Folosirea PCA pentru recunoasterea imaginilor cu ajutorul librariei Accord

Motrescu Radu

Contents

1 Libraria Accord.NET

Libraria Accord.Net contine clase pentru:

- Calcul stiintific: matematica, statistica si machine learning
- Procesare de imagini si semnale: imagini, semnale audio si recunoastere si urmarire faciala in timp real
- Librarii suport pentru controale specifice: histograme, scatter-plots, controale pentru fiecare clasa de procesare imagini si semnale

In contextul cerintelor pe care vrem noi sa le indeplinim, vom folosi clase dedicate matematicii, statisticii, si ale unor controale de afisare a datelor.

2 Ce este PCA?

PCA, sau Principal Component Analysis, sau pe romaneste, Analiza Componentelor Principale este o unealta matematica aplicata din algebra liniara.

Este o metoda non-parametrica (care nu depinde de statistici) de a extrage informatiile relevante dintr-un set de date complex sau confuz.

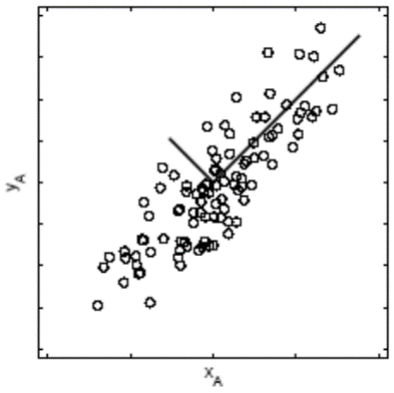
2.0.0.1 Un scurt exemplu: Sa presupunem ca am extras informatii pentru 100 de parametri pentru un student: inaltime, varsta, greutate, nota obtinuta la un test, culoarea parului etc. Vrem sa gasim cele mai importante caracteristici care definesc studentul. Cum facem asta? Folosim PCA pentru a selecta numai cele mai importante caracteristici.

- Ce parametri dorim sa indepartam:
  - Constanti: numarul de capete, care este 1 pentru toti studentii
  - Aproape constanti: grosimea firului de par: 0.003, 0.002, 0.0005 etc.
  - Care depind de alti parametri
- Ce parametri dorim sa pastram:
  - Care nu depind de alti parametri: culoarea ochilor
  - Care se schimba mult, au variatie mare: notele

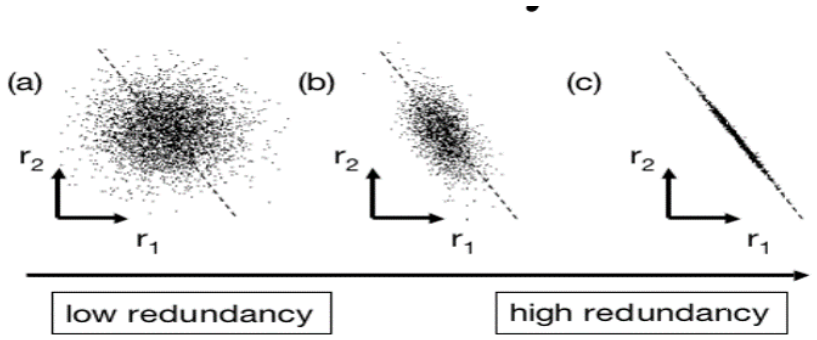
A putea elimina parametrii irelevanti si a-i pastra pe cei relevanti este usor pentru un om, el putand vedea clar care parametri nu exprima informatii relevante despre subiectul respectiv, dar cum putem face calculatorul sa isi dea seama de acesti parametri? Folosind matematica, desigur!

Dorim sa minimizam "sunetul de fundal" si redundanta datelor si sa maximizam variatia dintre parametri.

Se poate vedea in imaginea de mai jos maximizarea variatiei dintre axele norului de puncte respectiv.



In imaginile de mai jos se pot vedea inregistrarile unor informatii. In imaginea (a) si (b) se poate vedea cum informatiile nu sunt corelate, avand redunanta mica spre medie (ex: inaltimea unui student si media lui), iar in imaginea (c) se poate vedea o redundanta mare, insemnand ca ambii parametrii pot fi exprimati unul in functie de celalalt.



