VERİ MADENCİLİĞİ
Demetleme Yöntemleri

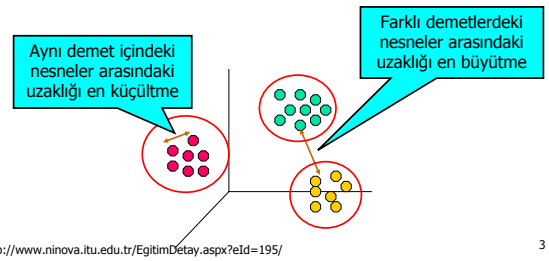
Yrd. Doç. Dr. Şule Gündüz Öğüdücü
http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=195/

Konular

- Demetleme işlemleri
  - Demetleme tanımı
  - Demetleme uygulamaları
- Demetleme Yöntemleri
  - Bölünmeli Yöntemler
  - Hiyerarşik Yöntemler
  - Yoğunluk Tabanlı Yöntemler
  - Model Tabanlı Yöntemler

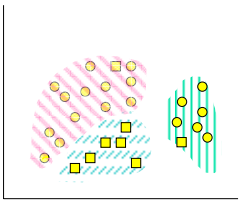
Demetleme

- Nesneleri demetlere (gruplara) ayırma
- Demet: birbirine benzeyen nesnelerden oluşan grup
  - Aynı demetteki nesneler birbirine daha çok benzer
  - Farklı demetlerdeki nesneler birbirine daha az benzer



Demetleme

- Gözetimsiz öğrenme: Hangi nesnenin hangi sınıfa ait olduğu ve sınıf sayısı belli değil
- Uygulamaları:
  - verinin dağılımını anlama
  - başka veri madenciliği uygulamaları için ön hazırlık



Demetleme Uygulamaları

- Örüntü tanıma
- Görüntü işleme
- Ekonomi
- Aykırılıkları belirleme
- WWW
  - Doküman demetleme
  - Kullanıcı davranışlarını demetleme
  - Kullanıcıları demetleme
- Diğer veri madenciliği algoritmaları için bir önişleme adımı
  - Veri azaltma – demet içindeki nesnelerin temsil edilmesi için demet merkezlerinin kullanılması

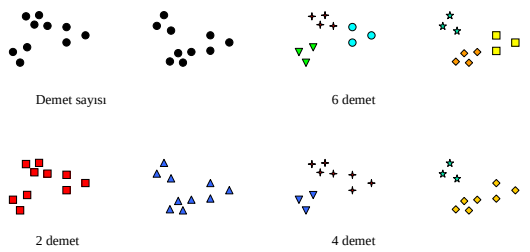
Veri Madenciliğinde Demetleme

- Ölçeklenebilirlik
- Farklı tipteki niteliklerden oluşan nesneleri demetleme
- Farklı şekillerdeki demetleri oluşturabilme
- En az sayıda giriş parametresi gereksinimi
- Hatalı veriler ve aykırılıklardan en az etkilenme
- Model oluşturma sırasında örneklerin sırasından etkilenmeme
- Çok boyutlu veriler üzerinde çalışma
- Kullanıcıların kısıtlarını göz önünde bulundurma
- Sonucun yorumlanabilir ve anlaşılabilir olması

İyi Demetleme

- İyi demetleme yöntemiyle elde edilen demetlerin özellikleri
  - aynı demet içindeki nesneler arası benzerlik fazla
  - farklı demetlerde bulunan nesneler arası benzerlik az
- Oluşan demetlerin kalitesi seçilen benzerlik ölçütüne ve bu ölçütün gerçeklenmesine bağlı
  - Uzaklık / Benzerlik nesnelerin nitelik tipine göre değişir
    - Nesneler arası benzerlik: s(i,j)
    - Nesneler arası uzaklık: d(i,j) = 1 – s(i,j)
- İyi bir demetleme yöntemi veri içinde gizlenmiş örüntüleri bulabilmeli
- Veriyi gruplama için uygun demetleme kriteri bulunmalı
  - demetleme = aynı demetteki nesneler arası benzerliği en büyüten, farklı demetlerdeki nesneler arası benzerliği en küçülten fonksiyon
- Demetleme sonucunun kalitesi seçilen demetlerin şekline ve temsil edilme yöntemine bağlı

