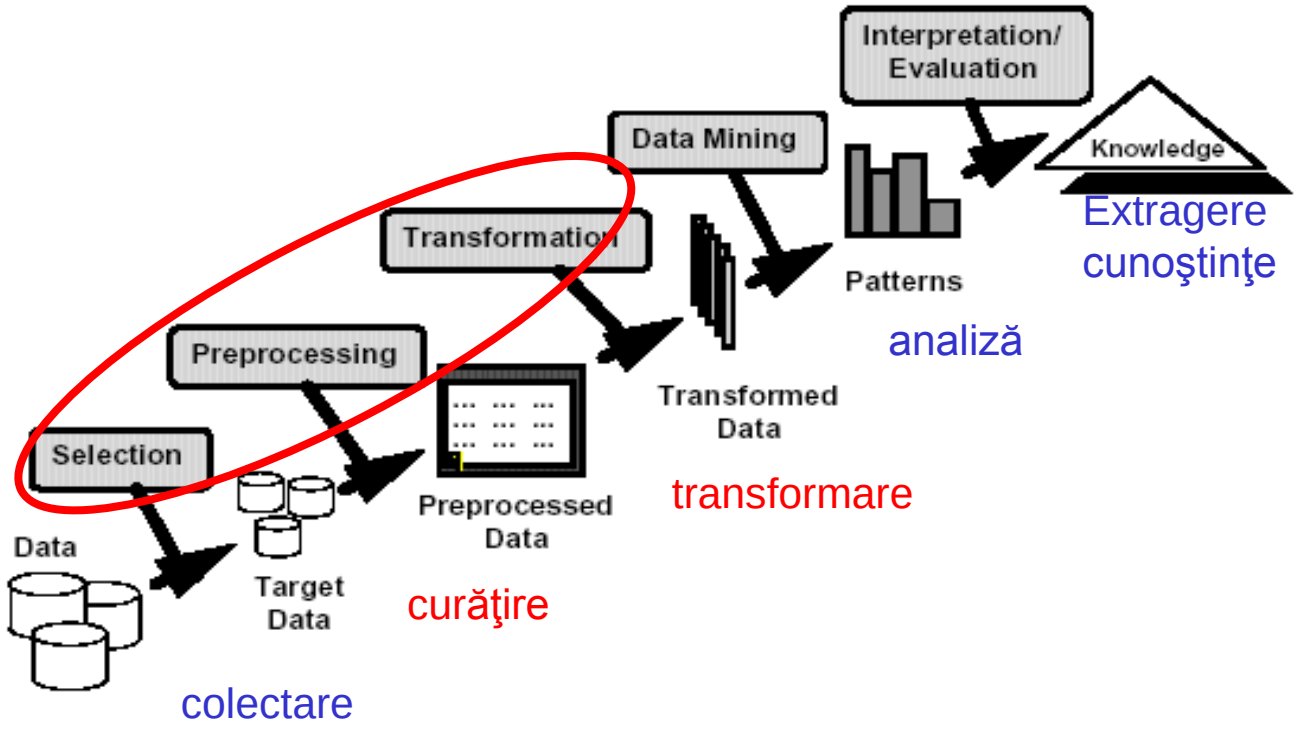
# Curs 2:

# Pre-procesarea datelor

# Structura

- Reminder: etape extragere cunoştinţe din date
- Extragerea caracteristicilor
- Tipuri de atribute
- Curatirea datelor
- Reducerea dimensiunii datelor
- Transformarea caracteristicilor

# Etape extragere cunoştinţe din date



# Etape extragere cunoştinţe din date

Exemplu: un comerciant care deţine un sistem de comerţ electronic este interesat să obţină informaţii referitoare la comportamentul clienţilor săi cu scopul de a recomanda anumite produse

Surse de date:

- Fişiere de tip log cu informaţii de conectare

```
98.206.207.157 - - [31/Jul/2013:18:09:38 -0700] "GET
/productA.htm HTTP/1.1" 200 328177 "-" "Mozilla/5.0 (Mac OS X)
AppleWebKit/536.26 (KHTML, like Gecko) Version/6.0
Mobile/10B329 Safari/8536.25” "retailer.net"
```

- Informaţii demografice colectate în procesul de înregistrare al utilizatorilor - stocate într-o bază de date (ex: e-mail, telefon, oraş, categorie de vârstă, categorie profesională)

Cum ar putea fi folosite aceste informaţii?

# Etape extragere cunoştinţe din date

Cum ar putea fi folosite aceste informaţii?

Comerciantul ar dori să determine care sunt produsele achiziţionate de către fiecare client

Aceasta necesită:

- Stabilirea corespondenţei între înregistrările din fişierele cu informaţii de logare şi baza de date cu informaţii privind clienţii (problema: erorile pot conţine erori care îngreunează procesul -> poate fi necesară curăţirea datelor)
- Agregarea tuturor informaţiilor de logare corespunzătoare unui client (Problema: nu toate informaţiile sunt neapărat utile -> ar putea necesita selecţie)
- Integrarea informaţiilor din ambele surse de date (ar putea necesita transformarea datelor)

# Etape extragere cunoştinţe din date

Principalele etape

- Colectarea datelor (din diferite surse)
- Pre-procesarea datelor
  - Extragerea caracteristicilor (specifice problemei de rezolvat)
  - Curăţirea datelor (ex: eliminarea înregistrărilor eronate sau completarea valorilor absente)
  - Selecţia caracteristicilor (ignoră atributele irelevante, redundante sau inconsistente)
  - Transformarea datelor/ atributelor
    - Transformarea valorilor unui atribut:
      - Numeric -> nominal/ordinal (e.g. valoarea vârstei este transformată într-o categori: foarte tânăr, tânăr, bătrân, foarte bătrân);
      - Nominal -> logic/binar (e.g. fiecărei valori posibile a unui atribut nominal i se asociază un atribut binar )
    - Transformă un set de atribute în alt set de atribute care poartă mai multă informaţie (e.g. explică mai bine variabilitatea din date)
- Analiza datelor (extragerea de cunoştinţe din date)

# Extragerea caracteristicilor

Scop:

- Extragerea caracteristicilor semnificative din datele brute (datele pot proveni din diferite surse)

Particularitate:

- Procesul de extragere depinde de specificul domeniului şi necesită expertiză în domeniul respectiv

Exemple: extragerea caracteristicilor din

- imagini
- documente  (XML, PDF)
- web logs
- date privind trafic în reţea

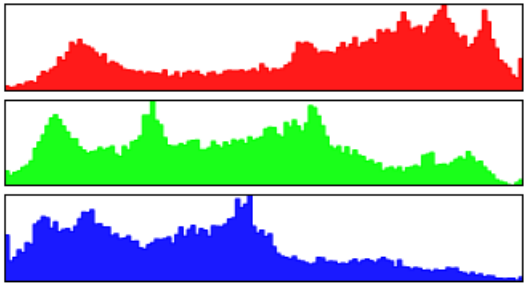
# Extragerea caracteristicilor

Extragerea informaţiilor privind textura dintr-o imagine:

Abordare bazată pe histogramă:

- Construirea histogramei color (pentru fiecare bandă de culoare şi pt fiecare regiune din imagine)

  H(v)=numărul de pixeli care au valoarea v
- Calcul valori statistice:
  - medie
  - varianţă
  - energie
  - entropie
  - [alţi indicatori statistici (skewness, kurtosis)]
- Obs: dacă imaginea este partiţionată în $K^2$ regiuni şi pt fiecare regiune şi fiecarea bandă de culoare sunt calculate 4 mărimi statistice atunci imaginii i se asociază un vector cu 12 $K^2$ caracteristici numerice



# Extragerea caracteristicilor

Extragerea caracteristicilor de textură dintr-o imagine:

Alte variante (ex: [http://www.eletel.p.lodz.pl/programy/cost/pdf_1.pdf]):