2.0.0.2 Variatia Este un mijloc de realizare a variabilitatii datelor dintr-un set de date cu media $\overline{X}$:

$$\sigma^2 = \frac{\sum_{i=1}^{n} \left(X_i - \overline{X}\right)^2}{n-1} \tag{1}$$

2.0.0.3 Covariatia Reprezinta variabilitatea fiecarei dimensiuni in relatie cu celelalte, si este masurata intre 2 dimensiuni pentru a se putea observa relatia dintre cele 2, spre exemplu numarul de ore studiate si nota obtinuta la examen.

$$\text{var}\left(X\right) = \frac{\sum_{i=1}^{n} \left(X_i - \overline{X}\right)\left(X_i - \overline{X}\right)}{n-1} \tag{2}$$

$$\text{cov}\left(X, Y\right) = \frac{\sum_{i=1}^{n} \left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{n-1} \tag{3}$$

2.0.0.4 Matricea de covariatie Consideram setul de date din care extragem valoarea medie (zero-mean data) si avand setul de vectori $\{x_1, x_2, ..., x_m\}$ care reprezinta liniile unei matrici $X_{m,n}$.

Fiecare linie a matricii reprezinta toate masuratorile unui anumit parametru, iar fiecare coloana reprezinta masuratorile care s-au intamplat la un moment dat.

De aici ajungem la definitia matricei de covariatie:

$$S_x \equiv \frac{1}{n-1} X X^T \text{ where } X = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} \tag{4}$$

S_x este o matrice simetrica $m \times m$, termenii de pe diagonala reprezentand variatia din acelasi parametru, iar termenii care nu sunt de pe diagonala reprezinta covariatia dintre parametri diferiti. Calculand S_x, cuantificam corelatia dintre toate posibilele perechi de masuratori. Observand elementele din matrice, o cavariatie mare reprezinta un caz de redundanta mare, iar o covariatie egala cu 0 reprezinta date complet necorelate.

$$C = \begin{bmatrix} cov(X,X) & cov(X,Y) & cov(X,Z) \\ cov(Y,X) & cov(Y,Y) & cov(Y,Z) \\ cov(Z,X) & cov(Z,Y) & cov(Z,Z) \end{bmatrix} \tag{5}$$

2.0.0.5 Valorile si vectorii proprii Urmatorul pas in calcularea PCA este aflarea valorilor si a vectorilor proprii ale matricii de covariatie. Extragand aceste informatii, ele ne vor arata componentele principale ale setului de date: vectorul propriu cu cea mai mare valoare proprie este componenta principala a setului de date. Se obisnuieste sortarea vectorilor proprii in functie de valoarea proprie pentru a determina ordinea de semnificativitate.

Vectorii si valorile proprii reies din probleme de urmatoarea forma:

$$A.v = \lambda.v \tag{6}$$

A: matrice $m \times m$
v: vector $m \times 1$ nenul
λ: constanta

Pentru orice valoare a lui λ pentru care ecuatia are solutie se numeste valoarea proprie a lui A, si vectorul **v** care corespunde acestei valori se numeste vectorul propriu a lui A.

2.0.0.6 Pasul final PCA este sa aflam valorile finale ale setului de date. Aici avem optiunea sa ignoram o parte din dimensiunile pe care le avem, deoarece ele pot fi nesemnificative, avand valori proprii mici, sau dorim afisarea lor in 1, 2 sau 3 dimensiuni. Dupa stabilirea vectorilor proprii doriti, aflarea datelor finale este simpla, proiectam punctele in spatiul lor:

FinalData = RowFeatureVector $\times$ RowZeroMeanData

RowFeatureVectore este matricea cu vectorii proprii transpusi, cu cel mai semnificati vector propriu pe prima linie.

RowZeroMeanData este matricea care contine setul de date initial din care s-a scazut valorea medie.

Matricea rezultata **FinalData** va contine setul de date dupa aplicarea algoritmului PCA.

