Lecture 12
Principal Components Analysis

Breitzman 8/12/2018

PCA

- Principal Components Analysis (PCA) is a dimension reduction method
- We’ve seen several cases, where it would be nice to reduce the number of variables (Neural Networks in Particular, but also decision trees, Naïve Bayes etc. benefit from dimension reduction)
- For those of you that know some linear algebra, the idea is that we take our original N variables and create a new set of variables that are independent (orthogonal) and a linear combination of the original variables
- The principal components actually form an orthonormal basis if you remember what that means. (If you don’t, it doesn’t matter)

Z-Score Normalization

- Recall from week 1, z-score normalization
- Normalizes an attribute so that the mean is 0 and standard deviation is 1
- For a vector $X=\langle x_1, x_2, x_3, \ldots, x_n \rangle$, $z_i=(x_i-\mathrm{mean}(X))/\mathrm{sd}(X)$
- PCA only works if all attributes are normalized this way

Houses Data

- We will use a data set from http://lib.stat.cmu.edu/datasets/
- Several interesting data sets can be found here
- The one we are using houses.zip contains block groups from 1990 California Census

Houses Data – First 15 Rows (20,640 total)

Median HomeVal	Median Income	Median HomeAge	Total Rooms	Total Bed Rooms	Populat ion	House holds	Lati tude	Longi tude
452600	8.3252	41	880	129	322	126	37.9	-122
358500	8.3014	21	7099	1106	2401	1138	37.9	-122
352100	7.2574	52	1467	190	496	177	37.9	-122
341300	5.6431	52	1274	235	558	219	37.9	-122
342200	3.8462	52	1627	280	565	259	37.9	-122
269700	4.0368	52	919	213	413	193	37.9	-122
299200	3.6591	52	2535	489	1094	514	37.8	-122
241400	3.12	52	3104	687	1157	647	37.8	-122
226700	2.0804	42	2555	665	1206	595	37.8	-122
261100	3.6912	52	3549	707	1551	714	37.8	-122
281500	3.2031	52	2202	434	910	402	37.9	-122
241800	3.2705	52	3503	752	1504	734	37.9	-122
213500	3.075	52	2491	474	1098	468	37.9	-122
191300	2.6736	52	696	191	345	174	37.8	-122
159200	1.9167	52	2643	626	1212	620	37.9	-122

Houses Data

- All data somewhat self-explanatory except
- Median Income seems to be already scaled
- 20,640 records from 1990 California Census

Houses Data – Same Data Normalized

zMed HomeVal	zMed Inc	zMed HomeAge	zTot Rooms	zBed rooms	zPop	zHouse holds	zLat itude	zLong itude
2.13	2.34	0.98	-0.80	-0.97	-0.97	-0.98	1.05	-1.33
1.31	2.33	-0.61	2.05	1.35	0.86	1.67	1.04	-1.32
1.26	1.78	1.86	-0.54	-0.83	-0.82	-0.84	1.04	-1.33
1.17	0.93	1.86	-0.62	-0.72	-0.77	-0.73	1.04	-1.34
1.17	-0.01	1.86	-0.46	-0.61	-0.76	-0.63	1.04	-1.34
0.54	0.09	1.86	-0.79	-0.77	-0.89	-0.80	1.04	-1.34
0.80	-0.11	1.86	-0.05	-0.12	-0.29	0.04	1.03	-1.34
0.30	-0.40	1.86	0.21	0.35	-0.24	0.39	1.03	-1.34
0.17	-0.94	1.06	-0.04	0.30	-0.19	0.25	1.03	-1.34
0.47	-0.09	1.86	0.42	0.40	0.11	0.56	1.03	-1.34
0.65	-0.35	1.86	-0.20	-0.25	-0.46	-0.26	1.04	-1.34
0.30	-0.32	1.86	0.40	0.51	0.07	0.61	1.04	-1.34
0.06	-0.42	1.86	-0.07	-0.15	-0.29	-0.08	1.04	-1.34
-0.13	-0.63	1.86	-0.89	-0.82	-0.95	-0.85	1.03	-1.34
-0.41	-1.03	1.86	0.00	0.21	-0.19	0.32	1.04	-1.34

