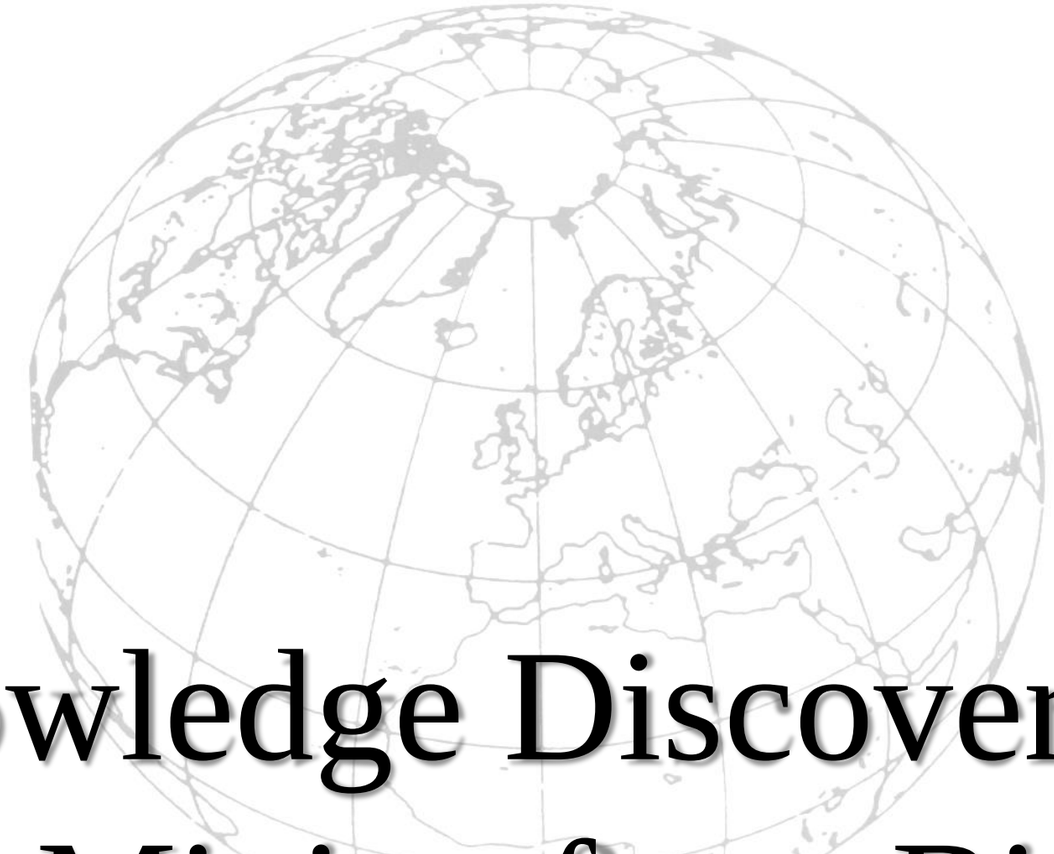


# Knowledge Discovery and Data Mining from Big Data

*Dr. Cahit Karakuş*

# Data Mining

# Veri Madenciliği

- Otomatize edilmiş veri toplama araçları ve yerleşik veri tabanlarında, veri ambarlarında ve diğer bilgi depolama alanlarında depolanmış durumda olan çok büyük miktarda veri bulunmaktadır.
- Özellikle dijital medyada kaydedilen ve saklanan verinin boyutu olağan üstü bir şekilde arttı ve git gide de artıyor. Yılda exabite(10^18)'ın üzerinde veri üretilmektedir. Sabit bir depolama kapasitesi her 9 ayda bir ikiye katlanıyor. Verinin içinde boğuluyoruz ama bilgi (knowledge) için can atıyoruz.
- Günümüzün en büyük sorunu, insanlara alakasızları görmezden gelmeyi, bir şeyleri boğulmadan önce reddetmeyi bilmeyi öğretmek. Çünkü çok fazla gerçek, hiç olmadığı kadar kötüdür. Çözüm ise veri ambarı (data warehousing) ve veri madenciliği (data mining).

# Bazı Büyük Verı (Big Data) Örnekleri

- Avrupa'da Very Long Baseline Interferomerty (VLBI) olarak adlandırılan ve radyo astronomide kullanılan bir gök bilimsel interferometre var. VLBI astronomik bir radyo(sinyal) kaynağından gelen sinyalleri, mesela quasar (galaksi dışındaki yıldızımsı gök cisimleri), dünyadaki birden çok radyo teleskop aracılığıyla topluyor ve kaydediyor. VLBI'de kullanılan toplam 16 teleskop var ve bu teleskopların her biri 1 Gigabit/saniye gibi son derece yüksek hızda ve boyutlarda veri üretiyor. Bu verinin depolanması ve analiz edilmesi ne kadar büyük bir problem değil mi?
- AT&T her gün milyarlarca arama isteği işliyor. Bu o kadar çok veriye sebep oluyor ki bütün veriler depolanamıyor ve depolanamadığı için de analizler "akan veri"(streaming data) üzerinden yapılmak zorunda.
- Alexa internet arşivinde 7 yıllık veri bulunuyor ve bu verinin boyutu 500 Terabyte civarlarında.
- Google hergün 4 milyarın üzerinde arama isteğini alıyor ve bu da bir sürü 100 Terabyte demek oluyor.
- Internat Archive (www.archive.org) neredeyse 300 terabytle'lık veriye sahip.
- Dünyadaki veritabanlarında depolanan veri miktarı her 20 ayda bir iki katına çıkıyor. Diğer büyüme oranları ise bundan daha fazla. Bu verilerden çok küçük bir miktarı insanlar tarafından inceleniyor. Bu yüzden böylesine devasa verilerden anlamlar çıkarmak için Bilgi Keşfine (Knowledge Discovery) ihtiyaç duyuluyor.
- Verinin boyutu her 10 katına çıktığında onu analiz etme biçimimizi tamamen yeniden şekillendirmek zorundayız.

# Veri Madenciliği

- Veri madenciliği veri tabanı sorgularından (database query) farklıdır. Geleneksel analizde, “X ürününün satışları Kasım ayında arttı mı?”, “X ürününün satışları Y ürününde bir promosyondan dolayı düştü mü?” gibi analizler yaparken veri madenciliğinde ise “X ürününün satışını belirleyen faktörler nelerdir?” gibi daha karmaşık ve sonuç odaklı analizler yapılarsınız.

- İş dünyasının anahtarı, kimsenin bilmediği bir şeyi bilmektir.
- *Anlamak, kalıpları algılamaktır.*

Diyelim ki X kişisi Amazon.com’dan bir kitap aldı.

- Görev: Bu kişiye almaya yatkın olabileceği başka kitaplar da öner.
- Amazon kitap satın alımları üzerinden bir kümeleme (clustering) işlemi yapıyor: “Advance in Knowledge Discovery and Data Mining” kitabını alan müşteriler ayrıca “Data Mining: Practical Machine Learning Tools and Techniques” kitabını da almaktadır. Öneri sistemleri son derece başarılıdır.

# Veri Madenciliği Nedir?

- Veri tabanında **bilgi keşfi (Knowledge Discovery in Databases-KDD)** veri yığınında
- **üstü kapalı** (saklı)
- **geçerli** (bulunan örüntüler yeni veriler üzerinde de geçerli olacak şekilde)
- **özgün** (beklenen değerlere kıyasla)
- **potansiyel olarak kullanışlı** (başarılı aksiyonlara neden olabilecek)
- **anlaşılabilir** (insanlar açısından)
- örüntülerin (pattern) bulunması ve tanımlanması olarak tanımlanabilecek, basit olmayan bir süreçtir.

# Veri madenciliği, KDD’nin yalnızca bir adımıdır.

- Veri madenciliği için bazı alternatif terimler kullanılabilir:
- Veritabanlarında (KDD) bilgi keşfi (madencilik),
- Bilgi çıkarma
- Veri/kalıp analizi
- Veri arkeolojisi
- Veri tarama
- Bilgi toplama
- İş zekası vb.

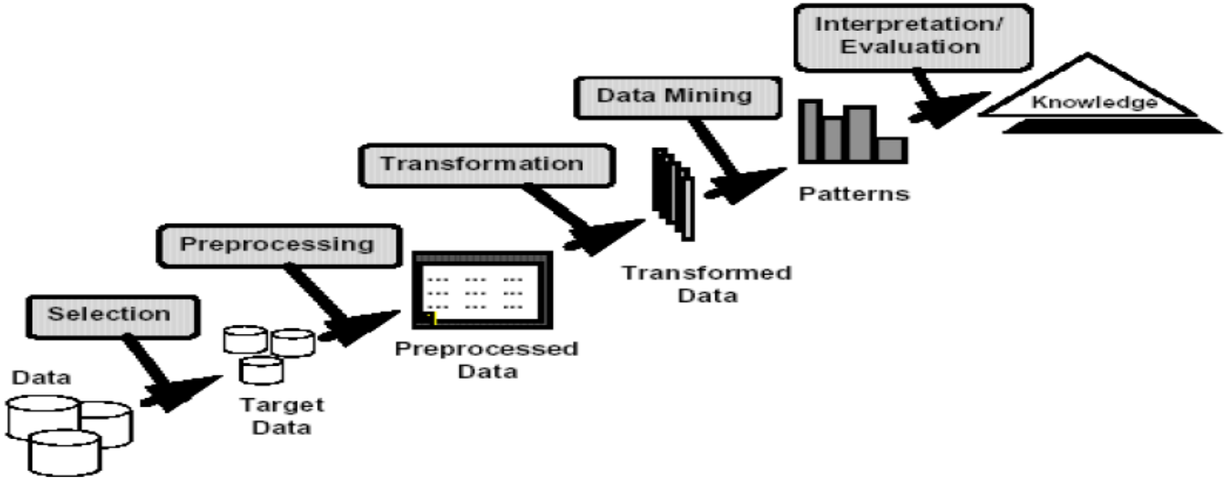
- Veri madenciliği için bazı alternatif terimler kullanılabilir:
- Knowledge discovery (mining) in databases (KDD),
- Knowledge extraction
- Data/patterns analysis
- Data archeology
- Data dredging
- Information harvesting
- Business intellingece, etc.

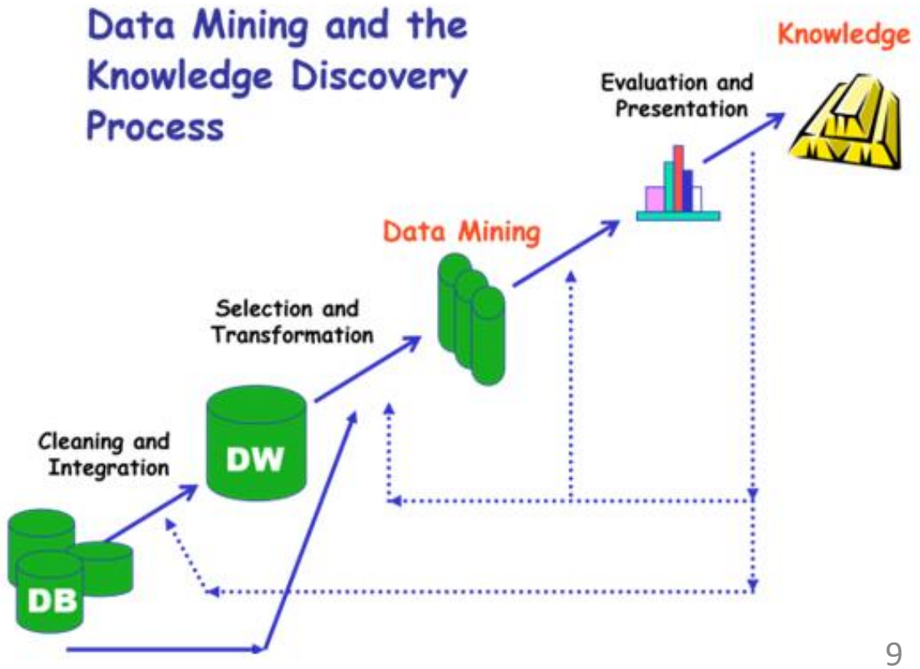
# Knowledge Discovery in Databases (KDD) süreci:

- Veri Madenciliği ve Bilgi Keşfi (Knowledge Discovery – KD) süreci bazı adımlardan oluşur. Bu adımlar şu şekilde özetlenebilir:
- Veri Temizliği (data cleaning): eksik değerlerin, gürültülü(noisy) verinin ve tutarsız(çelişkili) verinin temizlenmesi sürecidir.
- Veri Entegrasyonu (Data integration): Birden fazla veri kaynağından alınan verilerin birleştirilmesi sürecidir.
- Veri Seçimi (data selection): Yapılacak olan analize uygun(alakalı) verilerin seçilmesi sürecidir.
- Veri Dönüşümü (data transformation): Veriyi toplama (örneğin günlük satışları haftalık ya da aylık satışlara dönüştürme), veya genelleştirme (sokağı şehire; yaşı genç, orta, ve yaşlıya dönüştürme) gibi işlemler sürecidir.
- Veri Madenciliği (data mining): Elde edilen veriye akıllı algoritmalar uygulayarak örüntüleri (patterns) keşfetme sürecidir.
- Örüntü Değerlendirme (pattern evaluation): Elde edilen örüntülerin değerlendirilmesi ve hipotezlerin geçerliliğinin test edilmesi sürecidir.
- Bilgi Sunumu (knowledge presentation): Bulunan bilgileri (knowledge) görselleştirme (visualisation) ve sunum (representation) tekniklerini kullanarak kullanıcılara sunma sürecidir.

# What is Data Mining?

- Büyük verilerden örtük, önceden bilinmeyen ve faydalı bilgilerin önemsiz olmayan keşfi.



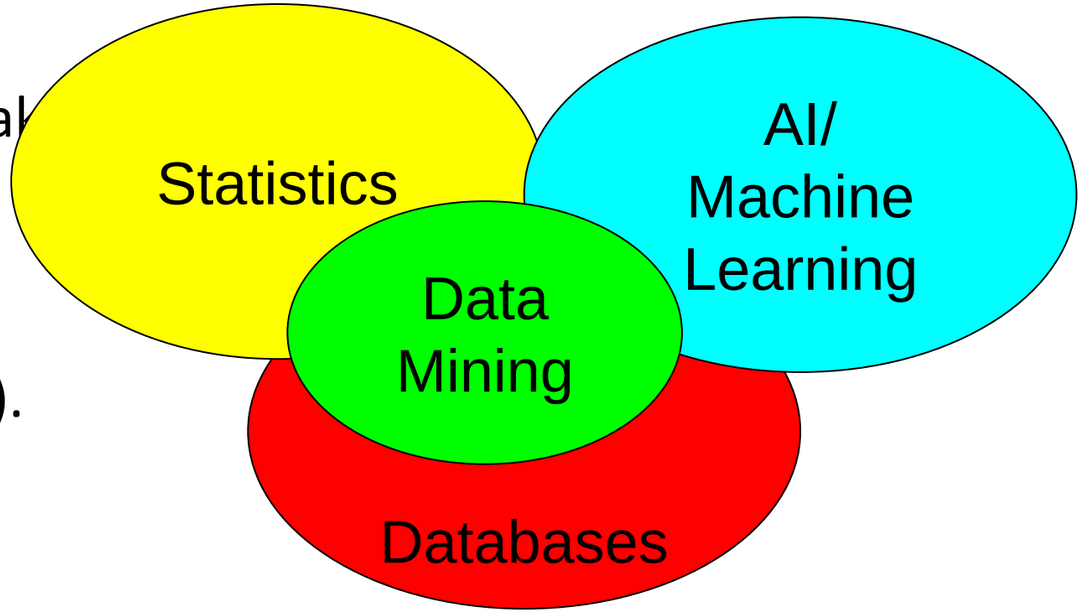



# *İstatistik, Makine Öğrenmesi ve Veri Madenciliği*

- *İstatistik daha çok teori odaklıdır ve hipotezlerin test edilmesine (hypotheses testing) odaklanır.*
- Makine Öğrenmesi, daha deneyimseldir (heuristic) ve öğrenme performansını iyileştirmeye odaklanır. Aynı zamanda robotik (robotics) ve gerçek zamanlı öğrenme (real-time learning) gibi alanları da kapsar ama bu alanlar veri madenciliği dışındaki alanlarıdır.
- Veri madenciliği ve bilgi keşfi, teoriyi(istatistik) ve deneyimi(otomatik öğrenme) birleştirir ve *bilgi keşfinin bütün süreçleri olan veri temizliği, öğrenme, entegrasyon ve görselleştirme alanlarını içerir.*
- İstatistik, Makine Öğrenmesi ve Veri Madenciliği kavramları arasındaki sınırlar son derece belirsizdir.
- Veri madenciliği bazen "ikincil veri analizi" olarak adlandırılır. Veri madenciliğinde kullanılan algoritmaların çoğu istatistikçiler ve veri madencileri tarafından ortaklaşa kullanılır. *Veri madenciliği örüntü tespitini (pattern detection) hedeflerken istatistik bulunan örüntülerin gerçekliğini belirlemeyi hedefler.* Örneğin, aynı hastalıktan muzdarip olan insanlar kümesinin bulunması, fakat bu hastalığın tamamen rastgele olup olmadığının tespiti.

# Cultures

- Databases:
  - büyük ölçekli (ana bellek dışı) verilere odaklanmak.
- AI (machine-learning):
  - karmaşık yöntemlere, küçük verilere odaklanmak
- Statistics:
  - Modellere odaklanmak (concentrate on models).



# Modeller ve Analitik İşleme

- Bir veritabanı çalışanı için veri madenciliği, büyük miktarda veriyi inceleyen sorgulamalardaki analitik işlemenin uç bir biçimidir.
  - Sonuç, sorgulamanın yanıtıdır.

- Bir istatistikçi için veri madenciliği, modellerin çıkarımıdır.
  - Sonuç, modelin parametreleridir.

# (Way too Simple) Example

- Bir milyar sayı verildiğinde, bir Veri Tabanı Uzmanı olan (DB) kişi ortalama ve standart sapmalarını hesaplayacaktır.
- Bir istatistikçi, en iyi Gauss dağılımına milyar adet veriyi sığdırabilir ve bu dağılımın ortalamasını ve standart sapmasını hesaplayıp rapor edebilir.

# Veri Madenciliği Görevleri

- Birliktelik kuralı keşfi
- Sınıflandırma
- Kümeleme
- Öneri sistemleri:
- İşbirliğine dayalı filtreleme
- Bağlantı analizi ve grafik madenciliği
- Web reklamlarını yönetme
- … …

# Öneri Sistemleri

- Netflix
  - Movie recommendation
- Amazon
  - Book recommendation

# Bağlantı Analizi ve Grafik madenciliği

- Sayfa Sıralaması
- Bağlantı tahmini
- Topluluk algılama

# Cevapların Anlamlılığı

- Büyük bir veri madenciliği riski, anlamsız kalıpları “keşfedeceksiniz”.
- İstatistikçiler buna Bonferroni ilkesi diyorlar: (kabaca) ilginç desenler için veri miktarınızın destekleyeceğinden daha fazla yere bakarsanız, saçmalık bulmanız kaçınılmazdır.

# Bonferroni Prensibi Örnekleri

1. Toplam Bilgi Farkındalığına (TIA:Total Information Awareness ) büyük bir itiraz o kadar çok belirsiz bağlantı arıyordu ki, sahte olan ve dolayısıyla masumların mahremiyetini ihlal eden şeyler bulacağından emindi.
2. Ren Paradoksu: Bilimsel araştırmanın nasıl yapılmayacağının harika bir örneği.

# Uzaktan İnsan Tanımlama (HumanID) projesi

Uzaktan İnsan Tanımlama (HumanID) projesi , "kuvvet koruması", suç önleme ve "yurt güvenliği/savunma" amaçları için büyük mesafelerdeki insanları tespit etmek, tanımak ve tanımlamak için otomatik biyometrik tanımlama teknolojileri geliştirildi . HumanID'nin hedefleri şunlardı:

- 150 metreye (500 ft) kadar olan nesneleri bulmak ve elde etmek için algoritmalar geliştirilmesi.
- Yüz ve yürüyüş tanımayı 7/24 insan tanımlama sisteminde birleştirilmesi.
- Görünür görüntüleri kullanarak 150 metreye (500 ft) kadar uzanan bir insan tanımlama sistemi geliştirilmesi ve gösterilmesi.
- Geniş görüş alanı algılama ve dar görüş alanı yürüyüş sınıflandırması için düşük güçlü bir milimetre dalga radar sistemi geliştirilmesi.
- Uzaktan insan tanımlaması için videodan yürüyüş performansını karakterize edilmesi.
- Çok spektral bir kızılötesi ve görünür yüz tanıma sistemi geliştirilmesi.

# Biyolojik Gözetim

Bio-Surveillance projesi, hayvan nöbetçileri, davranışsal göstergeler ve teşhis öncesi tıbbi veriler gibi geleneksel olmayan veri kaynaklarını izleyerek biyoterörizmi tahmin etmek ve yanıt vermek için tasarlanmıştır . Mevcut hastalık modellerinden yararlanarak, anormal sağlık erken göstergelerini belirleyecek ve anormal sağlık koşulları için en değerli erken göstergeleri belirlemek için mevcut veri tabanlarını araştıracaktır.

Gözetim kapsamı: Sanal, merkezileştirilmiş, büyük bir veri tabanı" olarak gözetim kapsamı, mülk kayıtlarını, kredi kartı satın alımlarını, dergi aboneliklerini, web tarama geçmişlerini , telefon kayıtlarını, akademik notları , banka mevduatlarını , kumar geçmişlerini, pasaport uygulamalarını, havayolu ve demiryolu biletlerini, ehliyetler , silah ruhsatları , otoyol geçiş kayıtları , adli sicil kayıtları ve boşanma kayıtları.

Sağlık ve biyolojik bilgiler TIA, ilaç reçetelerini, tıbbi kayıtları, parmak izleri, yürüyüş, yüz ve iris verileri,  ve DNA'yı içermektedir.

# The "TIA" Story

- Bazı kötü niyetli grupların ara sıra otellerde kötülük yapmayı planlamak için toplandıklarına inandığımızı varsayalım.

- Aynı gün aynı otelde en az iki kez kalmış (ilgisiz) kişileri bulmak istiyoruz.

# The "TIA" Story

- $10^9$ kişi izleniyor (1.000.000.000: Bir milyar kişi).
- 1000 günde.
- Bir kişi zamanın %1'inde bir otelde kalır. Bir kişi 1000 günde 10 gün otelde kalıyor.
- Bir kişi 1/100 gün otelde kalyor.
- Bir milyar kişi 1.000.000.000*1/100=10.000.000 gün otelde kalıyor.
- Oteller 100 kişilik ise 10.000.000/100= $10^5$ hotels (100.000 hotels).
- Herkes rastgele davranırsa (yani, kötü niyetli kimseler yok) veri madenciliği şüpheli bir şey tespit edecek mi?

# The "TIA" Story

- p ve q'nun belirli bir günde aynı otelde olma olasılığı:
  - $(1/100) \times (1/100) \times (1/ 10^5 ) = 10^{-9}$
- p ve q'nun iki gün kadar aynı otelde olma olasılığı:
  - $5 \times 10^5 \times (10^{-9} \times 10^{-9}) = 5 \times 10^{-13}$.
  - (Pairs of days is $5 \times 10^5$ )
- Pairs of people:
  - $5 \times 10^{17}$.
- Expected number of "suspicious" pairs of people:
  - $5 \times 10^{17} \times 5 \times 10^{-13} = 250{,}000$.

# Conclusion

- Diyelim ki aynı otelde kesinlikle iki kez kalan 10 çift kötülük var..

