11MAMY – Cvičení 11

Učení bez učitele

Jan Přikryl

ČVUT FD

středa 22. března 2022

Příklad 1

Analýza hlavních komponent

V této úloze vyzkoušíme PCA na datovém souboru `USArrests`. Řádky datové sady obsahují údaje o zločinnosti v 50 státech USA, seřazené abecedně podle státu. Její atributy jsou

- `Murder`: počet vražd na 100 000 obyvatel ročně
- `Assault`: počet napadení na 100 000 obyvatel ročně
- `UrbanPop`: procento populace státu ve městech
- `Rape`: počet znásilnění na 100 000 obyvatel ročně

Napřed jako vždy načteme CSV data do tabulky. První sloupec obsahuje názvy statů USA, proto `'ReadRowNames'`:

```
usatbl = readtable('islr_usaarrests.csv','ReadRowNames',true);
```

Sloupce datové sady obsahují čtyři proměnné:

```
usatbl.Properties.VariableNames
usatbl
```

Příklad 1

Analýza hlavních komponent

Nejprve stručně prozkoumáme data. Uvidíme, že proměnné mají obrovsky rozdílné průměry i rozptyly:

```
h = figure(1);
boxplot(usatbl.Variables, 'Orientation', 'horizontal', ...
        'Labels', usatbl.Properties.VariableNames);
```

Průměrné hodnoty atributů jsou

```
mean(usatbl.Variables)
```

Můžeme také prozkoumat razdílné rozptyly všech čtyř atributů:

```
var(usatbl.Variables)
```

Není divu, že proměnné mají tak odlišné charakteristiky: Proměnná `UrbanPop` není srovnatelná s počty zločinů na 100 000 obyvatel.

Příklad 1

Analýza hlavních komponent

Pokud se nám nepodaří sladit měřítko proměnných před provedením PCA, pak většina hlavních komponent bude odrážet proměnnou `Assault`, protože ta má zdaleka největší průměr a rozptyl.
Potřebujeme navíc tabulku čistě číselných hodnot, protože PCA vyžaduje na vstupu matici dat a ne tabulku.

```
zdata = zscore(usatbl.Variables); % spočte z-skóre, N(0,1)
mean(zdata) % měl by být okolo 0
var(zdata) % měla by být okolo 1
```

A teď konečně můžeme udelat PCA na matici o n řádcích a p sloupcích:

```
% Vlnovka ignoruje výstup
[coeffs,scores,latent,~,varexpl,muest] = pca(zdata);
```

Příklad 1

Analýza hlavních komponent

V proměnné `coeffs` jsou uvedené *zátěže* hlavních komponent. V každém sloupci je uložen jeden vektor zátěže:

```
coeffs
```

Vidíme, že existují čtyři odlišné hlavní komponenty. To bychom měli očekávat, protože v sadě dat s n pozorováními a p proměnnými obecně existuje $\min(n-1, p)$ hlavních komponent.
Skóre hlavních komponent již máme připraveno v matici `scores`

```
size(scores)
```

kde k-tý sloupec je vektor skóre k-té hlavní složky.

Příklad 1

Analýza hlavních komponent – biplot

Zobrazíme biplot prvních dvou komponent:

```
h = figure(2);
biplot(coeffs(:,[1,2]), 'Scores', scores(:,[1,2]), ...
       'VarLabels', usatbl.Properties.VariableNames);
xlim([-1,1]);
ylim([-1,1]);
axis equal;
```

Příklad 1

Analýza hlavních komponent

Standardni odchylka všech PCA komponent

```
sqrt(latent)
```

se spočte z jejího rozptylu, zámého z `pca()`

```
latent
```

Vysvětlená variance je zde rovnou v procentech:

```
varexpl
```

Příklad 1

Analýza hlavních komponent

Vykresleni variance

```
figure(3);
plot(varexpl,'-o');
xticks([1 2 3 4]);
xlim([0.9 4.1]);
xlabel('Pořadí␣hlavních␣komponent');
ylabel('Vysvětlený␣rozptyl␣[%]')
```

Příklad 1

Analýza hlavních komponent

Na závěr graf kumulativní vysvětlené variance

```
figure(4);
plot(cumsum(varexpl),'-o');
xticks([1 2 3 4]);
xlim([0.9 4.1]);
ylim([0 105]);
xlabel('Pořadí␣hlavních␣komponent');
ylabel('Kumulativní␣vysvětlený␣rozptyl␣[%]')
```

Odpovězte:

Kolik rozpylu vysvětlují první dvě hlavní komponenty?

