CPSC 340: Machine Learning and Data Mining

Principal Component Analysis

Fall 2015

Admin

- Midterm on Friday.
  - Assignment 3 solutions posted after class.
  - Practice midterm posted.
  - List of topics posted.
  - In class, 55 minutes, closed-book, cheat sheet: 2-pages each double-sided.

Last time: Stochastic Gradient Methods

- We want to fit a regression model:

$$\underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \sum_{i=1}^{n} g(y_i, w^T x_i) + \lambda r(w)$$

- If ‘g’ and ‘r’ are smooth, gradient descent allows huge ‘d’.
- When ‘n’ is huge/infinite, we can use stochastic gradient:

Set ‘i_t’ to a random training example.

$$w^{t+1} = w^t - \alpha_t \nabla f_{i_t}(w_t)$$

- For convergence, α_t must go to zero.
- Amazing theoretical properties in terms of test error:
  - Even for non-IID data, but in practice often doesn’t live up to expectations.
- Nevertheless, widely-used because it allows enormous datasets.

The Story So Far…

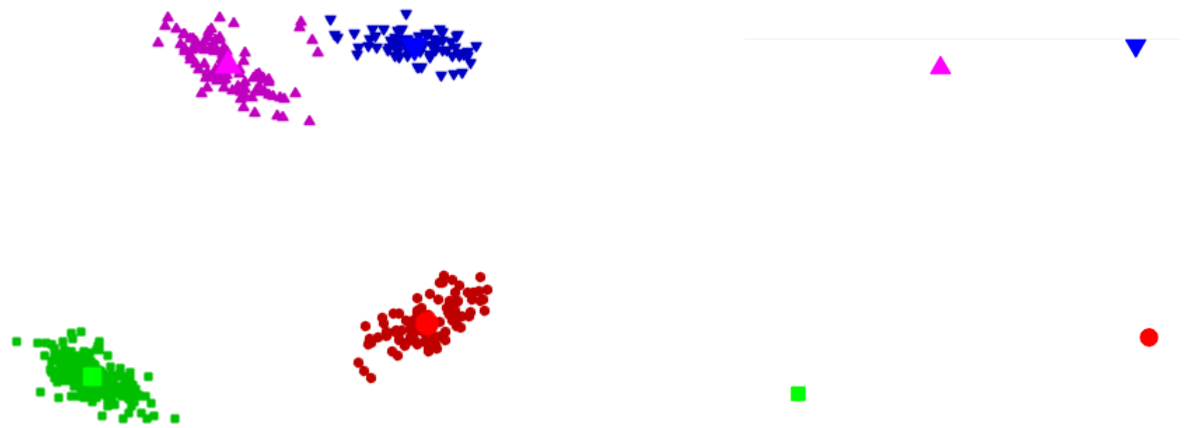
- Supervised Learning Part 1:
  - Methods based on counting and distances.
  - Training vs. testing, parametric vs. non-parametric, ensemble methods.
  - Fundamental trade-off, no free lunch.
- Unsupervised Learning Part 1:
  - Methods based on counting and distances.
  - Clustering and association rules.
- Supervised Learning Part 2:
  - Methods based on linear models and gradient descent.
  - Continuity of predictions, suitability for high-dimensional problems.
  - Loss functions, change of basis, regularization, features selection, big problems.
- Unsupervised Learning Part 2:
  - Methods based on linear models and gradient descent.

Unsupervised Learning Part 2

- Unsupervised learning:
  - We only have x_i values, and want to do 'something' with them.
- Some unsupervised learning tasks:
  - Clustering: What types of x_i are there?
  - Association rules: Which x_{ij} occur together?
  - Outlier detection: Is this a 'normal' x_i?
  - Data visualization: What does the high-dimensional X look like?
  - Ranking: Which are the most important x_i?
  - Latent-factors: What 'parts' are the x_i made from?

Motivation: Vector Quantization

- K-means was originally designed for vector quantization:
  - Find a set of 'means', so that each object is close to mean.
  - Compress the data by replacing each object by its mean:



  - You only need to store means, and cluster 'c_i' for each object.
  - But you lose a lot of information unless number of means is large.

Latent-Factor Models

Weird notation alert:
w_c is $d \times 1$
w_j is $k \times 1$

- Latent-factor models:
  - We don't call them 'means' μ_c, we call them factors w_c.
  - Approximate each object as a linear combination of factors:

K-means: $x_i \approx w_{c_i}$ or $x_{ij} \approx (w_{c_i})_j$ Latent-factor: $x_{ij} \approx \sum_c (w_c)_j z_{ic} = w_j^T z_i$

  - We still have 'k' by 'd' matrix 'W' of factors/means.
  - Instead of cluster 'c_i', we have 'k' by '1' weight vector 'z_i' for each 'I'.
  - K-means: special case where each (z_i = 1) for 'c_i' and (z_i = 0) zero otherwise.
- Matrix inner factorization notation:

$X \approx ZW$ where Z is 'n' by 'k' and W is 'k' by 'd'.

- Compresses if 'k' is much smaller than 'd'.
  - Above assumes features have been standardized (otherwise, need bias).

