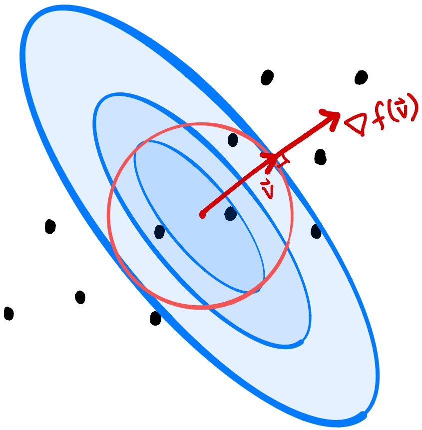


CSE 151A

Intro to Machine Learning

Lecture 17 – Part 01

Dimensionality Reduction

Announcements

- Midterm 02 is Friday! Covers Week 05 – Week 08.
  - Canvas Quiz. Tip: **don’t** use Safari.
- There are no more mandatory homeworks.
  - I **will** be posting more plus problems (maybe another competition...).
  - I’ll post some “essential” conceptual questions about Weeks 09 and 10, but they will not be turned in. Preparation for final exam.

Dimensionality Reduction

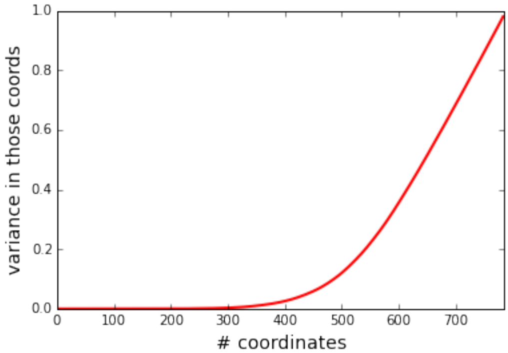
- Too many features hurts performance.
- Einstein: “Everything should be made as simple as possible, but no simpler.”

Dimensionality Reduction

- **Given**: a data set in high dimensions.
- **Reduce** dimensionality while preserving information.
- Why?
  - Faster, less memory.
  - High-dimensional data usually has redundancy.
  - Remove noisy/irrelevant features.

Example

Example


1.0
0.8
0.6
0.4
0.2
0.0
variance in those coords
0
100
200
300
400
500
600
700
coordinates

Assumption

- Variance is **interesting**.
    - More variable features are more useful.

D-R Approach #1

- Start with data in d dimensions.
- Compute variance of each feature.
- Keep only the k features with most variance.

This is OK...

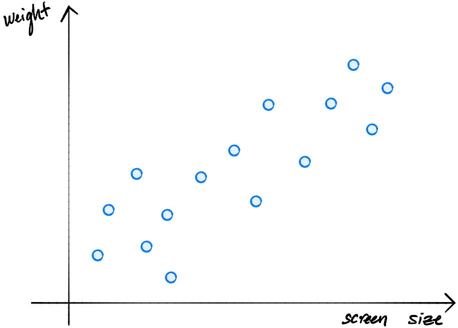
- ...but we can do better.
- **Problem**: features are often redundant.
- **Example**: height and weight

D-R Approach #2

- Find features which vary **together**.
  - **Example**: height and weight.
- Create **new** features which are combinations of old features.
- Keep best k combinations.

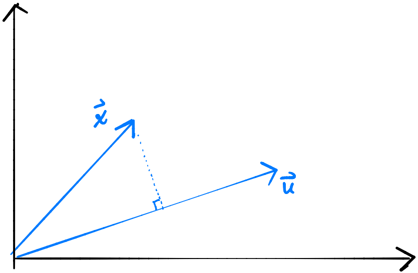
Example

Suppose we want just one feature.