- Analistler, 10 gerçek vakayı bulmak için 250.010 adayı elemek zorunda. Olmayacak. Ama düzeni nasıl iyileştirebiliriz?

# Moral

- Bir otel ararken (örneğin, “aynı otelde iki kez kalan iki kişi”), otellerin o kadar çok olasılığa izin vermediğinden emin olun ki, rastgele veriler kesinlikle “ilgi çekici” gerçekler üretecektir

# Rhine Paradox

- Rhine Paradoksu bir bilimsel araştırmanın nasıl yapılmaması gerektiğinin harika bir örneği.
- David Rhine bir parapsikologdu. 1950 yılında bir hipotez geliştirdi ve hipotezi de şuydu: Bazı insanlar 6. Hisse (Extra-Sensory Percetion-ESP) sahiptir. Bunu test etmek için bir deney tasarladı ve deneklere 10 adet kartın arka yüzlerinde kırmızı mı yoksa mavi renk mi olduğunu tahmin etmelerini istedi. Deneyinin sonucunda gördü ki neredeyse her 1000 kişiden 1'i 6. hisse sahipti, çünkü 10 kartın 10'unun da rengini doğru bilmişti!
- Daha sonra bu testi geçen insanlara 6. Hislerinin olduğunu söyledi ve onları tekrar test etmek için farklı bir deneye davet etti. Fakat bu sefer fark etti ki bu deneyde deneklerin neredeyse hepsi 6. Hislerini kaybetmişti. Ve testi nasıl mı sonuçlandırdı?
- "İnsanlara 6. Hisleri olduğunu söylememelisiniz; çünkü bu 6. Hislerini kaybetmelerine neden oluyor."
- Peki gerçekte ne oldu? Kırmızı veya mavi renklerden oluşan 10 kartın tam olaram 1024(2^10) kombinasyonu vardır. Bu yüzden her 1000 kişiden 1'inin 10 kartın rengini doğru tahmin etme olasılığı 1000/1024 = 0.98 yani %98'dir! Sanırım daha fazla açıklamaya gerek yok

# Rhine Paradox

- Joseph Rhine, 1950'lerde bazı insanların Ekstra Duyusal Algıya (ESP) sahip olduğunu varsayan bir parapsikologdu.Deneklerden kırmızı veya mavi olmak üzere 10 gizli kart tahmin etmelerinin istendiği bir deney tasarladı (gibi bir şey).Neredeyse 1000'de 1'inin ESP'ye sahip olduğunu keşfetti - 10'unu da doğru yapabildiler!
- Bu insanlara ESP'leri olduğunu söyledi ve onları aynı tipte başka bir test için çağırdı.Ne yazık ki, neredeyse hepsinin ESP'lerini kaybettiğini keşfetti.Ne sonuca vardı?
- İnsanlara ESP sahibi olduklarını söylememeniz gerektiği sonucuna vardı; kaybetmelerine neden olur.
- Bonferroni İlkesini anlamak, bir parapsikologdan biraz daha az aptal görünmenize yardımcı olacaktır.

# Uygulamalar

- Bankacılık: kredi/kredi kartı onayı: Eski müşterilere dayalı olarak iyi müşterileri tahmin edilmesi.
- Müşteri ilişkileri yönetimi: Bir rakip için ayrılma olasılığı yüksek olanları belirlenmesi.
- Hedefli pazarlama: Promosyonlara olası yanıt verenlerin belirlenmesi.
- Dolandırıcılık tespiti: Çevrimiçi bir olay akışından dolandırıcılık olaylarını tanımlanması.
- Üretim: Süreç parametresi değiştiğinde düğmelerin otomatik olarak ayarlanması.
- Tıp: Hastalık sonucu, tedavilerin etkinliği. Hastanın hastalık geçmişinin analiz edilmesi: Hastalık arasındaki ilişkiyi bulunması.
- Bilimsel veri analizi: Gen analizi
- Web sitesi/mağaza tasarımı ve tanıtımı: Ziyaretçinin sayfalara yakınlığının bulunması ve düzenin değiştirilmesi.

# Verı Madenciliği Uygulamaları

**Pazar analizi ve yönetimi**

- Hedef kitle seçimi, müşteri ilişkileri yönetimi, pazar basket analizi, çapraz satış, pazar segmentasyonu, vb.
- Aynı karakteristiklere sahip olan (örneğin ilgi alanı, gelir seviyesi, harcama alışkanlıkları, vs.) müşterileri bulma ve kümeleme
- Zaman boyunca müşterilerin satın alma örüntülerini (purchasing pattern) belirleme

**Risk analizi ve yönetimi**

- Tahminleme (forecasting), müşteri tutumu (customer retention), kalite kontrolü, rekabet analizi, kredi puanlama

**Sahtekarlık tespiti ve yönetimi**

Geçmişte gerçekleştirilen sahtekarlık işlemlerini kullanarak model geliştirme ve belirli davranışlarda bulunan müşterileri önceden tahminleme.

- Araba sigortası: sigorta parası almak için “sahte olarak” kaza yapan insanların tespiti
- Kara para aklama: şüpheli para transferlerini tespit etme
- Telefon araması modeli: aramanın hedefi, süresi, günü, saati, vs. Beklenen normlardan sapan örüntülerin analizi.
- British Telecom’un başına bununla ilgili bir olay gelmiş. Olayda, belirli kişilerden oluşan bir takım grup sürekli mobil telefonlarından grup içindeki kişileri arayarak birkaç milyon dolarlık bir kaçakçılık gerçekleştirmişler. Mesela, hapisanedeki mahkumlardan biri dışarıdaki bir arkadaşına terk edilmiş bir ev için telefon hattı kurdurtuyor. Aramalar mahkumun 3 eyalet ötedeki kız arkadaşına yönlendiriliyor. Bu sayede telefon şirketi bunu 90 gün sonra fark edinceye kadar bedava telefon görüşmesi yapıyorlar.

# Veri Madenciliğinin Görevleri

**Birliktelik (association):**

- Karşılıklı ilişki (correlation) ve nedensellik (causality)
- Çok boyutlu veya tek boyutlu birlikteliklerin tespit edilmesi.

**Sınıflandırma (classification) ve Tahminleme (prediction):**

- Sınıfları veya konseptleri tanımlayan ve ayrıştıran modellerin(fonksiyonları) bulunması. Örneğin, iklime göre ülkeleri sınıflandırma, kilometre başına yakıt tüketimine göre arabaları sınıflandırma, vb.

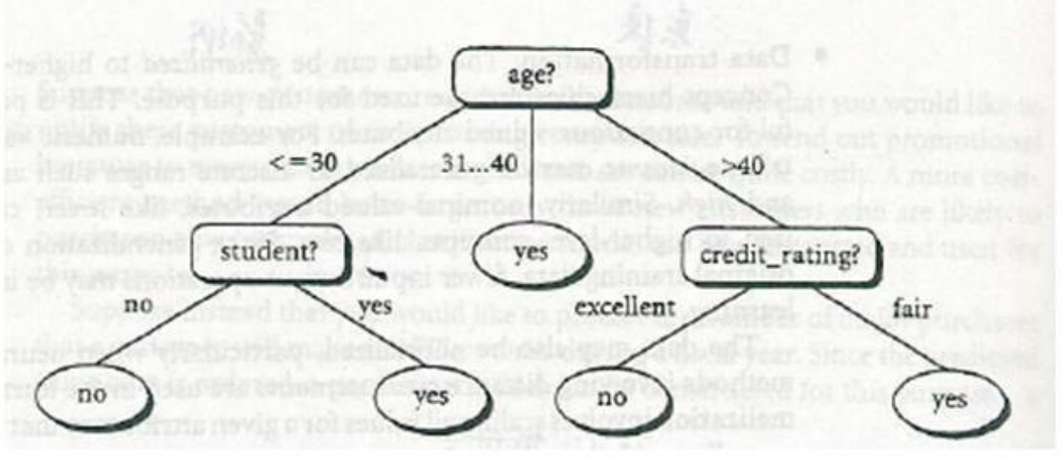
**Sunum: K**arar ağacı (decision tree), sınıflandırma kuralı, yapay sinir ağları (neural network)

**Tahminleme: B**azı bilinmeyen veya eksik olan sayısal verilerin tahminlenmesi.

# Veri Madenciliğinin Görevleri

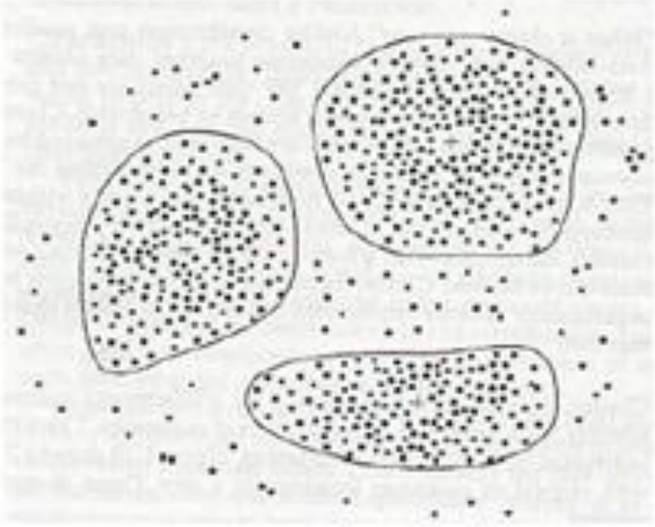
- Aşağıdaki tablo bir “öğretici küme” (training set) örneğini temsil ediyor. Bu veri üzerinde karar ağacı sınıflandırma algoritmasını uyguladığımızı düşünelim.
- Elde edeceğimiz sonuç aşağıdaki gibi bir sonuç olacak. Yani aşağıdaki sonuçlar diyor ki yaş faktörü bilgisayar alınmasındaki en önemli faktör. Yaşı 31..40 arasında olanların hepsi bilgisayar alıyor. Yaşı 30’dan küçük olan insanlar için ikinci bir ayrım yapılıyor ve öğrenci olup olmadığı sorgulanıyor. Eğer öğrenciyse bilgisayar alıyor; öğrenci değilse bilgisayar almıyor. Yaşı 40’tan büyük olan insanlar ise kredi derecelendirmelerine göre ayrıma uğratılıyor. Kredi derecesi mükemmel olanlar bilgisayar almazken normal düzeyde olanlar bilgisayar alıyor.

| age | income | student | credit_rating | buys_computer |
|---|---|---|---|---|
| <=30 | high | no | fair | no |
| <=30 | high | no | excellent | no |
| 30…40 | high | no | fair | yes |
| >40 | medium | no | fair | yes |
| >40 | low | yes | fair | yes |
| >40 | low | yes | excellent | no |
| 31…40 | low | yes | excellent | yes |
| <=30 | medium | no | fair | no |
| <=30 | low | yes | fair | yes |
| >40 | medium | yes | fair | yes |
| <=30 | medium | yes | excellent | yes |
| 31…40 | medium | no | excellent | yes |
| 31…40 | high | yes | fair | yes |
| >40 | medium | no | excellent | no |



# Küme Analizi (Cluster analysis)

- Sınıflandırma'nın aksine küme analizinde kümelerin etiketleri (label) bilinmiyor. Küme analizi genellikle sınıfların belirlenmesi ve tanımlanması için kullanılıyor.
- Kümeleme (clustering) işlemi şu prensip ile çalışıyor: bir sınıf (class) içindeki benzerlikleri maksimize et ve sınıflar arasındaki benzerliği minimize et.



# Aykırı Gözlem

- Aykırı Gözlem (Outlier) Tespiti:  Verinin genel davranışına uymayan örneklere aykırı gözlem (outlier) adı veriliyor. Bunlar gürültü (noise) veya istisna (exception) olarak düşünülebilir; fakat sahtekarlık gibi nadir olayların (aykırılıkların) analizinde oldukça kullanışlıdırlar.
- Trend, Sapma ve değişim (evrim, evolution) Analizi: regresyon analizi (regression analysis)
- Sıraları örüntü madenciliği (sequential pattern mining), dönemsellik analizi
- Benzerlik tabanlı analiz
- Görselleştirmenin Gücü: Tüm işlemler sonucu elde ettiğimiz sonuçları görselleştirmemiz gerekmektedir.

# Öngörülü Modelleme: Sınıflandırma

# Examples of Classification Task

- Tümör hücrelerinin iyi huylu veya kötü huylu olarak tahmin edilmesi
- Proteinin ikincil yapılarını alfa sarmalı, beta yaprağı veya rastgele bobin olarak sınıflandırma
- Proteinlerin tahmin etme işlevleri
- Kredi kartı işlemlerinin yasal veya hileli olarak sınıflandırılması
- Haberleri finans, hava durumu, eğlence, spor vb. olarak kategorize etme
- Siber uzaydaki davetsiz misafirleri belirleme

# Commonly Used Classification Models

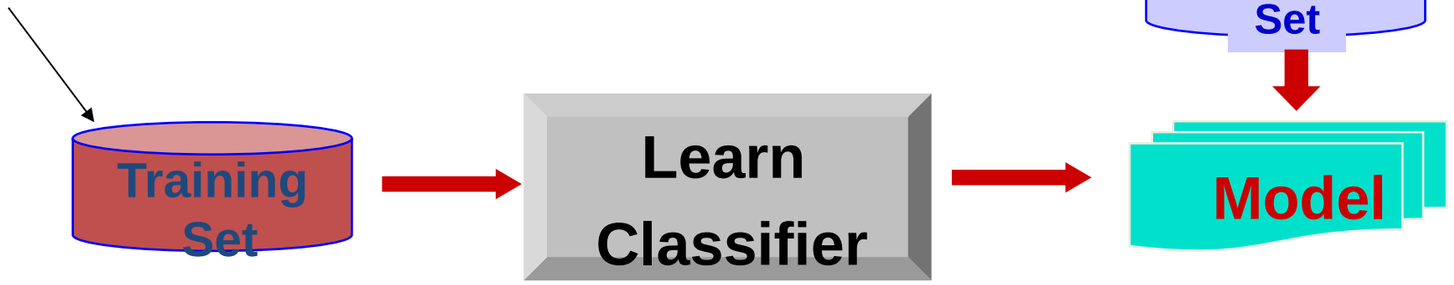
- **Temel Sınıflandırıcılar**
  - Decision Tree based Methods
  - Rule-based Methods
  - Nearest-neighbor
  - Neural Networks
  - Naïve Bayes and Bayesian Belief Networks
  - Support Vector Machines

- **Topluluk Sınıflandırıcıları**
  - Boosting, Bagging, Random Forests

# Bir Sınıflandırma Modeli Oluşturmak İçin Genel Yaklaşım

| | | categorical | categorical | quantitative | class |
|---|---|---|---|---|---|
| *Tid* | Employed | Level of Education | # years at present address | Credit Worthy |
| 1 | Yes | Graduate | 5 | Yes |
| 2 | Yes | High School | 2 | No |
| 3 | No | Undergrad | 1 | No |
| 4 | Yes | High School | 10 | Yes |
| … | … | … | … | … |

| *Tid* | Employed | Level of Education | # years at present address | Credit Worthy |
|---|---|---|---|---|
| 1 | Yes | Undergrad | 7 | ? |
| 2 | No | Graduate | 3 | ? |
| 3 | Yes | High School | 2 | ? |
| … | … | … | … | … |

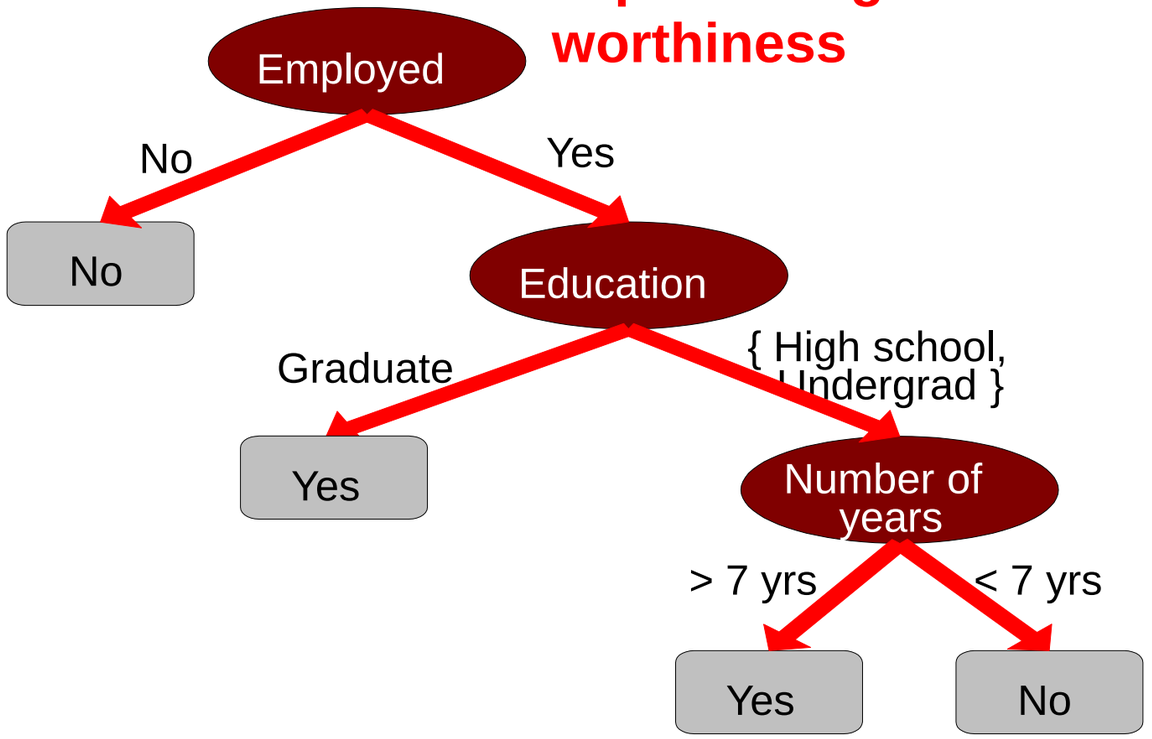


# Classification Model: Decision Tree

**Class**

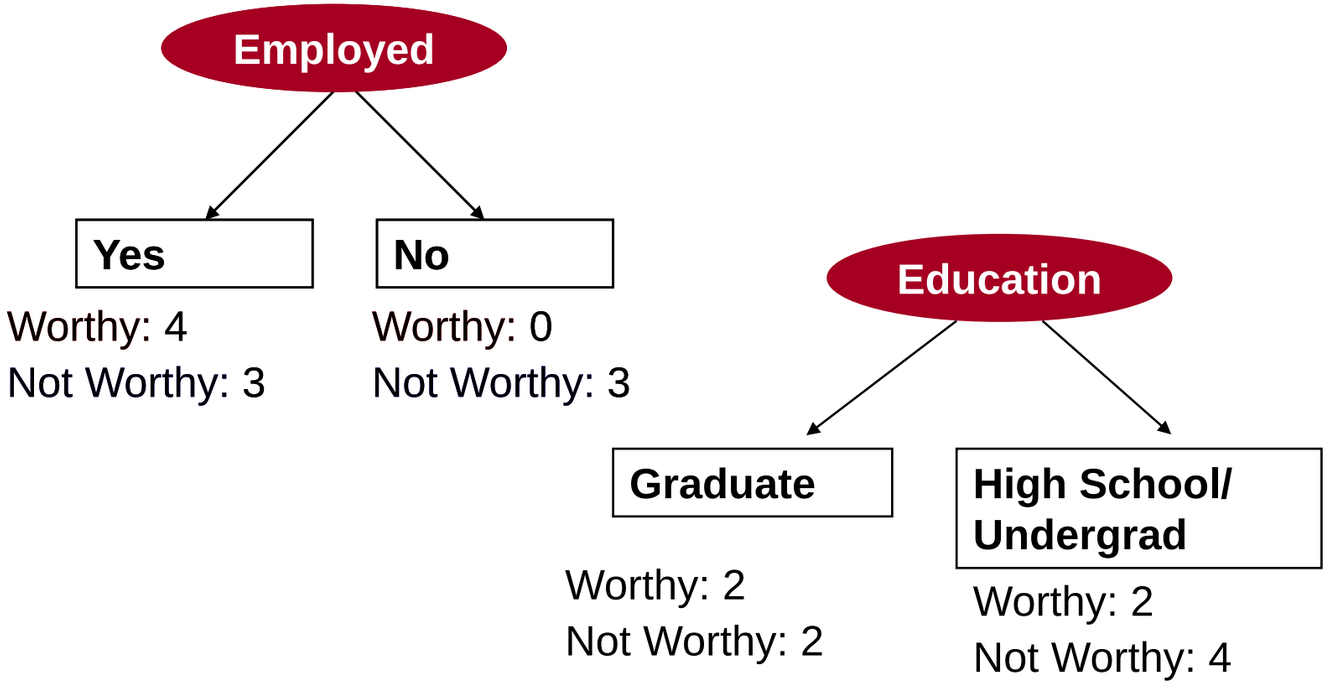
| *Tid* | Employed | Level of Education | # years at present address | Credit Worthy |
|---|---|---|---|---|
| 1 | Yes | Graduate | 5 | Yes |
| 2 | Yes | High School | 2 | No |
| 3 | No | Undergrad | 1 | No |
| 4 | Yes | High School | 10 | Yes |
| … | … | … | … | … |

**Model for predicting credit worthiness**



# Constructing a Decision Tree

| *Tid* | Employed | Level of Education | # years at present address | Credit Worthy |
|---|---|---|---|---|
| 1 | Yes | Graduate | 5 | Yes |
| 2 | Yes | High School | 2 | No |
| 3 | No | Undergrad | 1 | No |
| 4 | Yes | High School | 10 | Yes |
| 5 | Yes | Graduate | 2 | No |
| 6 | No | High School | 2 | No |
| 7 | Yes | Undergrad | 3 | No |
| 8 | Yes | Graduate | 8 | Yes |
| 9 | Yes | High School | 4 | Yes |
| 10 | No | Graduate | 1 | No |




**Key Computation**

Employed = Yes

Employed = No

| Worthy | Not Worthy |
|---|---|
| 4 | 3 |
| 0 | 3 |

# Constructing a Decision Tree

| *Tid* | Employed | Level of Education | # years at present address | Credit Worthy |
|---|---|---|---|---|
| 1 | Yes | Graduate | 5 | Yes |
| 2 | Yes | High School | 2 | No |
| 3 | No | Undergrad | 1 | No |
| 4 | Yes | High School | 10 | Yes |
| 5 | Yes | Graduate | 2 | No |
| 6 | No | High School | 2 | No |
| 7 | Yes | Undergrad | 3 | No |
| 8 | Yes | Graduate | 8 | Yes |
| 9 | Yes | High School | 4 | Yes |
| 10 | No | Graduate | 1 | No |

Employed = Yes

| *Tid* | Employed | Level of Education | # years at present address | Credit Worthy |
|---|---|---|---|---|
| 1 | Yes | Graduate | 5 | Yes |
| 2 | Yes | High School | 2 | No |
| 4 | Yes | High School | 10 | Yes |
| 5 | Yes | Graduate | 2 | No |
| 7 | Yes | Undergrad | 3 | No |
| 8 | Yes | Graduate | 8 | Yes |
| 9 | Yes | High School | 4 | Yes |

Employed = No

| *Tid* | Employed | Level of Education | # years at present address | Credit Worthy |
|---|---|---|---|---|
| 3 | No | Undergrad | 1 | No |
| 6 | No | High School | 2 | No |
| 10 | No | Graduate | 1 | No |

# Karar Ağacı Tümevarımının Tasarım Konuları

- How should training records be split?
  - Method for specifying test condition
    - depending on attribute types
  - Measure for evaluating the goodness of a test condition

- How should the splitting procedure stop?
  - Stop splitting if all the records belong to the same class or have identical attribute values
  - Early termination

# Karar Ağacı Tümevarımının Tasarım Konuları

- Eğitim kayıtları nasıl bölünmelidir?
  - Nitelik türlerine bağlı olarak test koşulunu belirleme yöntemi
  - Bir test koşulunun iyiliğini değerlendirmek için ölçü
- Bölme işlemi nasıl durdurulmalı?
  - Tüm kayıtlar aynı sınıfa aitse veya aynı öznitelik değerlerine sahipse bölmeyi durdur
  - Erken sonlandırma

# En İyi Bölme Nasıl Belirlenir

- Açgözlü yaklaşım:Daha saf sınıf dağılımına sahip düğümler tercih edilir.
- Düğüm kirliliğinin bir ölçüsüne ihtiyacınız var:

C0: 5
C1: 5

Yüksek derecede kirlilik (impurity)

C0: 9
C1: 1

Düşük kirlilik derecesi

# Measure of Impurity: GINI

- Gini Index for a given node t :

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

(NOTE: $p(j \mid t)$ is the relative frequency of class j at node t).