Principal Component Analysis

$$\|z\|^2 = \sum_{j=1}^{d} (z_i)^2$$

- Recall the k-means objective function:

$$\sum_{i=1}^{n} \|x_i - w_{c_i}\|^2 = \sum_{i=1}^{n} \sum_{j=1}^{d} (x_{ij} - (w_{c_i})_j)^2$$

  - The variables are the means 'W' and clusters c_i.
- Using the latent-factor approximation we obtain:

$$\sum_{i=1}^{n} \sum_{j=1}^{d} (x_{ij} - w_j^T z_i)^2$$

  - The variables are the factors 'W' and low-dimensional 'features' Z.
- Minimizing this is called principal component analysis (PCA):
  - The factors/means 'w_c' are called 'principal components'.

PCA Applications

- Dimensionality reduction: replace 'X' with lower-dimensional 'Z'.
- Outlier detection: if PCA gives poor approximation of x_i, could be 'outlier'.
- Basis for linear models: use 'Z' as features in regression models.

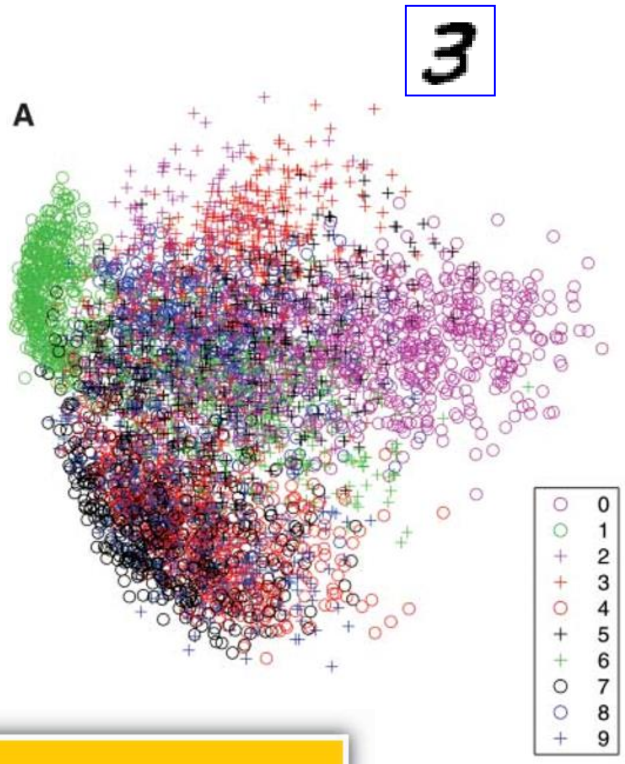
Compute approximation $X \approx ZW$

Now use Z as features in linear model:

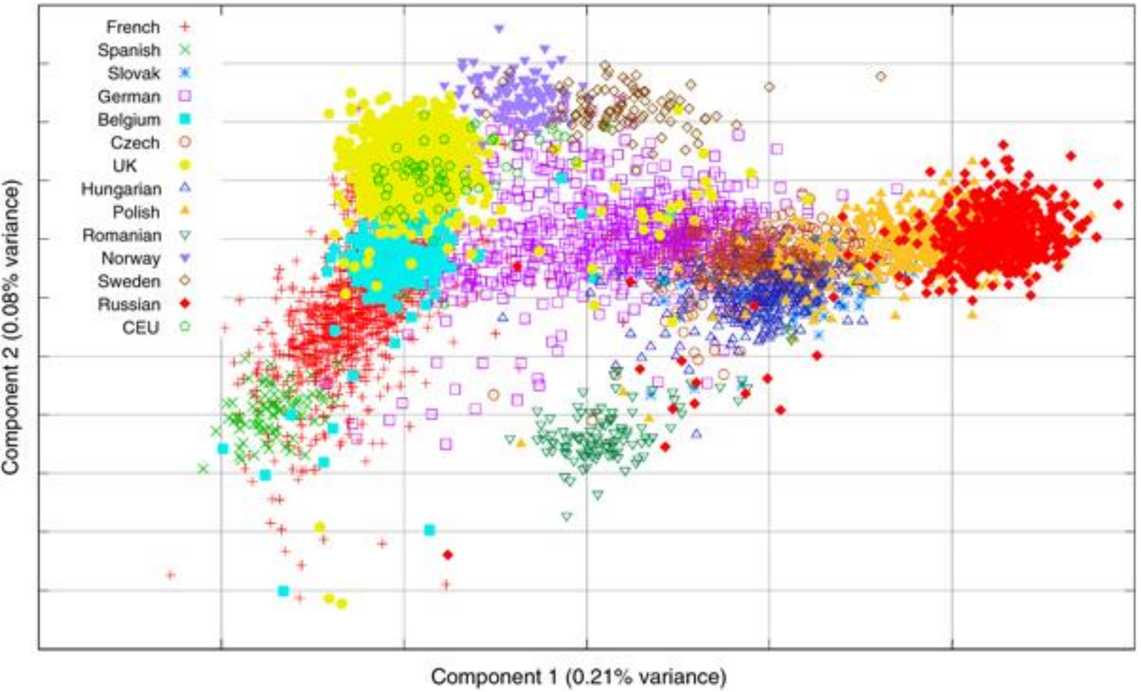
$$y_i = w^T z_i + \beta$$

N.B., different 'w': this one trained for regression.