Farklı Demetler



Temel Demetleme Yaklaşımları

- Bölünmeli yöntemler: Veriyi bölerek, her grubu belirlenmiş bir kritere göre değerlendirir
- Hiyerarşik yöntemler: Veri kümelerini (ya da nesneleri) önceden belirlenmiş bir kritere göre hiyerarşik olarak ayırır
- Yoğunluk tabanlı yöntemler: Nesnelerin yoğunluğuna göre demetleri oluşturur
- Model tabanlı yöntemler: Her demetin bir modele uyduğu varsayılır. Amaç bu modellere uyan verileri gruplamak

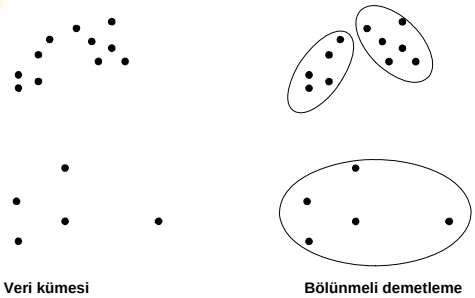
Konular

- Demetleme işlemleri
  - Demetleme tanımı
  - Demetleme uygulamaları
- Demetleme Yöntemleri
  - Bölünmeli Yöntemler
    - K-means demetleme yöntemi
    - K-medoids demetleme yöntemi
  - Hiyerarşik Yöntemler
  - Yoğunluk Tabanlı Yöntemler
  - Model Tabanlı Yöntemler

Bölünmeli Yöntemler

- Amaç: n nesneden oluşan bir veri kümesini (D) k ($k \leq n$) demete ayırmak
  - her demette en az bir nesne bulunmalı
  - her nesne sadece bir demette bulunmalı
- Yöntem: Demetleme kriterini en büyütücek şekilde D veri kümesi k gruba ayırma
  - Global çözüm: Mümkün olan tüm gruplamaları yaparak en iyisini seçme (NP karmaşık)
  - Sezgisel çözüm: k-means ve k-medoids
    - k-means (MacQueen'67): Her demet kendi merkezi ile temsil edilir
    - k-medoids veya PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Her demet, demette bulunan bir nesne ile temsil edilir

Bölünmeli Demetleme

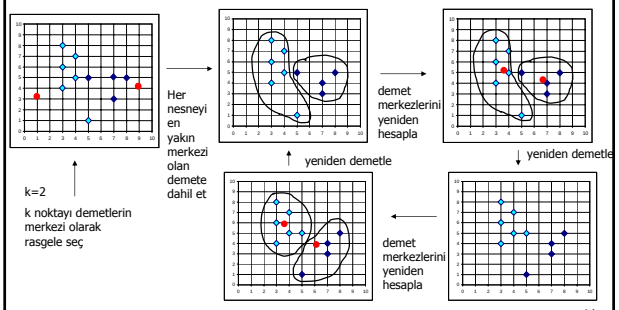


K-means Demetleme

- Bilinen bir *k* değeri için k-means demetleme algoritmasının 4 aşaması vardır:
  1. Veri kümesi *k* altkümeye ayrılır (her demet bir altküme)
  2. Her demetin ortalaması hesaplanır: merkez nokta (demetteki nesnelerin niteliklerinin ortalaması)
  3. Her nesne en yakın merkez noktanın olduğu demete dahil edilir
  4. Nesnelerin demetlenmesinde değişiklik olmayana kadar adım 2'ye geri dönülür.

K-means Demetleme Yöntemi

- Örnek

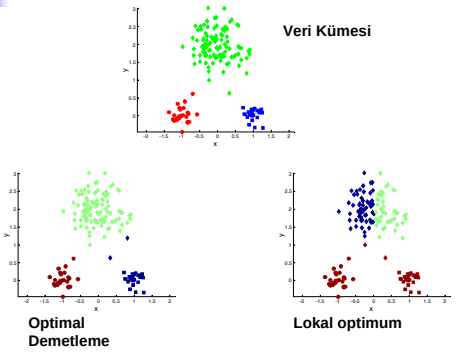


k-means Demetleme Yöntemi

- Demet sayısının belirlenmesi gerekir
- Başlangıçta demet merkezleri rasgele belirlenir
  - Her uygulamada farklı demetler oluşabilir
- Uzaklık ve benzerlik Öklid uzaklığı, kosinüs benzerliği gibi yöntemlerle ölçülebilir
- Az sayıda tekrarda demetler oluşur
  - Yakınsama koşulu çoğunlukla az sayıda nesnenin demet değiştirmesi şekline dönüştürülür
- Karmaşıklığı:
  - Yer karmaşıklığı - *O((n+k) d)*
  - Zaman karmaşıklığı - *O(ktnd)*