  - Maximum (1 - 1/$n_c$) when records are equally distributed among all classes, implying least interesting information
  - Minimum (0.0) when all records belong to one class, implying most interesting information

# Measure of Impurity: GINI

- Gini Index for a given node t :

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

(NOTE: $p(j \mid t)$ is the relative frequency of class j at node t).

  - For 2-class problem (p, 1 – p):
    - GINI = $1 - p^2 - (1 - p)^2 = 2p\,(1\text{-}p)$

| C1 | 0 |
|---|---|
| C2 | 6 |
| Gini=0.000 | |

| C1 | 1 |
|---|---|
| C2 | 5 |
| Gini=0.278 | |

| C1 | 2 |
|---|---|
| C2 | 4 |
| Gini=0.444 | |

| C1 | 3 |
|---|---|
| C2 | 3 |
| Gini=0.500 | |

# Computing Gini Index of a Single Node

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

| C1 | 0 |
|---|---|
| C2 | 6 |

**P(C1) = 0/6 = 0  P(C2) = 6/6 = 1**

**Gini = 1 – P(C1)$^2$ – P(C2)$^2$ = 1 – 0 – 1 = 0**

| C1 | 1 |
|---|---|
| C2 | 5 |

**P(C1) = 1/6  P(C2) = 5/6**

**Gini = 1 – (1/6)$^2$ – (5/6)$^2$ = 0.278**

| C1 | 2 |
|---|---|
| C2 | 4 |

**P(C1) = 2/6  P(C2) = 4/6**

**Gini = 1 – (2/6)$^2$ – (4/6)$^2$ = 0.444**

# Computing Gini Index for a Collection of Nodes

- When a node p is split into k partitions (children)

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

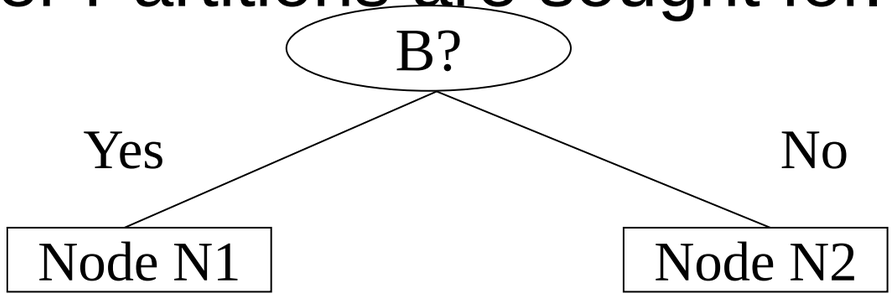
where, $n_i$ = number of records at child i,
n = number of records at parent node p.

- Choose the attribute that minimizes weighted average Gini index of the children

- Gini index is used in decision tree algorithms such as CART, SLIQ, SPRINT

# Binary Attributes: Computing GINI Index

- Splits into two partitions
- Effect of Weighing partitions:
  - Larger and Purer Partitions are sought for.




| | Parent |
|---|---|
| C1 | 7 |
| C2 | 5 |
| Gini = 0.486 | |

**Gini(N1)**
**= 1 – (5/6)² – (1/6)²**
**= 0.278**

**Gini(N2)**
**= 1 – (2/6)² – (4/6)²**
**= 0.444**

| | N1 | N2 |
|---|---|---|
| C1 | 5 | 2 |
| C2 | 1 | 4 |
| Gini=0.361 | | |

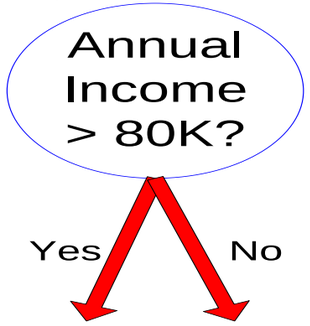
Weighted Gini of N1 N2
= 6/12 * 0.278 +
6/12 * 0.444
= 0.361

**Gain = 0.486 – 0.361 = 0.125**

# Continuous Attributes: Computing Gini Index

- Use Binary Decisions based on one value
- Several Choices for the splitting value
  - Number of possible splitting values = Number of distinct values
- Each splitting value has a count matrix associated with it
  - Class counts in each of the partitions, A < v and A ≥ v
- Simple method to choose best v
  - For each v, scan the database to gather count matrix and compute its Gini index
  - Computationally Inefficient! Repetition of work.

| ID | Home Owner | Marital Status | Annual Income | Defaulted |
|---|---|---|---|---|
| 1 | Yes | Single | 125K | No |
| 2 | No | Married | 100K | No |
| 3 | No | Single | 70K | No |
| 4 | Yes | Married | 120K | No |
| 5 | No | Divorced | 95K | Yes |
| 6 | No | Married | 60K | No |
| 7 | Yes | Divorced | 220K | No |
| 8 | No | Single | 85K | Yes |
| 9 | No | Married | 75K | No |
| 10 | No | Single | 90K | Yes |

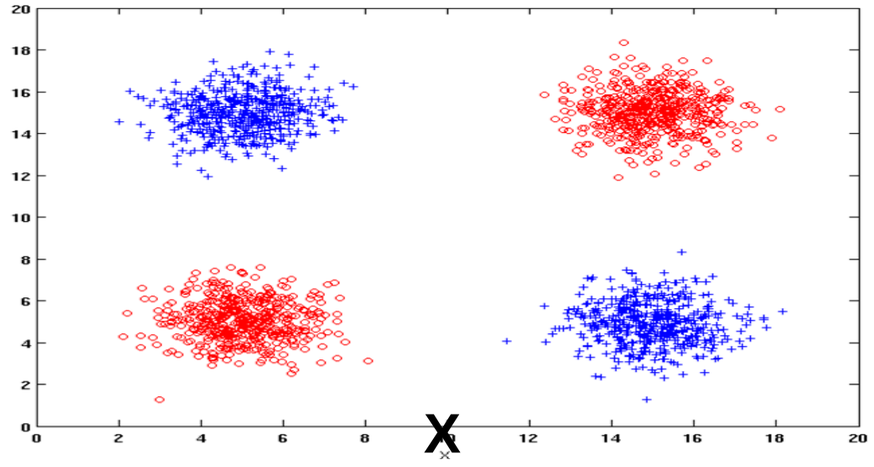



| | ≤ 80 | > 80 |
|---|---|---|
| Yes | 0 | 3 |
| No | 3 | 4 |

# Decision Tree Based Classification

- Advantages:
  - Inexpensive to construct
  - Extremely fast at classifying unknown records
  - Easy to interpret for small-sized trees
  - Robust to noise (especially when methods to avoid overfitting are employed)
  - Can easily handle redundant or irrelevant attributes (unless the attributes are interacting)
- Disadvantages:
  - Space of possible decision trees is exponentially large. Greedy approaches are often unable to find the best tree.
  - Does not take into account interactions between attributes
  - Each decision boundary involves only a single attribute

# Handling interactions



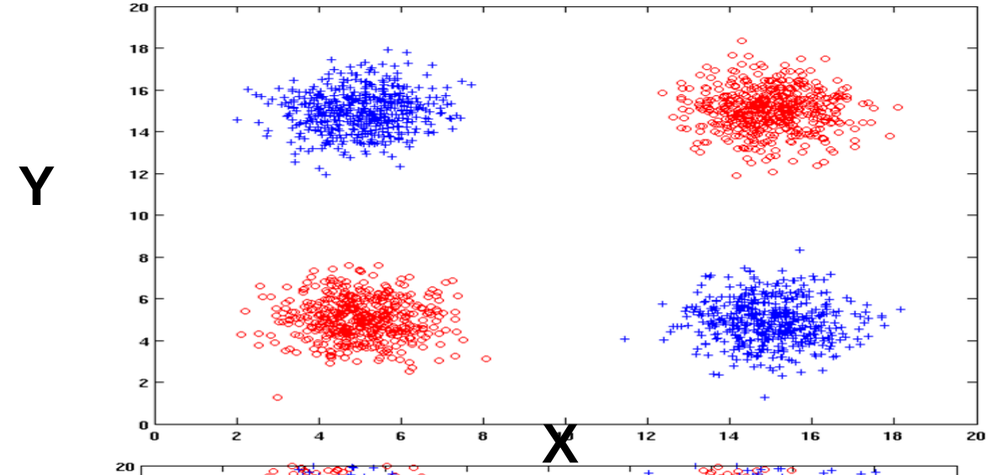

**+ : 1000 instances**

**o : 1000 instances**

**Entropy (X) : 0.99**

**Entropy (Y) : 0.99**

# Handling interactions



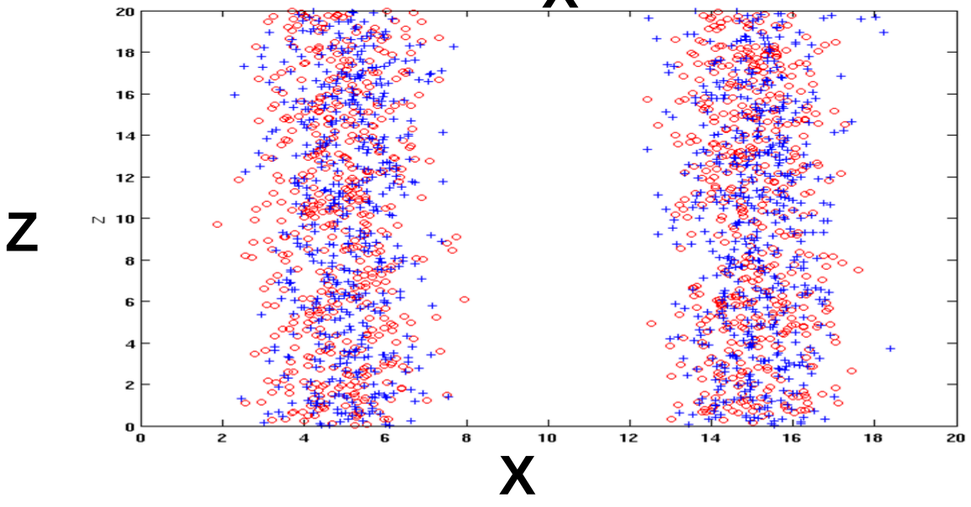

**+ : 1000 instances**

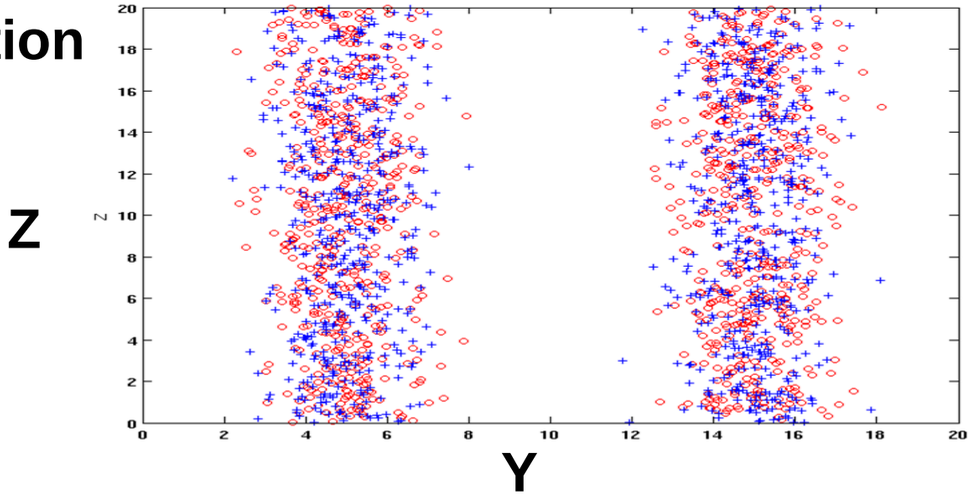
**o : 1000 instances**

**Adding Z as a noisy attribute generated from a uniform distribution**

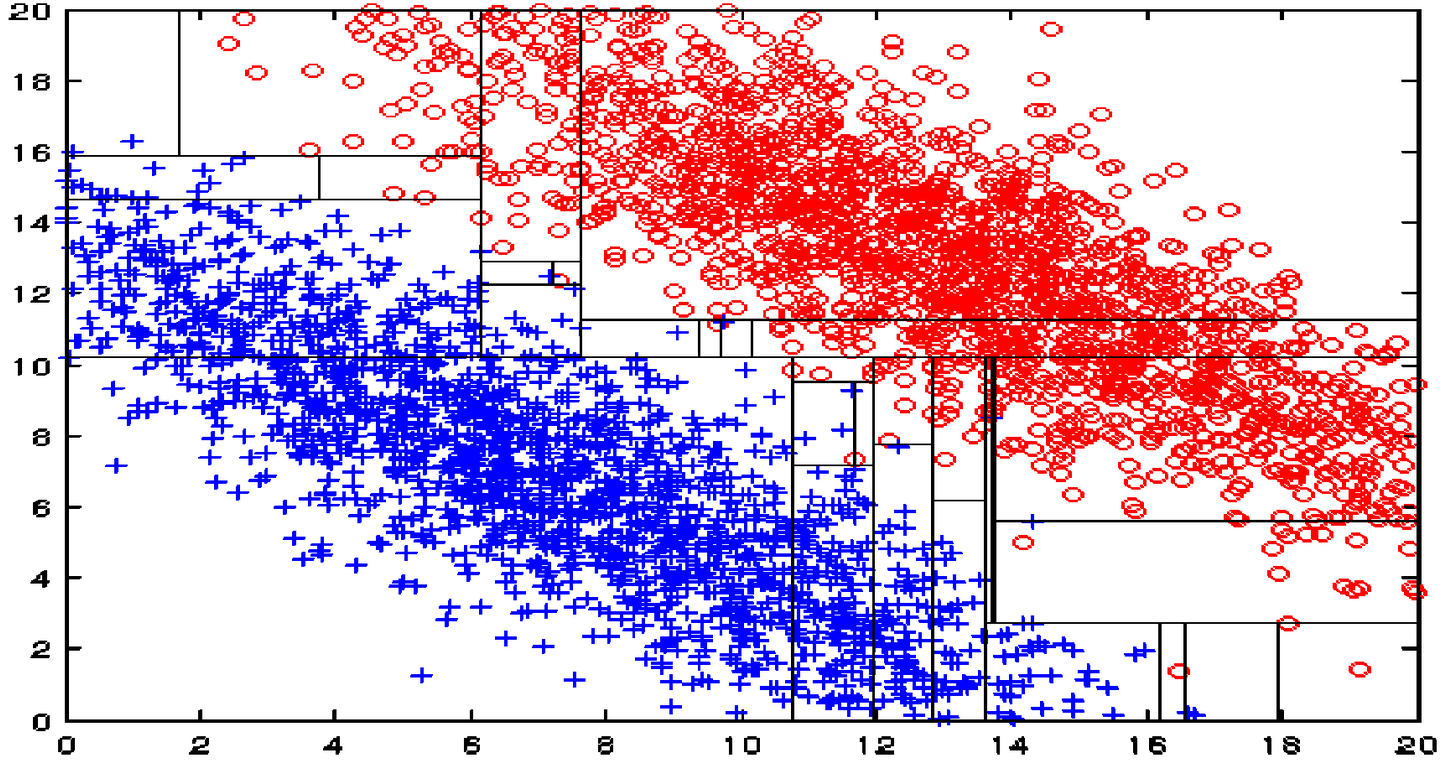
**Entropy (X) : 0.99**
**Entropy (Y) : 0.99**
**Entropy (Z) : 0.98**

**Attribute Z will be chosen for splitting!**



# Limitations of single attribute-based decision boundaries




**Both positive (+) and negative (o) classes generated from skewed Gaussians with centers at (8,8) and (12,12) respectively.**

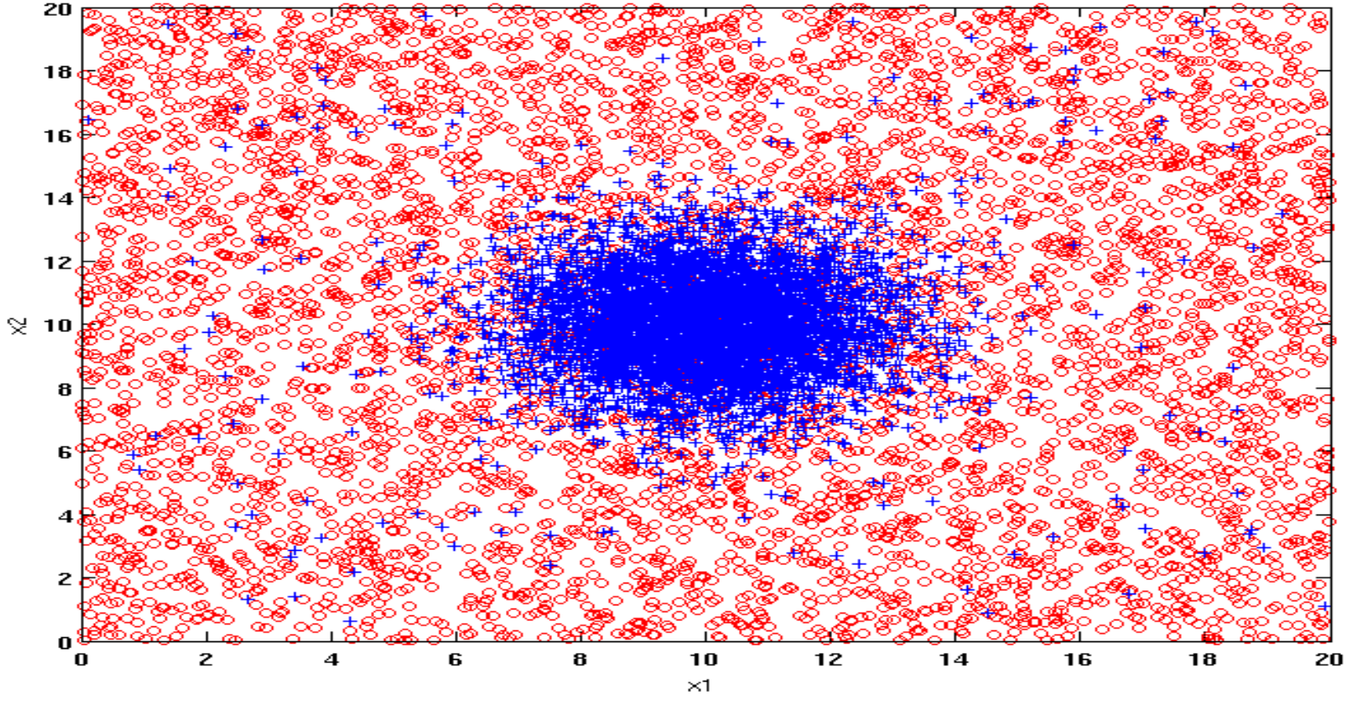
# Aşırı Uyum Modeli

Model Overfitting

# Sınıflandırma Hataları

- Training errors (apparent errors)
  - Errors committed on the training set
- Test errors
  - Errors committed on the test set
- Generalization errors
  - Expected error of a model over random selection of records from same distribution

# Example Data Set




**Two class problem:**

**+ : 5200 instances**

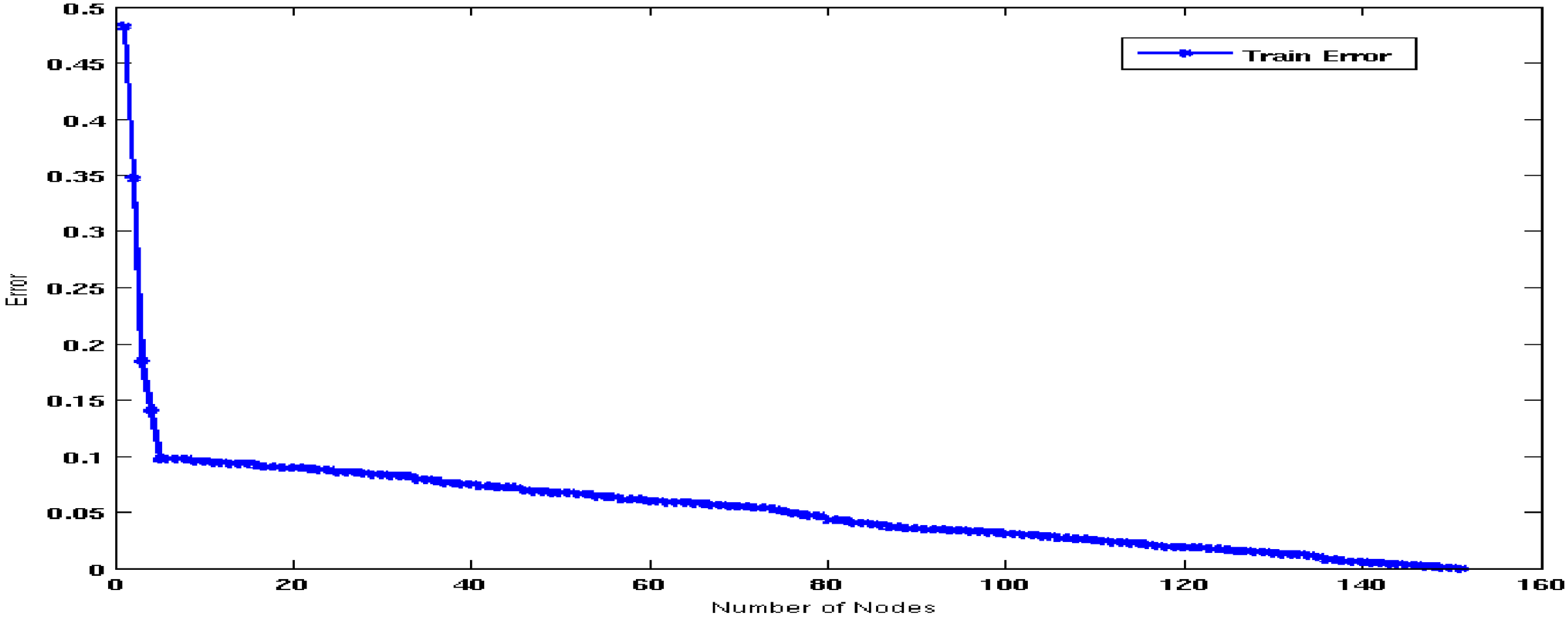
- **5000 instances generated from a Gaussian centered at (10,10)**
- **200 noisy instances added**

