Dimension Reduction

Data Mining
Prof. Dawn Woodard
School of ORIE
Cornell University

Outline

Learning

- Supervised Learning:
  - Observe pairs (X, Y) where X and/or Y can be multi-dimensional
  - Learn to estimate Y (the outcome) from X (the predictors)
- Unsupervised Learning:
  - Observe vectors X
  - Find “structure” in X

Unsupervised Learning

Several types of unsupervised learning:

1. Dimension reduction
   - I.e., obtain a lower-dimensional summary of high-dimensional data
   - E.g., the "principal components" method
2. Find groups (clusters) with most of the observations concentrated in these clusters
   - Includes K-means clustering, which is often covered in ORIE 3120
   - Can use the resulting clusters for, e.g. targeted marketing
   - <draw>
3. (Probability) density estimation
   - Estimate the distribution of a variable or variables

Dimension Reduction

Dimension Reduction: Sometimes we want to obtain a low-dimensional summary of the high-dimensional data we have on e.g. customers. This can be useful for:

- Obtaining a single number summarizing many variables like stock prices
- Allow visualization of high-dimensional data (by reducing to 2-4 dimensions and using scatterplots etc.)
- Removing redundancy in data to allow it to be interpreted more easily
  - Datasets often have many highly correlated / closely related variables (e.g. an individual's gross income and the amount of money in their bank account)
- As a preliminary step in another analysis like clustering or regression
  - in regression, would apply dimension reduction to just the predictor vars

Dimension Reduction

Examples:

- Stock indices
  - The Dow Jones average: replaces thousands of stock prices by a single numerical summary of 30 stock prices.
- Consumer Price Index
- Inflation indices

Dimension Reduction

Common-sense ways to reduce dimension of data:

- Keep only the variables that are deemed most likely to be relevant for the analysis of interest
  - e.g. financial variables are more likely to be useful than demographic variables when evaluating an individual's credit-worthiness
- Drop variables that have a lot of missing values or that are error-prone
  - E.g., self-reported overall health of an individual

Principal Components

An automatic approach: Principal Components Analysis.

- Only applied to a set of CONTINUOUS variables
- Capture most of the “information” (variability) in the original data using a low-dimensional vector of variables
- These variables are taken to be linear combinations of the original variables
- They are uncorrelated, and are called “principal components”

Singular Value Decomposition

Singular Value Decomposition:

For any real-valued $m \times n$ matrix A where $m > n$,

$\exists \underbrace{U}_{m \times n}, \underbrace{D}_{n \times n}, \underbrace{V}_{n \times n}$ such that $A = UDV^T$ and $U^T U = I_n$, $V^T V = I_n$,
and D is diagonal with nonegative entries
$d_1 \geq d_2 \geq \ldots \geq d_n \geq 0$.

Spectral Decomposition

Spectral Decomposition:

For any symmetric $n \times n$ matrix A,

$A = P\Lambda P^T$ where Λ is an $n \times n$ diagonal matrix with entries equal to the eigenvalues $\lambda_1, \ldots, \lambda_n$ of A

and $P = [w_1 \ w_2 \ \ldots \ w_n]$ and $\{w_i : i = 1, \ldots, n\}$ is a set of orthonormal eigenvectors of A that correspond to $\lambda_1, \ldots, \lambda_n$.

I.e., $\|w_i\| = 1$ and $w_i \perp w_j$ for all $i \neq j$.

I.e., $P^T P = I_n$.

Principal Components Analysis

Training data: have n observations and p variables. Want to reduce to $< p$ variables without losing much information.