- Matrici de co-ocurenţă

| | | | |
|---|---|---|---|
| 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 2 | 2 | 2 |
| 2 | 2 | 3 | 3 |

Image example

| $i\backslash j$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | #(0,0) | #(0,1) | #(0,2) | #(0,3) |
| 1 | #(1,0) | #(1,1) | #(1,2) | #(1,3) |
| 2 | #(2,0) | #(2,1) | #(2,2) | #(2,3) |
| 3 | #(3,0) | #(3,1) | #(3,2) | #(3,3) |

Construction of co-occurrence matrix

| | | | |
|---|---|---|---|
| 4 | 2 | 1 | 0 |
| 2 | 4 | 0 | 0 |
| 1 | 0 | 6 | 1 |
| 0 | 0 | 1 | 2 |

$h_{1,0^\circ}$

| | | | |
|---|---|---|---|
| 6 | 0 | 2 | 0 |
| 0 | 4 | 2 | 0 |
| 2 | 2 | 2 | 2 |
| 0 | 0 | 2 | 0 |

$h_{1,90^\circ}$

# Extragerea caracteristicilor

Extragerea caracteristicilor dintr-un document:

1. XML - date semistructurate

```
<PersonalData><PersonDescriptors><DemographicDescriptors><Nationality>francaise</Nationality>
</DemographicDescriptors>
<BiologicalDescriptors><DateOfBirth>1978-01-16</DateOfBirth>
<GenderCode>1</GenderCode>
</BiologicalDescriptors>
</PersonDescriptors>
</PersonalData>
```

…

Prin parsare, se pot extrage caracteristicile demografice:

| Nationalitz | Date of birth | Gender |
|---|---|---|
| Francaise | 1978-01-16 | 1 |

# Extragerea caracteristicilor

Extragerea caracteristicilor dintr-un document:

2. Fişier text – date nestructurate
3. Exemplu (abordarea bazată pe bag-of-words):

"In document classification, a bag of words is a sparse vector of occurrence counts of words; that is, a sparse histogram over the vocabulary. In computer vision, a bag of visual words is a vector of occurrence counts of a vocabulary of local image features."

a) Eliminarea cuvintelor de legătură (stop words)

"In document classification, a bag of words is a sparse vector of occurrence counts of words; that is, a sparse histogram over the vocabulary. In computer vision, a bag of visual words is a vector of occurrence counts of a vocabulary of local image features."

"document classification bag words sparse vector occurrence counts words sparse histogram vocabulary computer vision bag visual words vector occurrence counts vocabulary local image features."

# Extragerea caracteristicilor

Extragerea caracteristicilor dintr-un document – fişier text: abordarea de tip bag-of-words

b) Reducerea cuvintelor la rădăcina lor – stemming (algoritm Porter)

“document classification bag words sparse vector occurrence counts words sparse histogram vocabulariy computer vision bag visual words vector occurrence counts vocabulariy local image features”

[http://textanalysisonline.com/nltk-porter-stemmer]

“document classif bag word spars vector occurr count word spars histogram vocabulari comput vision bag visual word vector occurr count vocabulari local imag featur”

# Extragerea caracteristicilor

Extragerea caracteristicilor dintr-un document – fişier text: abordarea de tip bag-of-words

c) Calculul frecvenţelor:

“document classif bag word spars vector occurr count word spars histogram vocabulari comput vision bag visual word vector occurr count vocabulari local imag featur”

Caracteristici extrase:

(bag,2), (classif,1), (comput,1), (count,2), (document,1), (featur,1), (histogram,1), (imag,1), (local,1), (occurr,2), (spars,2), (vector,2), (vision,1), (visual,1), (vocabulari,2), (word,3)

# Extragerea caracteristicilor

Extragere caracteristici din fişier log:

```
192.168.198.92 - - [22/Dec/2002:23:08:37 -0400] "GET / HTTP/1.1" 200 6394
www.yahoo.com "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1...)" "-"
192.168.198.92 - - [22/Dec/2002:23:08:38 -0400] "GET /images/logo.gif HTTP/1.1"
200 807 www.yahoo.com "http://www.some.com/" "Mozilla/4.0 (compatible; MSIE
6...)" "-"
192.168.72.177 - - [22/Dec/2002:23:32:14 -0400] "GET /news/sports.html HTTP/1.1"
200 3500 www.yahoo.com "http://www.some.com/" "Mozilla/4.0 (compatible; MSIE
...)" "-"
```

Prin parsarea fişierului se pot extrage:

| Client IP address | Date | Time | Request command | etc. |
|---|---|---|---|---|
| 192.168.198.92 | 22/Dec/2002 | 23:08:37 | GET / HTTP/1.1 | |
| 192.168.198.92 | 22/Dec/2002 | 23:08:38 | GET /images/logo.gif HTTP/1.1 | |
| 192.168.72.177 | 22/Dec/2002 | 23:32:14 | GET /news/sports.htmlHTTP/1.1 | |

# Extragerea caracteristicilor

Rezultatul unui proces de extragere:

- Matrice de date: fiecare linie corespunde unei înregistrări (articol sau instanţă), fiecare coloană corespunde unei caracteristici (atribut)

Exemplu (CV data):

| | Nationality | Date of birth | Gender |
|---|---|---|---|
| CV 1: | Francaise | 1978-01-16 | 1 |
| CV 2: | Roman | 1965-09-01 | 2 |

….

- Set de instanţe, fiecare instanţa = listă de valori ale caracteristicilor

Exemplu (fişier text):

Fişier 1: (bag,2), (classif,1), (comput,1), (count,2), (document,1), (featur,1), (histogram,1), (imag,1), (local,1), (occurr,2), (spars,2), (vector,2), (vision,1), (visual,1), (vocabulari,2), (word,3)

Fişier 2: …

# Tipuri de caracteristici/ atribute

- Numerice (cantitative)

Exemple: vârsta, greutate, preţ, cantitate, temperatură etc.

Specific:

- Valorile atributelor cantitative sunt numere (întregi sau reale)
- Se poate defini o ordine între valori (i.e. se poate calcula: minim, maxim, mediana şi se pot ordona valorile)
- Se pot efectua operaţii aritmetice:
  - Calcul  medie, varianţă şi alţi indicatori statistici
  - Alte operaţii:  adunare, scădere, înmulţire, împărţire etc (e.g. valoare = preţ*cantitate)

Obs:  un caz particular este reprezentat de valorile de tip data calendaristică sau oră (ex: 1975-01-16); are sens să se compare sau să se calculeze diferenţa dintre date dar nu are sens să se înmulţească)

# Tipuri de caracteristici/ atribute

- Ordinale (valori discrete aparţinând unei mulţimi ordonate)

Exemple:

Nivele de calitate (e.g: inacceptabil, acceptabil, bun, foarte bun, excelent)

Nivele ale unei caracteristici (e.g: foarte scăzut, scăzut, mediu, ridicat, foarte ridicat)

Specific:

- Valorile pot fi numere, simboluri, şiruri
- Există relaţie de ordine pe mulţimea valorilot (i.e. se poate calcula minim, maxim, mediana şi se pot ordona valorile)
- Nu are sens să se efectueze operaţii aritmetice

# Tipuri de caracteristici/ atribute

- Nominale/ categoriale (valori discrete aparţinând unei mulţimi pe care nu se este definită o relaţie de ordine )

Exemple:

Gen (e.g: female, male)

Rasă (e.g. caucaziană, asiatică, africană etc)

Stare civilă

Specific:

- Valorile unei astfel de caracteristici pot fi simboluri, siruri de caractere etc
- Nu se pot aplica operaţii aritmetice sau de ordonare
- Operatii:
  - Verificare egalitate
  - Calcul frecvenţe

# Tipuri de caracteristici/ atribute

- Binar (doar două valori posibile: {0,1} sau {False, True})
  - Se utilizează pentru a codifica absenţa/prezenţa unor caracteristici
  - Permite specificarea unor submulţimi (interpretate ca funcţii indicator)