R Needed to Create Previous Table

```
m<-mean(houses$MedianHomeVal)
sigma<-sd(houses$MedianHomeVal)
houses$zMedHomeVal<-(houses$MedianHomeVal-m)/sigma
m<-mean(houses$MedianIncome)
sigma<-sd(houses$MedianIncome)
houses$zMedInc<-(houses$MedianIncome-m)/sigma
m<-mean(houses$MedianHomeAge)
sigma<-sd(houses$MedianHomeAge)
houses$zMedHomeAge<-(houses$MedianHomeAge-m)/sigma
m<-mean(houses$TotalBedRooms)
sigma<-sd(houses$TotalBedRooms)
houses$zBedrooms<-(houses$TotalBedRooms-m)/sigma
.
.
.
```

The Key to PCA is a Covariance Matrix

- The previous table is a giant matrix called the Z matrix (20,640 rows, 8 columns)

- 8 columns include all names except zMedHomeVal which is going to be our predicted variable

- Cov(Z)= $$\begin{bmatrix} \overset{*}{\sigma}_{1,1} & \overset{*}{\sigma}_{1,2} & \cdots & \overset{*}{\sigma}_{1,m} \\ \overset{*}{\sigma}_{1,2} & \overset{*}{\sigma}_{2,2} & \cdots & \overset{*}{\sigma}_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \overset{*}{\sigma}_{1,m} & \overset{*}{\sigma}_{2,m} & \cdots & \overset{*}{\sigma}_{m,m} \end{bmatrix}$$

- Where $$\overset{*}{\sigma}_{i,j} = \frac{\sum_{k=1}^{n}(x_{k,i} - \mu_i)(x_{k,j} - \mu_j)}{n}$$

Cov() in R

- Output of cov(houses)

	zMedInc	zMed HomeAge	zBed rooms	zTot Rooms	zPop	zHouse holds	zLati tude	zLongi tude
zMedInc	1.00	-0.12	-0.01	0.20	0.00	0.01	-0.08	-0.02
zMed HomeAge	-0.12	1.00	-0.32	-0.36	-0.30	-0.30	0.01	-0.11
zBedrooms	-0.01	-0.32	1.00	0.93	0.88	0.98	-0.07	0.07
zTotRooms	0.20	-0.36	0.93	1.00	0.86	0.92	-0.04	0.04
zPop	0.00	-0.30	0.88	0.86	1.00	0.91	-0.11	0.10
zHouseholds	0.01	-0.30	0.98	0.92	0.91	1.00	-0.07	0.06
zLatitude	-0.08	0.01	-0.07	-0.04	-0.11	-0.07	1.00	-0.92
zLongitude	-0.02	-0.11	0.07	0.04	0.10	0.06	-0.92	1.00

- Note number close to 1 implies variables are colinear
- Independent variables show 0, but 0 does not guarantee independence