Example: $p = 2$, so that our matrix of observed variables is

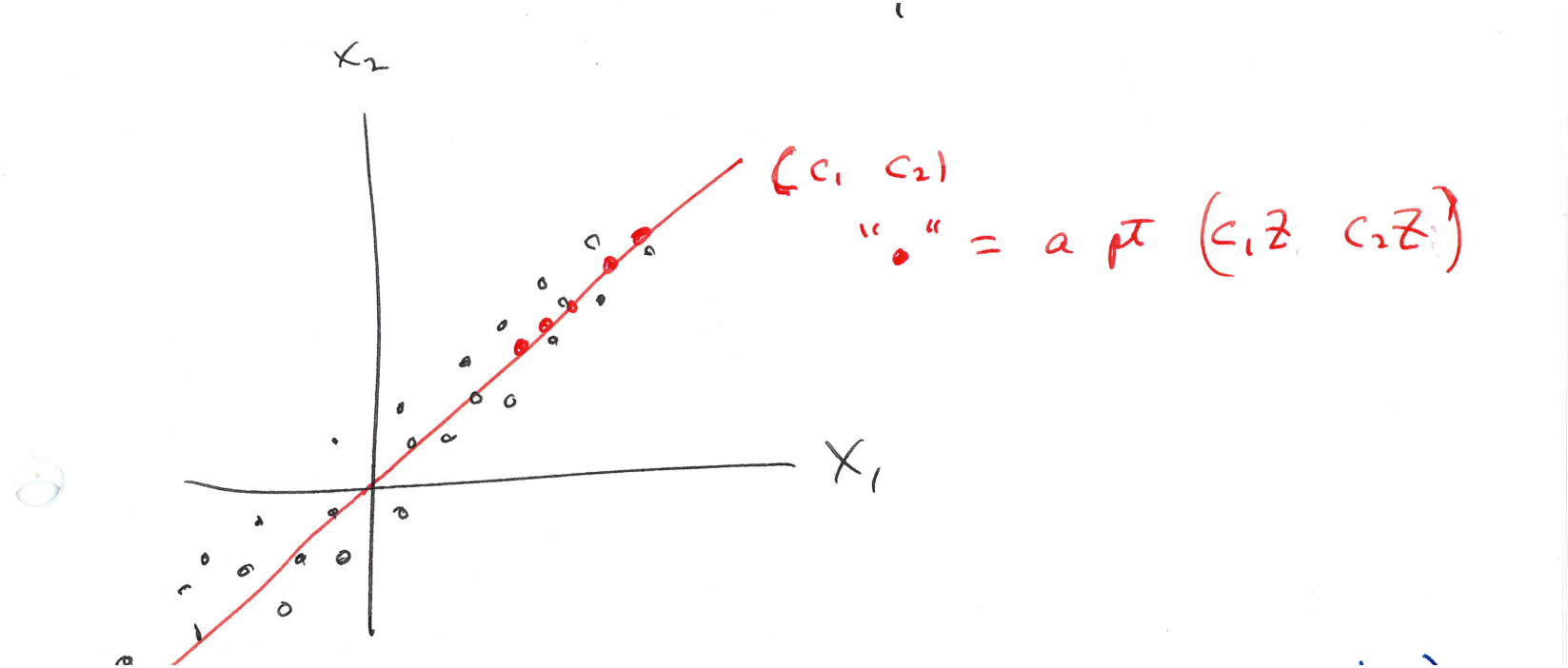
$$X = \begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ \vdots & \\ x_{n1} & x_{n2} \end{pmatrix}.$$

Assume that $\sum_{i=1}^{n} x_{i1} = 0$ and $\sum_{i=1}^{n} x_{i2} = 0$ for convenience. This can be easily obtained by subtracting the mean of each variable from that variable.

Principal Components Analysis

Want a single variable that is a linear combination of the two original variables. Geometrically this corresponds to projecting the points $(x_{i1}, x_{i2}) \in \mathbb{R}^2$ onto a line; the position on the line is the new variable.

Principal Components Analysis



Approximate each point $(x_{i1}\ x_{i2})$ by

$$(c_1 z_i\ c_2 z_i) = z_i(c_1\ c_2)$$

Here $(c_1\ c_2)$ is the direction vector in the plot and z_i is the position on that vector

Want $z_i(c_1\ c_2)$ to be close "on average" to $(x_{i1}\ x_{i2})$.