Exemplu: set de tranzacţii

T1: {lapte, pâine, carne}

T2: {pâine, apă}

T3: {unt, carne}

T4: {apă}

| Trans. | pâine | unt | carne | lapte | apă |
|---|---|---|---|---|---|
| T1 | 1 | 0 | 1 | 1 | 0 |
| T2 | 1 | 0 | 0 | 0 | 1 |
| T3 | 0 | 1 | 1 | 0 | 0 |
| T4 | 0 | 0 | 0 | 0 | 1 |

- Obs: este un exemplu de conversie a datelor (de la nominal la binar)

# Conversii între tipuri

Conversia unui atribut numeric într-unul categorial (discretizare)

- Motivation: anumite tehnici de data mining pot fi aplicate doar pt date categoriale
- Idee principală:
  - Domeniul de valori se împarte în subdomenii
  - Se asignează o valoare fiecărui subdomeniu

Exemplu: considerăm atributul “vârsta” care ia valori în intervalul [0,100]; atributul numeric se poate transforma într-unul categorial după cum urmează

| Subdomeniu | Valoare |
|---|---|
| [0, 10) | 1 |
| [10,20) | 2 |
| [20,30) | 3 |
| [30,40) | 4 |
| … | |
| [90,100] | 10 |

# Conversii între tipuri

Conversia unui atribut numeric într-unul categorial (discretizare)

Obs:

- Prin discretizare se pierde o parte din informaţie
- O discretizare uniformă (ca în ex. anterior) nu e întotdeauna cea mai adevată (de exemplu intervalul [90,100] conţine de regulă mai puţine valori decât celelalte intervale).

Alte variante:

- Equi-log: domeniul [a,b] este divizat în K subdomenii $[a_1,b_1),[a_2,b_2),\ldots[a_K,b_K]$ a.î. $\log(b_i)-\log(a_i)$ este constant (în loc de $b_i-a_i$)

# Conversii între tipuri

Conversia unui atribut numeric într-unul categorial (discretizare)

- Equi-depth: fiecare subdomeniu are acelaşi număr de înregistrări
- Equi-label: fiecare subdomeniu conţine valori care aparţin aceleiaşi clase (în contextul unei probleme de clasificare pentru care se cunoaşte un set de date etichetate)

Exemplu (valorile atributului “vârsta” sunt ordonate crescător):

Vârsta: 15, 16, 16, 20, 20, 20, 25,26,27,30,30,31

Clasa: c1, c2, c2, c1, c1, c1, c2,c2,c1, c2,c2,c1

Equi-depth: [15,18), [18,22.5), [22.5,28.5), [28.5,31)

Equi-label: [15,15.5), [15.5, 18), [18,22.5), [22.5,26.5), [26.5,28.5), [28.5,30.5), [30.5,31)

# Conversii între tipuri

Conversia atributelor nominale în atribute binare (binarizare)

Motivaţie: există tehnici de data mining (e.g. reţelele neuronale) care nu pot prelucra direct atribute nominale

Procedura: un atribut nominal A care ia valori în mulţimea $\{v_1,v_2,\ldots,v_r\}$ este transformat in r atribute binare $Av_1,Av_2,\ldots,Av_r$ a.î. într-o instanţă dată doar unul dintre atribute va avea valoarea 1 iar toate celelalte vor avea valoarea 0.

Exemplu: considerăm atributul social din setul “nursery”

@attribute social {nonprob,slightly_prob,problematic}

şi valori corespunzătoare câtorva instanţe:

| | A_nonprob | A_slightly_prob | A_problematic |
|---|---|---|---|
| slightly_prob | 0 | 1 | 0 |
| nonprob | 1 | 0 | 0 |
| nonprob | 1 | 0 | 0 |
| problematic | 0 | 0 | 1 |

# Curătirea datelor

Scop: eliminarea erorilor şi a inconsistenţelor din date

Tipuri de erori:

- Valori greşite
- Valori absente

Cauze ale erorilor:

- Defecte în dispozitivele de înregistrare a datelor (e.g. senyori)
- Erori umane (e.g. completare greşită)
- Absenţa răspunsurilor (e.g. date confidenţiale)

Valoare eronată

| Pacient | Vârsta | Inălţime [cm] | Greutate[kg] |
|---|---|---|---|
| P1 | 20 | 170 | 60 |
| P2 | 10 | 1.30 | 30 |
| P3 | 22 | 165 | ? |
| P4 | 8 | 190 | 80 |

Valoare absentă

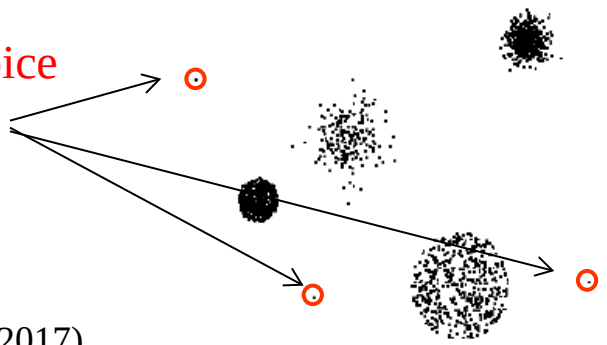
Date inconsistente

# Curătirea datelor

Descoperirea şi corecţia valorilor eronate:

- Utilizând cunoştinţe specifice domeniului (e.g. se pot defini domenii de valori normale)
- Căutând inconsistenţe între valorile aceluiaşi atribut folosind diferite surse de date (e.g. Numele unei persoane poate fi specificat în mai multe moduri, “Ioan Popescu”, “I. Popescu”, “Ioan Popesu”; rasa unei persoane este specificată diferit în diferite instanţe)
- Utilizând o abordare statistică (e.g. Se presupune că datele sunt generate de o distribuţie normală iar valorile atipice sunt considerate erori)

Excepţii,
valori atipice



[Tan, Steinbach, Kumar – Introduction to Data Mining]

# Curătirea datelor

Cauze ale valorilor absente:

- Omitere în procesul de colectare
- Informaţii care nu sunt furnizate (e.g. Vârsta  sau genul  într-un chestionar)
- Informaţii nerelevante în anumite contexte (e.g. valoarea venitului în cazul unor copii)

Tratarea valorilor absente:

- Eliminarea înregistrărilor care conţin valori absente
- Asignarea unor valori specifice (e.g. Valoarea absentă este marcată cu 0 iar 0 este considerată o valoare posibilă pt acel atribut)
- Estimarea valorii absente (o astfel de aboradre este denumită imputare) utilizând valori corespondent din înregistrări “similare”.  In exemplul anterior s-ar putea folosi 60 (întrucât P1 şi P3 sunt similare în raport cu celelalte atribute). Dacă sunt mai multe înregistrări “similare” atunci se poate folosi valoarea medie a atributului

# Selecţia atributelor

Scop:

- Reducerea dimensiunii datelor
- Imbunătăţirea modelului de analiză a datelor (prin eliminarea atributelor redundante)

Exemple:

- Atribute irelevante (e.g. ID)
- Atribute corelate (e.g. BMI=weight/height$^2$)

| **Pacient** | **Vârsta** | **Inalţime [m]** | **Greutate [kg]** | **BMI** | **ID** | **Clasa** |
|---|---|---|---|---|---|---|
| P1 | 20 | 1.70 | 60 | 20.8 | 111 | normal |
| P2 | 15 | 1.30 | 30 | 17.8 | 222 | subponderal |
| P3 | 22 | 1.65 | 100 | 36.7 | 333 | obez |
| P4 | 48 | 1.90 | 80 | 22.2 | 444 | normal |

Obs: in practică relaţia dintre atribute este ascunsă astfel că nu este evident criteriul de selecţie

# Selecţia atributelor

Scop:

- Reducerea dimensiunii datelor
- Imbunătăţirea modelului de analiză a datelor (prin eliminarea atributelor redundante)

Componente ale unei metode de selecţie a atributelor:

- Criteriu de selecţie
- Metoda de căutare (în spaţiul de submulţimi ale atributelor)

Obs:

- Tehnica de selecţie a atributelor (in particular criteriul de selecţie) depinde de caracteristicile tipului de analiză a datelor

Variante:

- Metode nesupervizate de selecţie (e.g. utilizate în contextul grupării datelor)
- Metode supervizate de selecţie (e.g. utilizate în contextul clasificării datelor)

# Selecţia atributelor

Căutarea în spaţiul atributelor:

- Considerăm  o matrice de date cu  n atribute
- Spaţiul de căutare (toate submulţimile posibile de atribute) are dimensiunea $2^n$

Abordări posibile:

- Căutare exhaustivă: se analizează impactul fiecărei submulţimi de atribute asupra rezultatului;  e fezabilă doar dacă n este relativ mic
- Selecţie înainte:
  - Se porneşte cu un set vid de atribute
  - Se adaugă secvenţial câte un nou atribut (se analizează impactul fiecăruia dintre atributele rămase şi se selectează cel mai bun) – dacă adăugarea nici unui atribut nu îmbunătăţeşte calitatea rezultatului procesul se opreşte
- Selecţie înapoi:
  - Se porneşte cu întreg setul de atribute
  - se elimină secvenţial câte unul dintre atribute (cel prin a cărui eliminare se obţine cea mai mare îmbunătăţire a performanţei)

# Selecţie / ierarhizare / ponderare

- In anumite situaţii este mai util doar să se ierarhizeze atributele în ordinea descrescătoare a relevanţei şi să fie lăsat la latitudinea utilizatorului decizia

- Criteriul de ierarhizare este similar celui de selecţie (are ca scop să exprime relevanţa atributului în contextul problemei de analiză tratate)

- Ierarhizarea poate fi realizată asignând ponderi atributelor (o valoarea mai mare a ponderii sugerează că atributul este mai important)
  - Estimarea ponderilor conduce de regulă la necesitatea de a rezolva o problemă de optimizare (e.g determinarea ponderilor care minimizează pierderea de informaţie sau maximizează acurateţea)
  - Ponderile sunt importante în cazul în care tehnica de analiză se bazează pe calculul unor măsuri de similaritate (e.g. Clasificatori de tip nearest neighbor, clustering)

  Exemplu: distanţa euclidiană ponderată $d_w(x,y) = \sqrt{\sum_{i=1}^{n} w_i (x_i - y_i)^2}$

# Selecţia atributelor

Criteriu de selecţie - cum se poate evalua un subset de atribute (sau valorile corespunzătoare ale ponderilor)

- Abordare de tip ‘filtru’
  - Selecţia se bazează pe relaţia dintre:
    - atribute (context nesupervizat)
    - atribute şi etichete ale claselor (context supervizat)
- Abordare de tip ‘înveliş’ (wrapper)
  - Calitatea subsetului de atribute este estimată pe baza performanţei clasificatorului sau a modelului de grupare construit pe baza subsetului de atribute

# Selecţia atributelor

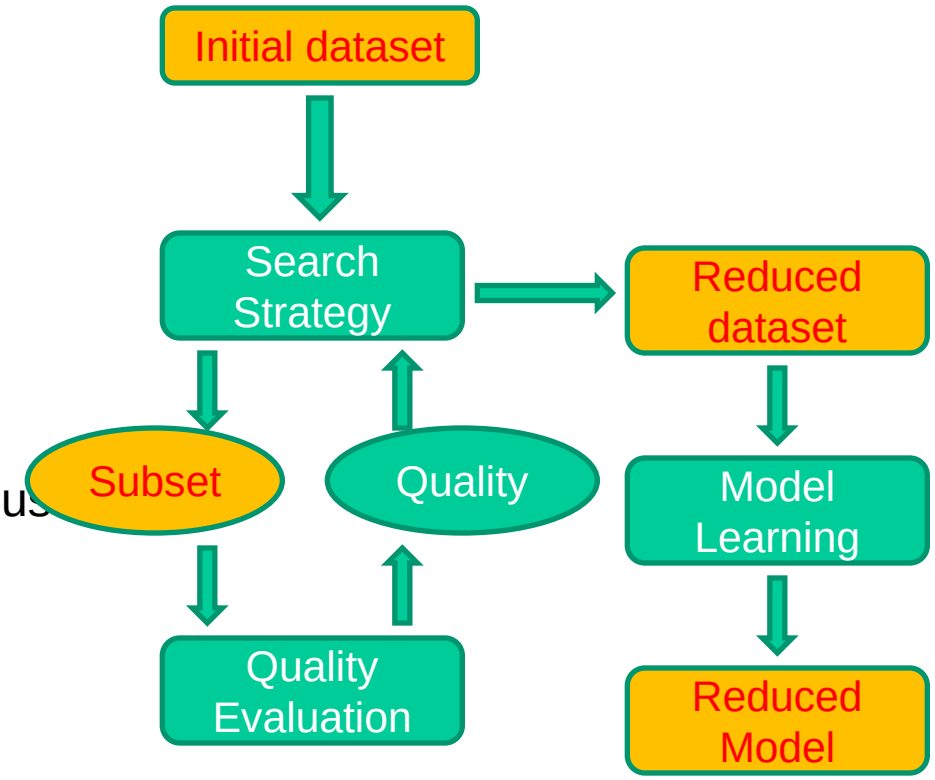
Abordare de tip filtru

Criterii bazate pe date

- Câştig informaţional
- Compacitate (within-class)
- Separare (between-classes)
- Corelare intre etichete de clase şi atribute
- Informaţie mutuală

Avantaj: cost computaţional relativ mic

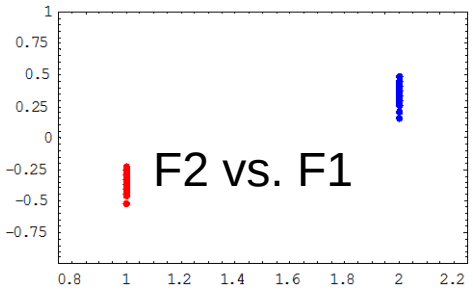
Dezavantaj: ignoră impactul setului redus de date asupra algoritmului de extragere a modelului din date



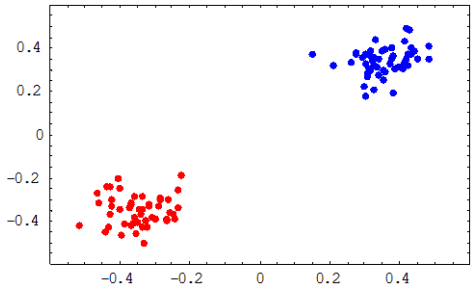
# Selecţia atributelor

Exemplu: set artificial de date: 10 atribute, 2 clase

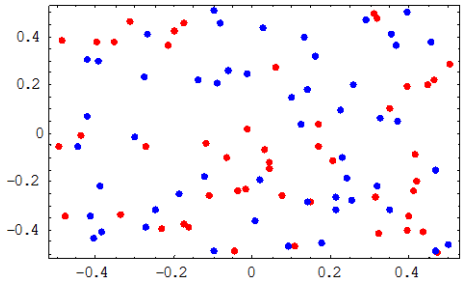
- Atribute 1: identic cu eticheta clasei
- Atribute 2-6: valori aleatoare cu rep. normală $N(m_1,s_1)$ (class 1), $N(m_2,s_2)$ (class 2)
- Atribute 7,8: valori constante pentru toate instanţele
- Atribute 9,10: valori aleatoare uniform repartizate (U(a,b)) pt toate instanţele