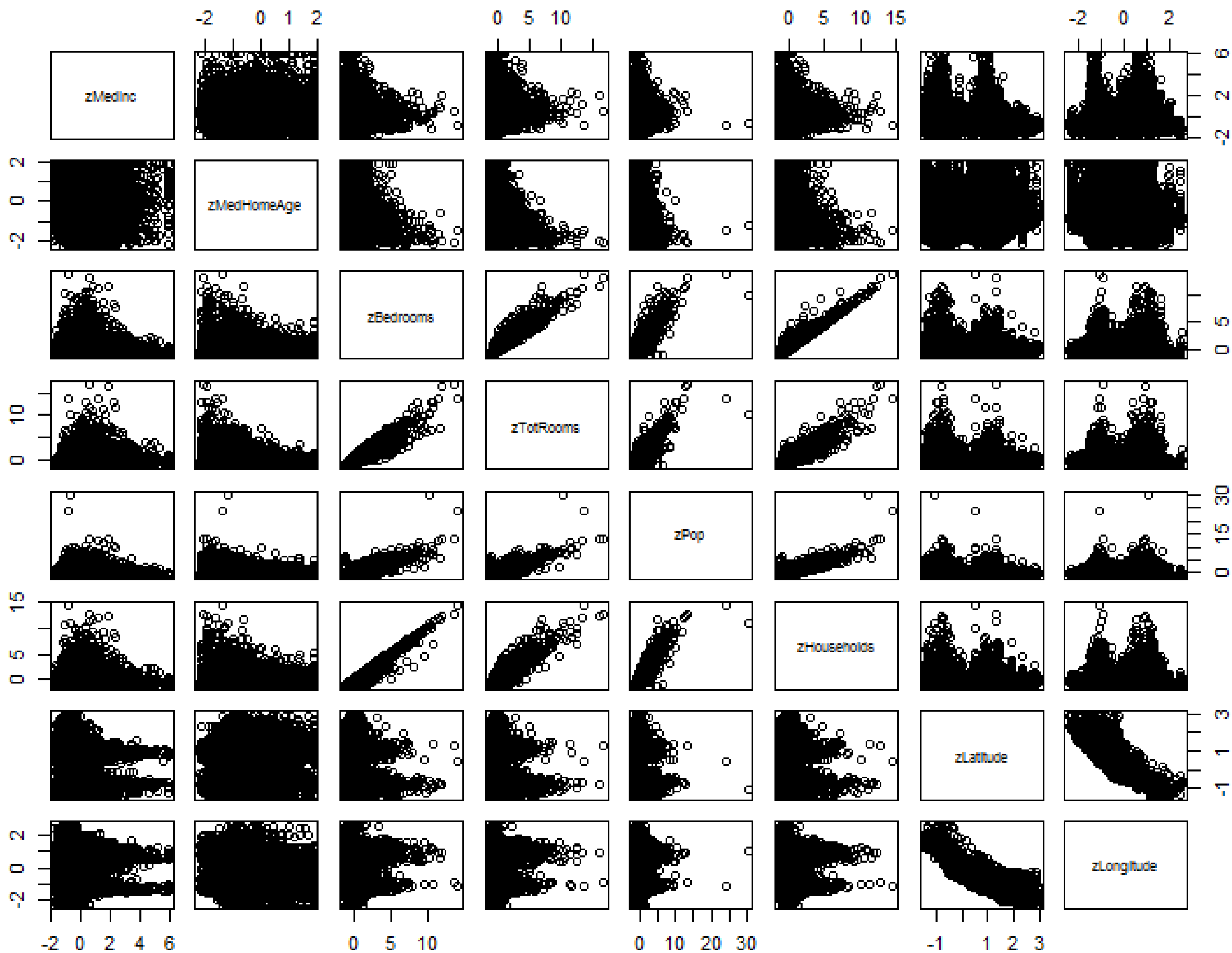
Covariance Matrix (II)

- It’s clear that Population, Households, Total Rooms, and Bedrooms are highly correlated

- Also Latitude is highly negatively correlated with longitude

- We can see this with pairs(houses) which we have used before to show a draftsman’s plot

Draftsman's Plot of Z-Scored Housing Data



The ith Principal Component is…

- $Yi=ei^TZ$ where Z is the standardized Matrix and ei^T is the transpose of the ith eigenvector of the covariance matrix
- If you haven't had linear alg, don't worry we can use R to compute eigenvectors
- If you had linear algebra then… we told you eigenvectors were important!

Eigenvalues and Eigenvectors

```
> eigen(cov(houses3))
$values
[1] 3.9066 1.907947 1.071961 0.821998 0.148054 0.081664 0.046899 0.014791

$vectors
          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]       [,8]
[1,] -0.045144 -0.035300  0.890917 -0.407818 -0.056213 -0.057214 -0.168837 -0.0415559
[2,]  0.218498  0.016026 -0.393864 -0.886341  0.034911  0.091478 -0.040843 -0.0039577
[3,] -0.490500  0.060715 -0.117157 -0.063410  0.377398 -0.231003 -0.221091 -0.7023544
[4,] -0.483771  0.074611  0.093019 -0.115204  0.317365  0.557980  0.550480  0.1529124
[5,] -0.471967  0.026036 -0.116247 -0.082506 -0.848952  0.131099 -0.022261 -0.1335842
[6,] -0.491718  0.063521 -0.109440 -0.096658  0.138694 -0.402963 -0.302346  0.6781067
[7,]  0.073022  0.701976  0.012535  0.098898  0.049269  0.464578 -0.521878  0.0366216
[8,] -0.075640 -0.701255 -0.055776  0.069705  0.100433  0.477759 -0.503636  0.0505843
```

