaces worst in population
9:    else
10:      discard child
11:   end if
12: end while
```

- İki objektif fonksiyonu:
  - *Min-max cut*
  - *Silhoutte index*

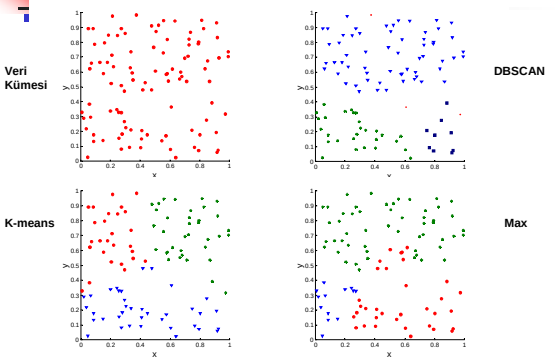
Demetleme Değerlendirme

- Hangi demetleme algoritması kullanılmalı?
- Farklı bir demetleme uygulanmalı mı?
- Farklı demetleme yöntemleri kullanılarak sonuç nasıl iyileştirilir?

Demetleme Değerlendirme

- Farklı demetleme teknikleri
- En iyi demetleme algoritmasını seçmek zor
- Verinin istatistiksel özelliğine, önişleme tekniklerine, nitelik sayısına bağlı olarak algoritmaların avantajları ve dezavantajları var
- Aynı veri kümesi üzerinde farklı algoritmalar farklı demetleme sonuçları üretebilir. Hangi demetlemenin daha iyi olduğuna karar vermek gerekiyor
  - uygulama alanını iyi incelemek gerekiyor
  - demetleme sonucunu iyi anlamak gerekiyor

Bir Veri Kümesi için Farklı Demetlemeler



Demetleme Değerlendirme

- Gözetimli öğrenme için kullanılan yöntemler:
  - Doğruluk, kesinlik, duyarlılık
- Demetleme yöntemlerinde değerlendirilmesi gerekenler:
  - Hatalı veriler için örüntü bulunmaması
  - Farklı demetleme algoritmalarını karşılaştırma
  - Farklı demetlemeleri karşılaştırma
  - Farklı demetleri karşılaştırma

Veri Kümesi Demetlemeye Uygun mu?

- Veri kümesi içinde gruplar olmayabilir.
  - Nesneler rasgele dağılmış
- Her demetleme algoritması veri kümesi üzerinde demetleme yapar
- Hopkins istatistiği: Veri kümesi içinde demetler bulunup bulunmadığını test etmek için kullanılır
  - Veri uzayında rasgele dağılmış p nokta üretilir
  - Veri kümesinden örnekleme ile p nokta seçilir
  - Her iki küme için veri kümesinden en yakın nesnler seçilir.
    - u_i yapay olarak üretilmiş noktalara olan uzaklık, w_i veri kümesinden seçilmiş olan noktalara olan uzaklık
  - Hopkins istatistiği

$$H = \frac{\sum_{i=1}^{p} w_i}{\sum_{i=1}^{p} u_i + \sum_{i=1}^{p} w_i}$$

Demetleme Değerlendirme Ölçütleri

- Üç yaklaşım:
  - Harici Gösterge:Veri kümesi için öngörülen bir yapıya dayanarak değerlendirme
  - Dahili Gösterge: Ek bir bilgi kullanmadan veri kümesinden elde edilen bilgiye dayanarak değerlendirme
  - Göreceli Değerlendirme: Aynı algoritmanın farklı parametrelerini kullanarak elde edilen demetleme sonuçlarını değerlendirme
- İki kriter:
  - Sıkılık: Her demette bulunan nesneler birbirine mümkün olduğunca yakın olmalı
  - Uzaklık: Demetler birbirinden mümkün olduğunca uzak olmalı
    - Tek bağ
    - Tam bağ
    - Demet merkezleri arasındaki uzaklık

Harici Değerlendirme

- Demetleme algoritması kullanılarak elde edilen demetleme $C=\{C_1,\ldots,C_k\}$
- Veri içinden bulunan gruplar $P=\{P_1,\ldots,P_m\}$
- Demetleme sonucundan elde edilen dağılım
  - SS: Eğer iki nesne C için aynı demette ve P için aynı grupta ise (a)
  - SD: Eğer iki nesne C için aynı demette ancak P için farklı gruplarda ise (b)
  - DS: Eğer iki nesne C için farklı demette ancak P için aynı grupta ise (c)
  - DS: Eğer iki nesne C için farklı demette ancak P için aynı grupta ise (d)

Harici Değerlendirme Ölçütleri

- Rand Statistics: $R = \frac{a+d}{a+b+c+d}$
- Jaccard katsayısı: $J = \frac{a}{a+b+c}$
- Folkes ve Mallows göstergesi: $FM = \sqrt{\frac{a}{a+b} \cdot \frac{a}{a+c}}$

Harici Değerlendirme Ölçütleri

- Entropi: Her demette sınıfların nasıl dağıldığı

$$e_i = -\sum_{j=1}^{m} p_{ij} \log_2 p_{ij}$$

$$e = \sum_{i=1}^{k} \frac{n_i}{n} e_i$$

m: sınıf sayısı
k: demet sayısı
$p_{ij} = n_{ij}/n_i$
n_i: i demedindeki nesne sayısı
n_{ij}: i demedinde j sınıfından nesne sayısı
n: toplam nesne sayısı

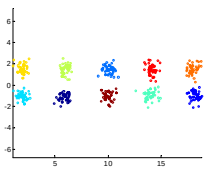
- Saflık: Bir demette ne kadar tek sınıftan örnek bulunduğu

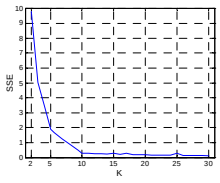
$$p_i = \max_j p_{ij}$$

$$purity = \sum_{i=1}^{k} \frac{n_i}{n} p_i$$

Dahili Değerlendirme Ölçütleri

- Sadece veri kümesi özellikleri kullanılarak yapılan değerlendirme
  - Hataların karelerinin toplamı (SSE)
- İki farklı demetlemeyi ya da iki demedi karşılaştırmak için iyi bir yöntem
- Demet sayısını tahmin etmek için de kullanılabilir.