Principal Components Analysis

How to do such a dimension reduction (in general)? Using singular value decomposition:

Assume that $n > p$ where the matrix X is $n \times p$, and assume that each column of X has mean zero.

$$X = \underbrace{U}_{n\times p}\underbrace{D}_{p\times p}\underbrace{V^T}_{p\times p}$$

$$= (u_1\ u_2 \ldots\ u_p) \begin{pmatrix} d_1 & 0 & \ldots & 0 \\ 0 & d_2 & & 0 \\ \vdots & & & \\ 0 & \ldots & 0 & d_p \end{pmatrix} (v_1\ v_2\ \ldots v_p)^T$$

where $d_1 \geq d_2 \geq \ldots \geq d_p \geq 0$. Also, $U^T U = I_p = V^T V$.

This means that $\|u_i\| = \|v_i\| = 1$.

Principal Components Analysis

The sample covariance matrix of X is (because the columns of X have mean zero)

$$\begin{aligned}\text{Cov}(X) &= \frac{1}{n-1}X^TX = \frac{1}{n-1}(VDU^T)(UDV^T) \\ &= \frac{1}{n-1}VD^2V^T.\end{aligned}$$

This is a spectral decomposition of X^TX! So the columns of V are the eigenvectors of X^TX. This means they are also the eigenvectors of $\text{Cov}(X)$.

Also, $d_1^2 \geq d_2^2 \geq \ldots \geq d_p^2 \geq 0$ are the ordered eigenvalues of X^TX, and $\frac{d_1^2}{n-1} \geq \frac{d_2^2}{n-1} \geq \ldots \geq \frac{d_p^2}{n-1} \geq 0$ are the ordered eigenvalues of $\text{Cov}(X)$.

Principal Components Analysis

$UD = (u_1 d_1 \;\; u_2 d_2 \;\; \ldots \;\; u_p d_p)$ are called the "principal components" (PCs)

$X = (u_1 d_1 \;\; \ldots \;\; u_p d_p) V^T$.

V is the matrix of PC "loadings."

Note: $XV = UDV^T V = UD$.

Principal Components Analysis

Another way to look at PCA: Where $u_1, \ldots, u_p$ are the columns of U and $v_1, \ldots, v_p$ are the columns of V (eigenvectors of $X^T X$),

$$X = (u_1 d_1 \ \ldots \ u_p d_p) \begin{pmatrix} v_1^T \\ \vdots \\ v_p^T \end{pmatrix}$$

$$= \sum_{j=1}^{p} u_j d_j v_j^T$$

$$= \sum_{j=1}^{p} \underbrace{d_j}_{\text{scalar}} \underbrace{u_j v_j^T}_{n \times p \text{ matrix}}$$

Principal Components Analysis

Recall that $d_1 \geq d_2 \geq \ldots \geq d_p \geq 0$ and that $\|u_j\| = \|v_j\| = 1$ for all j

So for $q < p$, can use the approximation:

$$X \approx \sum_{j=1}^{q} d_j u_j v_j^T.$$

When d_1 is much larger than the other d_j, it may be reasonable to even use the approximation:

$$X \approx \underbrace{d_1}_{\text{scalar}} \underbrace{u_1}_{n\times 1} \underbrace{v_1^T}_{1\times p}.$$