- Now the original variables are replaced by the principal components
- Component1 is -.05*zMedianInc+.22*zMedHomeAge - .49*zBedrooms etc.
- In other words… (next slide)

PCA

	Component							
	1	2	3	4	5	6	7	8
zMedIncome	-0.05	-0.04	0.89	-0.41	-0.06	-0.06	-0.17	-0.04
zMedAge	0.22	0.02	-0.39	-0.89	0.03	0.09	-0.04	0.00
zBedrooms	-0.49	0.06	-0.12	-0.06	0.38	-0.23	-0.22	-0.70
zRooms	-0.48	0.07	0.09	-0.12	0.32	0.56	0.55	0.15
zPop	-0.47	0.03	-0.12	-0.08	-0.85	0.13	-0.02	-0.13
zHouseholds	-0.49	0.06	-0.11	-0.10	0.14	-0.40	-0.30	0.68
zLat	0.07	0.70	0.01	0.10	0.05	0.46	-0.52	0.04
zLong	-0.08	-0.70	-0.06	0.07	0.10	0.48	-0.50	0.05

- We've now replaced our original 8 variables with 8 independent variables
- The key though is we don't need all of them

Remember the eigenvalues?

- 3.9, 1.9, 1.1 etc. are the first, second and third eigenvalues which correspond to the first, second, and third principal component.

- They explain the % of variance.

- So 3.9 out of 8 variables means the first principal component explains 3.9/8 = 49% of the variance (almost half)

- Continued on next slide

Dimension Reduction

Component	Eigenvalue	% of Variance	Cumulative %
1	3.91	48.8%	48.8%
2	1.91	23.8%	72.7%
3	1.07	13.4%	86.1%
4	0.82	10.3%	96.4%
5	0.15	1.9%	98.2%
6	0.08	1.0%	99.2%
7	0.05	0.6%	99.8%
8	0.01	0.2%	100.0%

- This suggests we get 96% of the model explained by the 4 new independent variables (principal components)

