6 Analiza Componentelor Principale

6.1 Obiective

În această lucrare de laborator se descrie metoda de Analiză a Componentelor Principale (*Principal Component Analysis* – PCA). Această metodă se utilizează pentru reducerea dimensionalității, compresia și vizualizarea datelor. Pentru realizarea acestei lucrări de laborator este necesară o bibliotecă care să calculeze valorile și vectorii proprii ale unei matrice (descompunerea în valori proprii).

6.2 Fundamente teoretice

Se consideră un set de puncte de date într-un spațiu de dimensiune mare (ND). Fiecare vector reprezintă trăsăturile unui exemplu de antrenare. Scopul acestei metode este reducerea dimensionalității punctelor la o dimensiune mai mică KD în așa fel încât să se păstreze cât mai multă informație cu putință. Ideea PCA este de a găsi principalele K axe de variație, astfel încât după proiecția datelor într-un spațiu mai redus, varianța datelor proiectate sa fie maximizată.

Inițial vom considera un exemplu bidimensional: se afișează datele colectate despre cât de mult apreciază anumite persoane niște activități și aptitudinea lor în domeniul respectiv. Figura 6.1 ilustrează un exemplu simplificat [2].

Să analizăm cei doi vectori $\boldsymbol{u}_1$ și $\boldsymbol{u}_2$. Dacă se proiectează punctele 2D pe vectorul $\boldsymbol{u}_2$ se obțin valori scalare cu o împrăștiere redusă (deviație standard mică). În schimb, dacă se proiectează punctele pe $\boldsymbol{u}_1$ punctele sunt mult mai împrăștiate. Dacă ar trebui să reducem datele la o singură dimensiune, atunci ar fi preferabil să le proiectăm pe $\boldsymbol{u}_1$ întrucât datele sunt mai ușor separabile iar varianța este mai mare.

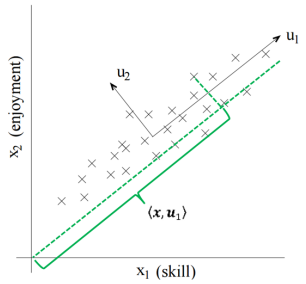

Figura 6.1 – Proiecţia punctului x pe vectorul u_1

Exprimat într-un mod mai formal, fiecare punct bidimensional poate fi scris ca:

$$\boldsymbol{x} = \langle \boldsymbol{x}, \boldsymbol{u}_1 \rangle \, \boldsymbol{u}_1 / \|\boldsymbol{u}_1\| + \langle \boldsymbol{x}, \boldsymbol{u}_2 \rangle \, \boldsymbol{u}_2 / \|\boldsymbol{u}_2\|$$

În ecuația de mai sus punctul $\boldsymbol{x}$ a fost proiectat pe fiecare vector și apoi rezultatele obținute au fost însumate. Produsul scalar $\langle \boldsymbol{x}, \boldsymbol{u}_i \rangle$ definește magnitudinea proiecției și trebuie normalizat cu norma vectorului $\|\boldsymbol{u}_i\|$; cei doi vectori definesc direcțiile. Această exprimare este posibilă deoarece $\boldsymbol{u}_1$ și $\boldsymbol{u}_2$ sunt vectori perpendiculari. Dacă se pune condiția ca cei doi vectori să fie vectori unitate, atunci termenul de normalizare dispare. Pentru mai multe exemple de proiecţie vedeți [4].

Ideea principală a reducerii dimensionalității datelor este să se utilizeze cele mai mari proiecții. Întrucât proiecțiile pe $\boldsymbol{u}_2$ vor fi mai mici, $\boldsymbol{x}$ se poate aproxima folosind doar primul termen:

$$\tilde{\boldsymbol{x}}_1 = \langle \boldsymbol{x}, \boldsymbol{u}_1 \rangle \, \boldsymbol{u}_1 / \|\boldsymbol{u}_1\|$$

În general, fiind dată o bază ortonormală a unui spațiu vectorial cu *d* dimensiuni $\boldsymbol{B}$ cu vectorii de bază $\boldsymbol{b}_i$, orice vector se poate scrie ca:

$$\boldsymbol{x} = \sum_{i=1}^{d} \langle \boldsymbol{x}, \boldsymbol{b}_i \rangle \boldsymbol{b}_i = \sum_{i=1}^{d} (\boldsymbol{x}^T \boldsymbol{b}_i)\, \boldsymbol{b}_i$$