  k: demet sayısı, *t*: tekrar sayısı, *n*: nesne sayısı, *d*: nitelik sayısı

K-Means: İki Farklı Demetleme



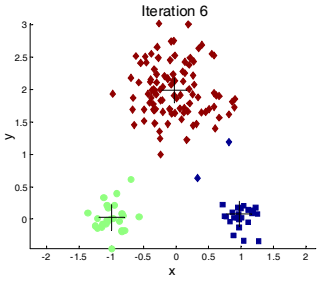
K-Means Demetleme Yöntemini Değerlendirme

- Yaygın olarak kullanılan yöntem hataların karelerinin toplamı (Sum of Squared Error SSE)
  - Nesnelerin bulundukları demetin merkez noktalarına olan uzaklıklarının karelerinin toplamı

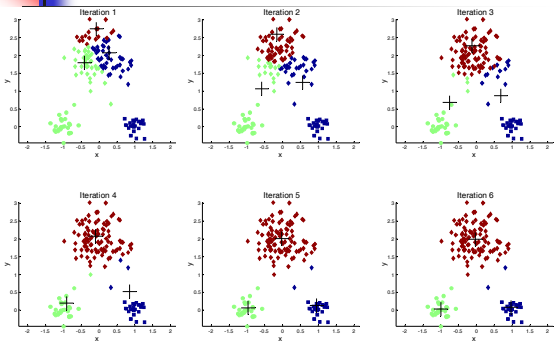
$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} dist^2(m_i, x)$$

  x: C_i demetinde bulunan bir nesne, m_i : C_i demetinin merkez noktası
- Hataların karelerinin toplamını azaltmak için *k* demet sayısı artırılabilir
  - Küçük k ile iyi bir demetleme, büyük k ile kötü bir demetlemeden daha az SSE değerine sahip olabilir.
- Başlangıç için farklı merkez noktaları seçerek farklı demetlemeler oluşturulur
- En az SSE değerini sahip olan demetleme seçilir

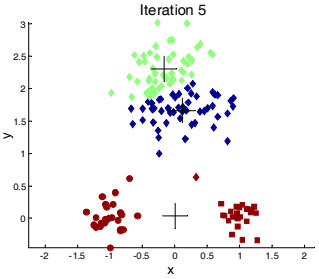
Merkez Noktaların Seçimi



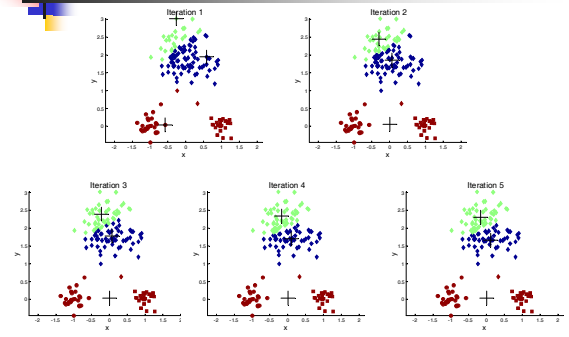
Merkez Noktaların Seçimi



Merkez Noktaların Seçimi



Merkez Noktaların Seçimi



K-Means Demetleme Çeşitleri

- K-Means demetlemeye başlamadan önce yapılanlar
  - Veri kümesini örnekleyerek hiyerarşik demetleme yap. Oluşan *k* demetin ortalamasını başlangıç için merkez nokta seç
  - Başlangıçta *k*'dan fazla merkez nokta seç. Daha sonra bunlar arasından *k* tane seç.
- K-Means demetleme işlemi sonrasında yapılanlar
  - Küçük demetleri en yakın başka demetlere dahil et
  - En büyük toplam karesel hataya sahip olan demeti böl
  - Merkez noktaları birbirine en yakın demetleri birleştir
  - Toplam karesel hatada en az artışa neden olacak iki demeti birleştir