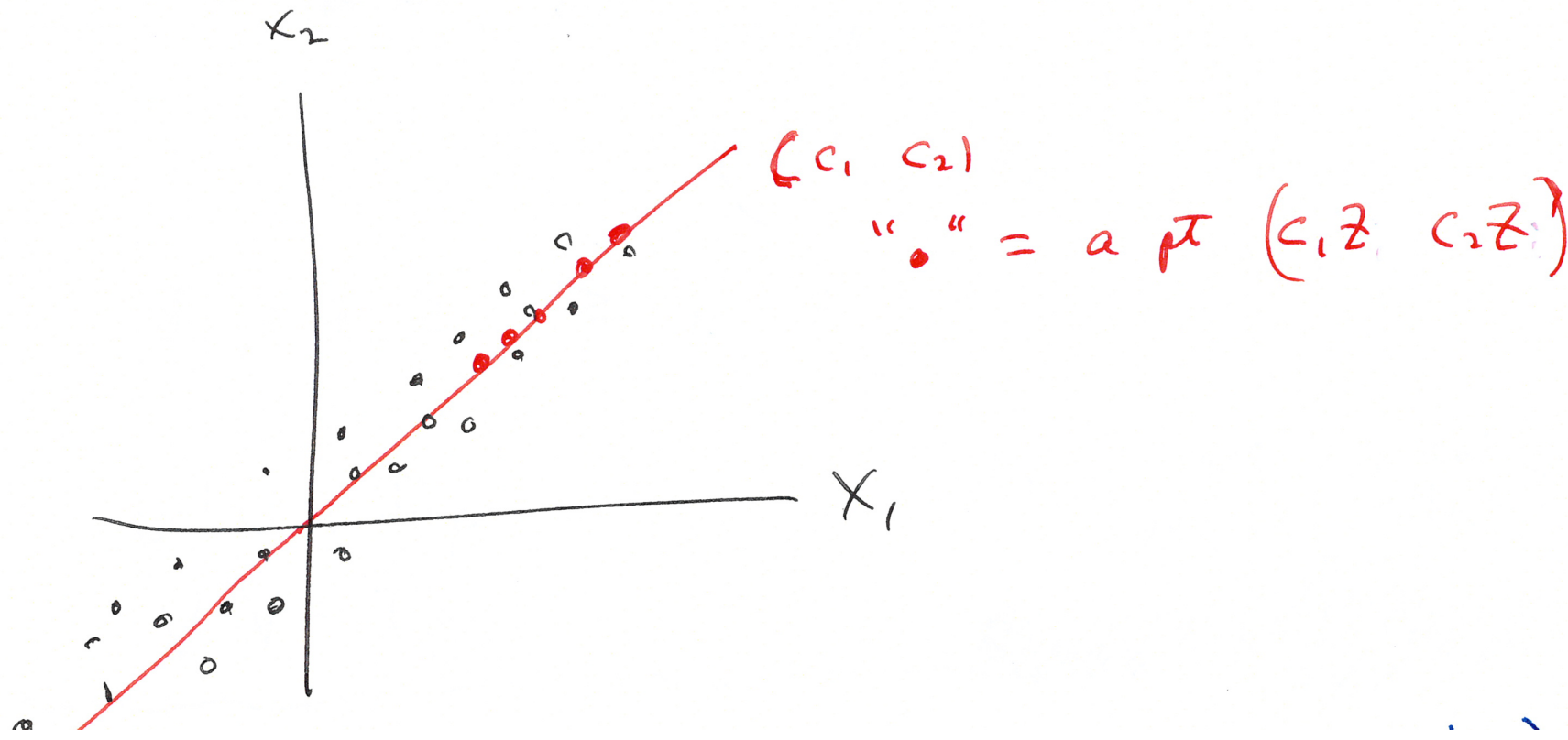
Principal Components Analysis

So for each observation i, we are approximating the p-dimensional observation vector x_i as:

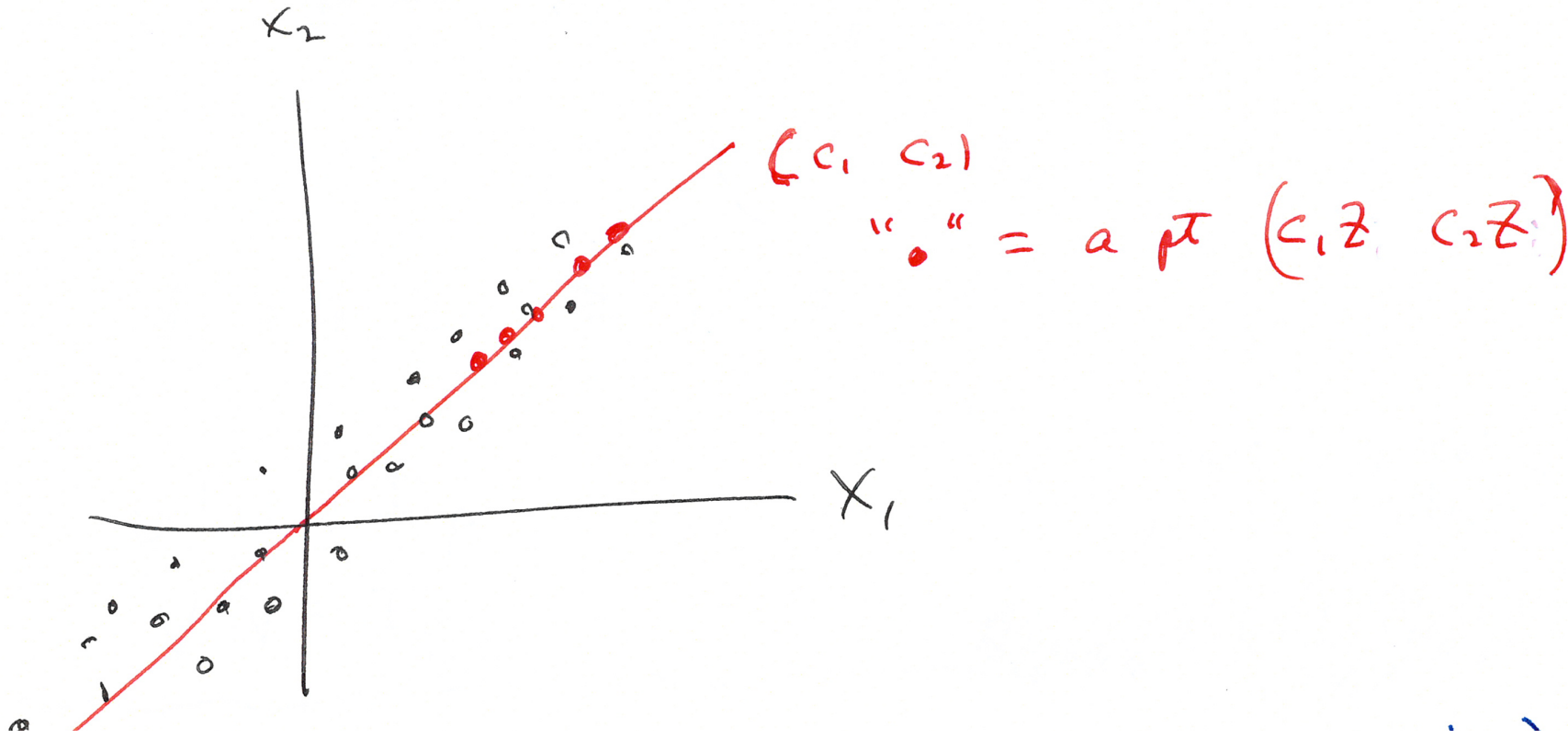
$$x_i \approx \underbrace{d_1 u_{1i}}_{\text{1st PC for } ith \text{ obs.}} \underbrace{v_1^T}_{\text{loadings vector for 1st PC}}.$$

Principal Components Analysis

Ex: when $p = 2$. Then $(x_{i1}\ x_{i2}) \approx d_1 u_{1i}(v_{11}\ v_{12})$.
Remember our picture:



Principal Components Analysis

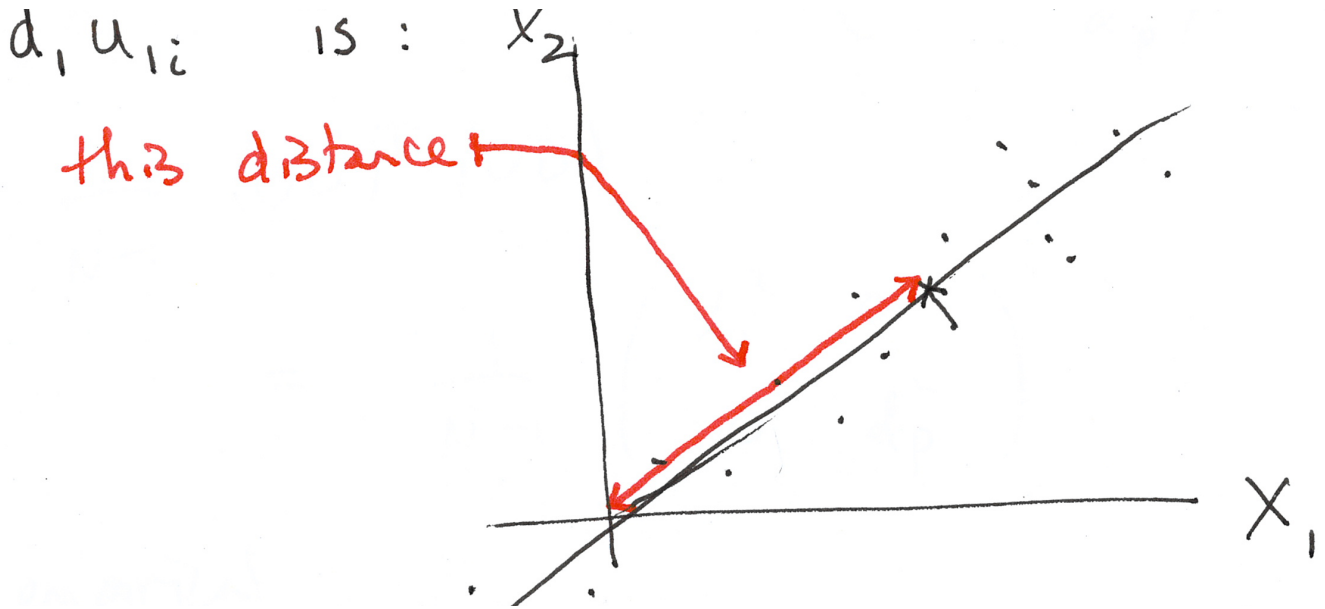


We wanted to find c_1, c_2, z_i so that $(x_{i1}\ x_{i2}) \approx z_i(c_1\ c_2)$.

We have found an approximation of this form, which is $(c_1\ c_2) = (v_{11}\ v_{12})$ and $z_i = d_1 u_{1i}$.

Principal Components Analysis

(v_{11} v_{12}) is the vector defining the line in our plot.



Principal Components

This data set contains nutritional and consumer rating data on 77 cereals (figures from Shmueli, Patel, and Bruce (2007)):

Cereal Name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass
100% Bran	N	C	70	4	1	130	10	5	6	280
100% Natural Bran	Q	C	120	3	5	15	2	8	8	135
All-Bran	K	C	70	4	1	260	9	7	5	320
All-Bran with Extra Fiber	K	C	50	4	0	140	14	8	0	330
Almond Delight	R	C	110	2	2	200	1	14	8	
Apple Cinnamon Cheerios	G	C	110	2	2	180	1.5	10.5	10	70
Apple Jacks	K	C	110	2	0	125	1	11	14	30
Basic 4	G	C	130	3	2	210	2	18	8	100
Bran Chex	R	C	90	2	1	200	4	15	6	125
Bran Flakes	P	C	90	3	0	210	5	13	5	190
Cap'n'Crunch	Q	C	120	1	2	220	0	12	12	35
Cheerios	G	C	110	6	2	290	2	17	1	105
Cinnamon Toast Crunch	G	C	120	1	3	210	0	13	9	45
Clusters	G	C	110	3	2	140	2	13	7	105
Cocoa Puffs	G	C	110	1	1	180	0	12	13	55
Corn Chex	R	C	110	2	0	280	0	22	3	25
Corn Flakes	K	C	100	2	0	290	1	21	2	35
Corn Pops	K	C	110	1	0	90	1	13	12	20
Count Chocula	G	C	110	1	1	180	0	12	13	65
Cracklin' Oat Bran	K	C	110	3	3	140	4	10	7	160