Problema revine acum să determinăm vectorii de bază pe care se vor realiza proiecțiile. Întrucât scopul principal este maximizarea varianței dintre punctele rezultate în urma transformării, matricea de covarianță ne poate oferi informațiile necesare. Covarianța dintre două trăsături este definită ca:

$$C(x_i, x_j) = \frac{1}{n-1} \sum_{k=1}^{n} (X_{ki} - \mu_i)(X_{kj} - \mu_j)$$

unde μ_i este media trăsăturii *i*. Matricea de covarianță stochează covariațele pentru toate perechile de trăsături. Se poate demonstra că matricea de covarianță poate fi exprimată ca un simplu produs de matrice:

$$C = \frac{1}{n-1} (X - \boldsymbol{\mu} \mathbf{1}_{1xn})^T (X - \boldsymbol{\mu} \mathbf{1}_{1xn})$$

unde $\boldsymbol{\mu}$ este un vector care conține valorile medii ale trăsăturilor și $\mathbf{1}_{1xn}$ este un vector rând ce conține doar valori de 1. Dacă eliminăm media din datele de intrare, într-un pas de preprocesare, ecuația se simplifică și mai mult:

$$C = \frac{1}{n-1} X^T X$$

Pasul următor este găsirea axelor de-a lungul cărora covarianța este maximă. Descompunerea în valori proprii ale unei matrice ne furnizează aceste informații. Astfel, vectorul propriu corespunzător celei mai mare valori proprii va reprezenta prima axă principală și așa mai departe.

Intuitiv, (aproape) orice matrice poate fi vizualizată ca o rotație urmată de o scalare de-a lungul axelor și rotația inversă. Descompunerea în vectori și valorilor proprii calculează această descompunere a matricei de covarianță:

$$C = Q \Lambda Q^T = \sum_{i=1}^{d} \lambda_i Q_i Q_i^T$$

unde *Q* este o matrice de rotație de dimensiune *dxd* (ortonormală) și *Λ* este o matrice diagonală ale cărei elemente reprezintă scalarea de-a lungul fiecărei axe. Elementele se numesc valori proprii și fiecare coloană corespunzătoare din *Q* este vectorul propriu corespunzător. Deoarece scopul principal este menținerea proiecțiilor cu varianță maximă, valorile proprii se ordonează descrescător în funcție de magnitudinea lor și se aleg primele *k* valori proprii. Astfel *C* poate fi aproximat ca:

$$\tilde{C}_k = Q_{1:k} \Lambda_{1:k} Q_{1:k}^T = \sum_{i=1}^{k} \lambda_i Q_i Q_i^T$$

unde $Q_{1:k}$ este o matrice de dimensiune *dxk* cu primii *k* vectori proprii și $\Lambda_{1:k}$ este o matrice diagonală de dimensiune *kxk* ce conține primele *k* valori proprii. Dacă *k* este egal cu *d* se obține matricea originală și, pe măsură ce valoarea lui *k* scade, se obțin aproximări tot mai grosiere ale lui *C*.

Astfel am determinat axele de-a lungul cărora varianța proiecțiilor este maximizată. În cazul general un vector poate fi aproximat cu *k* vectori astfel:

$$\widetilde{\boldsymbol{x}}_{\boldsymbol{k}} = \sum_{i=1}^{k} \langle \boldsymbol{x}, Q_i \rangle Q_i = \sum_{i=1}^{k} (\boldsymbol{x}^T Q_i)\, Q_i$$

unde Q_i este coloana *i* a matricei de rotație *Q*.

Coeficienţii PCA pot fi calculaţi ca:

$$X_{coef} = XQ$$

Proiecția PCA (de la *d* la *k* dimensiuni, $k < d$) poate fi calculată ca:

$$X_k = XQ_{1:k}$$

unde $Q_{1:k}$ este matricea formată din primele *k* coloane din *Q*.

Aproximarea PCA (reconstrucția PCA de la *k* la *d* dimensiuni, $k < d$**)** poate fi calculată pentru toți vectorii de intrare simultan (dacă ei sunt stocați ca rânduri în *X*) utilizând formula:

$$\widetilde{\boldsymbol{X}}_{\boldsymbol{k}} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{k}} \boldsymbol{X Q_i Q_i^t} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{k}} \boldsymbol{X_{coef_i} Q_i^t} = \boldsymbol{X Q_{1:k} Q_{1:k}^T}$$

unde $Q_{1:k}$ este matricea formată din primele *k* coloane din *Q*. Este important să se facă distincția între aproximare și coeficienți: aproximarea este suma coeficienților înmulțite cu componentele principale.

În finalul acestei prezentări teoretice vom trece în revistă mai multe exemple în care PCA se poate aplica cu succes:

- Reducerea dimensionalității trăsăturilor: în unele cazuri, vectori de trăsături cu o dimensionalitate mare pot să încetinească procesul de predicție
- Vizualizarea datelor – datele pot fi analizate în 3D sau în 2D; pentru date cu o dimensionalitate mai mare este necesară proiecția datelor;
- Aproximarea vectorilor de date;
- Detecția trăsăturilor redundante și a dependențelor liniare dintre trăsături;
- Reducerea zgomotului – dacă zgomotul din date are o varianță mai mică decât datele, adică raportul dintre semnal și zgomot (SNR) este mare, atunci PCA elimină zgomotul din datele de intrare.