**o : 5200 instances**

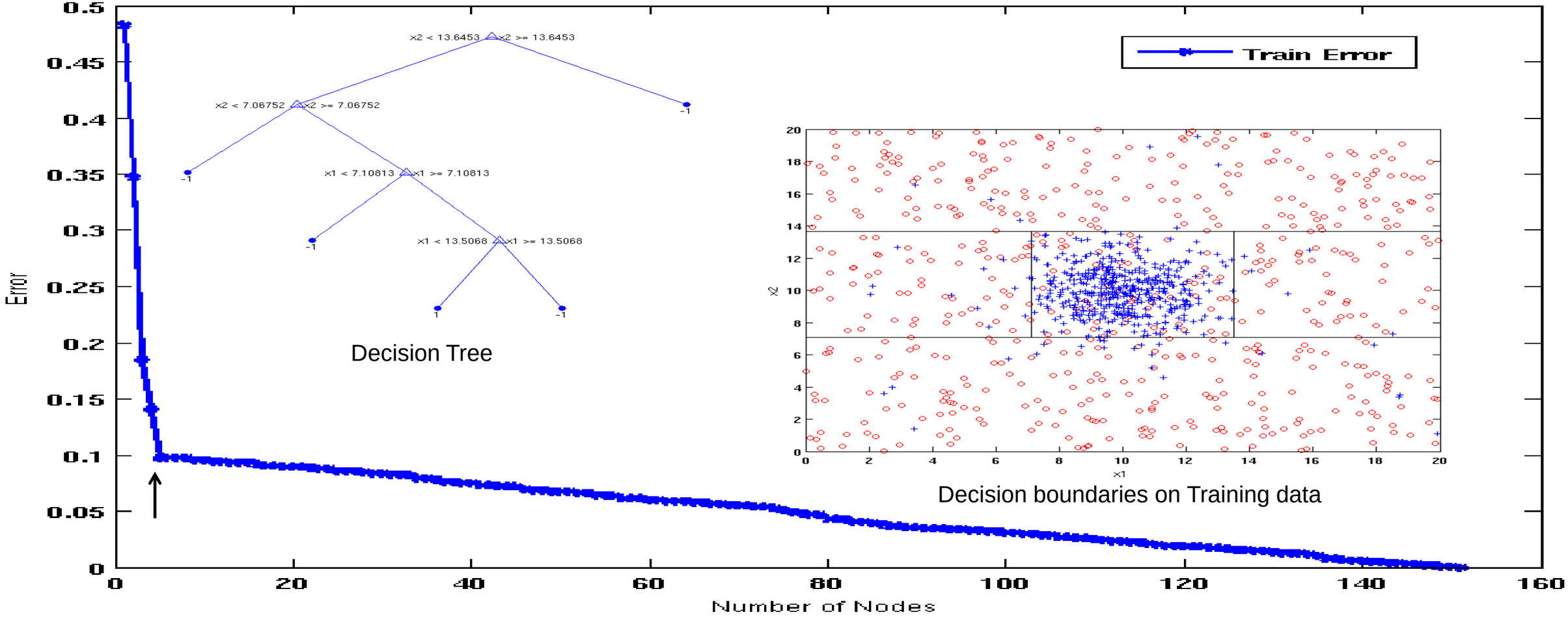
- **Generated from a uniform distribution**

**10 % of the data used for training and 90% of the data used for testing**

# Increasing number of nodes in Decision Trees



# Decision Tree with 4 nodes





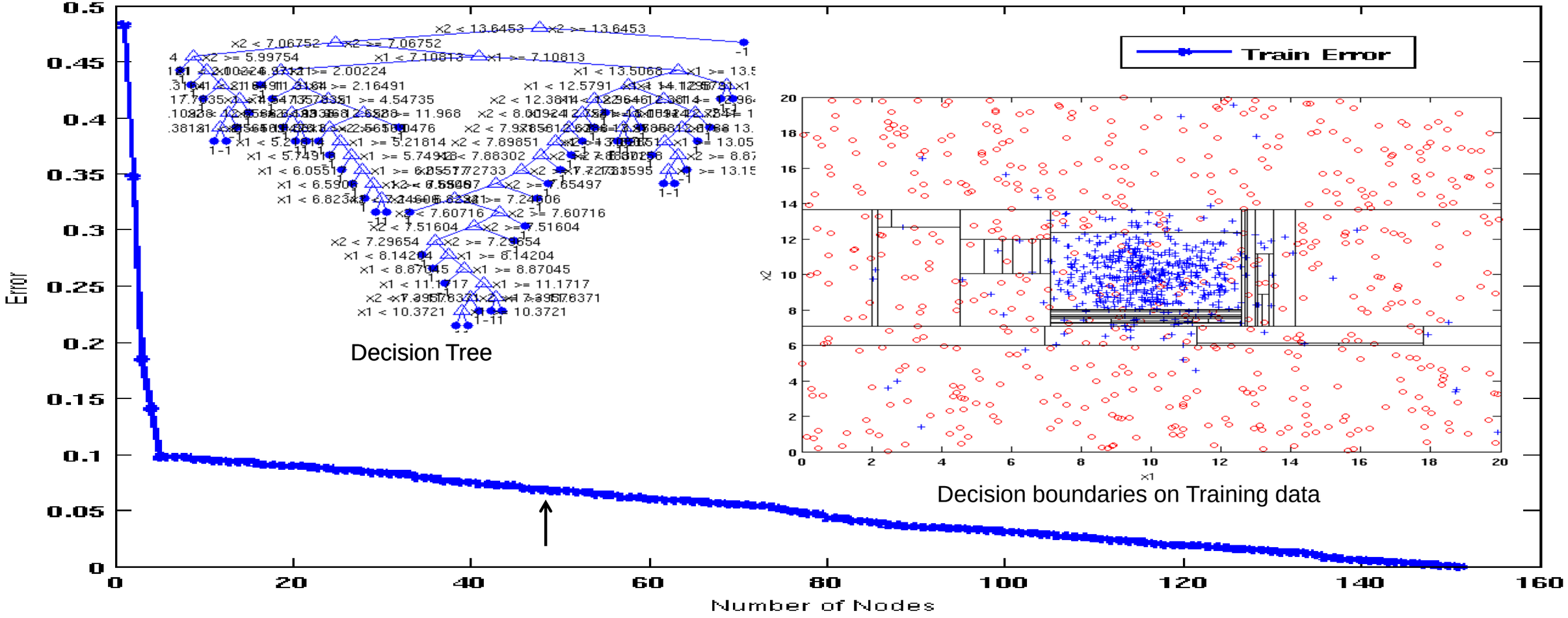


Decision Tree

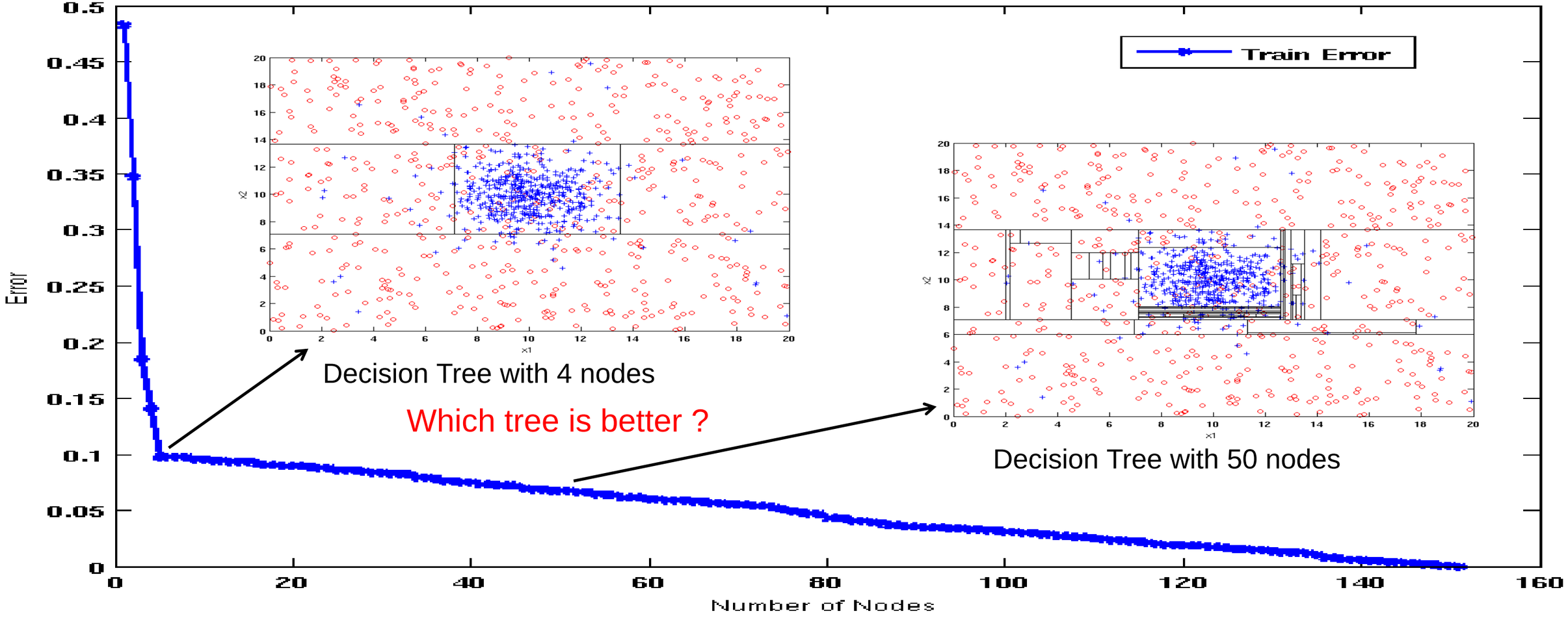



Decision boundaries on Training data

# Decision Tree with 50 nodes



# Which tree is better?

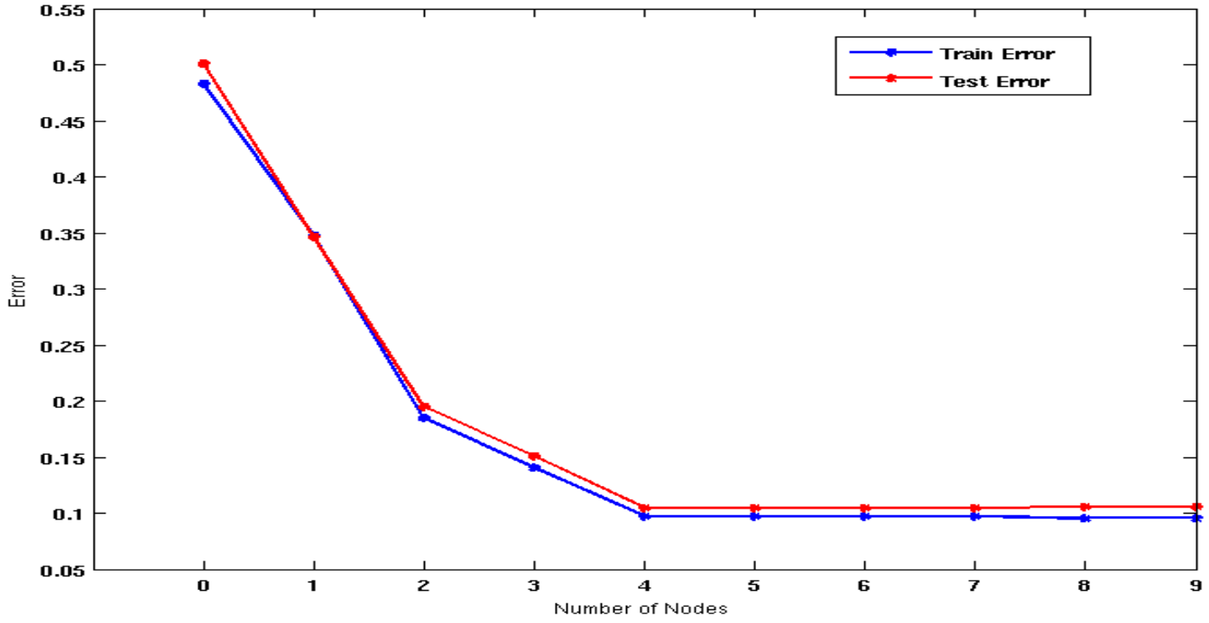


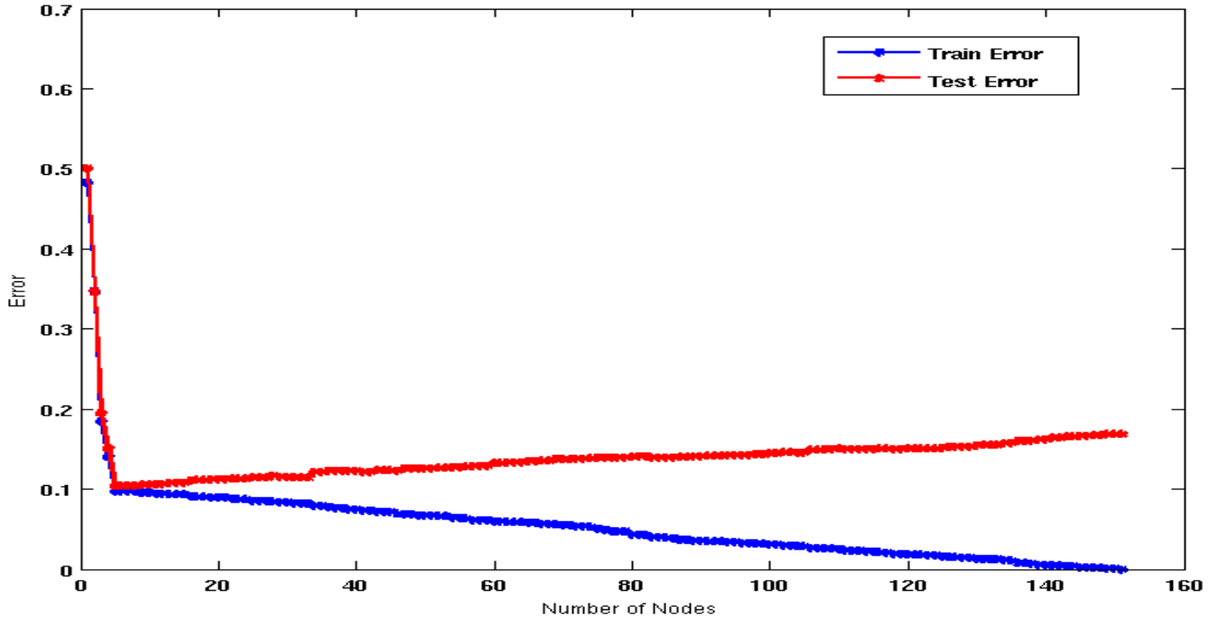

Decision Tree with 4 nodes

Which tree is better ?

Decision Tree with 50 nodes

# Model Overfitting

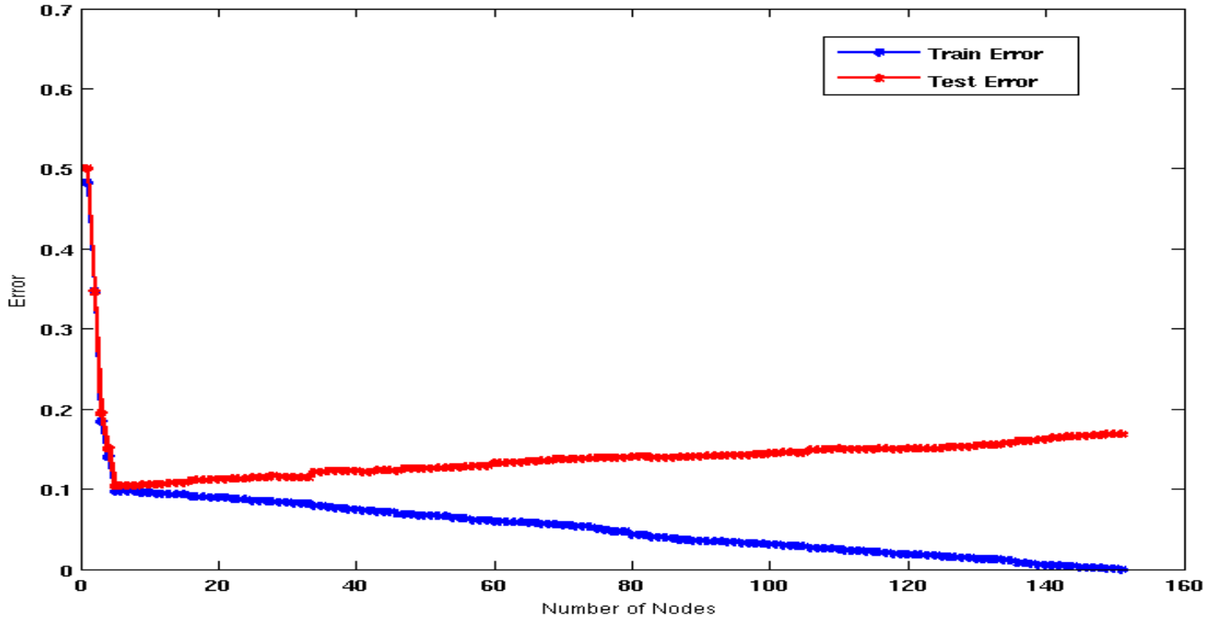


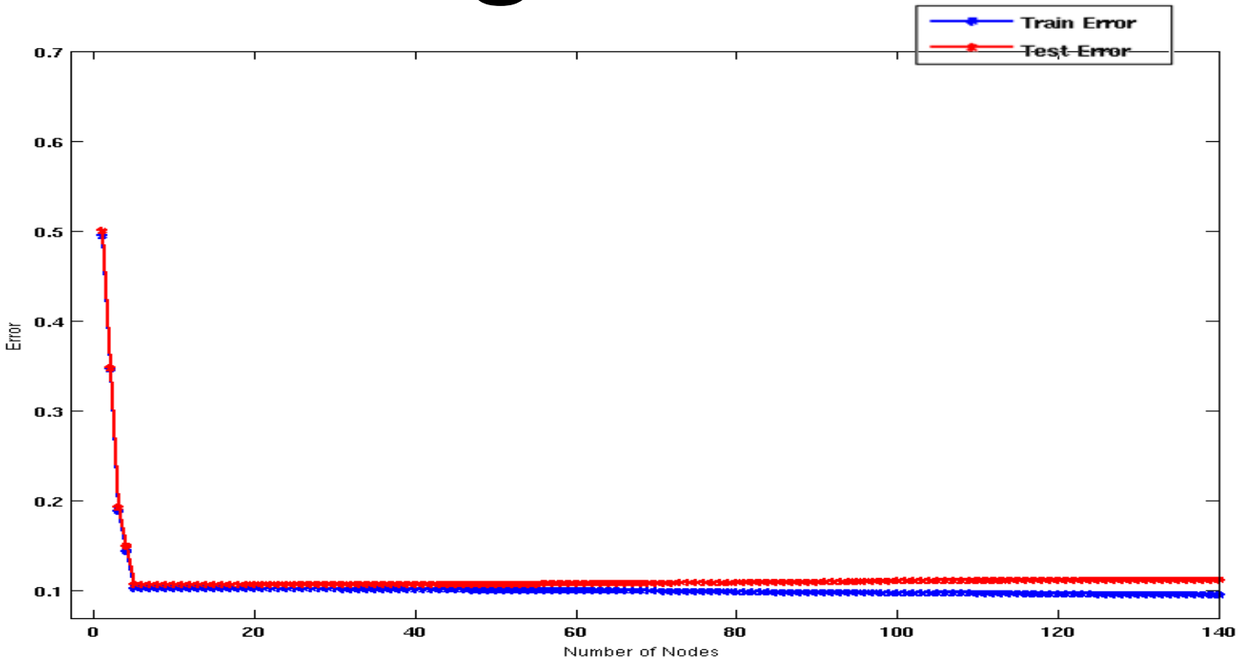




Underfitting: when model is too simple, both training and test errors are large

Overfitting: when model is too complex, training error is small but test error is large

# Model Overfitting




Using twice the number of data instances

- If training data is under-representative, testing errors increase and training errors decrease on increasing number of nodes
- Increasing the size of training data reduces the difference between training and testing errors at a given number of nodes

# Reasons for Model Overfitting

- Presence of Noise
- Lack of Representative Samples
- Multiple Comparison Procedure

# Effect of Multiple Comparison Procedure

- Consider the task of predicting whether stock market will rise/fall in the next 10 trading days

- Random guessing:

  $P(correct) = 0.5$

- Make 10 random guesses in a row:

$$P(\#correct \geq 8) = \frac{\binom{10}{8} + \binom{10}{9} + \binom{10}{10}}{2^{10}} = 0.0547$$

| | |
|---|---|
| Day 1 | Up |
| Day 2 | Down |
| Day 3 | Down |
| Day 4 | Up |
| Day 5 | Down |
| Day 6 | Down |
| Day 7 | Up |
| Day 8 | Up |
| Day 9 | Up |
| Day 10 | Down |

# Effect of Multiple Comparison Procedure

- Approach:
  - Get 50 analysts
  - Each analyst makes 10 random guesses
  - Choose the analyst that makes the most number of correct predictions

- Probability that at least one analyst makes at least 8 correct predictions

$$P(\#correct \geq 8) = 1 - (1 - 0.0547)^{50} = 0.9399$$

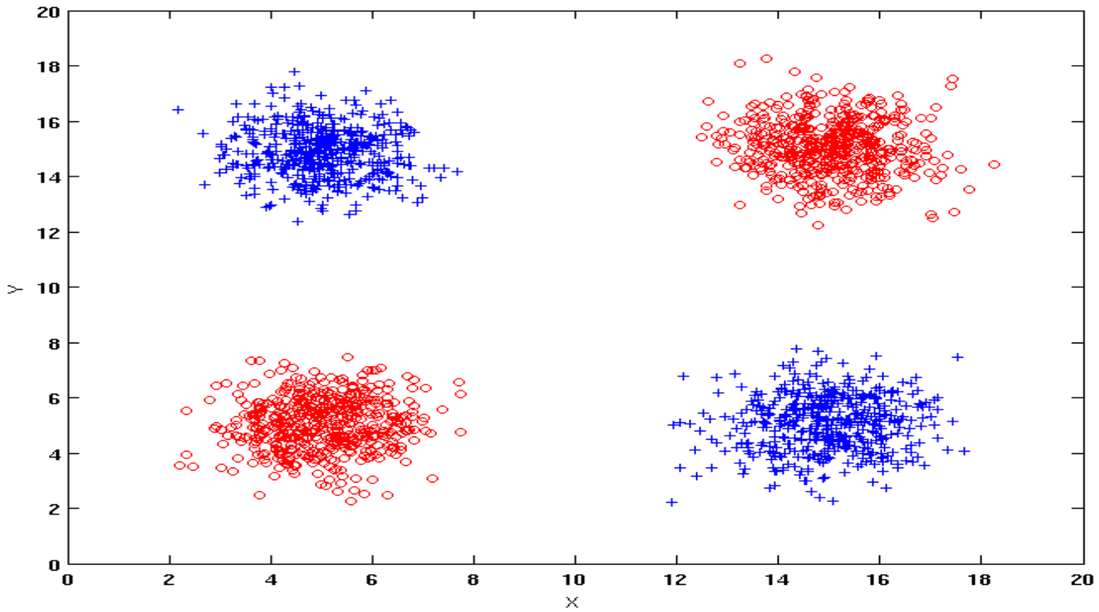
# Effect of Multiple Comparison Procedure

- Many algorithms employ the following greedy strategy:
  - Initial model: M
  - Alternative model: M' = M $\cup$ $\gamma$, where $\gamma$ is a component to be added to the model (e.g., a test condition of a decision tree)
  - Keep M' if improvement, $\Delta$(M,M') > $\alpha$

- Often times, $\gamma$ is chosen from a set of alternative components, $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_k\}$

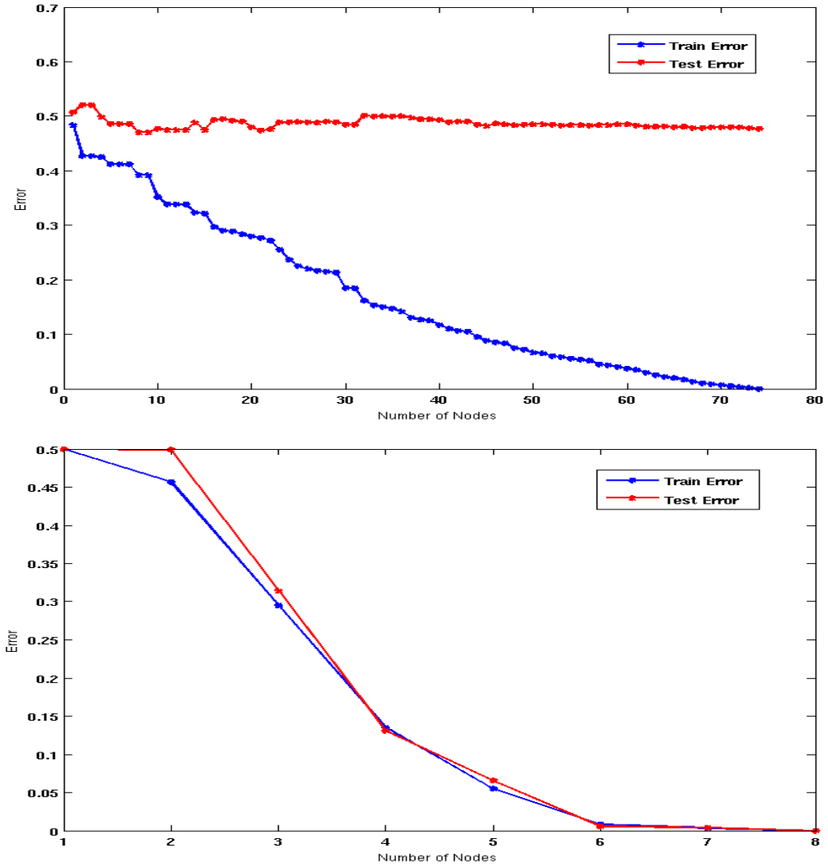
- If many alternatives are available, one may inadvertently add irrelevant components to the model, resulting in model overfitting

# Effect of Multiple Comparison - Example




Use additional 100 noisy variables generated from a uniform distribution along with X and Y as attributes.