F2 vs. F1



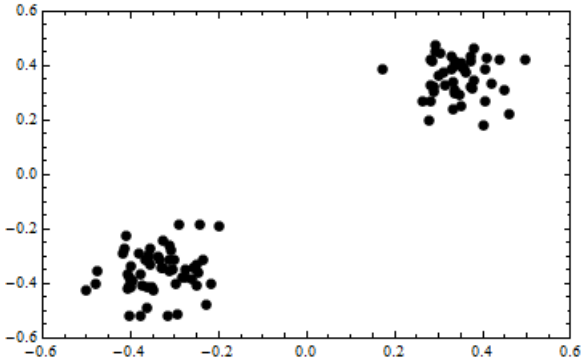

F6 vs. F5




F10 vs. F9

# Selecţia atributelor

Criteriu nesupervizat de selecţie (se bazează doar pe date – fără a se cunoaşte etichetele claselor)




Notaţii:

M={$x_1$,$x_2$,…,$x_N$} set de date cu N instanţe, fiecare conţinând n atribute

A= set de atribute

Idee:

- Se calculează similarităţile între perechile de date din set
- Se calculează entropia asociată matricii de similaritate (este o măsură a informaţiei conţinute în setul de date)
- Se analizează efectul fiecărui atribut asupra valorii entropiei şi se elimină atributele care au cel mai mic impact asupra entropiei

# Selecţia atributelor

## Criteriu nesupervizat de selecţie

Măsuri de similaritate (calculate folosind un set de atribute A)

Atribute numerice

$$S_{ij}(A) = \exp(-\alpha d(x_i, x_j)), \qquad d(x_i, x_j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2}$$

$$\alpha = \text{ct. } (\text{e.g. } 0.5)$$

Atribute nominale/ordinale/binare

$$S_{ij}(A) = \frac{1}{n} \sum_{k=1}^{n} I(x_{ik}, x_{jk}), \quad I(a,b) = 1 \text{ daca } a = b;$$

$$I(a,b) = 0 \text{ daca } a \neq b$$

# Selecţia atributelor

Criteriu nesupervizat de selecţie

Entropia $$E(S,A) = -\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}(S_{ij}(A)\ln(S_{ij}(A)) + (1-S_{ij}(A))\ln(1-S_{ij}(A)))$$

Obs: dpdv intuitiv, entropia măsoară impredictibilitatea conţinutului informaţional sau gradul de dezordine

Algoritm

Pas 1. se porneşte cu întreg setul de atribute A

Pas 2. pt fiecare atribut $a_i$ se calculează $E(S,A-\{a_i\})$ şi se ierarhizează atributele crescător după valoarea $E(S,A)- E(S,A-\{a_i\})$

Pas 3. se elimină primul atribut din lista ordonată (atributul a cărui eliminare a condus la cea mai mică pierdere în entropie) şi se repetă Pas 2 – Pas 3 până când rămâne un singur atribut în A (sau până când reducerea în entropie la eliminarea unui atribut depăşeşte un prag)

# Selecţia atributelor

Criteriu supervizat de selecţie – atribute cu valori discrete

Gini index: măsoară puterea de discriminare a unui atribut

Notaţii:

$A_1, A_2, \ldots, A_n$ - atribute, $C_1, C_2, \ldots, C_K$ – clase

$v_{i1}, v_{i2} \ldots, v_{ir}$ – valori posibile ale atributului i (se poate utiliza doar pt atribute cu valori discrete; $r_i$ numărul de valori ale atributului $A_i$)

index Gini pt atributul $A_i$

$$G(A_i) = \frac{1}{N} \sum_{j=1}^{r_i} n_{ij} G(v_{ij}), \; G(v_{ij}) = 1 - \sum_{k=1}^{K} p_{ijk}^2$$

$n_{ij}$ = numarul de instante pt care $A_i$ are valoarea $v_{ij}$

$$p_{ijk} = \frac{\text{numar de instante in } C_k \text{ cu } A_i = v_{ij}}{\text{numar de instante cu } A_i = v_{ij}}$$

Interpretare: valori mici ale lui $G(A_i)$ sugerează o putere mare de discriminare a lui $A_i$

# Selecţia atributelor

Criteriu supervizat de selecţie – atribute cu valori discrete

Scor Fisher: măsoară puterea de discriminare a unui atribut

Notaţii: $A_1, A_2, \ldots, A_n$ - atribute, $C_1, C_2, \ldots, C_K$ – clase

$v_{i1}, v_{i2} \ldots, v_{ir}$ – valori posibile ale atributului i (se poate utiliza doar pt atribute cu valori discrete; $r_i$ numărul de valori ale atributului $A_i$)

$$F(A_i) = \frac{\sum_{k=1}^{K} n_k (\mu_{ik} - \mu_i)^2}{\sum_{k=1}^{K} n_k \rho_{ik}^2}$$

$n_k$ = numar de instante in clasa $C_k$

$\mu_{ik}$ = media valorilor lui $A_i$ corespunzatoare instantelor din $C_k$

$\rho_{ik}^2$ = varianta valorilor

$\mu_i$ = media valorilor atributului $A_i$

Interpretare: valori mari ale lui $F(A_i)$ sugerează putere mare de discriminare pt $A_i$

# Selecţia atributelor

## Selecţie supervizată/criteriu de ponderare– atribute numerice

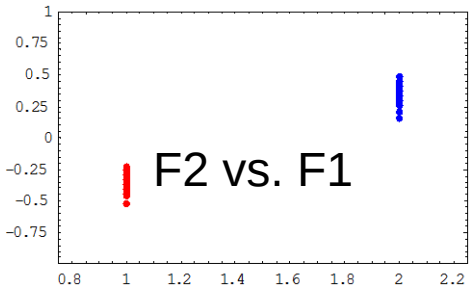
$\{x_i^c; i = \overline{1,N}\}$, $x_i^c \in R^n$, $c \in \{1,...,k\}$ eticheta clasa

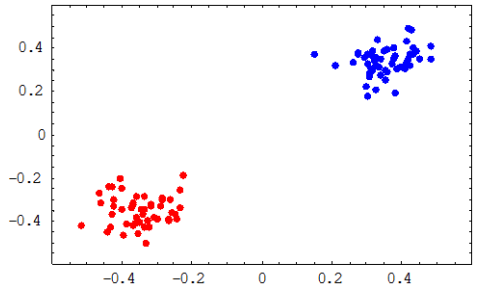
$w = (w_1,...,w_n)$, vector pondere

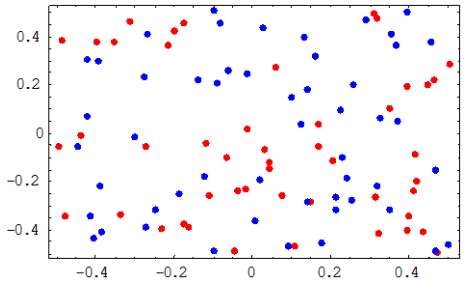
$d_w$ - masura disimilaritate

- Compacitate (within-class)

$$C_1(w) = \frac{1}{N}\sum_{c=1}^{k}\sum_{i=1}^{n_r} d_w(x_i^c, m_c),\ m_c = \frac{1}{n_c}\sum_{i=1}^{n_c} x_i^c$$




Exemplu:

Set artificial de date: 10 atribute, 2 clase

- Atribut 1: identic cu eticheta clasei
- Atribute 2-6: valori aleatoare din $N(m_1,s_1)$ (clasa 1), $N(m_2,s_2)$ (clasa 2)
- Atributes 7,8: valori constante
- Atribute 9,10: valori aleatoare cu repartiţia uniformă (U(a,b))

# Selecţia atributelor

Selecţie supervizată/criteriu de ponderare– atribute numerice

$\{x_i^c; i=\overline{1,N}\}$, $x_i^c \in R^n$, $c \in \{1,...,k\}$ eticheta clasa