PCA Applications



– Data visualization: display the z_i in a scatterplot:



– Interpret factors:

Trait	Description
Openness	Being curious, original, intellectual, creative, and open to new ideas.
Conscientiousness	Being organized, systematic, punctual, achievement-oriented, and dependable.
Extraversion	Being outgoing, talkative, sociable, and enjoying social situations.
Agreeableness	Being affable, tolerant, sensitive, trusting, kind, and warm.
Neuroticism	Being anxious, irritable, temperamental, and moody.

https://www.cs.toronto.edu/~hinton/science.pdf
http://infoproc.blogspot.ca/2008/11/european-genetic-substructure.html
https://new.edu/resources/big-5-personality-traits

Maximizing Variance vs. Minimizing Error

- PCA has been reinvented many times:

PCA was invented in 1901 by Karl Pearson,[1] as an analogue of the principal axis theorem in mechanics; it was later independently developed (and named) by Harold Hotelling in the 1930s.[2] Depending on the field of application, it is also named the discrete Kosambi-Karhunen–Loève transform (KLT) in signal processing, the Hotelling transform in multivariate quality control, proper orthogonal decomposition (POD) in mechanical engineering, singular value decomposition (SVD) of **X** (Golub and Van Loan, 1983), eigenvalue decomposition (EVD) of $\mathbf{X}^T\mathbf{X}$ in linear algebra, factor analysis (for a discussion of the differences between PCA and factor analysis see Ch. 7 of [3]), Eckart–Young theorem (Harman, 1960), or Schmidt–Mirsky theorem in psychometrics, empirical orthogonal functions (EOF) in meteorological science, empirical eigenfunction decomposition (Sirovich, 1987), empirical component analysis (Lorenz, 1956), quasiharmonic modes (Brooks et al., 1988), spectral decomposition in noise and vibration, and empirical modal analysis in structural dynamics.

standard deviation of 3 in roughly the (0.878, 0.478) direction and of 1 in th orthogonal direction. The vectors shown are the eigenvectors of the covariance matrix scaled by the squa root of the corresponding eigenvalue, and shifted so their tails are at the mean.

- There are many ways to arrive at the same model:
  - Classic ‘analysis’ view: PCA maximizes variance in compressed space.
    - You pick the ‘w_c’ to explain as much variance as possible.
  - We take the ‘synthesis’ view: PCA minimizes error of approximation.
    - Makes connection to k-means and least squares.
    - We can use tricks from linear regression to fix PCA’s many problems.

https://en.wikipedia.org/wiki/Principal_component_analysis

PCA with 1 Principal Component

- PCA with one principal component (PC) ‘w’:

$$X = \underbrace{\begin{bmatrix} \quad \end{bmatrix}}_{n \times d} \quad w = \underbrace{\begin{bmatrix} \quad \end{bmatrix}}_{1 \times d} \quad z = \underbrace{\begin{bmatrix} \circ \end{bmatrix}}_{n \times 1}$$

the "principal component"

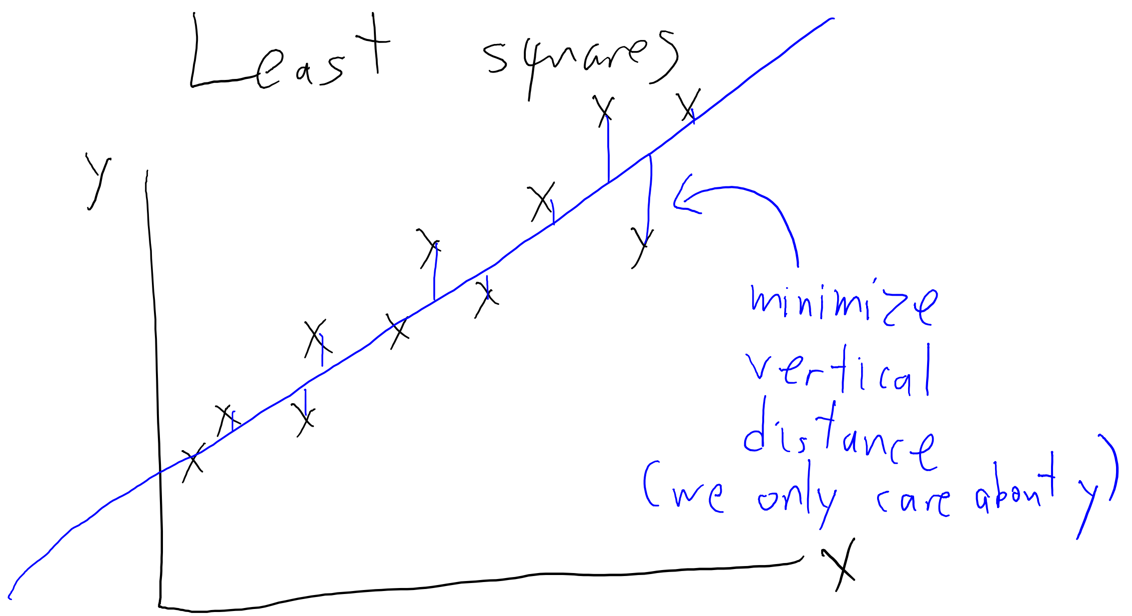
new "feature" for example 'i'.