Příklad 2 – Shlukování přes k průměrů (k-means)

Vyzkoušejme si nyní shlukování na synteticky generovaných datech: Napřed musíme napevno nastavit násadu generátoru náhodných čísel, aby byly výsledky pro všechny pokaždé stejné.

Vygenerujeme 50 záznamů o dvou prediktorech ve dvou shlucích:

```
rng(42); % Pevná násada generátoru náhodných čísel
ndat = 50; % Zapamatujeme si, kolik je dat
% Kde budou středy syntetických skluků?
c_center_1 = [1, -2];
c_center_2 = [-1, 2];
% Polovina dat kvůli dělení na dva shluky
ndat_h = floor(ndat/2);
```

Příklad 2

k-means – pokračování

Generujeme 2D data z $\mathcal{N}(\mathbf{o}, \mathbf{I})$: normální rozdělení ve 2D, střední hodnota [0,0] a kovarianční matice jednotková:

```
data = randn(ndat,2);
```

Zobrazíme je jako mrak bodů

```
figure(1);
scatter(data(:,1),data(:,2));
title('Původní data nerozdělená na shluky');
```

Data upravíme na dva shluky přičtením souřadnic středů:

```
data(1:ndat_h,:) = data(1:ndat_h,:) + c_center_1;
data(ndat_h+1:end,:) = data(ndat_h+1:end,:) + c_center_2;
```

Příklad 2

k-means – pokračování

Vyrobíme vektor příšlušnosti, sloužící i k obarvení obou shluků:

```
cn = ones(n_dat,1); % Základní příslušnost ke shluku 1
cn(ndat_h+1:end) = 2; % Druhá polovina je shluk 2
```

A zobrazíme znovu, tentokrát jako dvě skupiny barevných puntíků podle hodnot v `cn`:

```
figure(2);
scatter(data(:,1),data(:,2), 30, cn, 'filled');
title('Dva shluky');
```

Otázka

Hodnota parametru `sz` je 30, ale puntíky roznodně nejsou 30 jednotek veliké. Zjistěte, co vlastně parametr `sz` očekává za hodnoty.

Příklad 2

k-means – pokračování

Víme, že jsme vytvořili dva shluky. Najdeme je pomocí `kmeans()`?

```
[km2, centroids2, withinss2] = kmeans(data, 2);
```

Vykreslíme vše obarvené podle příslušnosti ke shlukům, včetně středů shluků

```
figure(3);
s = scatter(data(:,1),data(:,2), 30, km2, 'filled');
hold on;
plot(centroids2(:,1),centroids2(:,2),'r*');
hold off;
title('Obarvené dva shluky');
```

Celkový součet rozptylu mezi body shluků

```
tot2 = sum(withinss2);
fprintf('sum(withinss2)=%f\n', tot2);
```

Příklad 2

k-means – pokračování

Co když jsou shluky 3?

```
[km3, centroids3, withinss3] = kmeans(data, 3);
```

Vykreslíme

```
figure(4);
scatter(data(:,1),data(:,2), 30, km3, 'filled');
hold on;
plot(centroids3(:,1),centroids3(:,2),'r*');
title('Obarvené␣tři␣shluky');
```

Celkový součet rozptylu mezi body shluků

```
tot3 = sum(withinss3);
fprintf('sum(withinss3)=%f\n', tot3);
```

Příklad 2

k-means – pokračování

Identifikujeme 3 optimální shluky po 50 restartech

```
[km3b, centroids3b, withinss3b] = kmeans(data, 3, 'Replicates', 50);
```