Use 30% of the data for training and 70% of the data for testing




Using only X and Y as attributes

# Notes on Overfitting

- Overfitting results in decision trees that are more complex than necessary
- Training error does not provide a good estimate of how well the tree will perform on previously unseen records
- Need ways for incorporating model complexity into model development

# Evaluating Performance of Classifier

- Model Selection
  - Performed during model building
  - Purpose is to ensure that model is not overly complex (to avoid overfitting)
- Model Evaluation
  - Performed after model has been constructed
  - Purpose is to estimate performance of classifier on previously unseen data (e.g., test set)

# Methods for Classifier Evaluation

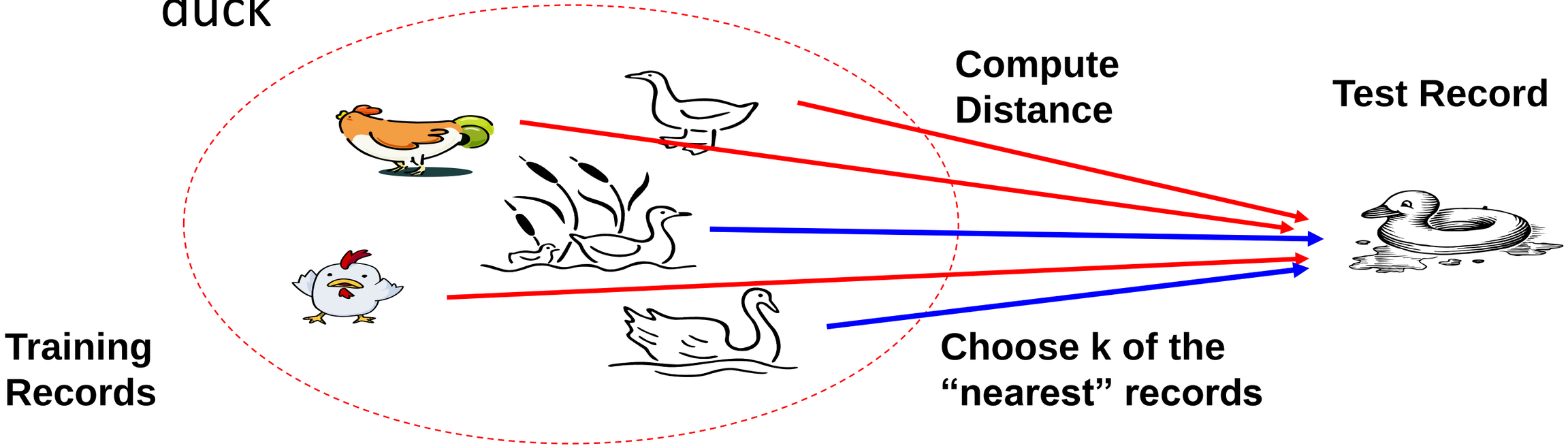
- Holdout
  - Reserve k% for training and (100-k)% for testing
- Random subsampling
  - Repeated holdout
- Cross validation
  - Partition data into k disjoint subsets
  - k-fold: train on k-1 partitions, test on the remaining one
  - Leave-one-out: k=n
- Bootstrap
  - Sampling with replacement
  - .632 bootstrap:

$$acc_{boot} = \frac{1}{b}\sum_{i=1}^{b}(0.632 \times acc_i + 0.368 \times acc_s)$$

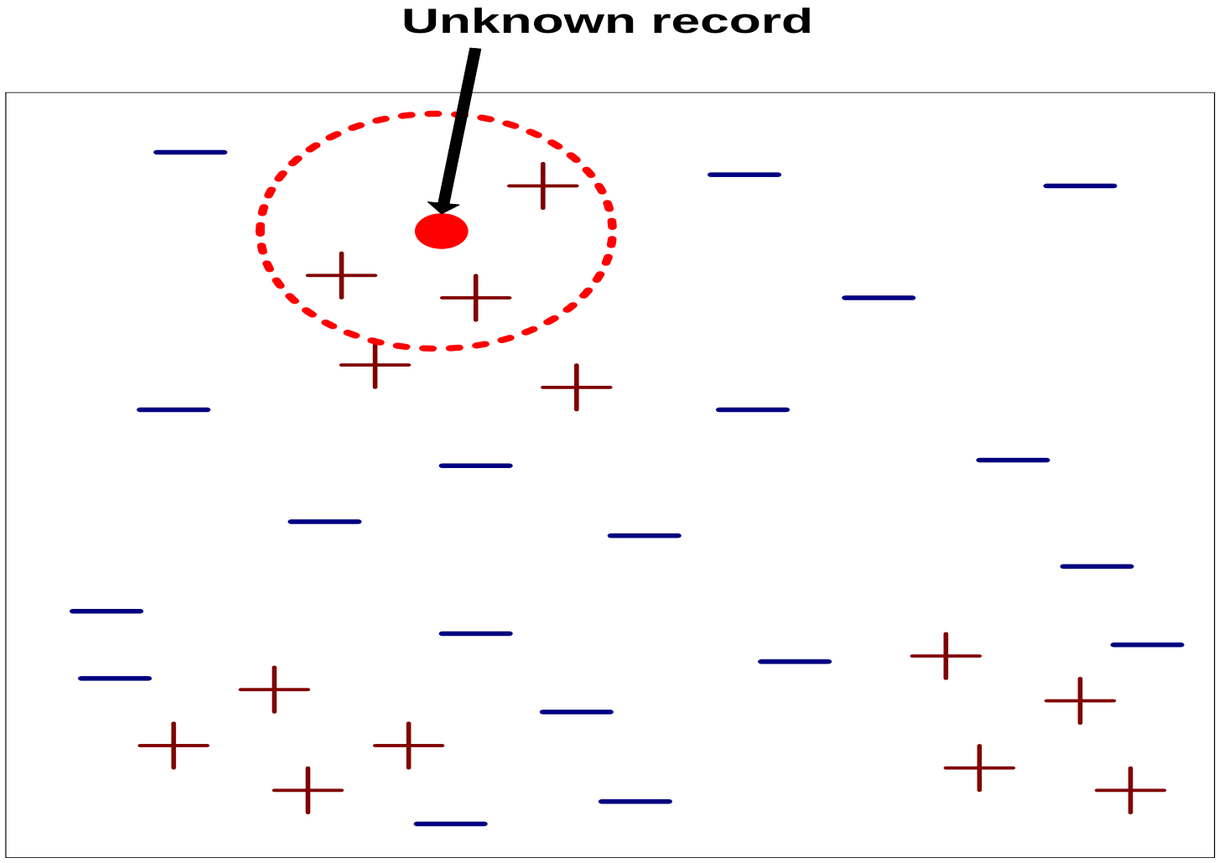
# Nearest Neighbor Classifier

# Nearest Neighbor Classifiers

- Basic idea:
  - If it walks like a duck, quacks like a duck, then it's probably a duck



# Nearest-Neighbor Classifiers

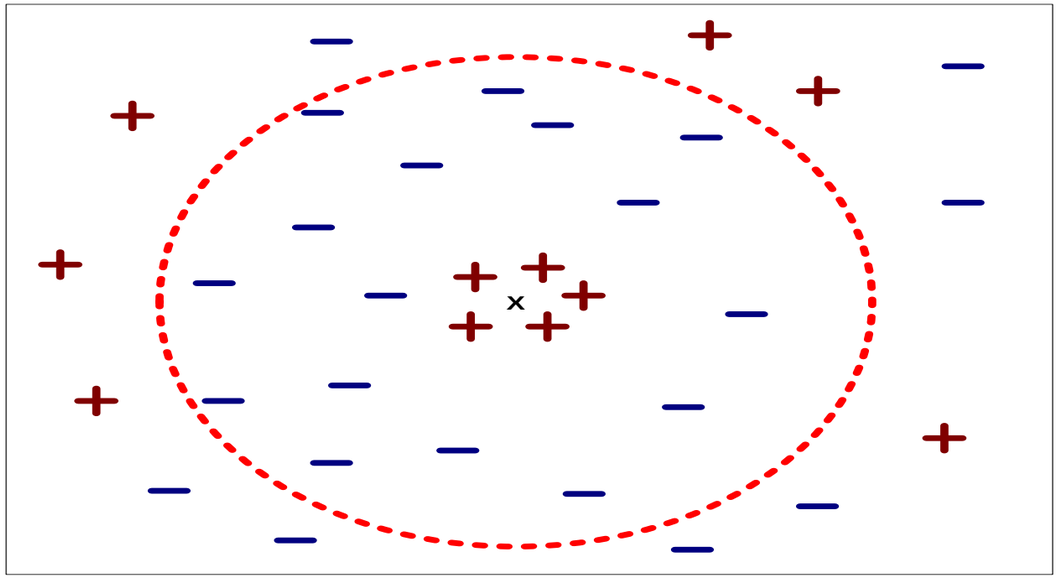



- Requires three things
  - The set of stored records
  - Distance metric to compute distance between records
  - The value of *k*, the number of nearest neighbors to retrieve

- To classify an unknown record:
  - Compute distance to other training records
  - Identify *k* nearest neighbors
  - Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

# Nearest Neighbor Classification…

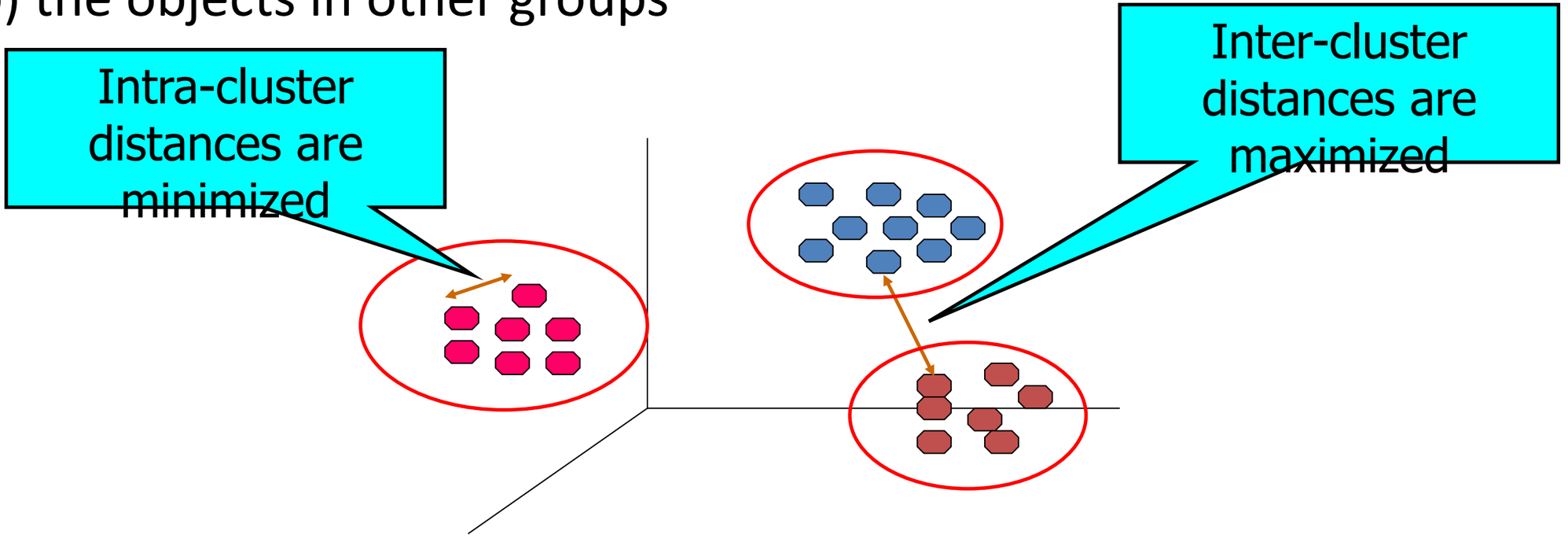
- Choosing the value of k:
  - If k is too small, sensitive to noise points
  - If k is too large, neighborhood may include points from other classes



# Clustering

# Clustering

- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups



# Applications of Clustering

- Applications:
  - Gene expression clustering
  - Clustering of patients based on phenotypic and genotypic factors for efficient disease diagnosis
  - Market Segmentation
  - Document Clustering
  - Finding groups of driver behaviors based upon patterns of automobile motions (normal, drunken, sleepy, rush hour driving, etc)

# Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters

Two Clusters

Four Clusters

# Similarity and Dissimilarity Measures

- Similarity measure
  - Numerical measure of how alike two data objects are.
  - Is higher when objects are more alike.
  - Often falls in the range [0,1]
- Dissimilarity measure
  - Numerical measure of how different are two data objects
  - Lower when objects are more alike
  - Minimum dissimilarity is often 0
  - Upper limit varies
- Proximity refers to a similarity or dissimilarity

# Euclidean Distance

- Euclidean Distance

$$dist(x, y) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}$$

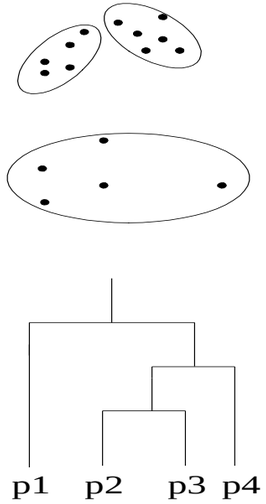
Where $n$ is the number of dimensions (attributes) and $x_k$ and $y_k$ are, respectively, the k$^{th}$ attributes (components) or data objects $x$ and $y$.

- Correlation

$$corr(x, y) = \frac{\sum_{k=1}^{n} (x_k - \bar{x})(y_k - \bar{y})^2}{\sqrt{\sum_{k=1}^{n} (x_k - \bar{x})^2 \sum_{k=1}^{n} (y_k - \bar{y})^2}} = \frac{\mathrm{cov}(x, y)}{std(x)std(y)}$$

# Types of Clusterings

- A clustering is a set of clusters
- Important distinction between hierarchical and partitional sets of clusters
- Partitional Clustering
  - A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset
- Hierarchical clustering
  - A set of nested clusters organized as a hierarchical tree



# Other Distinctions Between Sets of Clusters

- Exclusive versus non-exclusive
  - In non-exclusive clusterings, points may belong to multiple clusters.
  - Can represent multiple classes or ‘border’ points
- Fuzzy versus non-fuzzy
  - In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1
  - Weights must sum to 1
  - Probabilistic clustering has similar characteristics
- Partial versus complete
  - In some cases, we only want to cluster some of the data
- Heterogeneous versus homogeneous
  - Clusters of widely different sizes, shapes, and densities

# Clustering Algorithms

- K-means and its variants
- Hierarchical clustering
- Other types of clustering

# K-means Clustering

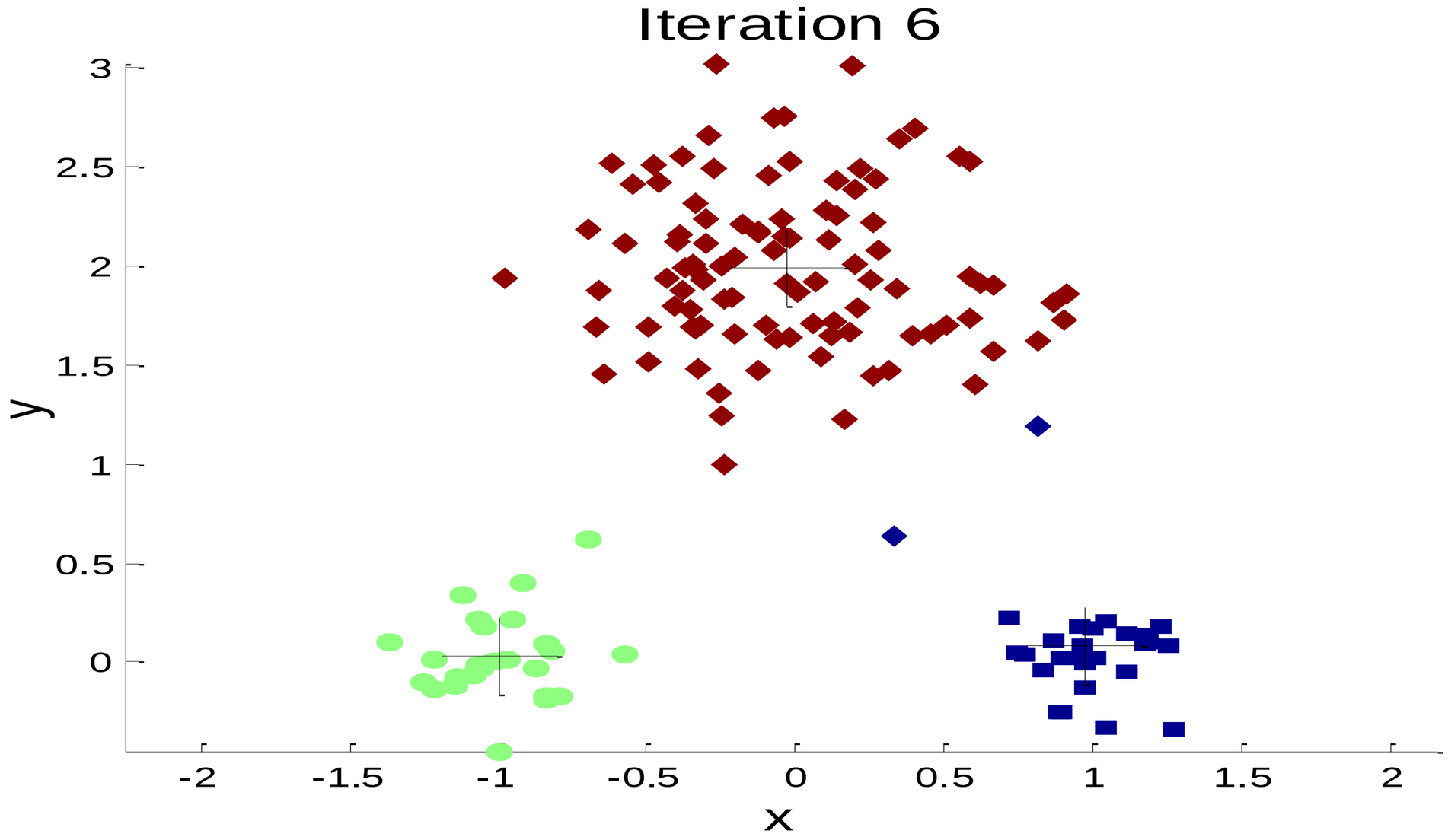
- Partitional clustering approach
- Number of clusters, K, must be specified
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- The basic algorithm is very simple

---

1: Select $K$ points as the initial centroids.
2: **repeat**
3: Form $K$ clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

---

# Example of K-means Clustering



# K-means Clustering – Details

- The centroid is (typically) the mean of the points in the cluster
- Initial centroids are often chosen randomly
  - Clusters produced vary from one run to another
- ‘Closeness’ is measured by Euclidean distance, cosine similarity, correlation, etc
- Complexity is O( n * K * I * d )
  - n = number of points, K = number of clusters,
    I = number of iterations, d = number of attributes

# Evaluating K-means Clusters

- Most common measure is Sum of Squared Error (SSE)
  - For each point, the error is the distance to the nearest cluster
  - To get SSE, we square these errors and sum them

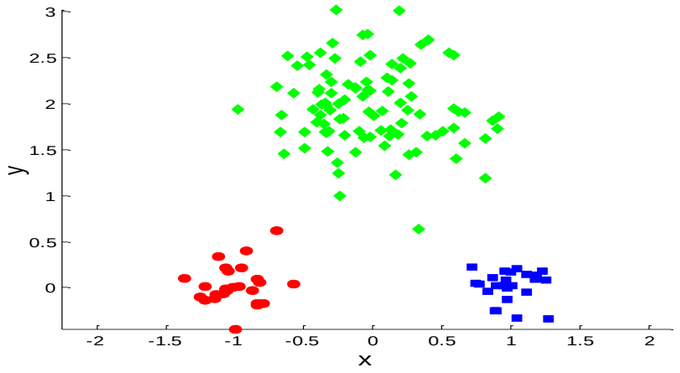
$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

    - $x$ is a data point in cluster $C_i$ and $m_i$ is the representative point for cluster $C_i$

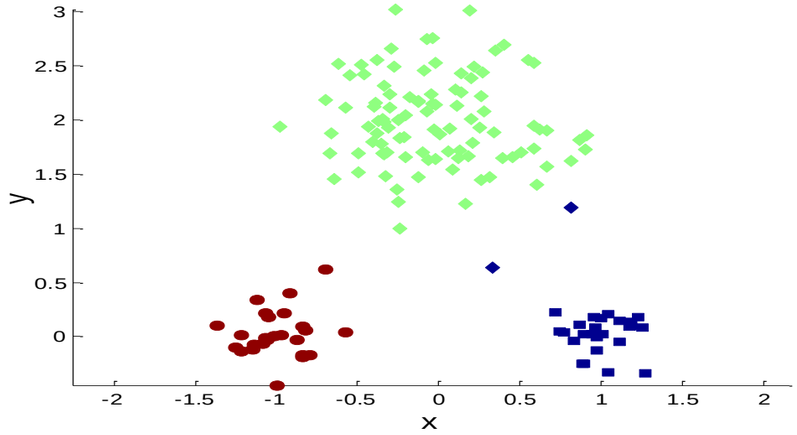
  - Given two sets of clusters, we prefer the one with the smallest error