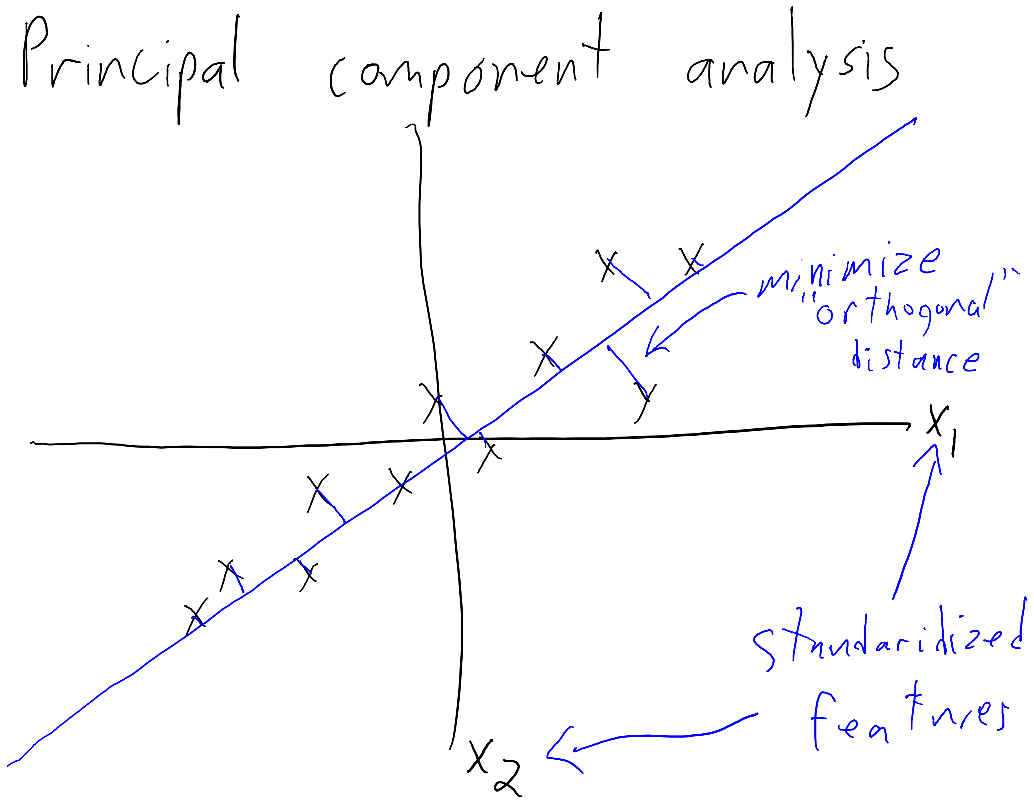
$$f(w,z) = \sum_{i=1}^{n}\sum_{j=1}^{d}(x_{ij} - w_j z_i)^2$$

- Very similar to a least squares problem, but note that:
  - We have no ‘y_i’, we are trying to predict each vector feature x_{ij} from the z_i.
  - Latent feaures ‘z_i’ are also variables, we are learning the z_i too.

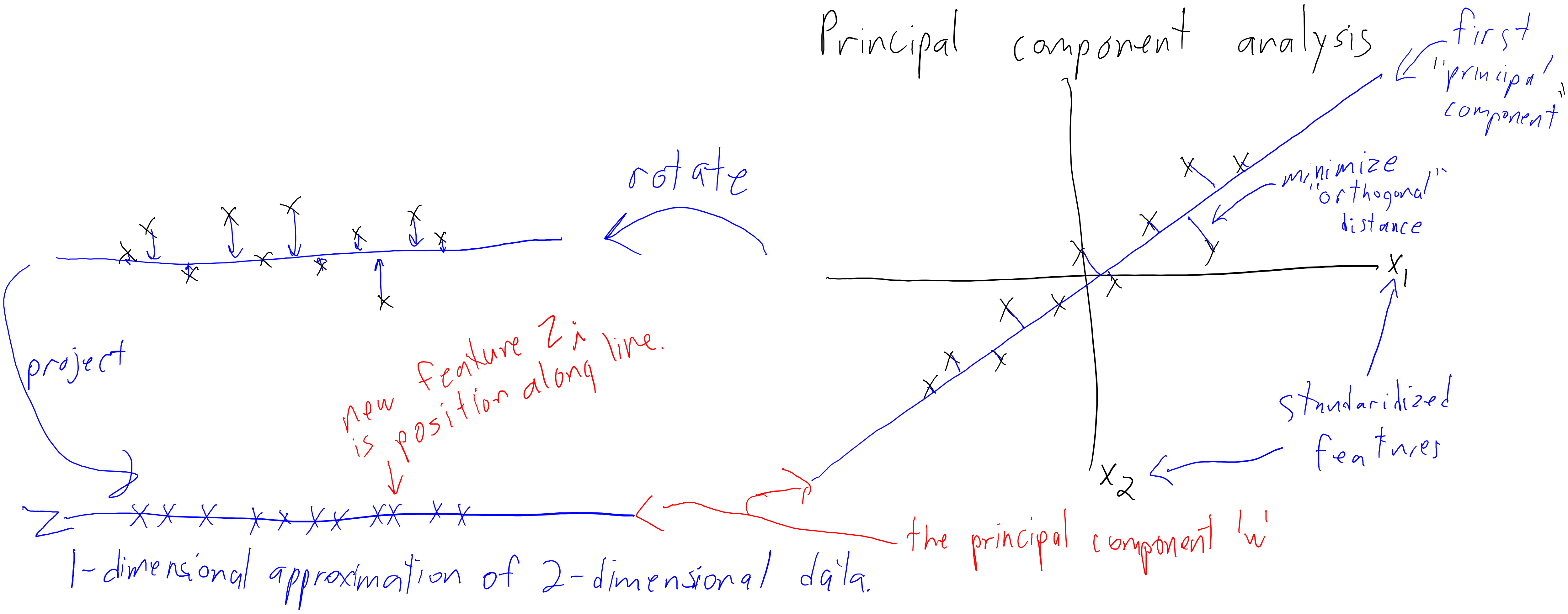
(if you know the z_i, equivalent to least squares)