K-Means Demetleme Algoritmasının Özellikleri

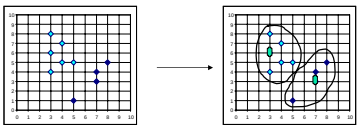
- Gerçeklemesi kolay
- Karmaşıklığı diğer demetleme yöntemlerine göre az
- K-Means algoritması bazı durumlarda iyi sonuç vermeyebilir
  - Veri grupları farklı boyutlarda ise
  - Veri gruplarının yoğunlukları farklı ise
  - Veri gruplarının şekli küresel değilse
  - Veri içinde aykırılıklar varsa

Konular

- Demetleme işlemleri
  - Demetleme tanımı
  - Demetleme uygulamaları
- Demetleme Yöntemleri
  - Bölünmeli Yöntemler
    - K-means demetleme yöntemi
    - K-medoids demetleme yöntemi
  - Hiyerarşik Yöntemler
  - Yoğunluk Tabanlı Yöntemler
  - Model Tabanlı Yöntemler

K-Medoids Demetleme Yöntemi

- Her demeti temsil etmek için demet içinde orta nokta olan nesne seçilir.
  - 1, 3, 5, 7, 9 ortalama: 5
  - 1, 3, 5, 7, 1009 ortalama 205
  - 1, 3, 5, 7, 1009 orta nokta 5



K-Medoids Demetleme Yöntemi

- PAM (Partitioning Araound Medoids 1987)
  1. Başlangıçta *k* adet nesne demetleri temsil etmek üzere rasgele seçilir x_{ik}
  2. Kalan nesneler en yakın merkez nesnenin bulunduğu demete dahil edilir
  3. Merkez nesne olmayan rasgele bir nesne seçilir x_{rk}
  4. x_{rk} merkez nesne olursa toplam karesel hatanın ne kadar değiştiği bulunur

$$TC_{ik} = \sum_{j=1}^{n_k}(x_{ik} - x_{jk})^2 - \sum_{j=1}^{n_k}(x_{rk} - x_{jk})^2$$

n_k: *k* demeti içindeki nesne sayısı
x_{jk}: *k* demeti içindeki *j.* nesne

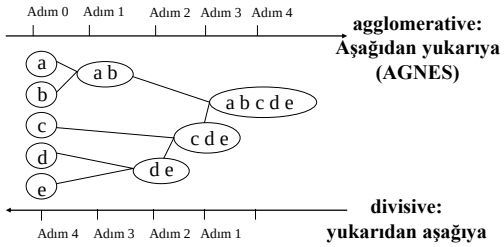
  5. $TC_{ik}<0$ ise O_{rk} merkez nesne olarak atanır.
  6. Demetlerde değişiklik oluşmayana kadar 3. adıma geri gidilir.
- Küçük veri kümeleri için iyi sonuç verebilir, ancak büyük veri kümeleri için uygun değil
- *CLARA* (Kaufmann & Rousseeuw, 1990)
- *CLARANS* (Ng & Han, 1994)

Konular

- Demetleme işlemleri
  - Demetleme tanımı
  - Demetleme uygulamaları
- Demetleme Yöntemleri
  - Bölünmeli Yöntemler
  - Hiyerarşik Yöntemler
  - Yoğunluk Tabanlı Yöntemler
  - Model Tabanlı Yöntemler

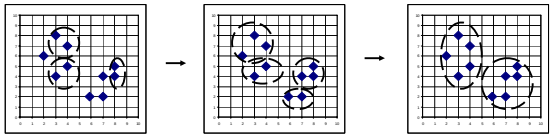
Hiyerarşik Demetleme

- Demet sayısının belirlenmesine gerek yok
  - Sonlanma kriteri belirlenmesi gerekiyor



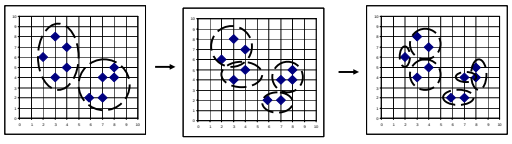
Hiyerarşik Yöntemler

- *AGNES* (*AG*glomerative *NES*ting):
  - Kaufmann ve Rousseeuw tarafından 1990 yılında önerilmiştir.
  - Birinci adımda her nesne bir demet oluşturur.
  - Aralarında en az uzaklık bulunan demetler her adımda birleştirilir.
  - Bütün nesneler tek bir demet içinde kalana kadar ya da istenen sayıda demet elde edene kadar birleştirme işlemi devam eder.