  - One easy way to reduce SSE is to increase K, the number of clusters

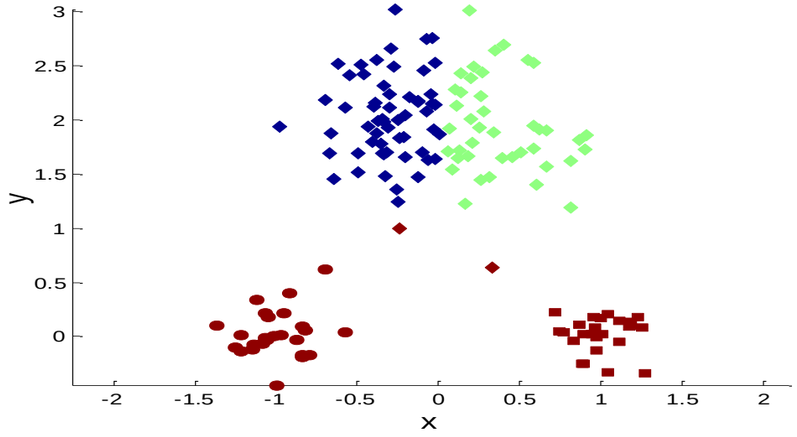
# Two different K-means Clusterings




**Original Points**




**Optimal Clustering**



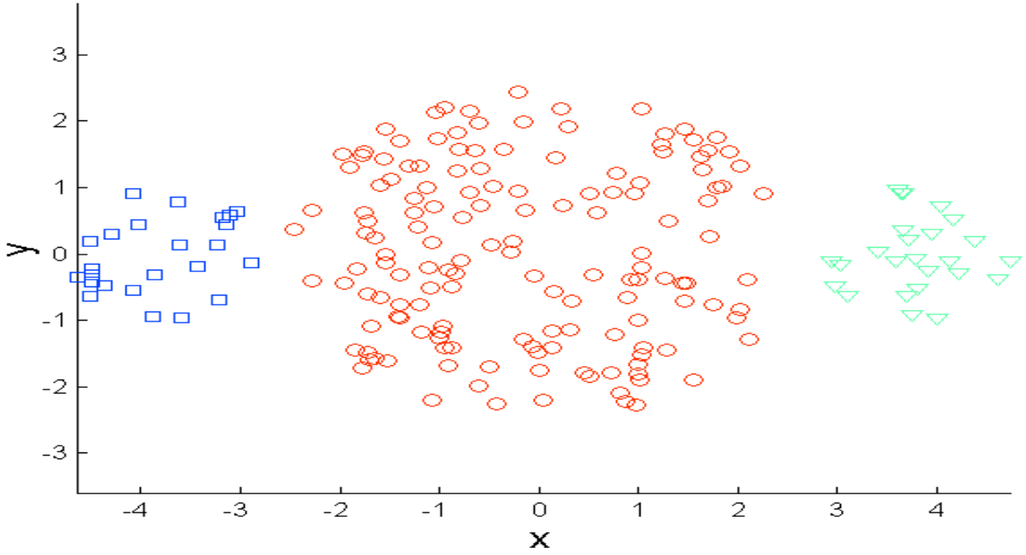

**Sub-optimal Clustering**

# Limitations of K-means

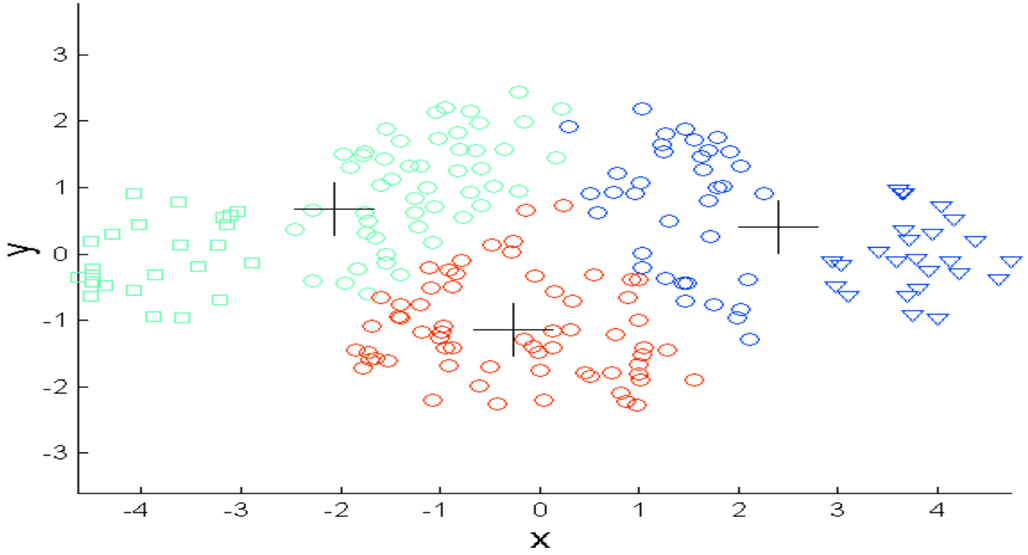
- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes

- K-means has problems when the data contains outliers.

# Limitations of K-means: Differing Sizes

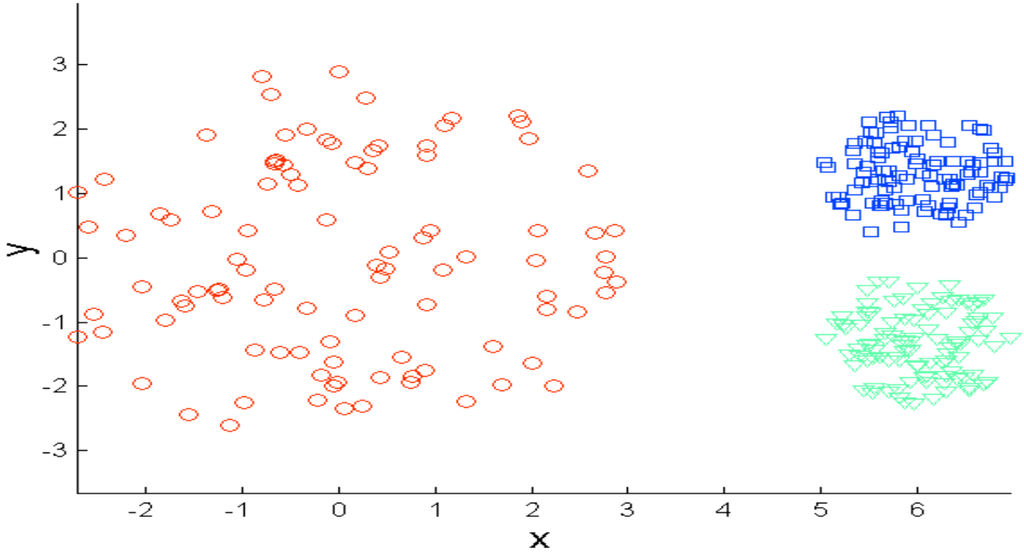



**Original Points**

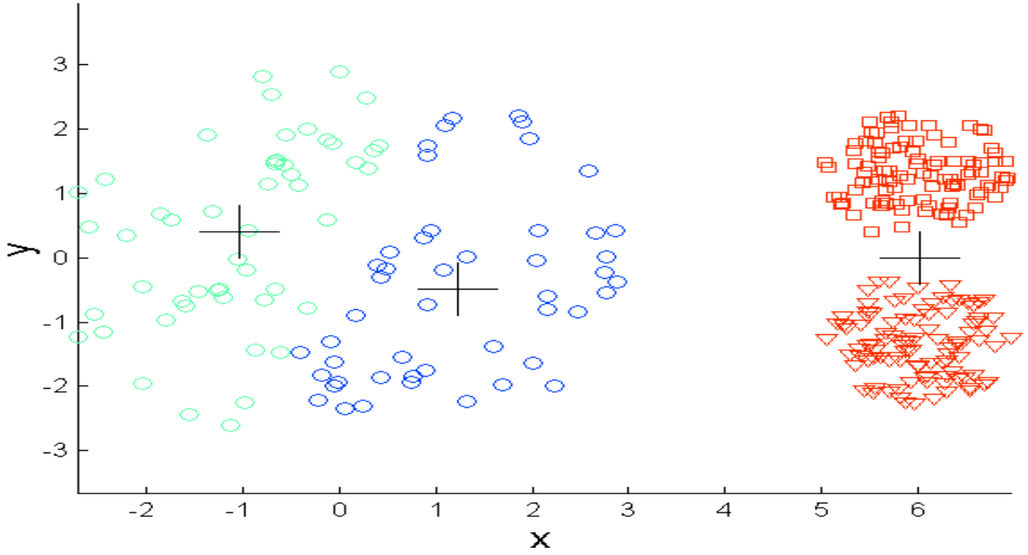



**K-means (3 Clusters)**

# Limitations of K-means: Differing Density

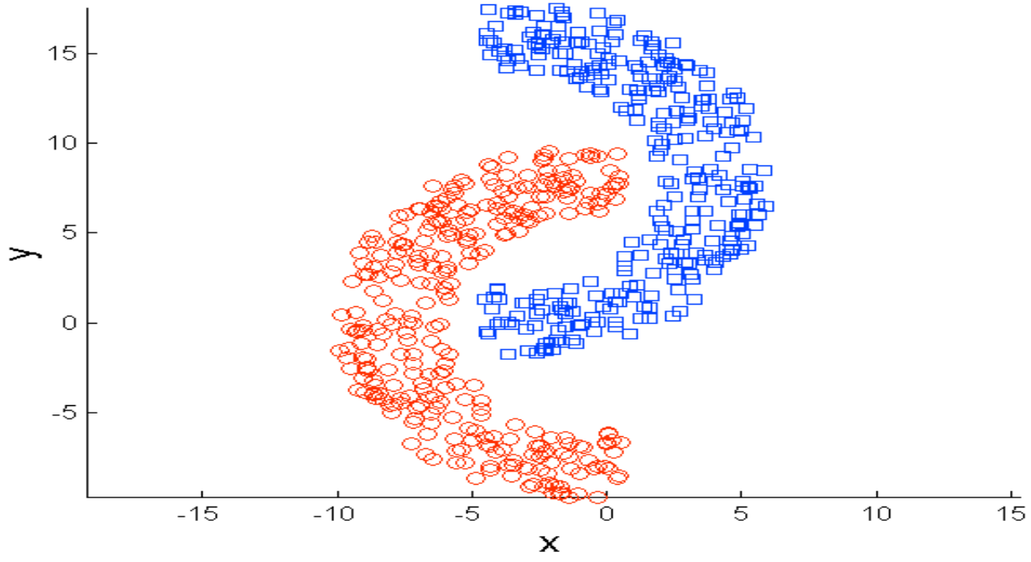



**Original Points**

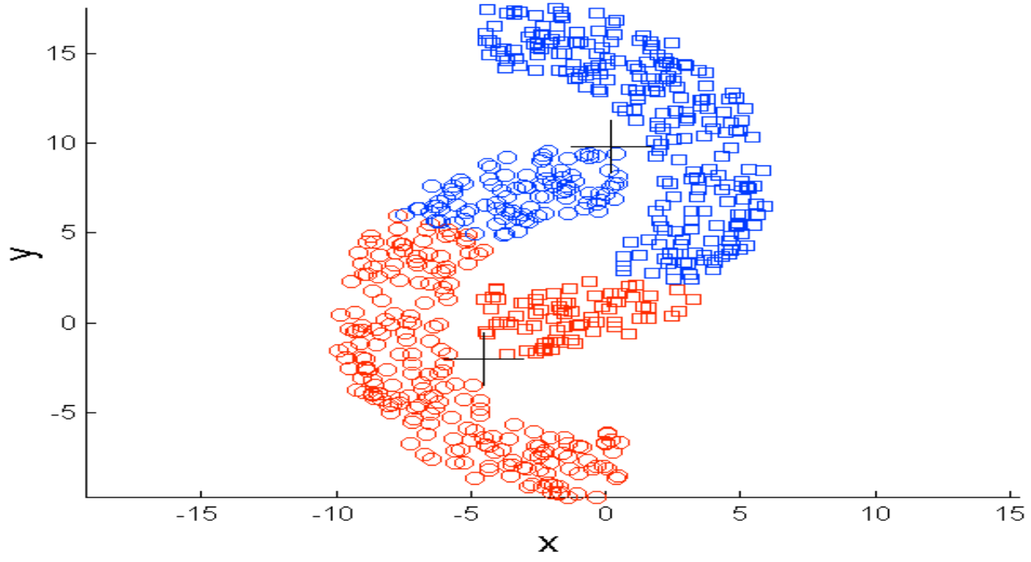



**K-means (3 Clusters)**

# Limitations of K-means: Non-globular Shapes



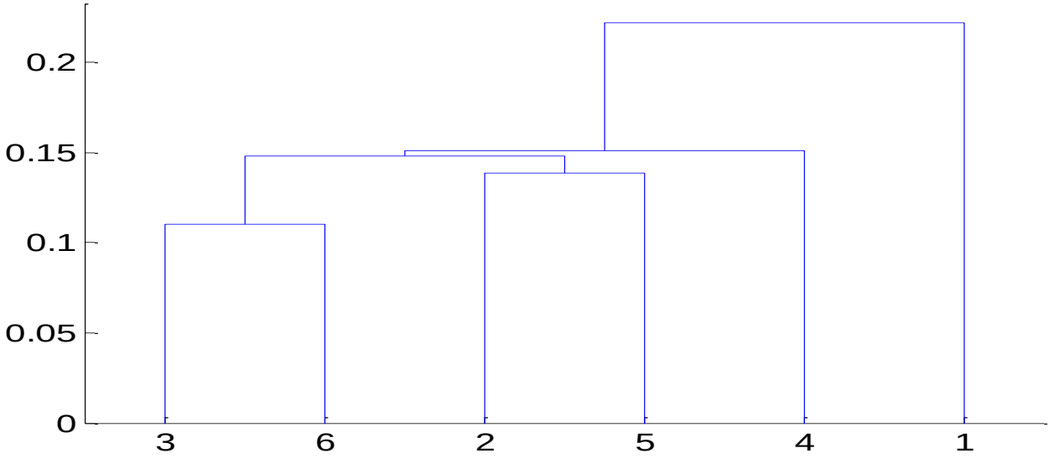

**Original Points**

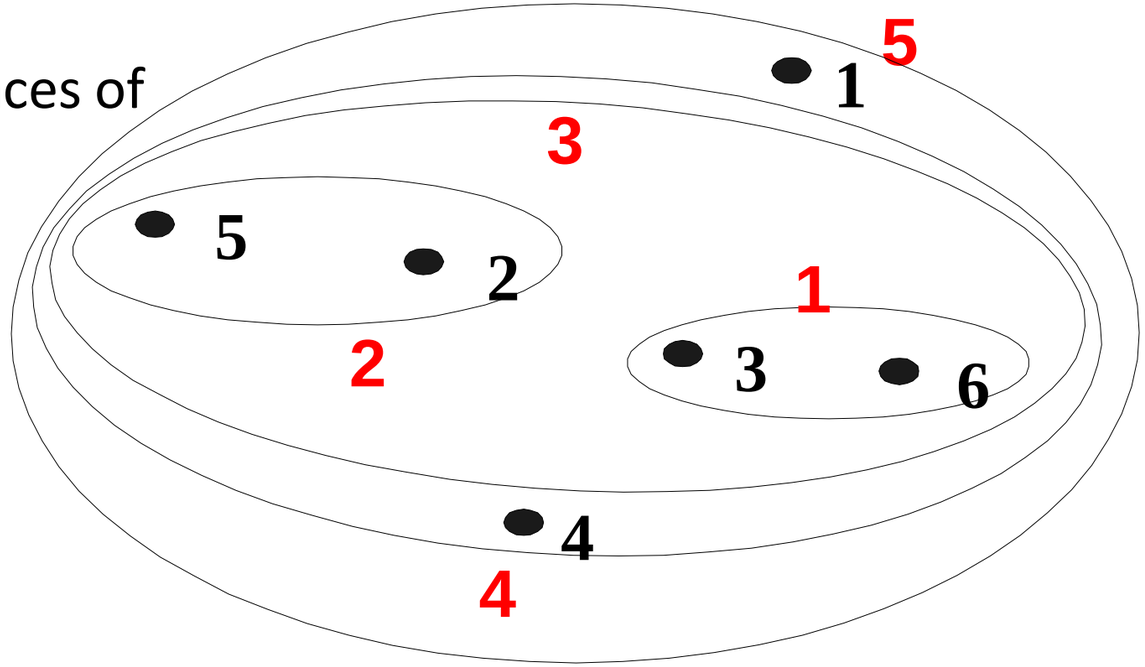



**K-means (2 Clusters)**

# Hierarchical Clustering

- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequences of merges or splits



# Strengths of Hierarchical Clustering

- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by ‘cutting’ the dendrogram at the proper level

- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)

# Hierarchical Clustering

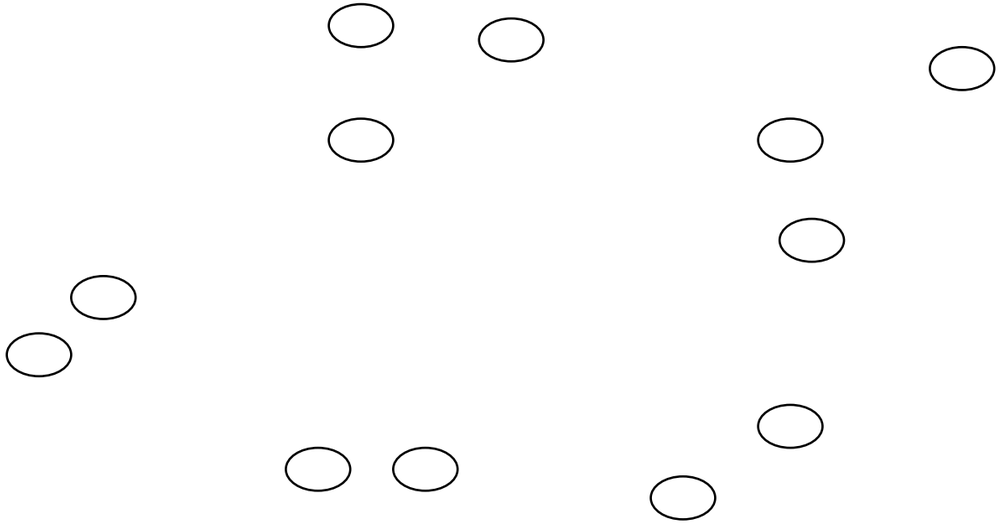
- Two main types of hierarchical clustering
  - Agglomerative:
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are k clusters)
- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge or split one cluster at a time

# Agglomerative Clustering Algorithm

- More popular hierarchical clustering technique
- Basic algorithm is straightforward
  1. Compute the proximity matrix
  2. Let each data point be a cluster
  3. **Repeat**
  4. Merge the two closest clusters
  5. Update the proximity matrix
  6. **Until** only a single cluster remains
- Key operation is the computation of the proximity of two clusters
  - Different approaches to defining the distance between clusters distinguish the different algorithms

# Starting Situation

- Start with clusters of individual points and a proximity matrix

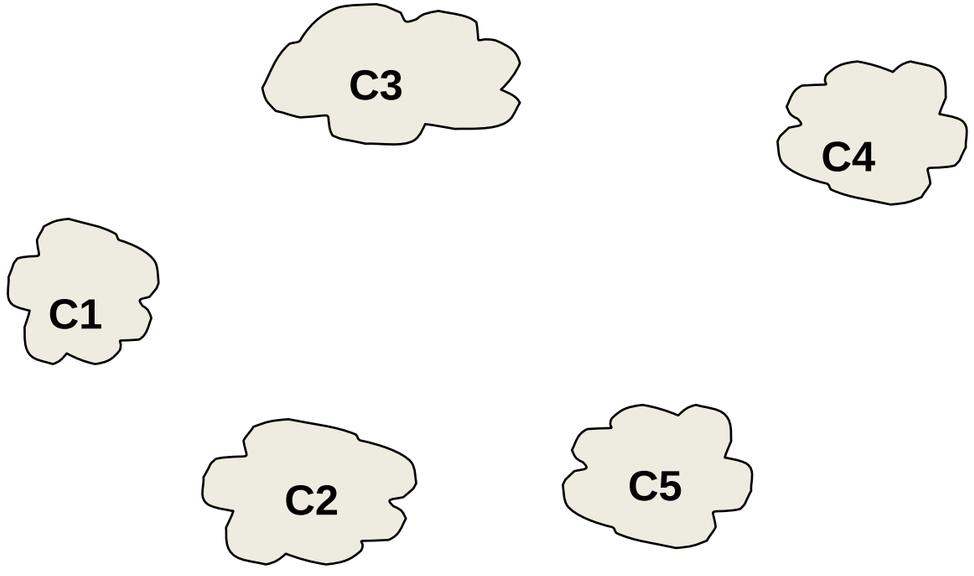


| | p1 | p2 | p3 | p4 | p5 | . . . |
|---|---|---|---|---|---|---|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |
| . | | | | | | |
| . | | | | | | |

**Proximity Matrix**

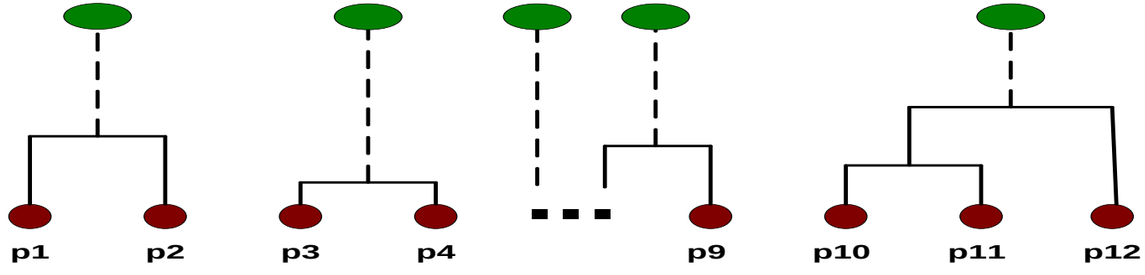
# Intermediate Situation

- After some merging steps, we have some clusters



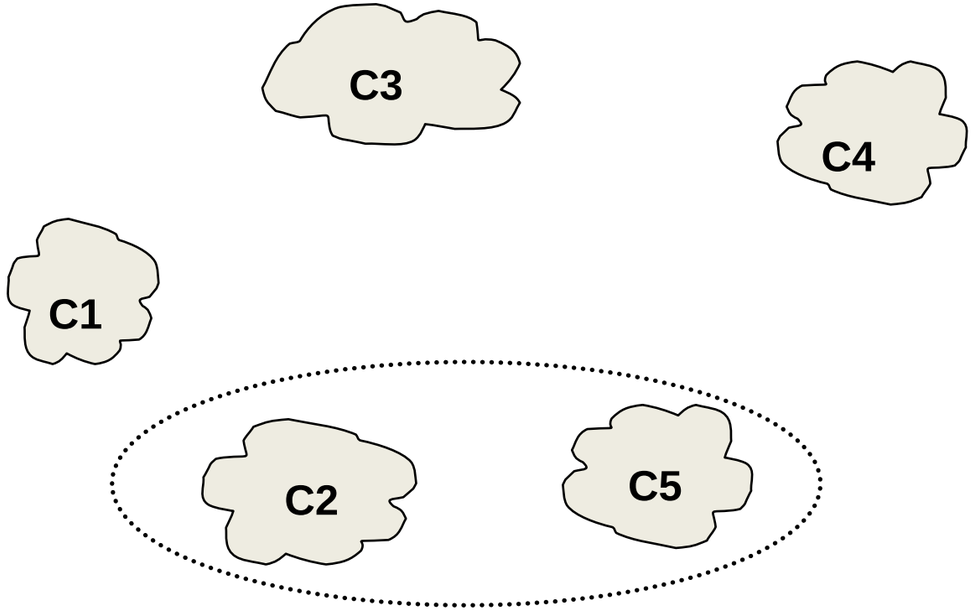

| | C1 | C2 | C3 | C4 | C5 |
|---|---|---|---|---|---|
| C1 | | | | | |
| C2 | | | | | |
| C3 | | | | | |
| C4 | | | | | |
| C5 | | | | | |

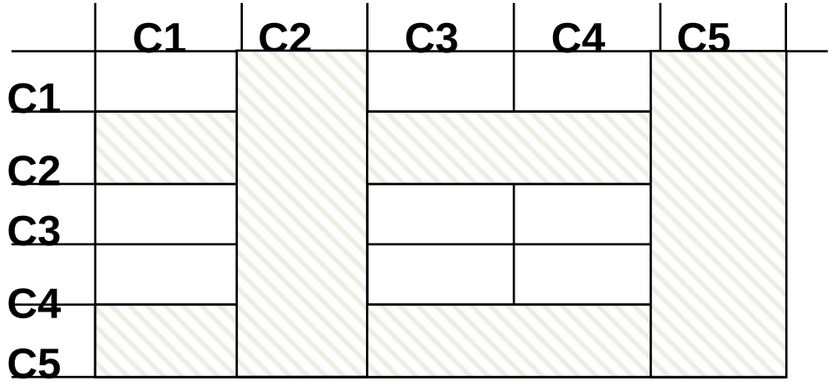
**Proximity Matrix**



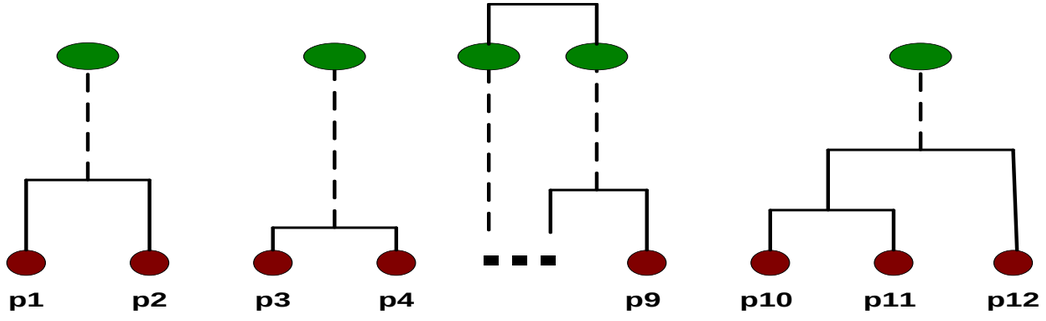
# Intermediate Situation

- We want to merge the two closest clusters (C2 and C5) and update the proximity matrix.