$w=(w_1,...,w_n)$, vector pondere

$d_w$ - masura disimilaritate

- Compacitate (within-class)

$$C_1(w)=\frac{1}{N}\sum_{c=1}^{k}\sum_{i=1}^{n_c} d_w(x_i^c, m_c),\ m_c=\frac{1}{n_c}\sum_{i=1}^{n_c} x_i^c$$

(de minimizat)

- $C_1$(1,1,1,1,1,1,1,1,1,1,1)=0.88
- $C_1$(1,1,1,1,1,1,1,1,1,0,0)=0.78
- $C_1$(1,1,1,1,1,1,1,0,0,0,0)=0.78
- $C_1$(1,1,1,0,0,0,0,0,0,0,0)=0.49
- $C_1$(1,1,0,0,0,0,0,0,0,0,0)=0.34
- $C_1$(1,0,0,0,0,0,0,0,0,0,0)=0

Exemplu:

Set artificial de date: 10 atribute, 2 clase

- Atribut 1: identic cu eticheta clasei
- Atribute 2-6: valori aleatoare din N($m_1$,$s_1$) (clasa 1), N($m_2$,$s_2$) (clasa 2)
- Atributes 7,8: valori constante
- Atribute 9,10: valori aleatoare cu repartiţia uniformă (U(a,b))

# Selecţia atributelor

## Selecţie supervizată/criteriu de ponderare– atribute numerice

$\{x_i^c; i=\overline{1,N}\}$, $x_i^c \in R^n$, $c \in \{1,...,k\}$ eticheta clasa

$w=(w_1,...,w_n)$, vector pondere

$d_w$ - masura disimilaritate

- Separare (between-class)

$$C_2(w)=\frac{1}{N}\sum_{c=1}^{k} n_c d_w(m_c,m),\ m=\frac{1}{N}\sum_{c=1}^{k} n_c m_c$$

(de maximizat)

- $C_2$(1,1,1,1,1,1,1,1,1,1,1)=0.51
- $C_2$(1,1,1,1,1,1,1,1,1,1,0,0)=0.50
- $C_2$(1,1,1,1,1,1,1,1,0,0,0,0)=0.50
- $C_2$(1,1,1,0,0,0,0,0,0,0,0,0)=0.49
- $C_2$(1,1,0,0,0,0,0,0,0,0,0,0)=0.49
- $C_2$(1,0,0,0,0,0,0,0,0,0,0,0)=0.49
- $C_2$(1,0,0,0,0,0,0,0,0,0,1,1)=0.50

Exemplu:

Set artificial de date: 10 atribute, 2 clase

- Atribut 1: identic cu eticheta clasei
- Atribute 2-6: valori aleatoare din N($m_1$,$s_1$) (clasa 1), N($m_2$,$s_2$) (clasa 2)
- Atributes 7,8: valori constante
- Atribute 9,10: valori aleatoare cu repartiţia uniformă (U(a,b))

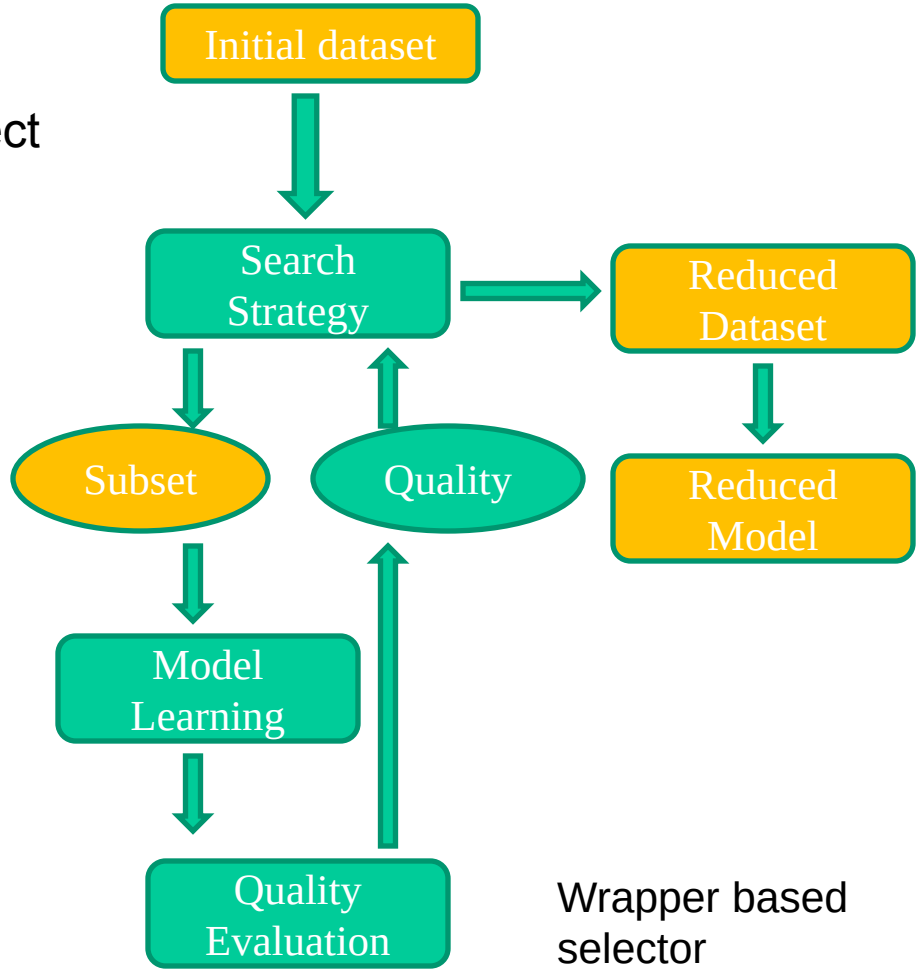
# Selecţia atributelor

Abordare de tip “înveliş”

- Acurateţe = număr de date corect clasificate/ număr total de date
- Obs: evaluarea fiecărei submulţimi necesită antrenarea întregului model

Avantaj: se foloseşte de impactul submulţimii de atribute asupra performanţei modelului

Dezavantaj: evaluarea este costisitoare




Wrapper based selector

# Selecţia instanţelor

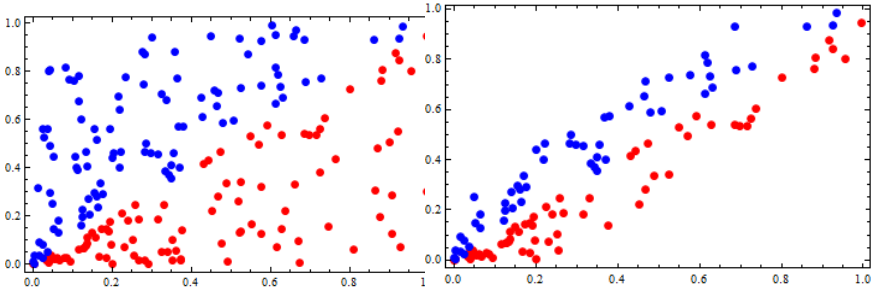
Selecţia poate fi aplicată nu doar atributelor ci şi instanţelor.