Vykreslíme

```
figure(5);
scatter(data(:,1),data(:,2), 30, km3b, 'filled');
hold on;
plot(centroids3b(:,1),centroids3b(:,2),'r*');
hold off
title('Obarvené tři optimální shluky po 50 restartech');
```

Celkový součet rozptylu mezi body shluků

```
tot3r = sum(withinss3b);
fprintf('sum(withinss3r)=%f\n', tot3r);
```

Příklad 2

k-means – pokračování

Identifikujeme 3 optimální shluky po 50 restartech při použití cityblock normy místo euklidovské vzdálenosti

```
[km3c, centroids3c, withinss3c] = ...
       kmeans(data, 3, 'Replicates', 50, 'Distance', 'cityblock');
```

Vykreslíme

```
figure(6);
scatter(data(:,1),data(:,2), 30, km3c, 'filled');
hold on;
plot(centroids3c(:,1),centroids3c(:,2),'r*');
hold off;
title('Obarvené␣tři␣optimální␣shluky,␣50␣restartů,␣cityblock');
tot3c = sum(withinss3c);
fprintf('sum(withinss3c)=%f\n', tot3c);
```

Příklad 3

Hierarchické shlukování

Přesuneme pozornost na hierarchické shlukování a to na totožných syntetických datech

V Matlabu používáme na hierachické shlukování postupně funkce

- `pdist()` ... vzdálenost (implicitně Euklidovská) mezi všemi páry shlukovaných záznamů
- `linkage()` ... tvoří binární vazby mezi shluky na základě vzdálenosti (implicitně té nejkratší) záznamů, zjištěných `pdist()`
- `cluster()` ... z datové struktury vazeb generuje přiřazení ke shlukům
- `dendrogram()` ... vykresluje dendrogram odpovídající zjištěným vazbám

Alternativně máme k dispozici metodu `clusterdata()`, jež zajistí provedení většiny výše uvedených úkonů.

Příklad 3

Hierarchické shlukování – pokračování

Napřed určíme Euklidovskou vzdálenost mezi všemi páry bodů

```
dist = pdist(data, 'euclid');
```

Najdeme hierarchické shluky minimalizací maximální vzdálenosti mezi prvky shluku

```
lnk_complete = linkage(dist, 'complete');
```

Vykreslíme pomocí `lnk_complete`

```
figure(3);
dendrogram(lnk_complete);
```

Příklad 3

Hierarchické shlukování

Najdeme hierarchické shluky minimalizací průměrné vzdálenosti mezi prvky shluku

```
lnk_average = linkage(dist, 'average');
```

Vykreslíme pomocí `lnk_average` s omezením na `nc` shluků:

```
nc = floor(n_dat/4); % Čtvrtina dat, pro n_dat=50 je to 12
% A vlastní obrázek
figure(4);
dendrogram(lnk_average, nc);
```

Nyní hierarchicky shlukneme na maximální počet `nc` shluků, shluky by měly odpovídat listům výše zobrazeného dendrogramu.

```
cluster_complete = cluster(lnk_average, 'maxclust', nc);
```

Příklad 3

Hierarchické shlukování

Nyní označíme 2 shluky v datech

```
cluster2 = cluster(lnk_average, 'maxclust', 2);
figure(5);
scatter(data(:,1),data(:,2), 30, cluster2, 'filled');
title('Hierarchické shlukování, 2 shluky, average');
```

Nyní označíme 3 shluky v datech

```
cluster3 = cluster(lnk_average, 'maxclust', 3);
figure(6);
scatter(data(:,1),data(:,2), 30, cluster3, 'filled');
title('Hierarchické shlukování,3 shluky, average');
```

Příklad 3

Hierarchické shlukování

Nyní označíme 4 shluky v datech

```
cluster4 = cluster(lnk_average, 'maxclust', 4);
figure(7);
scatter(data(:,1),data(:,2), 30, cluster4, 'filled');
title('Hierarchické shlukování, 4 shluky, average');
```

Nyní označíme 7 shluků v datech

```
cluster7 = cluster(lnk_average, 'maxclust', 7);
figure(8);
scatter(data(:,1),data(:,2), 30, cluster7, 'filled');
title('Hierarchické shlukování, 7 shluků, average');
```