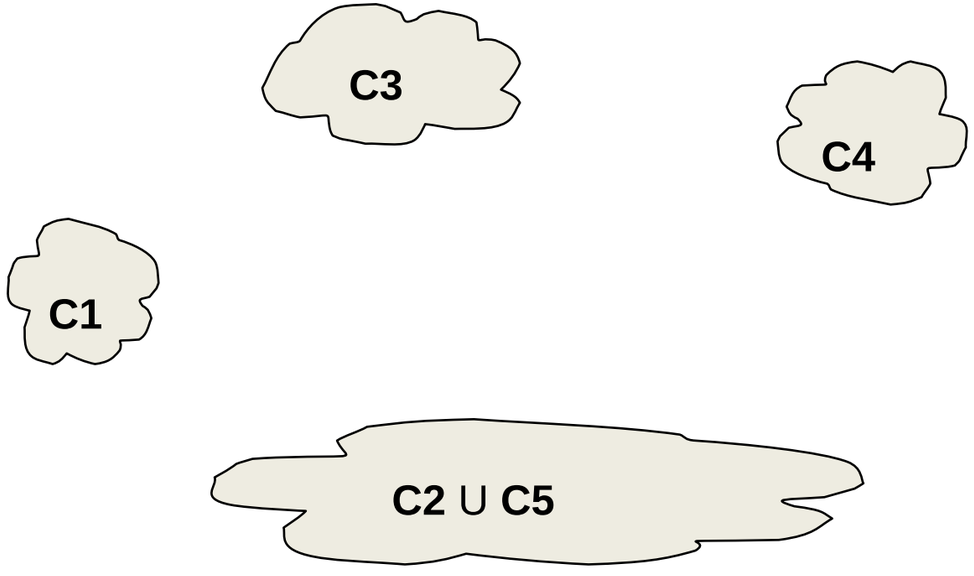




**Proximity Matrix**



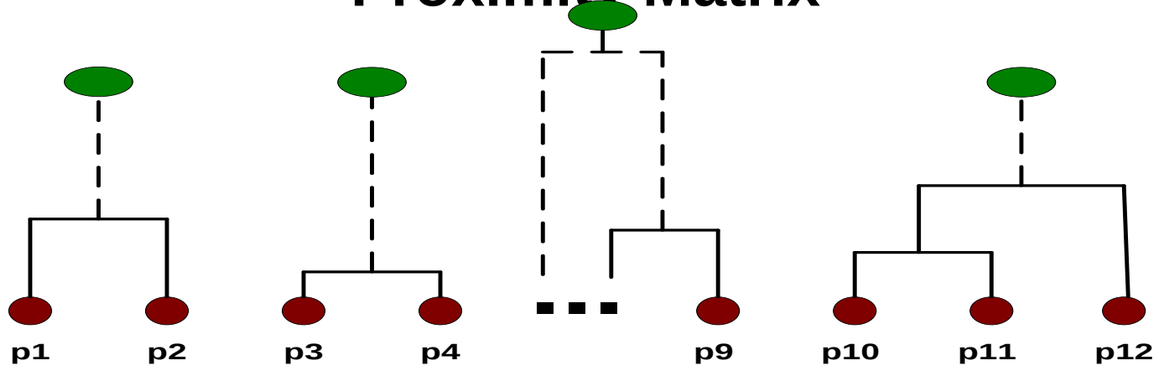
# After Merging

- The question is “How do we update the proximity matrix?”

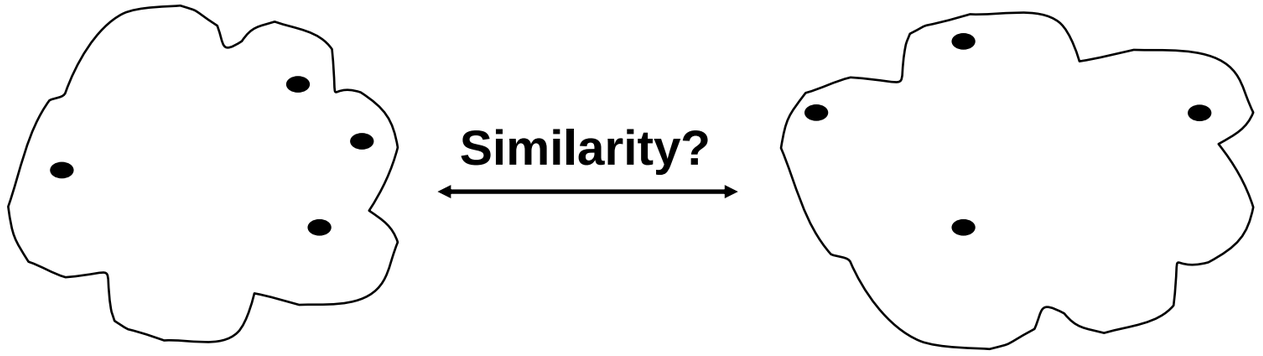



| | C1 | C2 U C5 | C3 | C4 |
|---|---|---|---|---|
| C1 | | ? | | |
| C2 U C5 | ? | ? | ? | ? |
| C3 | | ? | | |
| C4 | | ? | | |

Proximity Matrix



# How to Define Inter-Cluster Distance

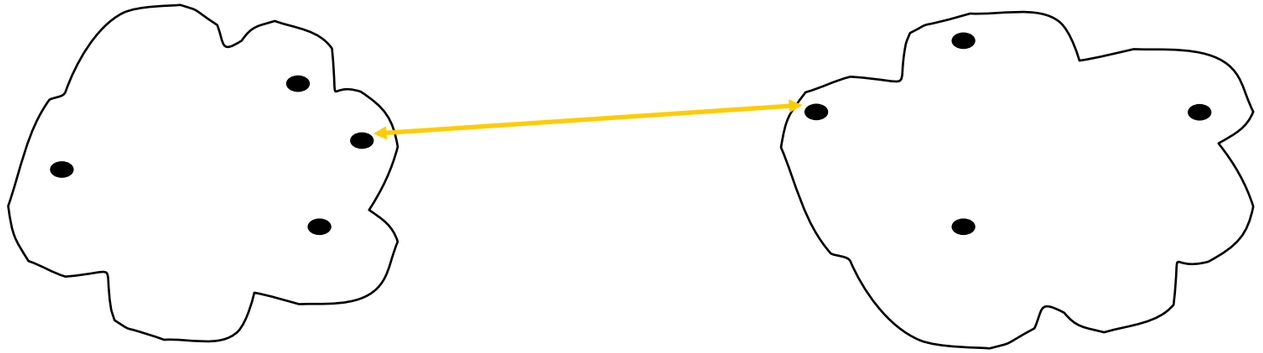



| | p1 | p2 | p3 | p4 | p5 | . . . |
|---|---|---|---|---|---|---|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |
| . | | | | | | |
| . | | | | | | |

**Proximity Matrix**

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

# How to Define Inter-Cluster Similarity

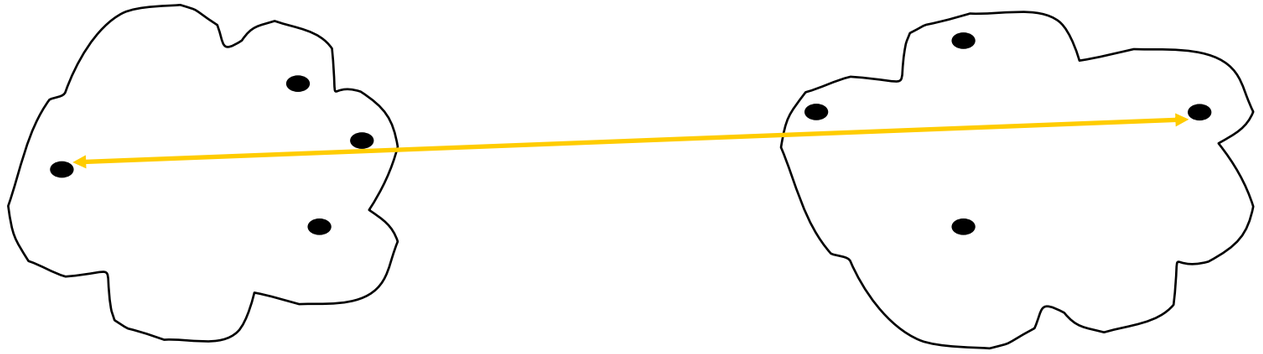


| | p1 | p2 | p3 | p4 | p5 | . . . |
|---|---|---|---|---|---|---|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |
| . | | | | | | |
| . | | | | | | |

**Proximity Matrix**

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

# How to Define Inter-Cluster Similarity

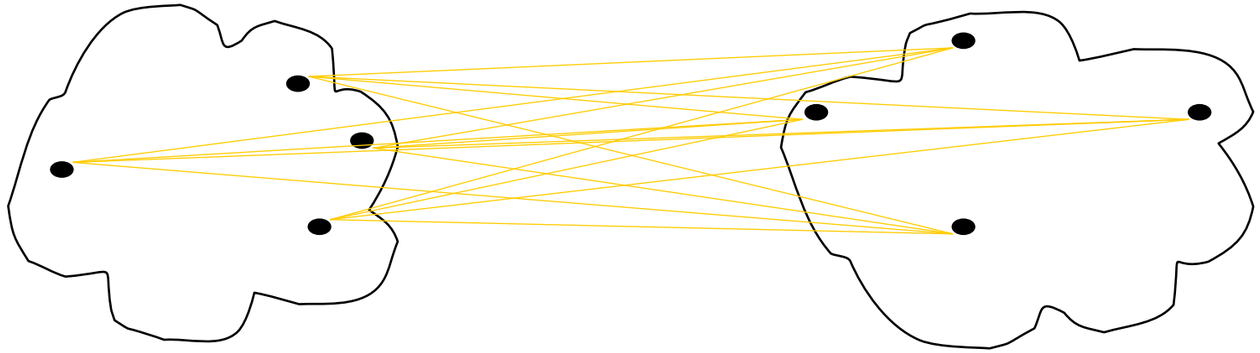


| | p1 | p2 | p3 | p4 | p5 | . . . |
|---|---|---|---|---|---|---|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |
| . | | | | | | |
| . | | | | | | |

**Proximity Matrix**

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

# How to Define Inter-Cluster Similarity

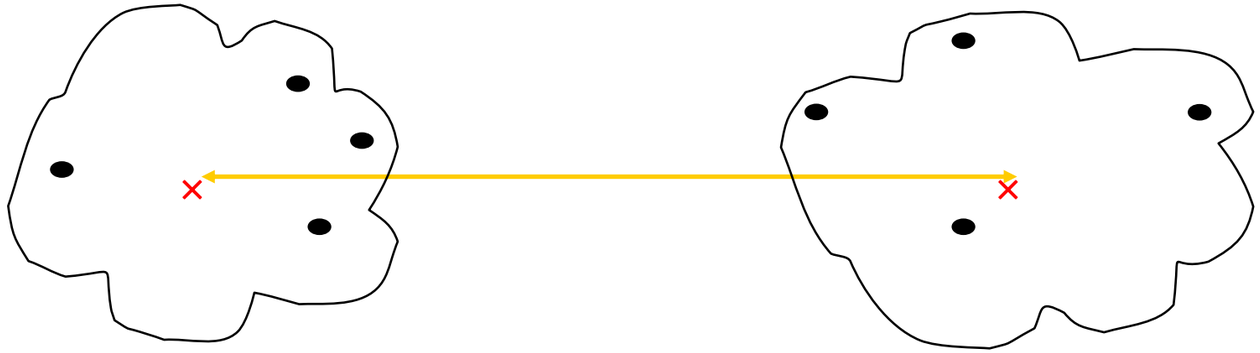


| | p1 | p2 | p3 | p4 | p5 | . . . |
|---|---|---|---|---|---|---|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |
| . | | | | | | |
| . | | | | | | |

**Proximity Matrix**

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

# How to Define Inter-Cluster Similarity



| | p1 | p2 | p3 | p4 | p5 | . . . |
|---|---|---|---|---|---|---|
| p1 | | | | | | |
| p2 | | | | | | |
| p3 | | | | | | |
| p4 | | | | | | |
| p5 | | | | | | |
| . | | | | | | |
| . | | | | | | |
| . | | | | | | |

**Proximity Matrix**

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

# Other Types of Cluster Algorithms

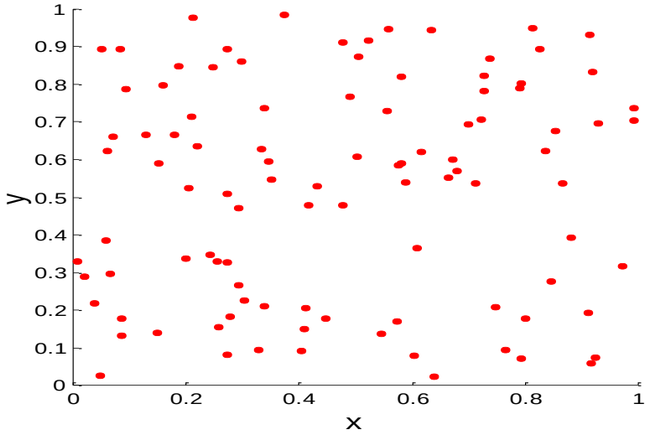
- Hundreds of clustering algorithms
- Some clustering algorithms
  - K-means
  - Hierarchical
  - Statistically based clustering algorithms
    - Mixture model based clustering
  - Fuzzy clustering
  - Self-organizing Maps (SOM)
  - Density-based (DBSCAN)
- Proper choice of algorithms depends on the type of clusters to be found, the type of data, and the objective

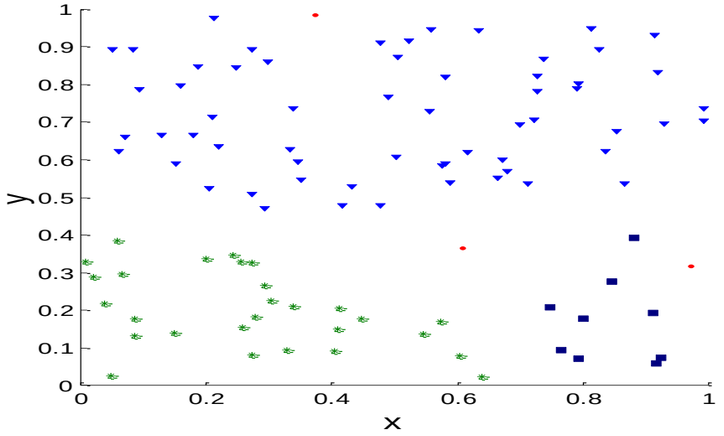
# Cluster Validity

- For supervised classification we have a variety of measures to evaluate how good our model is
  - Accuracy, precision, recall
- For cluster analysis, the analogous question is how to evaluate the "goodness" of the resulting clusters?
- But "clusters are in the eye of the beholder"!
- Then why do we want to evaluate them?
  - To avoid finding patterns in noise
  - To compare clustering algorithms
  - To compare two sets of clusters
  - To compare two clusters

# Clusters found in Random Data

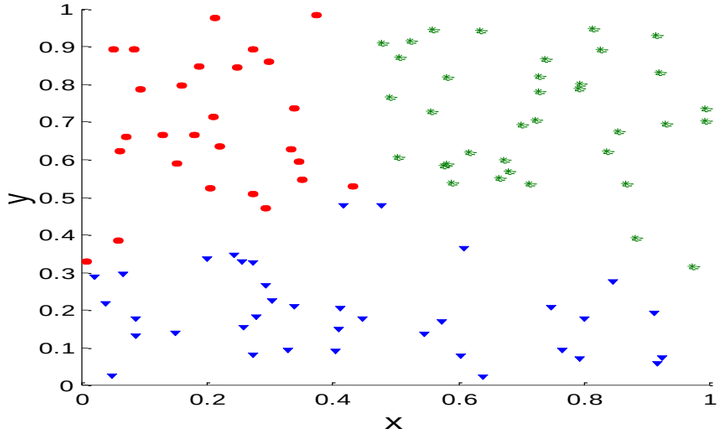
**Random Points**

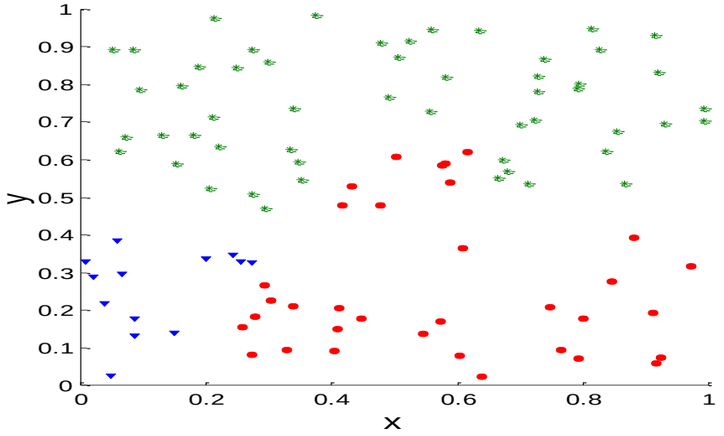



**DBSCAN**

**K-means**



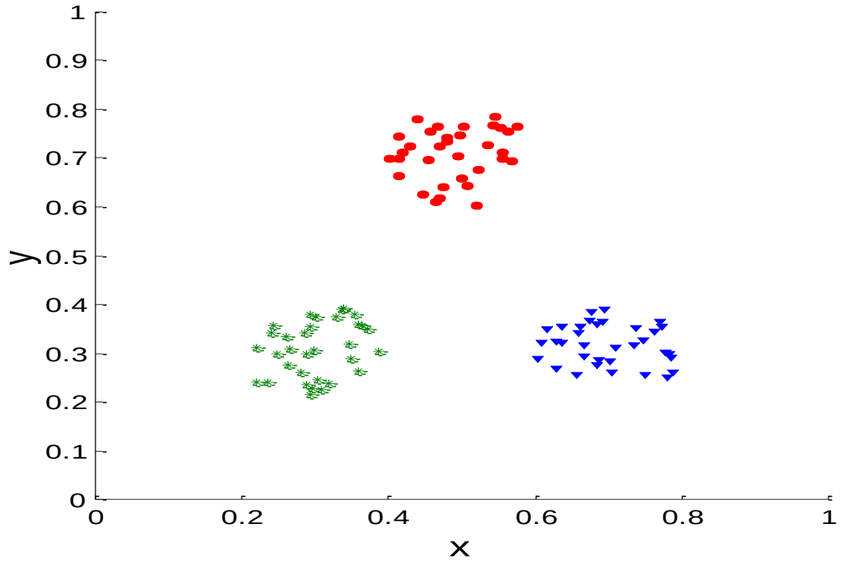

**Complete Link**

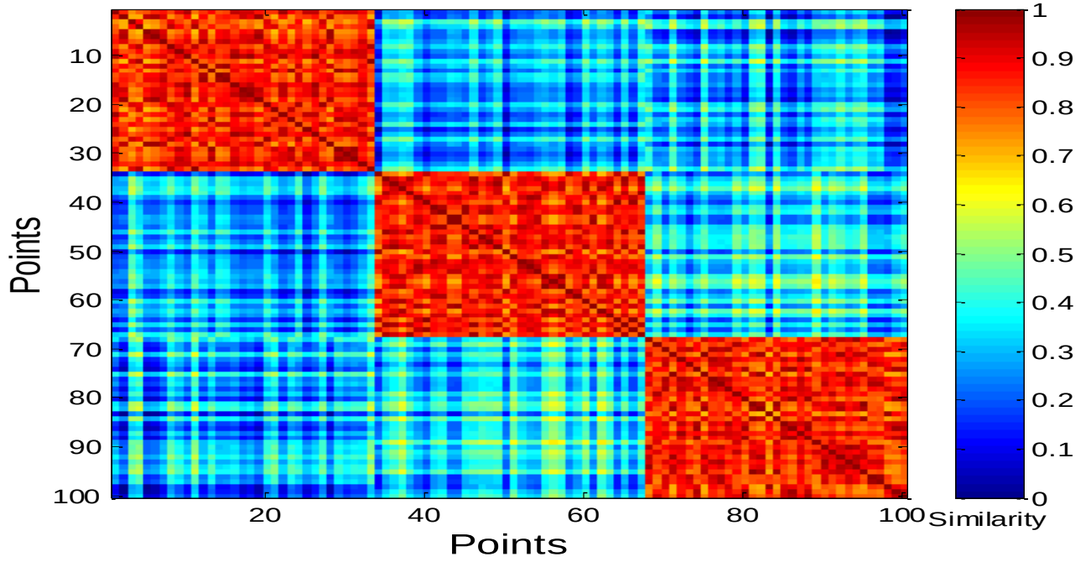
# Different Aspects of Cluster Validation

- Distinguishing whether non-random structure actually exists in the data
- Comparing the results of a cluster analysis to externally known results, e.g., to externally given class labels
- Evaluating how well the results of a cluster analysis fit the data *without* reference to external information
- Comparing the results of two different sets of cluster analyses to determine which is better
- Determining the ‘correct’ number of clusters

# Using Similarity Matrix for Cluster Validation

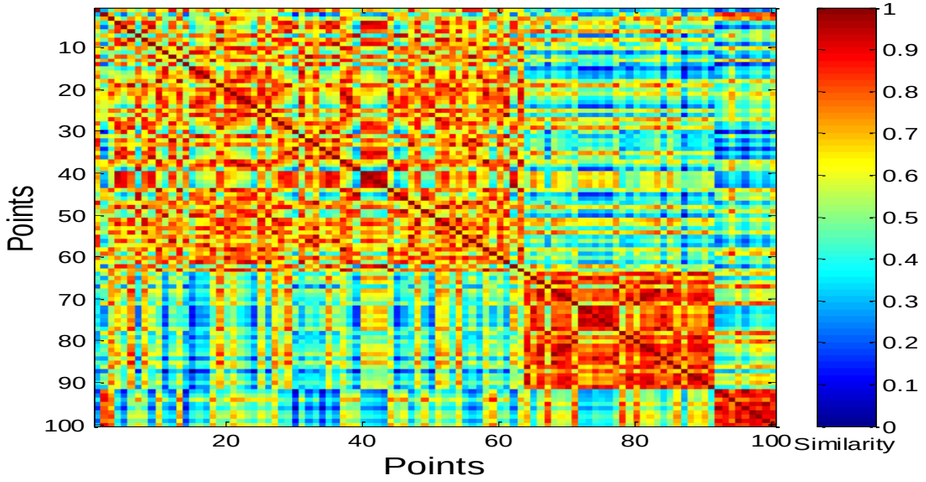
- Order the similarity matrix with respect to cluster labels and inspect visually.