PCA with 1 Principal Component





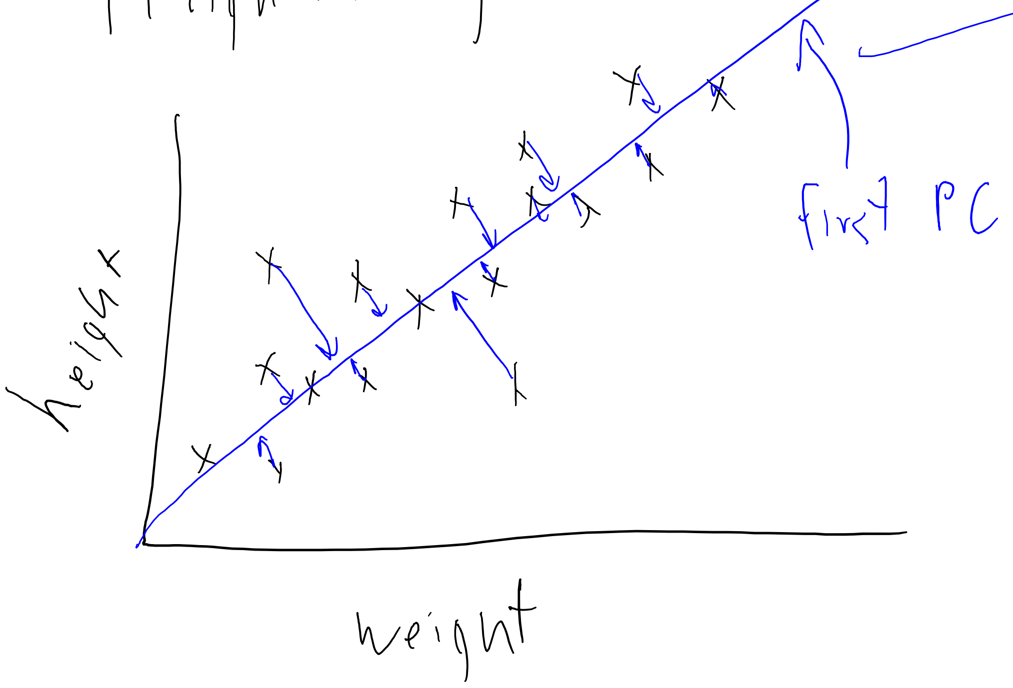
PCA with 1 Principal Component

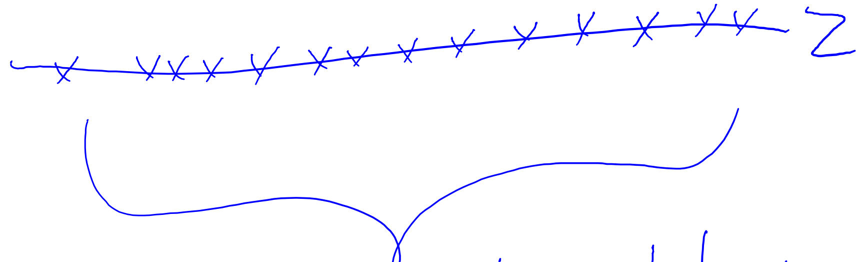


PCA with 1 Component

Height/weight of children:

transformed data





PC can be interpreted as overall 'size' or maybe 'age'.

PCA with 1 Component

- Our PCA objective function with one PC:

$$f(w, z) = \sum_{i=1}^{n} \sum_{j=1}^{d} (x_{ij} - w_j z_i)^2$$

- For small problems use closed-form solution:
  - First 'right singular vector' of X is a solution.
  - Equivalently, eigenvector of $X^T X$ with largest eigenvalue.
- For problems where 'd' is large, alternating minimization:
  - Update w given the z_i, then update the z_i given w (similar to k-means)
  - Convex in w, convex in z_i, but not jointly convex.
  - But, only stable local minimum is a global minimum.
- When 'n' is large, recent provably-correct stochastic gradient methods.