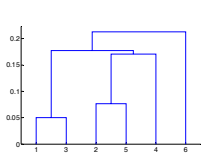
Hiyerarşik Yöntemler

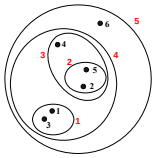
- *DIANA* (*DI*visive *ANA*lysis):
  - Kaufmann ve Rousseeuw tarafından 1990 yılında önerilmiştir.
  - AGNES'in yaptığı işlemlerin tersini yapar.
  - En sonunda her nesne bir demet oluşturur.
  - Her nesne ayrı bir demet oluşturana ya da istenilen demet sayısı elde edene kadar ayrılma işlemi devam eder.



Hiyerarşik Demetleme

- Dendogram: Demetler hiyerarşik olarak ağaç yapısı şeklinde görüntülenebilir
- Ara düğümler çocuk düğümlerdeki demetlerin birleşmesiyle elde edilir
  - Kök: bütün nesnelerden oluşan tek demet
  - Yapraklar: bir nesneden oluşan demetler
- Dendogram istenen seviyede kesilerek demetler elde edilir

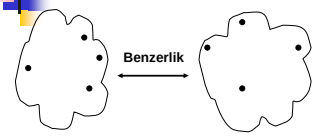


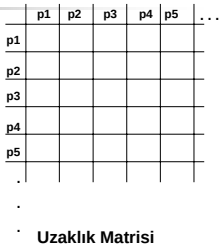


Aşağıdan Yukarıya Demetleme

- Algoritma
  1. Uzaklık matrisini hesapla
  2. Her nesne bir demet
  3. Tekrarla
  4. En yakın iki demeti birleştir
  5. Uzaklık matrisini yeniden hesapla
  6. Sonlanma: Tek bir demet kalana kadar
- Uzaklık matrisini hesaplarken farklı yöntemler farklı demetleme sonuçlarına neden olurlar

Demetler Arası Uzaklık

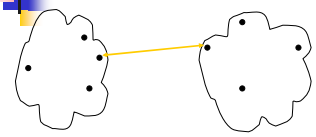


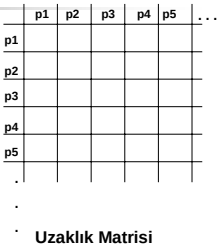


Uzaklık Matrisi

- MIN (Tek Bağ)
- MAX (Tam Bağ)
- Ortalama
- Merkezler arası uzaklık

Demetler Arası Uzaklık

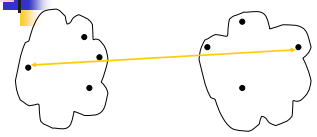


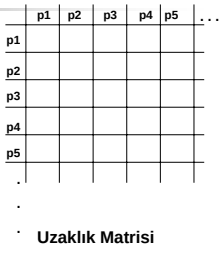


Uzaklık Matrisi

- MIN (Tek Bağ)
- MAX (Tam Bağ)
- Ortalama
- Merkezler arası uzaklık

Demetler Arası Uzaklık

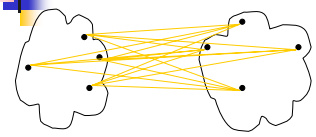


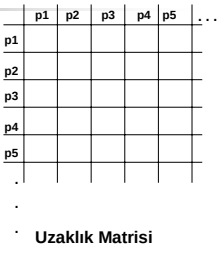


Uzaklık Matrisi

- MIN (Tek Bağ)
- MAX (Tam Bağ)
- Ortalama
- Merkezler arası uzaklık

Demetler Arası Uzaklık

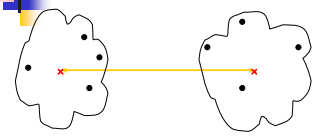


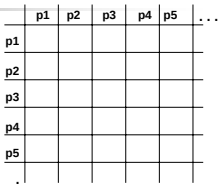


Uzaklık Matrisi

- MIN (Tek Bağ)
- MAX (Tam Bağ)
- Ortalama
- Merkezler arası uzaklık

Demetler Arası Uzaklık

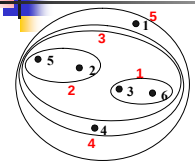


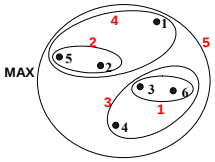


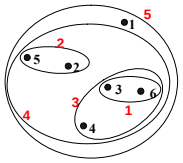
Uzaklık Matrisi

- MIN (Tek Bağ)
- MAX (Tam Bağ)
- Ortalama
- Merkezler arası uzaklık

Farklı Uzaklık Yöntemlerinin Etkisi







Hiyerarşik Demetleme Yöntemlerinin Özellikleri

- Demetleme kriteri yok
- Demet sayılarının belirlenmesine gerek yok
- Aykırılıklardan ve hatalı verilerden etkilenir
- Farklı boyuttaki demetleri oluşturmak problemli olabilir
- Yer karmaşıklığı – $O(n^2)$
- Zaman karmaşıklığı – $O(n^2 \log n)$