Principal Components

There are 15 variables in the data set:

mfr	Manufacturer of cereal (American Home Food Products, General Mills, Kellogg, etc.)
type	Cold or hot
calories	Calories per serving
protein	Grams of protein
fat	Grams of fat
sodium	Milligrams of sodium
fiber	Grams of dietary fiber
carbo	Grams of complex carbohydrates
sugars	Grams of sugars
potass	Milligrams of potassium
vitamins	Vitamins and minerals: 0, 25, or 100, indicating the typical percentage of FDA recommended
shelf	Display shelf (1, 2, or 3, counting from the floor)
weight	Weight in ounces of one serving
cups	Number of cups in one serving
rating	Rating of the cereal calculated by *Consumer Reports*

Principal Components

- 12 of the 15 variables are continuous.
- We wish to summarize the 12 continuous attributes with just a few variables that are linear combinations of those attributes
- We want these few variables to capture the original structure of the data as closely as possible
- For instance, we want the cereals that are close to each other in the original 12-dimensional space to be close to each other in the new low-dimensional space...
- and we want the cereals that are far apart in the original space to still be far apart.

Principal Components

- First let's consider just two of the original variables, calories and consumer rating, so that we can visualize the results
- The two variables are strongly negatively correlated: correlation = -0.69
- So there is redundancy in these two variables
- It might be possible to reduce these 2 variables to 1 variable without loosing too much information

Principal Components

Here is the PCA output (from XLMiner) for the 2 variables:

	Components	
Variable	**1**	**2**
calories	-0.84705347	0.53150767
rating	0.53150767	0.84705347

Variance	498.0244751	78.932724
Variance%	86.31913757	13.68086338
Cum%	86.31913757	100
P-value	0	1

The first principal component is given by
$(-0.85\ \text{calories}_i + 0.53\ \text{rating}_i)$

How do we calculate the second principal component?

Principal Components

What is the V matrix?

$$V = \begin{pmatrix} -0.85 & 0.53 \\ 0.53 & 0.85 \end{pmatrix}$$

The elements are called the *principal component loadings*

Recall that $XV = UD$, where UD are the principal components

Principal Components

Here are the values of the principal components for some of the cereals:

100% Bran	44.92152786	2.19717932
100% Natural Bran	-15.7252636	-0.38241446
All-Bran	40.14993668	-5.40721178
All-Bran with Extra Fiber	75.31076813	12.99912071
Almond Delight	-7.04150867	-5.35768652
Apple Cinnamon Cheerios	-9.63276863	-9.48732758
Apple Jacks	-7.68502998	-6.38325357
Basic 4	-22.57210541	7.52030993
Bran Chex	17.7315464	-3.50615811
Bran Flakes	19.96045494	0.04600986
Cap'n'Crunch	-24.19793701	-13.88514996
Cheerios	1.66467071	8.5171833
Cinnamon Toast Crunch	-23.25147057	-12.37678337
Clusters	-3.84429598	-0.26235023
Cocoa Puffs	-13.23272038	-15.2244997
Corn Chex	-3.28897071	0.62266076
Corn Flakes	7.5299263	-0.94987571

This is the matrix $UD = XV$.

Principal Components

- Now let's apply PCA to all 13 of the continuous variables (we will include "shelf" as a continuous variable to be consistent with Shmueli et al.)