PCA with 1 Component

- Our PCA objective function with one PC:

$$f(w, z) = \sum_{i=1}^{n} \sum_{j=1}^{d} (x_{ij} - w_j z_i)^2$$

- Even with 1 PC, solution is never unique:

Same model if replace w by αw and z with $(\frac{1}{\alpha})z$.

- To address this issue, we usually put a constraint on 'w':

$\|w\| = 1$ or equivalently $w^T w = 1$.

- For iterative methods, can do this afterwards (then update the z_i).

General PCA

- Our general PCA framework:

each row is a PC

row 'i' gives compressed version of example 'i'

$$X = \underbrace{\begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}}_{n \times d} \quad W = \underbrace{\begin{bmatrix} & \\ \hline & \end{bmatrix}}_{k \times d} \quad Z = \underbrace{\begin{bmatrix} & \\ \hline & \end{bmatrix}}_{n \times k}$$

- General objective function: $f(W, Z) = \sum_{i=1}^{n} \sum_{j=1}^{d} (x_{ij} - w_j^T z_i)^2$

- Same solution methods (closed-form is top 'k' singular vectors).
- With multiple components, even directions are not unique.

Non-Uniqueness of PC Directions

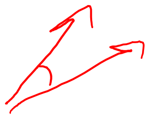
- We still have the scaling problem:

We get same model if you replace W by αW and Z with $(\frac{1}{\alpha})Z$.

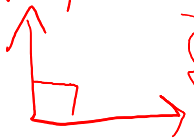
Usual fix is to require $||w_c|| = 1$ for all factors 'c', or equivalently $w_c^T w_c = 1$

- But with multiple PCs, we have new problems:
  - Factors could be non-orthogonal (components interfere with each other):
    - Usual fix to make the PCs orthogonal: $w_c^T w_{c'} = 0$ for $c \neq c'$.
  - Label switching: could swap w_c and $w_{c'}$ (if swap columns c and c' of z_i):
    - Usual fix is to fit the PCs sequentially.

"Orthogonal" vectors: high-dimensional version of vectors being perpendicular:

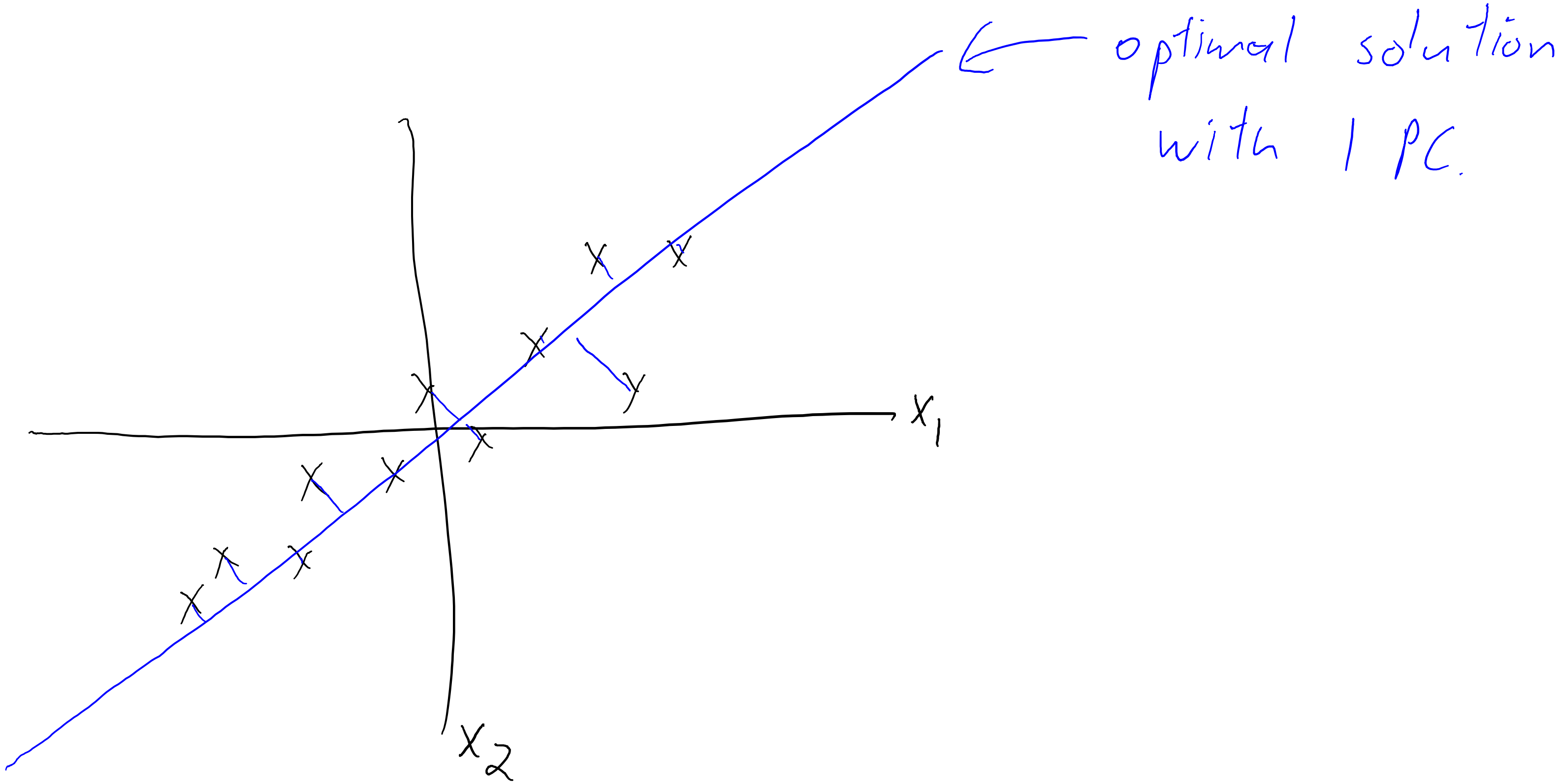


not orthogonal



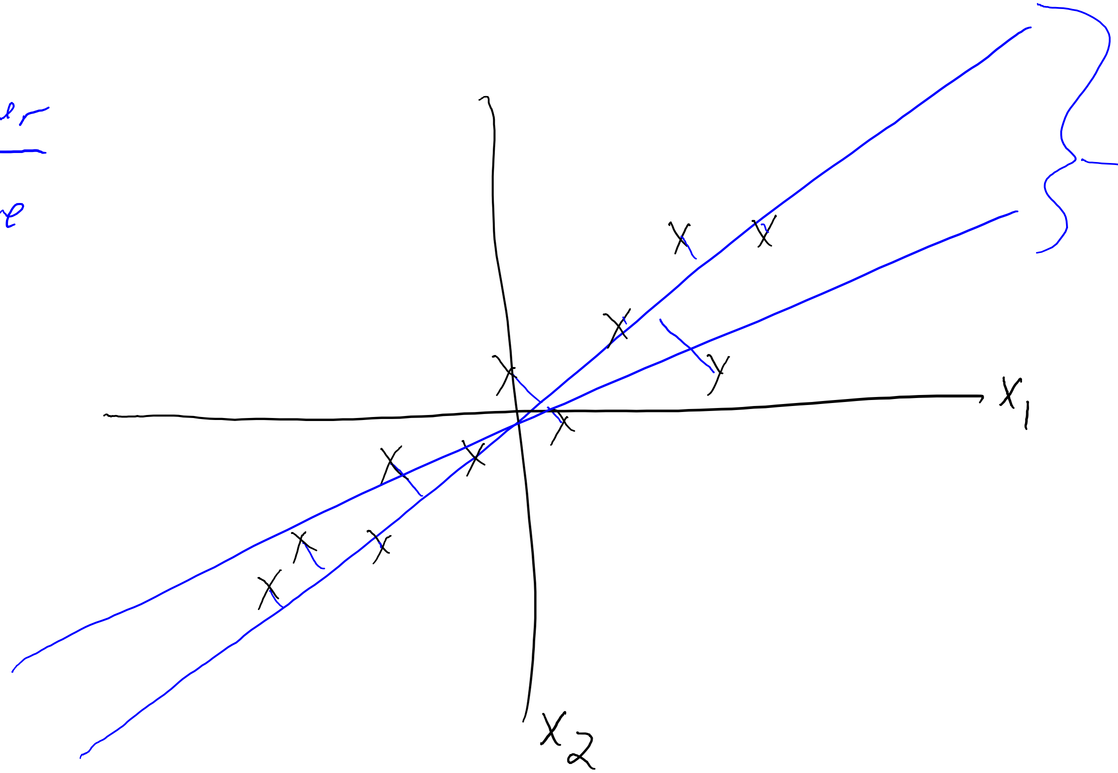
orthogonal

Basis, Orthogonality, Sequential Fitting



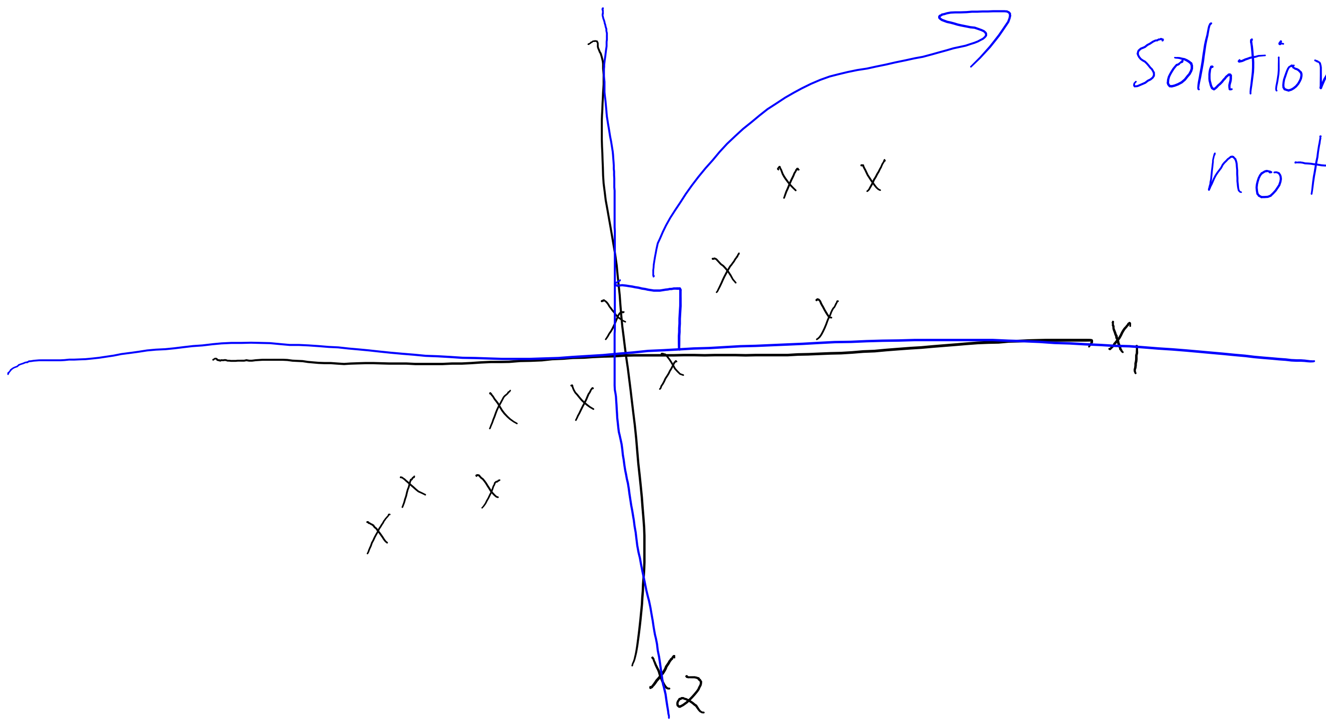
Basis, Orthogonality, Sequential Fitting