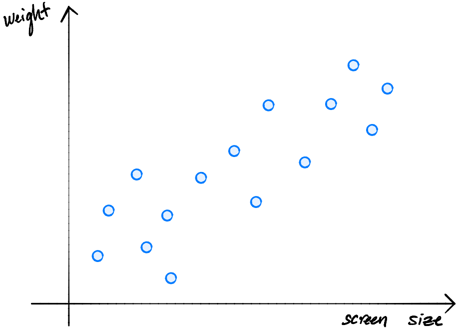


Projections

- The **projection** of $\vec{x} \in \mathbb{R}^d$ along the direction $\vec{u} \in \mathbb{R}^d$ (where $\vec{u}$ is a unit vector) is $(\vec{x} \cdot \vec{u})\vec{u}$.



Projections



The Problem

- **Given**: data $\vec{x}^{(1)}, \ldots, \vec{x}^{(n)} \in \mathbb{R}^d$
- **Map**: each data point $\vec{x}^{(i)}$ to a single feature, z_i.
  - Later: $\vec{z}^{(i)} \in \mathbb{R}^{d'}$, $d' \leq d$.
- **Idea**: map $\vec{x}^{(i)}$ by projecting it onto direction $\vec{u}$ of maximum variance.
  - $z_i = \vec{x}^{(i)} \cdot \vec{u} = \sum_{j=1}^{d} u_j x_j^{(i)}$

Variance in a Direction

- Let $\vec{u}$ be a unit vector.
- $\vec{x}^{(i)} \cdot \vec{u}$ is the new feature for $\vec{x}^{(i)}$.
- The variance of the new features is:

$$\begin{aligned}\mathrm{Var}(z_1, \ldots, z_n) &= \frac{1}{n}\sum_{i=1}^{n} z_i^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}\left(\vec{x}^{(i)} \cdot \vec{u}\right)^2\end{aligned}$$

Claim

- Suppose the data is **centered**.
  - The average of each feature is zero.
- Let C be the data's covariance matrix.
- Then the variance in the direction of $\vec{u}$ is:

$$\mathrm{Var}(z_1, \ldots, z_n) = \vec{u}^T C \vec{u}$$

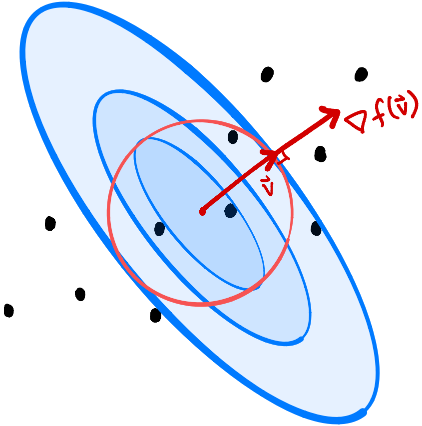
The Problem (More Formally)

- **Given**: covariance matrix C of centered data $\vec{x}^{(1)}, \ldots, \vec{x}^{(n)} \in \mathbb{R}^d$

- **Find**: the **unit** vector $\vec{u}$ maximizing $\vec{u}^T C \vec{u}$

The Problem (More Formally)

- **Given**: covariance matrix C of centered data $\vec{x}^{(1)}, \ldots, \vec{x}^{(n)} \in \mathbb{R}^d$
- **Find**: the **unit** vector $\vec{u}$ maximizing $\vec{u}^T C \vec{u}$
- **How**?



CSE 151A
Intro to Machine Learning

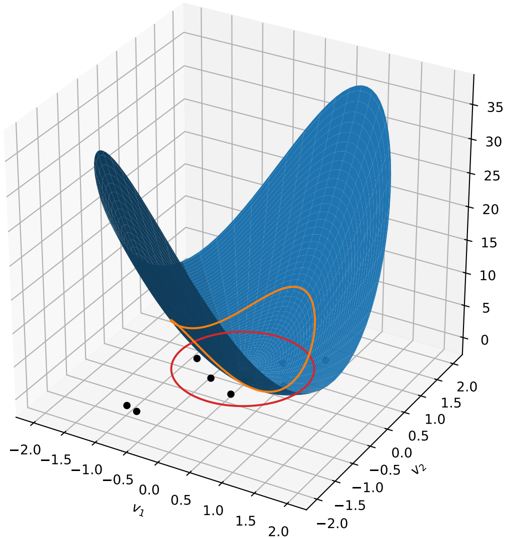
Lecture 17 – Part 02
Optimization

The Problem

- **Given**: covariance matrix C of centered data $\vec{x}^{(1)}, \ldots, \vec{x}^{(n)} \in \mathbb{R}^d$
- **Find**: the **unit** vector $\vec{u}$ maximizing $\vec{u}^T C \vec{u}$