Principal Components

Here is the PCA output showing the loadings for the first several principal components:

Variable	1	2	3	4	5	6	7
calories	0.07798425	-0.00931156	0.62920582	-0.60102159	0.45495847	0.11884782	0.09385654
protein	-0.00075678	0.00880103	0.00102611	0.00319992	0.05617596	0.11274506	0.25810272
fat	-0.00010178	0.00269915	0.01619579	-0.02526222	-0.01609845	-0.13181572	0.37258437
sodium	0.98021454	0.14089581	-0.13590187	-0.00096808	0.01394816	0.02279307	0.00450823
fiber	-0.00541276	0.03068075	-0.01819105	0.0204722	0.01360502	0.2628414	0.0431139
carbo	0.01724625	-0.0167833	0.01736996	0.02594825	0.34926692	-0.53783643	-0.67243195
sugars	0.00298888	-0.00025348	0.09770504	-0.11548097	-0.29906642	0.64792335	-0.5669753
potass	-0.13490002	0.98656207	0.03678251	-0.0421758	-0.04715054	-0.04999856	-0.01795866
vitamins	0.09429332	0.01672884	0.69197786	0.714118	-0.03700861	0.01575723	0.01210225
shelf	-0.00154142	0.0043604	0.01248884	0.00564718	-0.00787646	-0.0599014	0.09221537
weight	0.000512	0.00099922	0.00380597	-0.00254643	0.00302211	0.00905157	-0.02361298
cups	0.00051012	-0.00159098	0.00069433	0.00098539	0.00214846	-0.01030537	-0.01959434
rating	-0.07529629	0.07174215	-0.30794701	0.33453393	0.75770795	0.41302064	0.01832427

How would we calculate the first principal component for “All-Bran” cereal?

Principal Components

- Which variables contribute the most to the first principal component?
- Which variables contribute the most to the second?
- Why do you think this is?

Principal Components

- Would this change if we rescaled the variables?

- This suggests that we need to standardize the variables before applying PCA to the cereals data

Principal Components

Here is the PCA output after standardizing the variables:

Variable	1	2	3	4	5	6	7
calories	0.2995424	0.39314792	0.11485746	0.20435865	0.20389892	-0.25590625	-0.02559552
protein	-0.30735639	0.16532333	0.27728197	0.30074316	0.319749	0.120752	0.28270504
fat	0.03991544	0.34572428	-0.20489009	0.18683317	0.58689332	0.34796733	-0.05115468
sodium	0.18339655	0.13722059	0.38943109	0.12033724	-0.33836424	0.66437215	-0.28370309
fiber	-0.45349041	0.17981192	0.06976604	0.03917367	-0.255119	0.0642436	0.11232537
carbo	0.19244903	-0.14944831	0.56245244	0.0878355	0.18274252	-0.32639283	-0.26046798
sugars	0.22806853	0.35143444	-0.35540518	-0.02270711	-0.31487244	-0.15208226	0.22798519
potass	-0.40196434	0.30054429	0.06762024	0.09087842	-0.14836049	0.02515389	0.14880823
vitamins	0.11598022	0.1729092	0.38785872	-0.6041106	-0.04928682	0.12948574	0.29427618
shelf	-0.17126338	0.26505029	-0.00153102	-0.63887852	0.32910112	-0.05204415	-0.17483434
weight	0.05029929	0.45030847	0.24713831	0.15342878	-0.22128329	-0.39877367	0.01392053
cups	0.29463556	-0.21224795	0.13999969	0.04748911	0.12081645	0.09946091	0.74856687
rating	-0.43837839	-0.25153893	0.1818424	0.0383162	0.05758421	-0.18614525	0.06344455

Variance	3.63360572	3.1480546	1.90934956	1.01947618	0.98935974	0.72206175	0.67151642
Variance%	27.95081329	24.21580505	14.6873045	7.84212446	7.61045933	5.55432129	5.16551113
Cum%	27.95081329	52.16661835	66.85391998	74.69604492	82.3065033	87.86082458	93.02633667

Principal Components

Here is a scatterplot of the first two principal components (after standardization):