  n : nesne sayısı

Konular

- Demetleme işlemleri
  - Demetleme tanımı
  - Demetleme uygulamaları
- Demetleme Yöntemleri
  - Bölünmeli Yöntemler
  - Hiyerarşik Yöntemler
  - Yoğunluk Tabanlı Yöntemler
  - Model Tabanlı Yöntemler

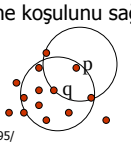
Yoğunluk Tabanlı Yöntemler

- Demetleme nesnelerin yoğunluğuna göre yapılır.
- Başlıca özellikleri:
  - Rasgele şekillerde demetler üretilebilir.
  - Aykırı nesnelerden etkilenmez.
  - Algoritmanın son bulması için yoğunluk parametresinin verilmesi gerekir.
- Başlıca yoğunluk tabanlı yöntemler:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98)

DBSCAN

- İki parametre:
  - *Eps*: En büyük komşuluk yarıçapı
  - *MinPts*: Eps yarıçaplı komşuluk bölgesinde bulunan en az nesne sayısı
- $N_{eps}(p)$: $\{q \in D \mid d(p,q) \leq Eps\}$
- Doğrudan erişilebilir nesne: *Eps* ve *MinPts* koşulları altında bir *q* nesnesinin doğrudan erişilebilir bir *p* nesnesi şu şartları sağlar:
  - $p \in N_{eps}(q)$
  - *q* nesnesinin çekirdek nesne koşulunu sağlaması

$N_{eps}(q) \geq MinPts$

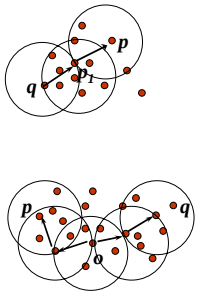


MinPts = 5

Eps = 1 cm

DBSCAN

- Erişilebilir nesne:
  - *Eps* ve *MinPts* koşulları altında *q* nesnesinin erişilebilir bir *p* nesnesi olması için:
    - $p_1, p_2, ..., p_n$ nesne zinciri olması,
    - $p1=q$, $pn=p$,
    - p_i nesnesinin doğrudan erişilebilir nesnesi: p_{i+1}
- Yoğunluk bağlantılı Nesne:
  - *Eps* ve *MinPts* koşulları altında *q* nesnesinin yoğunluk bağlantılı nesnesi *p* şu koşulları sağlar:
    - *p* ve *q* nesneleri *Eps* ve *MinPts* koşulları altında bir *o* nesnesinin erişilebilir nesnesidir.



Yoğunluk Tabanlı Yöntemler: DBSCAN

- Veri tabanındaki her nesnenin *Eps* yarıçaplı komşuluk bölgesi araştırılır.
- Bu bölgede *MinPts*'den daha fazla nesne bulunan *p* nesnesi çekirdek nesne olacak şekilde demetler oluşturulur.
- Çekirdek nesnelerin doğrudan erişilebilir nesneleri bulunur.
- Yoğunluk bağlantılı demetler birleştirilir.
- Hiçbir yeni nesne bir demete eklenmezse işlem sona erer.
- Yer karmaşıklığı – *O(n)*
- Zaman karmaşıklığı – *O(nlogn)*
  n: nesne sayısı

Konular

- Demetleme işlemleri
  - Demetleme tanımı
  - Demetleme uygulamaları
- Demetleme Yöntemleri
  - Bölünmeli Yöntemler
  - Hiyerarşik Yöntemler
  - Yoğunluk Tabanlı Yöntemler
  - Model Tabanlı Yöntemler

Model Tabanlı Demetleme Yöntemleri

- Veri kümesi için öngörülen matematiksel model en uygun hale getiriliyor.
- Verinin genel olarak belli olasılık dağılımlarının karışımından geldiği kabul edilir.
- Model tabanlı demetleme yöntemi
  - Modelin yapısının belirlenmesi
  - Modelin parametrelerinin belirlenmesi
- Örnek *EM (Expectation Maximization)* Algoritması

