Principal component analysis applications

Oliver W. Layton

CS252: Mathematical Data Analysis and Visualization

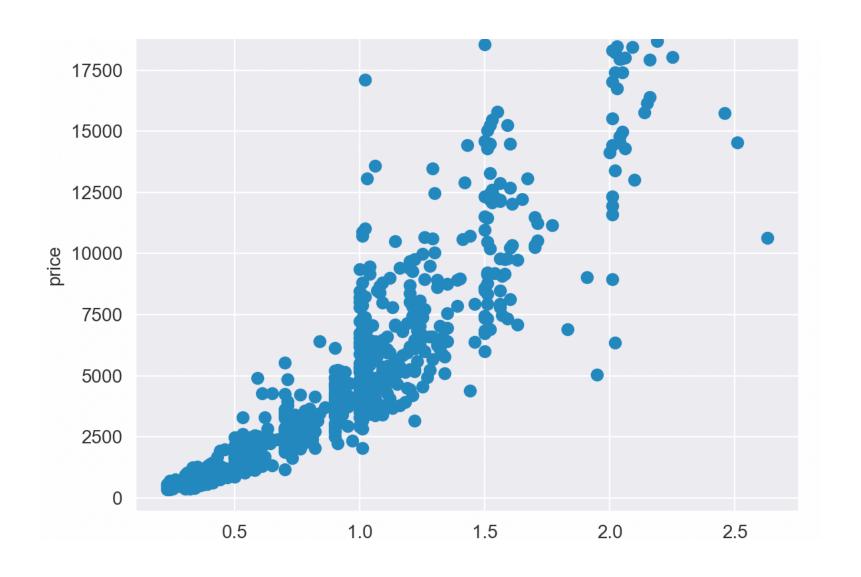
Lecture 20, Spring 2021

Monday March 29

Reconstruction of data from top principal components

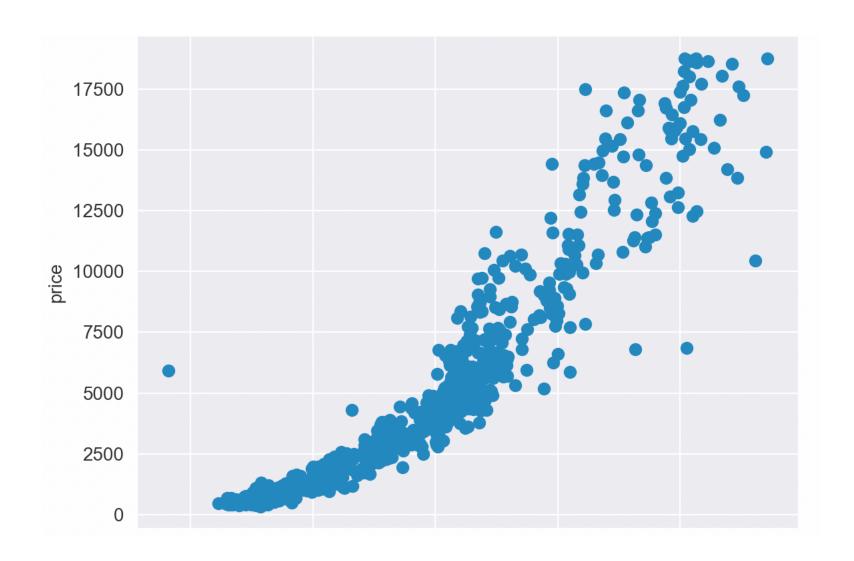
- PCA space useful for analysis, but sometimes we want to ask the question: "If I represent the data according to the top k eigenvectors, what would the data look like in their native space?"
- What might happen in the reconstruction process if we toss out eigenvectors?
 - This reconstruction process is **lossy** because we toss out information by dropping M-k out of M eigenvectors.

Original diamond data



3 variables selected (2 shown): price, carat, Z (depth measurement) (Not shown): Carat highly correlated with Z (doesn't account for much extra variance)

Reconstructed diamond data from top 2 PCs



3 variables selected (2 shown): price, carat, Z (depth measurement (Not shown): Carat highly correlated with Z (doesn't account for much extra variance)

Eigenfaces and facial recognition

- Project dataset: Celebrity Face images (LFWcrop)
- Paper on Eigenfaces: Turk & Pentland (1991).

Random sample of 500 faces

	500 randomly sampled faces																		
T	25	1	O.	管				1	E	<u>e</u>	4		•	0	*	0	营	1	1
1	Ø		D	•	0	•	৩	•	Ø	6	9	•	•	1		1		1	T.
	A.	3	51		3	1			重			9	•	1				9	· E
		***		1			25	0		1	9	4	Q _	1	1		•		1
.	A	S.	1	0	T	4	1	(Ed)	意		•	9	*		3				1
	堂	ø.	9	•	1	1	1	1	重	*			1		P	0	0.	1	9
1	0	1	Ď	100	1	(4)		40a					意		Ó	1	D	•	更
	(648)	8	意			9	1	1	色。	1	10	9		225			1		
	1	9	9	靊	10%	0		Ð	Ĭ	1	6	1	ē	1	•		1	Ø	1
1	ė		È.	1	重			N.		£.	Ž.	Ě	1	D		1		The second	(g)
J.		<u>Ú</u>	1	1	2		1	1		9	To.	3	0	1			2	A.D.	
			3		O	0	6	1		Ø.	9		ø,			•	3		0
Đ	•				1		9		9	9		*	1		0				
	夔		Ø.	9	-		1	T.		0	藝				TO	Ø		1	2
1	9	ė.		3	5	0	₫.	1	0	M.		1	1		Ð			0	(
9	9	9	ē.			9		0	ø.	10		1			9	3	Æ.		TO TO
Ø		e	3	ė		0	1	置、	e	e.	T)					0	8	9	9
靊					1		0 1	1	1	10	1	4	9	1	9	Ž	0	T.F	
1	1				6		CTO	ė.				12	意	1	9	Q .		1	
	0	1			*	3	0				8		•			3		1	
	9	Ú	(3)	•	1	2			1	•	1	0	1		•	Q.		1	0
1		0	1	1	1	0	Ø.	也	1		0	*	Ē	CE.	查	*	*	4	0
	•	1	•	0	躗	9	₫.			1		1	ě.	9			9	1	1
•	1	9	1	0	9		Ú	1	1	1			•	蔥	*	5-1)	1	0	質
1	0			1	1		Æ,		1	E.		Ú	1	9	•	D.	1	9	1

Eigenface algorithm: PCA on face images

- 1. Load in grayscale images, all the with same width and height: I_1, I_2, \ldots, I_N .
- 2. Collapse each 2D image into 1D vectors \vec{x}_i (e.g. 16x16 2D image \Rightarrow 256 1D vector). So, number of samples N = number of images. Variables are each of the pixels (e.g. if length(\vec{x}_i) is 256. M = 256). Like usual, A = [$\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_M$] (rows: images/samples, cols: 1D pixel value variables)
- 3. Center the images (subtract grand mean image): $A_c = A \vec{\mu}$, where $\vec{\mu}$ is the column means of A (i.e. the mean pixel value at the same position across all images in the dataset).
- 4. Compute covariance matrix Σ then recover eigenvalues and eigenvectors.
- 5. Project images onto top k of principal components.

Grand mean of 500 faces $(\vec{\mu})$



Because $\vec{\mu}$ is 1D vector, I had to **reshape** it into a 2D image format (e.g. 256 1D vector -> 16x16 2D image)

Variance explained by top eigenvalues/PCs