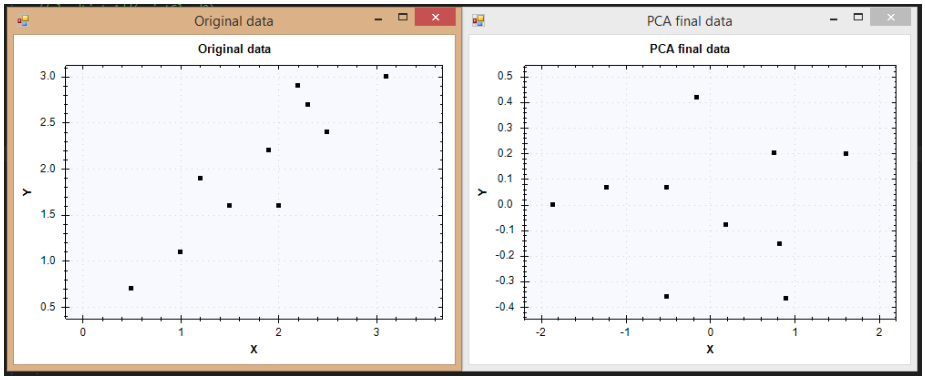
3 Jurnalul dezvoltarii aplicatiei

3.1 Aplicatia AccordPCA

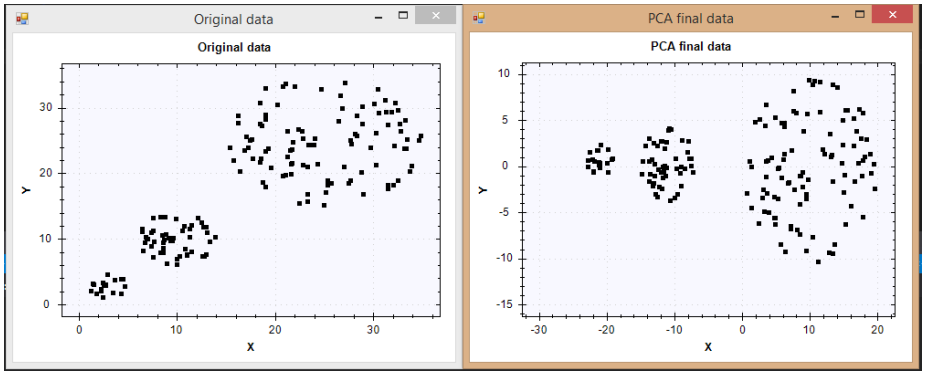
Primii pasi pe care i-am facut in cadrul acestui proiect a fost implementarea algoritmului PCA. Acest lucru a fost realizat usor, folosind libraria Accord, mai specific, pentru a realiza diferitele operatii cu matrici.

```
public void Compute()
{
    mean = initialData.Mean(0);

    dataAdjusted = initialData.Subtract(mean, 0);
    covarianceMatrix = dataAdjusted.Covariance();

    evd = new EigenvalueDecomposition(covarianceMatrix);

    eigenvalues = evd.RealEigenvalues;
    eigenvectors = evd.Eigenvectors;

    eigenvectors = Matrix.Sort(eigenvalues, eigenvectors, new GeneralComparer(ComparerDirection.Descending, true));

    finalData = dataAdjusted.Dot(eigenvectors);

    // data2D = finalData.GetColumns(0, 1);

}
```

Pentru a testa acest algoritm, am introdus prima data un set de date deja testat de catre altcineva, unde se poate observa clar rezultatul algoritmului. (referinta catre http://dai.fmph.uniba.sk/courses/ml/sl/PCA.pdf)



Urmatorul lucru pe care l-am dorit sa il facem a fost sa vedem cum mai multe noruri de puncte, de diferite dimensiuni, se modifica dupa aplicarea algoritmului. Pentru acest lucru, am creat o clasa **PointCloud** care descrie un nor de puncte. In interiorul clasei am generat norul de puncte in jurul unor coordonate (x, y), sub forma unui disc, in care punctele au distributie normala. Rezultatele au fost urmatoarele:



La fel ca in primul test, se poate observa aranjarea dupa prima componenta principala a norilor.

3.2 Aplicatia PlotPCA

In acest moment, am decis sa realizam o aplicatie care are un GUI usor de folosit si unde fiecare nor de puncte este reprezentat in alta culoare, pentru a se vedea mai clar ce se intampla, acest lucru fiind mai important pentru seturile de date in care masuratorile pentru fiecare entitate sunt intrepatrunse, spre exemplu, setul IRIS.