Model Tabanlı Demetleme Yöntemleri

- İstatistiksel yaklaşım:
  - K nesneden oluşan bir veri kümesi $D=\{x_1, x_2, ..., x_K\}$
  - her $\mathbf{x}_i$ $(i \in [1,...K])$ nesnesi $\boldsymbol{\Theta}$ parametre kümesiyle tanımlanan bir olasılık dağılımından oluşturulur.
  - Olasılık dağılımının, $c_j \in \mathbf{C}=\{c_1, c_2, ..., c_G\}$ şeklinde G adet bileşeni vardır.
  - Her $\boldsymbol{\Theta}_g$, $g \in [1, ..., G]$ parametre kümesi g bileşeninin olasılık dağılımını belirleyen, $\boldsymbol{\Theta}$ kümesinin ayrışık bir alt kümesidir.
  - Herhangi bir $\mathbf{x}_i$ nesnesi öncelikle, $p(c_g|\boldsymbol{\Theta}) = \tau_g$ ($\Sigma_G \tau_g = 1$ olacak şekilde) bileşen katsayısına (ya da bileşenin seçilme olasılığına) göre bir bileşene atanır.
  - Bu bileşen $p(\mathbf{x}_i|c_g; \boldsymbol{\Theta}_g)$ olasılık dağılımına göre $\mathbf{x}_i$ değişkenini oluşturur.
  - Böylece bir $\mathbf{x}_i$ nesnesinin bu model için olasılığı bütün bileşenlerin olasılıklarının toplamıyla ifade edilebilir:

$$p(x_i \mid \boldsymbol{\Theta}) = \sum_{g=1}^{G} p(c_g \mid \boldsymbol{\Theta}) p(\mathbf{x}_i \mid c_g; \boldsymbol{\Theta}_g)$$

$$p(x_i \mid \boldsymbol{\Theta}) = \sum_{g=1}^{G} \tau_g \, p(\mathbf{x}_i \mid c_g; \boldsymbol{\Theta}_g)$$

Model Tabanlı Demetleme Problemi

- Model parametrelerinin belirlenmesi
  - Maximum Likelihood (ML) yaklaşımı

$$\ell_{ML}(\Theta_1,...,\Theta_G;\tau_1,...,\tau_G \mid D) = \prod_{i=1}^{K} \sum_{g=1}^{G} \tau_g \, p(x_i \mid c_g, \Theta_g)$$

  - Maximum Aposteriori (MAP) yaklaşımı

$$\ell_{MAP}(\Theta_1,...,\Theta_G;\tau_1,...,\tau_G \mid D) = \prod_{i=1}^{K} \sum_{g=1}^{G} \frac{\tau_g \, p(x_i \mid c_g, \Theta_g) \, p(\Theta)}{p(D)}$$

  - Uygulamada her ikisinin logaritması

$$L(\Theta_1,...,\Theta_G;\tau_1,...,\tau_G \mid D) = \sum_{i=1}^{K} \ln \sum_{g=1}^{G} \left(\tau_g \, p(x_i \mid c_g, \Theta_g) \right)$$

$$L(\Theta_1,...,\Theta_G;\tau_1,...,\tau_G \mid D) = \sum_{i=1}^{K} \ln \sum_{g=1}^{G} \left(\tau_g \, p(x_i \mid c_g, \Theta_g) \right) + \ln p(\Theta)$$

EM Algoritması

- Veri kümesi: $D=\{x_1,x_2,\ldots,x_K\}$
- Gizli değişkenler $H=\{z_1,z_2,\ldots,z_K\}$ (her nesnenin hangi demete dahil olduğu bilgisi)
- Verinin eksik olduğu durumda, tam verinin beklenen değeri hesaplanır:

$$Q(\Theta,\Theta') = E[L_c(D,H \mid \Theta) \mid D,\Theta')$$
$$= \sum_{i=1}^{K}\sum_{g=1}^{G} p(c_g \mid x_i)[\ln p(x_i \mid c_g) + \ln \tau_g]$$

- EM Algoritmasının adımları:
  - Θ' için başlangıç değerleri atama
  - (E) Expectation: Q(Θ| Θ') hesaplanması
  - (M) Maximization: argmax Q(Θ| Θ')

http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=195/ 49