In 2D, any <u>other</u> direction will give an optimal solution. (error of 0)



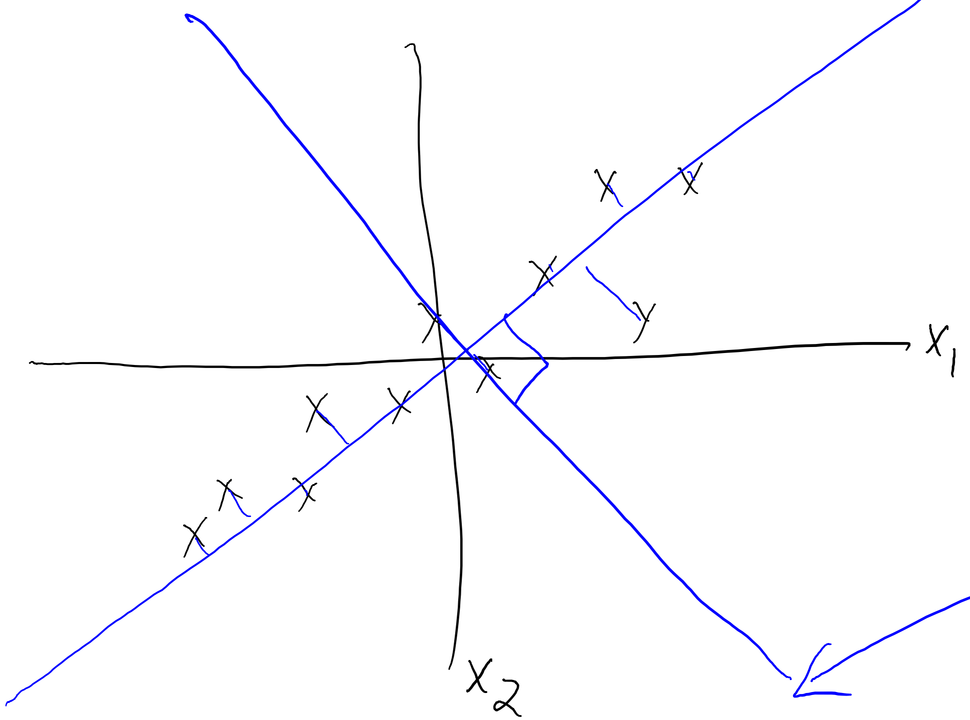
an optimal solution, but <u>not orthogonal:</u> PC2 is giving almost same information as PC1

Basis, Orthogonality, Sequential Fitting



An orthogonal optimal solution, but PCs have nothing to do with data.

Basis, Orthogonality, Sequential Fitting



optimal solution with 1 PC.

optimal PC2 that is orthogonal to PC1.
(in 2D, there is only one choice)

PCA with Singular Value Decomposition

- Under constraints that $w_c^T w_c = 1$ and $w_c^T w_{c'} = 0$, use:

$$U \Sigma V^T = SVD(X)$$

$$W = V(:, 1{:}k)^T \qquad Z = XW^T$$

- You can also quickly get compressed version of new data:

$$\hat{Z} = \hat{X} W^T$$

- If W was not orthogonal, could get Z by least squares.

Application: Face Detection

- 'Eigenfaces' classically used as basis for face detection:

1024 pixels

reconstructing based on first 100 PCs.

PC1 PC2

top 36 principal components

http://mikedusenberry.com/on-eigenfaces/

Summary

- Latent-factor models compress data as linear combination of ‘factors’.
- Principal component analysis: most common variant based on squared reconstruction error.
- Orthogonal basis is useful for interpretation and identifying of PCs.

- Next time: the discovering a hole in the ozone layer.