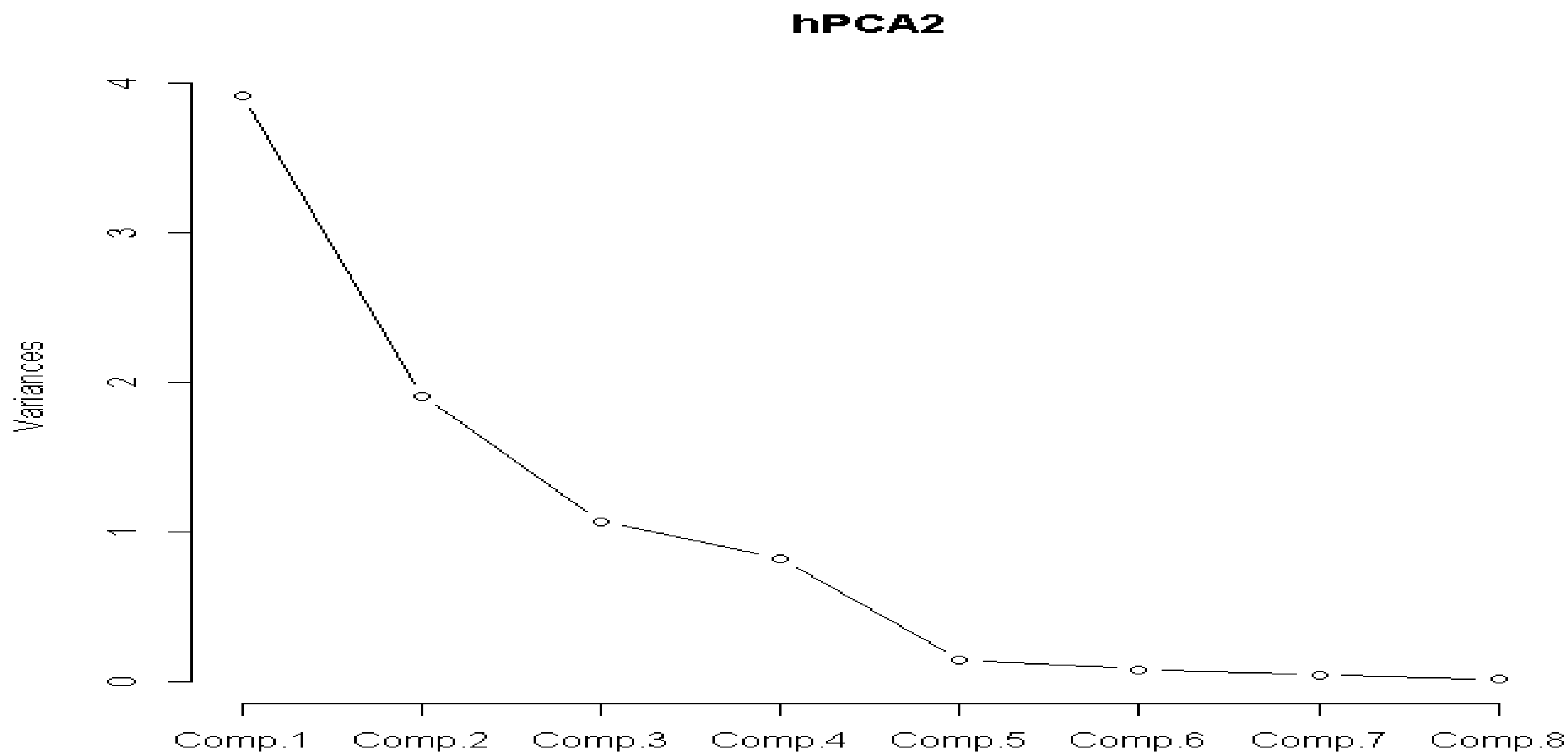
- Or 98% is explained by 5 principal components

How Many Principal Components do we use?

- You can make a table like the previous slide and decide to cut off at a certain percentage 95% or 98%

- Or you can make a scree plot like on next page with R: > screeplot(hPCA2,type="lines")

How Many Principal Components (II)?



- Typically, you stop when the scree plot gets flat (5)
- But it might be worth trying the model with both 4 and 5 components and seeing if it makes a difference

PCA in R

- You can do all of this directly in R without computing eigenvalues or covariant matrices etc.
  ```
  > hPCA<-princomp(houses)
  ```

- We can plot a screeplot as before, and we can get the components with
  ```
  > hPCA$loadings
  ```

Output of hPCA$Loadings

```
> hPCA2$loadings

Loadings:
              Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
MedianIncome                 0.891 -0.408               -0.169       
MedianHomeAge -0.218        -0.394 -0.886                            
TotalRooms     0.484               -0.115  0.317  0.558  0.550  0.153
TotalBedRooms  0.491        -0.117         0.377 -0.231 -0.221 -0.702
Population     0.472        -0.116        -0.849  0.131        -0.134
Households     0.492        -0.109         0.139 -0.403 -0.302  0.678
Latitude              0.702                       0.465 -0.522       
Longitude            -0.701                0.100  0.478 -0.504       
```

- Our previously defined components are same except our Comp1 = -Comp1 here. But that won’t affect results

User Defined Composites

- Notice in this case we had 8 variables where 4 were obviously highly correlated
- It’s a lot less complicated to make your own composite variable
- M=(zTotRooms/4+zBedrooms/4+zPop/4+zHouseholds/4)
- N=(zLatitude/2 –zLongitude/2)
- Note the latter has a negative because they were negatively correlated
- In this case we’ve replaced 6 variables with the 2 variables M and N so that we have 4 total instead of 8, and the 4 variables should be largely independent so that we can do Naïve Bayes or Regression without getting into trouble
- This is less complicated than PCA, but if you don’t know what you are doing, you can screw it up.

Go To R