6.3 Detalii de implementare

Declararea și alocarea unei matrice de dimensiune *n* x *d* cu valori flotante exprimate în dublă precizie:

```
Mat X(n,d,CV_64FC1);
```

Calculul matricei de covarianță după ce mediile au fost scăzute din valorile de intrare:

```
Mat C = X.t()*X/(n-1);
```

Pentru a calcula descompunerea în valori proprii, *Lambda* va conține valorile proprii și *Q* va conține vectorii proprii. Este necesară transpunerea deoarece *X* conține datele de intrare de-a lungul rândurilor.

```
Mat Lambda, Q;
eigen(C, Lambda, Q);
Q = Q.t();
```

Produsul scalar este implementat ca o simplă înmulțire. Atenție, datorită faptului că indexarea începe de la 0, primul rând este *row*(0). Produsul scalar dintre rândul *i* din X și coloana *i* din *Q* este dat de:

```
Mat prod = X.row(i)*Q.col(i)
```

6.4 Activitate practică

1. Deschideți fișierul de intrare și citiți punctele de date. Pe prima linie este stocat numărul de puncte *n* și dimensionalitatea datelor de intrare *d.* Liniile următoare din fișier conțin câte un punct cu *d* coordonate. Calculați vectorul cu valorile medii și scădeți-l din punctele de intrare.
2. Calculați matricea de covarianță ca un produs de matrice.
3. Efectuați descompunerea în valori proprii a matricei de covarianță apelând funcţia din bibliotecă.
4. Afișați valorile proprii.
5. Calculați coeficienții PCA și aproximarea $\tilde{X}_k$ de ordinul *k* (folosind primele *k* valori proprii) pentru datele de intrare.
6. Calculați valoarea medie a diferenței absolute dintre punctele originale și aproximarea lor utilizând primele *k* componente principale.
7. Găsiți minimele și maximele pe coloanele matricei de coeficienți.
8. Pentru datele de intrare din fișierul *pca2d.txt*, afișați coeficienţii PCA din primele două coloane ca puncte negre 2D pe fundal alb. Pentru a obține coordonate pozitive scădeți valorile minime.
9. Pentru datele de intrare din fișierul *pca3d.txt*, afişaţi coeficienţii PCA din primele trei coloane sub forma unei imagini cu niveluri de gri. Utilizați primele 2 componente ca și coordonatele *x* și *y*, iar cea de-a treia valoare ca intensitate în punctul (*x*, *y*). Pentru a obține coordonate pozitive trebuie să scădeți valoarea minimă din primele două coordonate. Normalizați a treia componentă astfel încât să ia valori între 0:255.
10. Determinați automat numărul de componente principale *k* care trebuie să fie păstrate astfel încât să se reţină un anumit procent din varianța inițială. De exemplu, găsiți valoarea lui *k* pentru care aproximarea de ordinul *k* reține 99% din varianța inițială. Procentajul variației păstrate este dat de $\sum_{i=1}^{k} \lambda_i / \sum_{i=1}^{d} \lambda_i$.

6.5 Exemple de rezultate

Pentru *pca2d*

- Prima valoare proprie este aproximativ 8090.21
- Eroarea medie absolută folosind o singură componentă: 22.43

Pentru *pca3d*

- Prima valoare proprie este aproximativ 5462.33
- Eroarea medie absolută folosind o singură componentă: 14.50

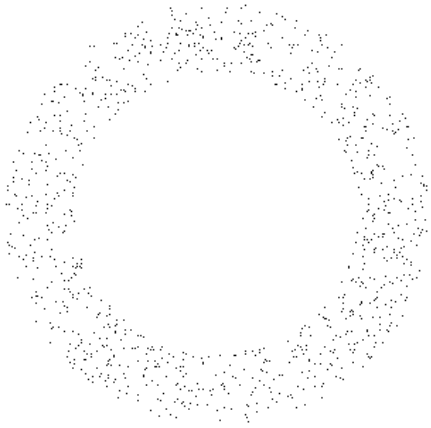

Figura 6.2. Vizualizarea punctelor ce rezultă după aplicarea metodei PCA pe datele din fişierul *pca2d.txt*

6.6 Referințe

[1] Wikipedia article PCA - https://en.wikipedia.org/wiki/Principal_component_analysis
[2] Stanford CS229 Machine Learning course notes - https://cs229.stanford.edu/main_notes.pdf
[3] Lindsay Smith - PCA tutorial - http://faculty.iiit.ac.in/~mkrishna/PrincipalComponents.pdf
[4] PCA in R (animation of projection) - https://poissonisfish.wordpress.com/2017/01/23/principal-component-analysis-in-r/