Exemplu (clasificare în 2 clase):

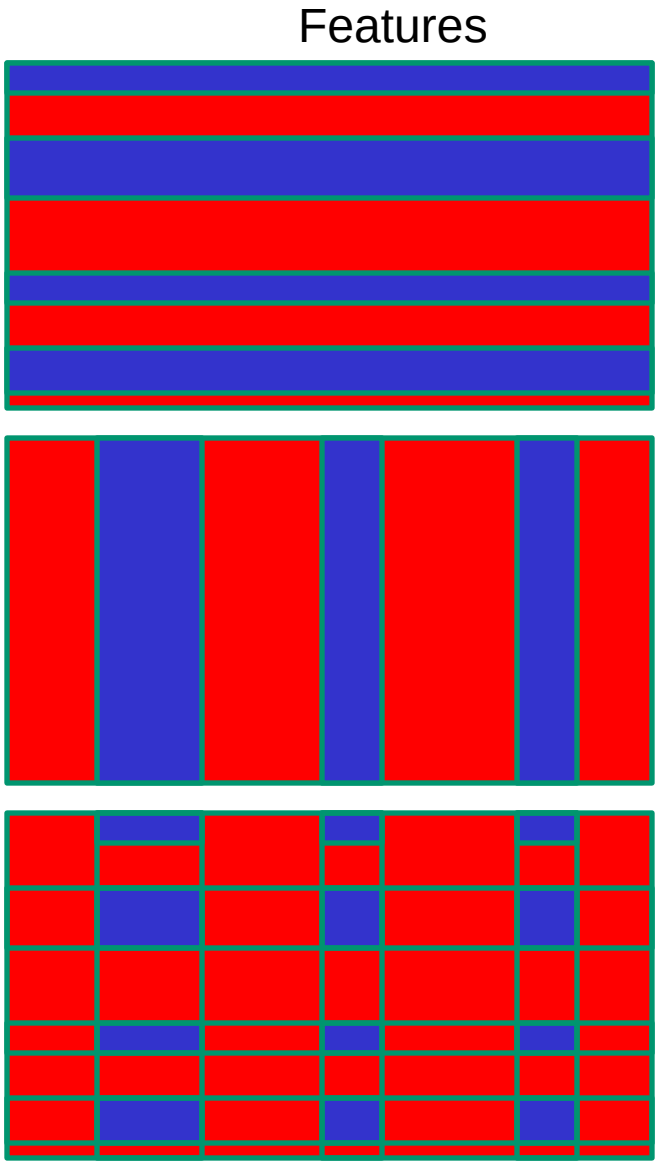
ar fi suficient să se folosească doar datele din vecinătatea frontierei celor două clase

Abordări:

- Selecţie aleatoare (cu sau fără revenire)
- Selecţie stratificată




simultan



# Transformarea atributelor

Scop:

- Îmbunătăţirea calităţii modelului extras din date prin eliminarea influenţei induse de scale diferite pt diferite atribute sau de corelaţii între atribute

Variante:

- Scalare
- Standardizare
- Normalizare
- Proiecţie – analiza componentelor principale (Principal Component Analysis)

Obs: aceste transformări pot fi aplicate doar atributelor numerice

# Normalizare

Scalare :

- Scalare liniară
- Este sensibilă la valorile atipice

Standardizare:

- Se scade media şi se împarte la abaterea standard
- Mai robustă decât scalarea liniară

Normalizare euclidiană:

- Se împarte fiecare componentă la normă (e.g. Norma euclidiană)

Scalare liniara :

$$z_i^j = \frac{x_i^j - \min_j}{\max_j - \min_j},\ i = \overline{1,n} \quad j = \overline{1,d}$$

Standardizare :

$$z_i^j = \frac{x_i^j - m(X^j)}{s(X^j)}, i = \overline{1,n} \quad j = \overline{1,d}$$

$$m(X^j) = \frac{1}{n}\sum_{i=1}^{n} x_i^j,\ s(X^j) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i^j - m(X^j))^2}$$

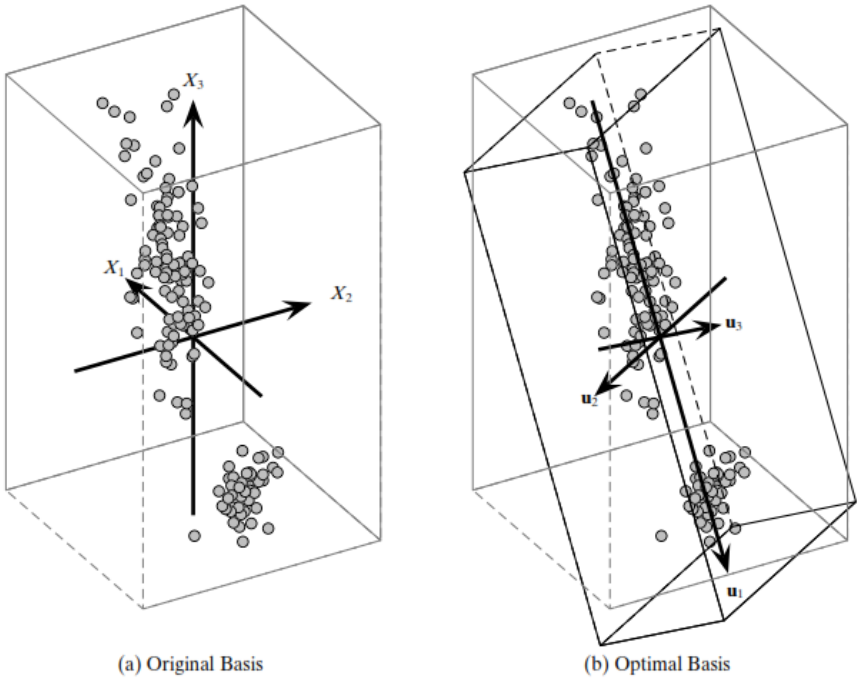
Normalizare :

$$Z_i = X_i / \|X\|,\ \|X\| = \sqrt{\sum_{j=1}^{d}(x_i^j)^2},\ i = \overline{1,n}$$

# Analiza componentelor principale

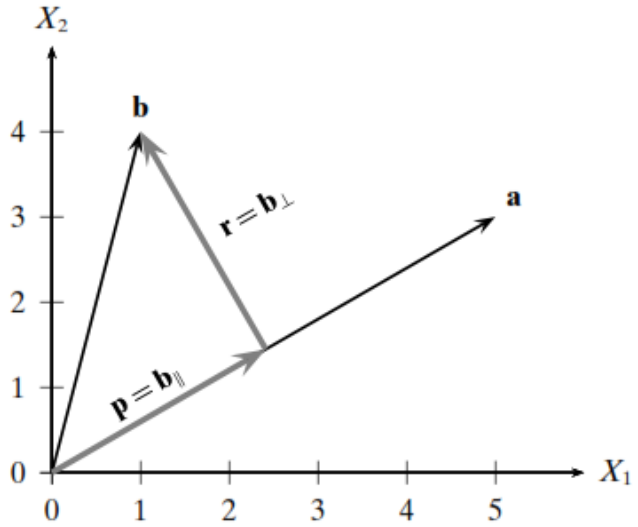
Principal Component Analysis (PCA):

- Se proiectează datele pe direcţia de variabilitate maximă




(a) Original Basis (b) Optimal Basis

Iris dataset – 3D bases [Zaki, 2014]



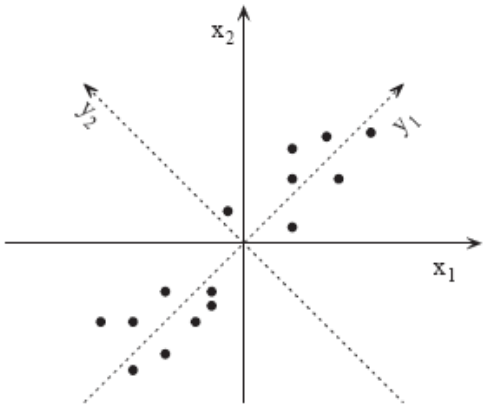

Orthogonal projection [Zaki, 2014]

PCA visualization:
http://setosa.io/ev/principal-component-analysis/

# Analiza componentelor principale

Principal Component Analysis (PCA)

Se proiectează datele pe direcţiile care captează cea mai mare variabilitate din date




Intrare: set de dat cu N instanţe având n atribute numerice (matrice de date D cu N linii şi n coloane)

Ieşire: matrice de date cu N instanţe având m<n atribute (a.î. este conservată cât mai mult din variabilitatea datelor)

Obs:

- PCA concentrează informaţia privind diferenţele dintre instanţe într-un număr mic de atribute
- PCA se utilizează pt a proiecta un set de date n-dimensional într-un spaţiu m-dimensional astfel încât atributele în noul spaţiu sunt necorelate şi este conservată cât mai mult din variabilitatea datelor

# Analiza componentelor principale

Principal Component Analysis (PCA)

Etape principale:

- Se calculează matricea de covarianţă C (matrice nxn cu elementele: C(i,j)=cov(D(i),D(j)), unde D(i) este coloana i a matricii de date D);
- Se calculează valorile proprii şi vectorii proprii ai matricii C vectorii proprii descrescătoare după valoarea proprie corespunzătoare
- Se selectează vectorii proprii care corespund celor mai mari m valori proprii
- Se proiectează setul de date D pe hiper-spaţiul definit de cei m vectori proprii

# Analiza componentelor principale

Principal Component Analysis (PCA) – elemente de statistică si algebra liniară

Matrice de covarianta $C = (c_{ij})_{i=\overline{1,n},\, j=\overline{1,n}}$

$$c_{ij} = \frac{1}{N}\sum_{k=1}^{N}(x_{ki}-\mu_i)(x_{kj}-\mu_j) = \frac{1}{N}\sum_{k=1}^{N}x_{ki}x_{kj} - \mu_i\mu_j,\ i=\overline{1,n},\ j=\overline{1,n}$$

$\mu_i$ = media valorilor atributului $i$, $c_{ii}$ = varianta atributului $i$

$C$ are $n$ vectori proprii $v_1, v_2, ..., v_n$ corespunzatori

celor $n$ valori proprii $\lambda_1, \lambda_2, ..., \lambda_n$

$Cv_i = \lambda_i v_i$ (C transforma vectorii proprii doar prin scalare)

Obs : C este simetrica si pozitiv - semidefinita

($x^T Cx \geq 0$ pt orice vector $x$) $\Rightarrow$ toate valorile proprii sunt pozitive

# Analiza componentelor principale

Principal Component Analysis (PCA) – elemente de statistică si algebra liniară

Proiecţia pe spaţiul definit de un vector (transformare în date uni-dimensionale)

- dacă v este direcţia corespunzătoare unui vector propriu atunci proiecţia lui D pe v est Dv (produsul dintre matricea cu N linii şi n coloane şi a vectorului v cu n elemente)
- Covarianţa noului set de date (e de fapt varianţa întrucât datele sunt uni-dimensionale)

$$\frac{(Dv)^T (Dv)}{N} - (\mu v)^2 = v^T C v = v^T \lambda v = \lambda \|v\|^2 = \lambda$$

$$\text{obs : vectorii proprii sunt ortonormati : } v_i^T v_j = 0,\ \|v\| = 1$$

- Varianţa proiecţiei uni-dimensionale pe un vector propriu este egală cu valoarea proprie corespunzătoare, deci pt a capta cât mai multă variabilitate trebuie aleasă cea mai mare valoare proprie

# Analiza componentelor principale

Principal Component Analysis (PCA) – elemente de statistică si algebra liniară

Proiecţia setului de date pe hiper-spaţiul definit de mai mulţi vectori proprii

- Rezultat util din algebra liniară:

$C = P\Lambda P^T$ ($C$ poate fi descompusa folosind matricea de vectori propri)

$P$ are vectorii proprii pe coloane

$P$ este matrice ortogonală : $PP^T = I_{n\times n}$

$\Lambda$ este o matrice diagonala care contine valorile proprii

- Vectorii proprii (fiind ortogonali) definesc un sistem de axe de coordonate Proiecţia setului D în noul sistem de coordonate este D'=DP
- Intrebare: care este matricea de covarianţă a noului set D' ?

# Analiza componentelor principale

Principal Component Analysis (PCA) – elemente de statistică si algebra liniară

Proiecţia setului de date pe hiper-spaţiul definit de mai mulţi vectori proprii

- Intrebare: care este matricea de covarianţă a noului set D’

$$D' = DP,\ \ X_k^{'} = P^T X_k$$

$$C' = \frac{1}{N}\sum_{k=1}^{N}(P^T X_k - P^T M)(P^T X_k - P^T M)^T =$$

$$= \frac{1}{N}\sum_{k=1}^{N} P^T (X_k - M)(X_k - M)^T P =$$

$$= P^T C P = P^T P \Lambda P^T P = \Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_n)$$

Deci din D’ sunt necorelate (matricea de covarianţă este diagonală) şi varianţa corespunzătoare atributului i este a i-a valoare proprie

# Analiza componentelor principale

Principal Component Analysis (PCA) – elemente de statistică si algebra liniară

Proiecţia setului de date pe hiper-spaţiul definit de mai mulţi vectori proprii

- Intrebare: ce se întâmplă dacă se păstrează doar m dintre componentele unui vector transformat?
- Raspuns: se conservă doar o fracţiune din variabilitatea datelor
- Ipoteza: valorile proprii sunt sortate descrescător
- Procedura: se calculează raportul varianţelor (R) şi se alege m a.î. R>prag prestabilit (e.g. R>0.95)
- Rezultat: noul set de date (cu cele m atribute obţinute prin proiecţia pe hiper-spaţiul definit de vectorii proprii corespunzători celor mai mari m valori proprii) captează cea mai mare parte din variabilitate (e.g. 95%)

$$\text{Proportia variantelor}: R = \frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{n} \lambda_i}$$

# Analiza componentelor principale

Exemplu: setul de date iris

4 atribute numerice:

A1=lungime sepale, A2=lăţime sepale, A3=lungime petale, A4=lăţime petale,

3 clase

150 instanţe

Matrice de covarianţă:

| | | | |
|---|---|---|---|
| 1 | -0.11 | 0.87 | 0.82 |
| -0.11 | 1 | -0.42 | -0.36 |
| 0.87 | -0.42 | 1 | 0.96 |
| 0.82 | -0.36 | 0.96 | 1 |

Vectori proprii (coloane)

| | | | |
|---|---|---|---|
| 0.52 | -0.37 | 0.72 | 0.26 |
| -0.26 | -0.93 | -0.24 | -0.12 |
| 0.58 | -0.02 | -0.15 | -0.80 |
| 0.57 | -0.06 | -0.63 | -0.53 |

Valori proprii: 2.91 0.92 0.14 0.02

R= (2.91+0.92)/(2.91+0.92+0.14+0.02)=0.96 (prin proiecţia datelor pe spaţiul definit de primii 2 vectori proprii se conservă 96% din variabilitatea datelor)

# Analiza componentelor principale

Exemplu: setul de date iris

4 atribute numerice:

A1=lungime sepale, A2=lăţime sepale, A3=lungime petale, A4=lăţime petale,

3 clase

150 instanţe

Matrice de covarianţă:

| | | | |
|---|---|---|---|
| 1 | -0.11 | 0.87 | 0.82 |
| -0.11 | 1 | -0.42 | -0.36 |
| 0.87 | -0.42 | 1 | 0.96 |
| 0.82 | -0.36 | 0.96 | 1 |

Vectori proprii (coloane)

| | | | |
|---|---|---|---|
| 0.52 | -0.37 | 0.72 | 0.26 |
| -0.26 | -0.93 | -0.24 | -0.12 |
| 0.58 | -0.02 | -0.15 | -0.80 |
| 0.57 | -0.06 | -0.63 | -0.53 |

Atribute noi: 0.52*A1-0.26*A2+0.58*A3+0.57*A4

-0.37*A1-0.93*A2-0.02*A3-0.06*A4

# Curs următor

Modele de clasificare:

- Concepte de bază
- Clasificatori
  - Vot simplu (ZeroR)
  - Reguli cu un atribut(OneR)
  - Clasificatori bazaţi pe instanţe (kNN)
- Măsuri de evaluare a calităţii clasificării