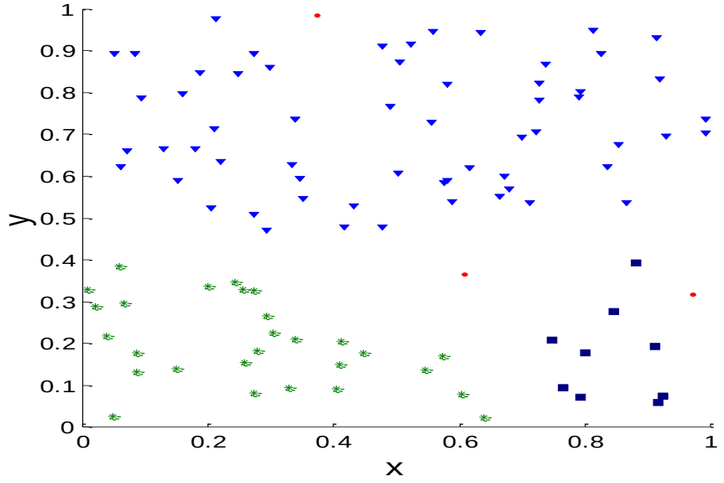


# Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp

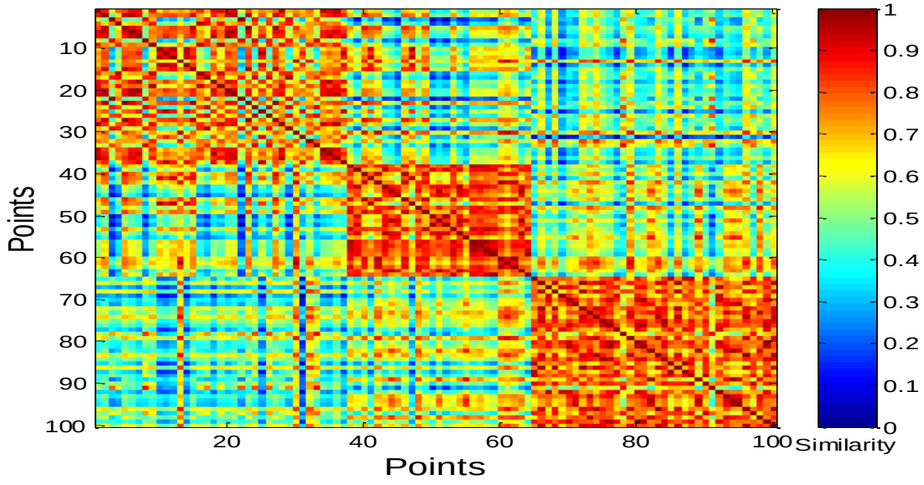


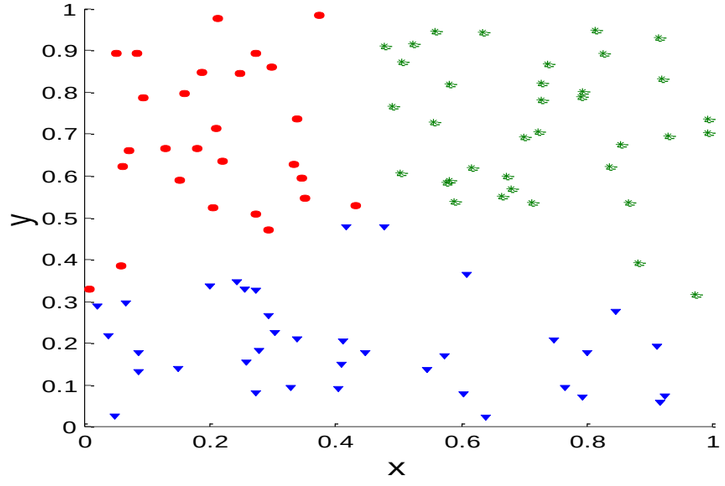




**DBSCAN**

# Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp

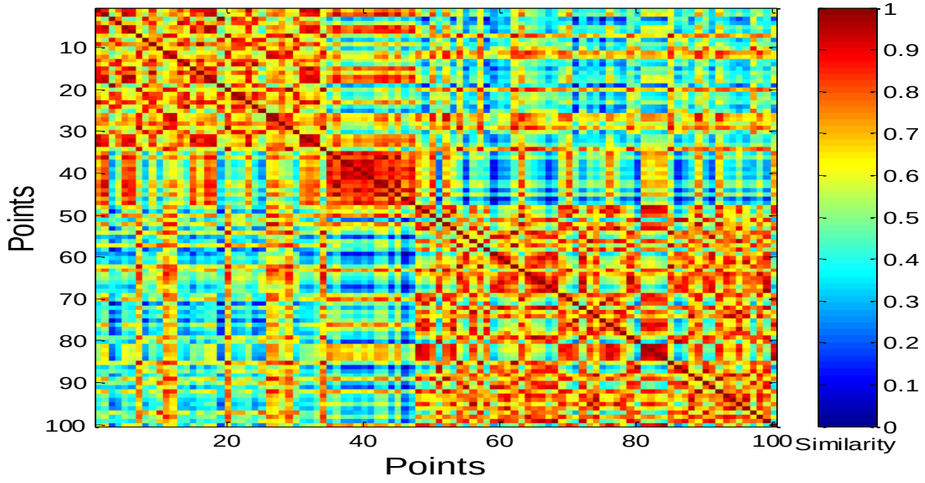


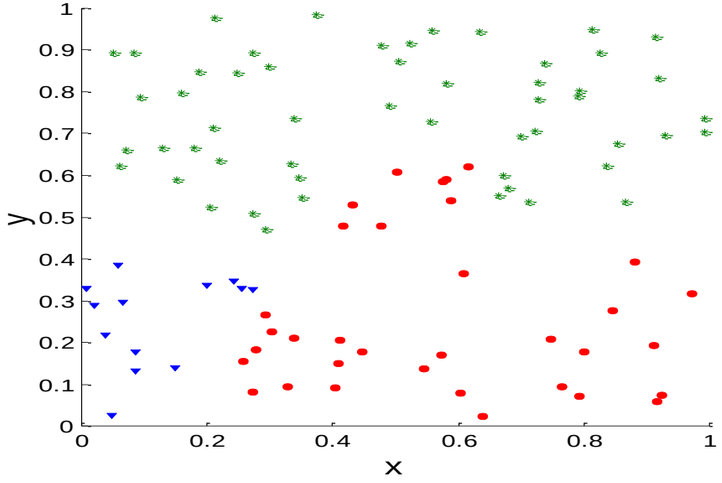




**K-means**

# Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp




**Complete Link**

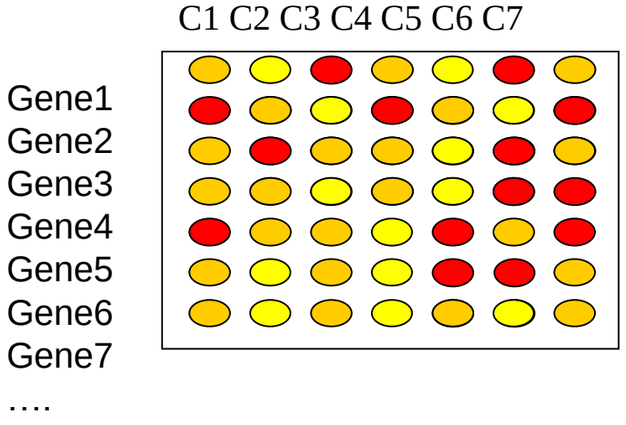
# Measures of Cluster Validity

- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types of indices.
  - External Index: Used to measure the extent to which cluster labels match externally supplied class labels.
    - Entropy
  - Internal Index: Used to measure the goodness of a clustering structure *without* respect to external information.
    - Sum of Squared Error (SSE)
  - Relative Index: Used to compare two different clusterings or clusters.
    - Often an external or internal index is used for this function, e.g., SSE or entropy

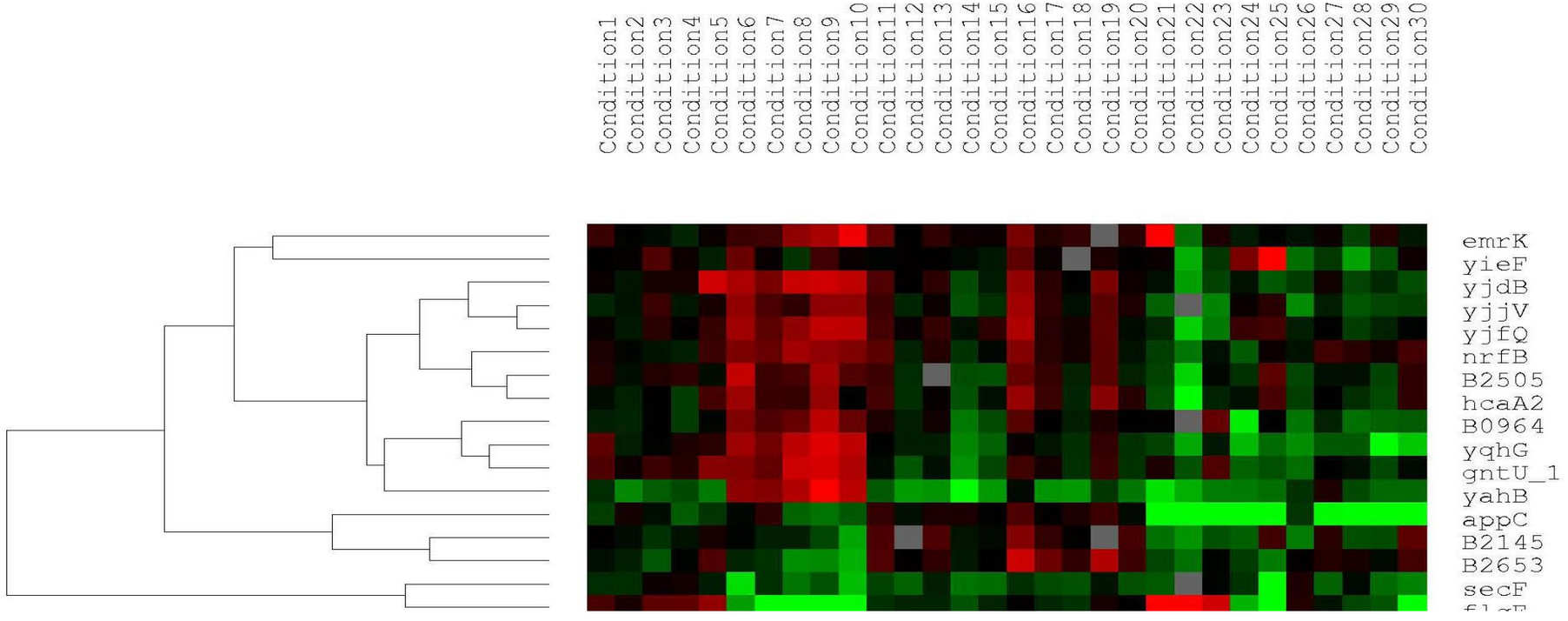
# Clustering Microarray Data

# Clustering Microarray Data

- Microarray analysis allows the monitoring of the activities of many genes over many different conditions
- Data: Expression profiles of approximately 3606 genes of E Coli are recorded for 30 experimental conditions
- SAM (Significance Analysis of Microarrays) package from Stanford University is used for the analysis of the data and to identify the genes that are substantially differentially upregulated in the dataset – 17 such genes are identified for study purposes
- Hierarchical clustering is performed and plotted using TreeView

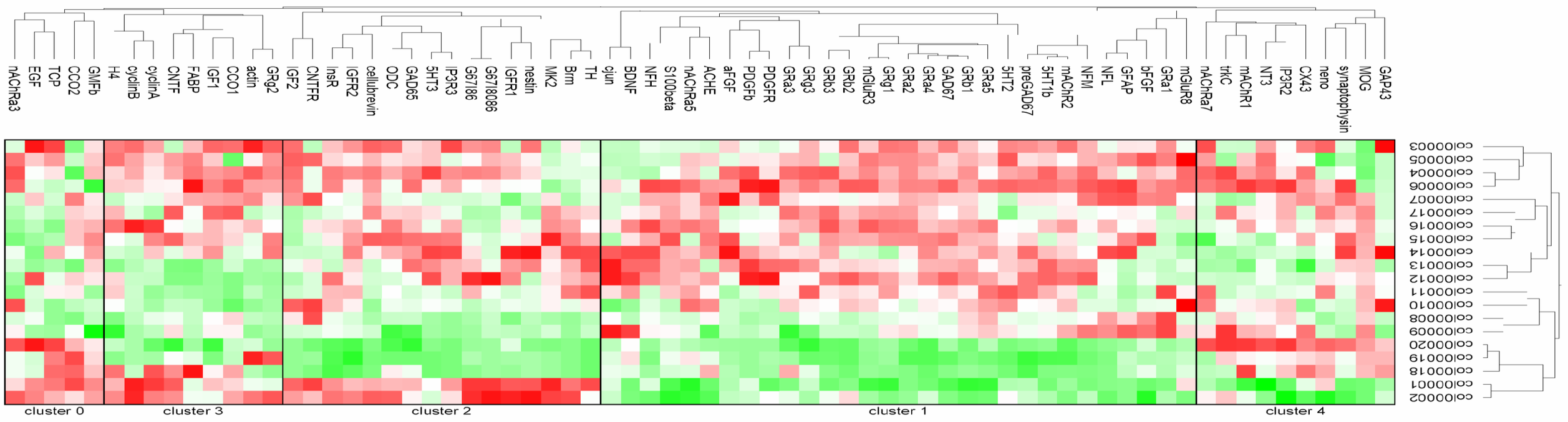


# Clustering Microarray Data…



# CLUTO for Clustering for Microarray Data

- CLUTO (Clustering Toolkit) George Karypis (UofM) CLUTO can also be used for clustering microarray data



# Issues in Clustering Expression Data

- Similarity uses all the conditions
  - We are typically interested in sets of genes that are similar for a relatively small set of conditions
- Most clustering approaches assume that an object can only be in one cluster
  - A gene may belong to more than one functional group
  - Thus, overlapping groups are needed
- Can either use clustering that takes these factors into account or use other techniques
  - For example, association analysis

# Clustering Packages

- Mathematical and Statistical Packages
  - MATLAB
  - SAS
  - SPSS
  - R

# Association Analysis

# Birliktelik kuralı keşfi

| TID | Items |
|---|---|
| 1 | Bread, Coke, Milk |
| 2 | Beer, Bread |
| 3 | Beer, Coke, Diaper, Milk |
| 4 | Beer, Bread, Diaper, Milk |
| 5 | Coke, Diaper, Milk |

Rules Discovered:
{Milk} --> {Coke}
{Diaper, Milk} --> {Beer}

Beş müşterinin market alış veriş listesi incelendiğinde süt alan kola da alıyor. Bebek bezi ve süt alan bira da alıyor.

# Association Analysis

- Bir dizi kayıt verildiğinde, kayıttaki diğer öğelerin oluşumlarına dayalı olarak bir öğenin oluşumunu tahmin edecek bağımlılık kurallarını bulunur.

| TID | Items |
|---|---|
| 1 | Bread, Coke, Milk |
| 2 | Beer, Bread |
| 3 | Beer, Coke, Diaper, Milk |
| 4 | Beer, Bread, Diaper, Milk |
| 5 | Coke, Diaper, Milk |

Rules Discovered:

**{Milk} --> {Coke} (s=0.6, c=0.75)**

**{Diaper, Milk} --> {Beer} (s=0.4, c=0.67)**

$$\text{Support, } s = \frac{\#\text{ transactions that contain X and Y}}{\text{Total transactions}}$$

$$\text{Confidence, } c = \frac{\#\text{ transactions that contain X and Y}}{\#\text{ transactions that contain X}}$$

- Applications
  - Pazarlama ve Satış Promosyonu
  - Süpermarket raf yönetimi
  - Trafik paterni analizi (örneğin, "58. Kavşaktaki yüksek sıkışıklık, sola dönüş trafiği için yüksek kaza oranları anlamına gelir" gibi kurallar)

# Birliktelik Kuralı Madenciliği Görevi
# ( Association Rule Mining Task )

- Given a set of transactions T, the goal of association rule mining is to find all rules having
  - support ≥ *minsup* threshold
  - confidence ≥ *minconf* threshold

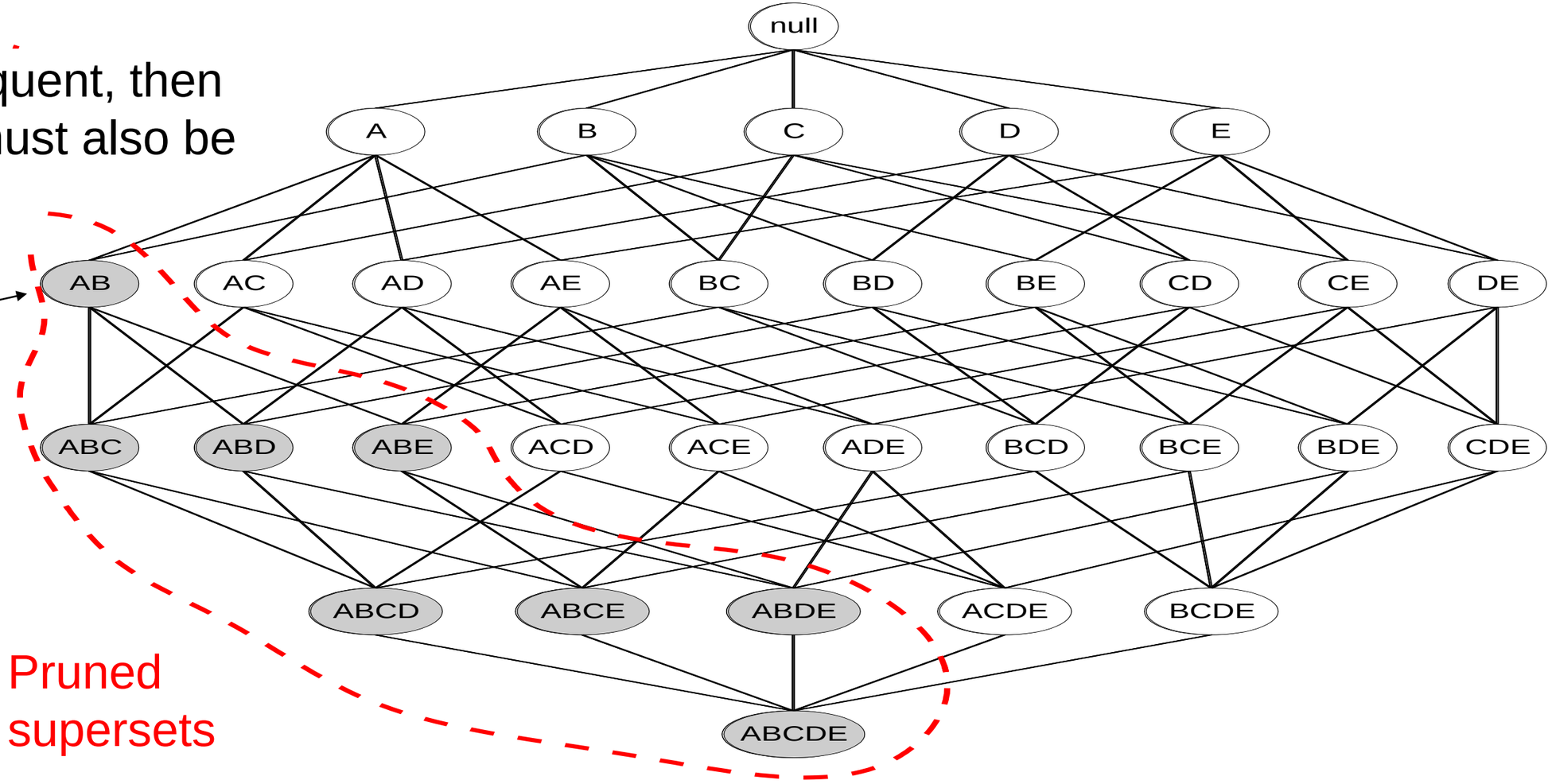
- Brute-force approach: Two Steps
  - Frequent Itemset Generation
    - Generate all itemsets whose support ≥ minsup
  - Rule Generation
    - Generate high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset

- Frequent itemset generation is computationally expensive

# Efficient Pruning Strategy

If an itemset is infrequent, then all of its supersets must also be infrequent

Found to be Infrequent




Pruned supersets

# Illustrating Apriori Principle

| Item | Count |
|---|---|
| Bread | 4 |
| Coke | 2 |
| Milk | 4 |
| Beer | 3 |
| Diaper | 4 |
| Eggs | 1 |

Items (1-itemsets)

| Itemset | Count |
|---|---|
| {Bread,Milk} | 3 |
| {Bread,Beer} | 2 |
| {Bread,Diaper} | 3 |
| {Milk,Beer} | 2 |
| {Milk,Diaper} | 3 |
| {Beer,Diaper} | 3 |

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

Triplets (3-itemsets)

| Itemset | Count |
|---|---|
| {Bread,Milk,Diaper} | 3 |

Minimum Support = 3

If every subset is considered,

$${}^6C_1 + {}^6C_2 + {}^6C_3 = 41$$

With support-based pruning,

$$6 + 6 + 1 = 13$$

# Association Measures

- Association measures evaluate the strength of an association pattern
  - Support and confidence are the most commonly used
  - The support, $\sigma(X)$, of an itemset $X$ is the number of transactions that contain all the items of the itemset
    - Frequent itemsets have support > specified threshold
    - Different types of itemset patterns are distinguished by a measure and a threshold
  - The confidence of an association rule is given by $conf(X \rightarrow Y) = \sigma(X \cup Y) / \sigma(X)$
    - Estimate of the conditional probability of $Y$ given $X$
- Other measures can be more useful
  - H-confidence
  - Interest

# Usage Notes

- A lot of slides are adopted from the presentations and documents published on internet by experts who know the subject very well.
- I would like to thank  who prepared slides and documents.
- Also, these slides are made publicly available on the web for anyone to use
- If you choose to use them, I ask that you alert me of any mistakes which were made and allow me the option of incorporating such changes (with an acknowledgment) in my set of slides.

Sincerely,
Dr. Cahit Karakuş
**cahitkarakus@gmail.com**