Dahili Değerlendirme Ölçütleri

- Silhouette Göstergesi:
  - x_i nesnesi C_j demedinde
  - Ortalama uzaklığa göre x_i nesnesine en yakın demet C_h
  - x_i nesnesi için silhouette göstergesi

$$s(x_i) = \frac{d(x_i, C_h) - d(x_i, C_j)}{\max(d(x_i, C_h), d(x_i, C_j))}$$

  - $-1 \le s(v_i) \le 1$
  - 1'e yakın olursa x_i doğru demette
  - Demetleme için silhouette göstergesi:

$$S_j = \frac{\sum_{i=1}^{|Vj|} s(x_i)}{|Vj|}$$

$$GS = \frac{\sum_{j=1}^{k} S_j}{k}$$

Göreceli Değerlendirme

- P_{alg} seçilen demetleme algoritmasının parametreleri
- P_{alg}'deki parametrelerin farklı değerleri ile elde edilen demetlemeler C_i, $i=1,...,nc$ arasında veriye en çok uyanı seçme
- İki durum:
  - Demet sayısı nc P_{alg}'deki parametereler arasında değil:
    - *Palg*'deki parametrelerin değerleri geniş bir aralıkta değiştirilerek demetleme algoritması çalıştırılır. $nc << N$ (nesne sayısı) sabit kaldığı en geniş aralık seçilir. Parametre değerleri olarak bu aralığın orta noktası seçilir. Bu yöntemle demet sayısı da belirlenmiş olur.
  - Demet sayısı nc P_{alg}'deki parametereler arasında:
    - En iyi demetleme, demetleme göstergesi q kullanarak seçilir.
      - nc_{min} ve nc_{max} arasında değişen farklı demet sayıları için algoritma çalıştırılır.
      - nc'nin her farklı değeri için algoritma diğer parametreleri değiştirerek r defa çalıştırılır.
      - Her nc için q'nun en büyük değeri seçilir ve nc'nin fonksiyonu olarak çizilir. Bu çizim kullanılarak nc değeri belirlenir.

Göreceli Değerlendirme Ölçütleri

- Birbiri ile örtüşmeyen demetler için tanımlanmış göstergeler:
  - Hubert istatistiği:

$$\Gamma = (1/M)\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} P(i,j)\cdot Q(i,j)$$

N= veri kümesindeki nesne sayısı
M=N(N-1)/2
P: yakınlık matrisi
Q: (i,j) elemanı x_i ve x_j nesnelerinin bulundukları demetler arasındaki uzaklık olan matris

- Dunn göstergesi:

$$Dnc = \min_{i=1,\ldots,nc}\left\{\min_{j=i+1,\ldots,nc}\frac{d(c_i,c_j)}{\max_{k=1,\ldots,nc} diam(c_k)}\right\}$$

$d(c_i,c_j)=\min_{x\in ci, y\in cj} d(x,y)$

$diam(C)=\max_{x,y\in C} d(x,y)$

Hiyerarşik Demetleme için Gösterge

- Hiyerarşik demetleme için 4 gösterge
  - Demetlerin standard sapmalarının karakökü (RMSSTD)
  - Semi-partial R-squared (SPR)
  - R-Squared (RS)
  - İki demet arası uzaklık (CD)

Örtüşen Demetleri Değerlendirme

- Örtüşen demetleme için $U=[u_{ij}]$ matrisi: x_i nesnesinin c_j demedine dahil olma olasılığı
- Bölme katsayısı:

$$PC = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{nc} u_{ij}^2$$

- [1/nc,1] arasında değişir.