The Variance Function

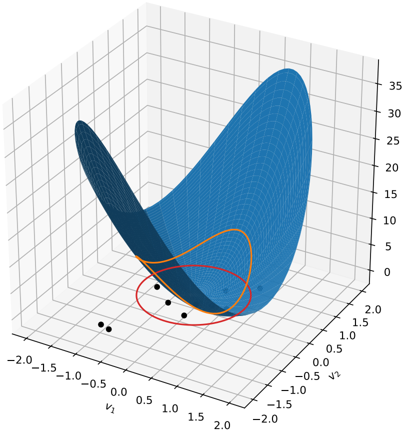
- Define $f(\vec{v}) = v^T C v$.
- **Claim**: f is **paraboloidal**.



35
30
25
20
15
10
5
0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
v_1
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
v_2

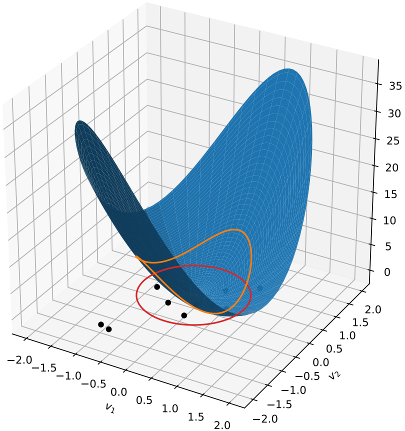
Optimization

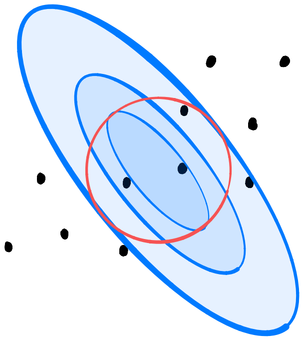
- Set gradient to zero, solve?

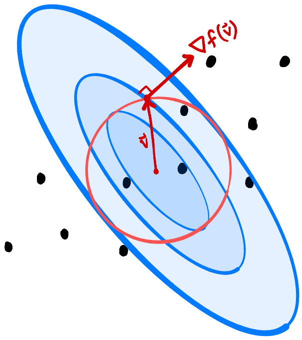


Optimization

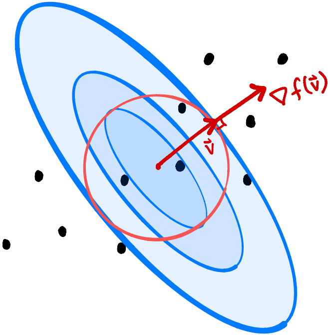
- Set gradient to zero, solve? **No.**







$\nabla f(\vec{v})$
$\vec{v}$



∇f(v⃗)
v⃗

The Solution

- We want to maximize $f(\vec{v})$ subject to $\|\vec{v}\| = 1$.
- Necessary: $\vec{v}$ is in same direction as $\nabla f(\vec{v})$.

$$\nabla f(\vec{v}) = \lambda \vec{v} \tag{1}$$

- **Lagrange Multipliers**

Claim

- Remember: $f(\vec{v}) = \vec{v}^T C \vec{v}$
- Claim: $\nabla f(\vec{v}) = 2C\vec{v}$
- Condition (1) becomes:

$$2C\vec{v} = \lambda\vec{v}$$

- $\vec{v}$ must be an **eigenvector** of C.

Remember: Eigenvectors

- An **eigenvector** of a matrix A is a vector $\vec{u}$ such that $A\vec{u} = \lambda\vec{u}$. λ is called the **eigenvalue**.
- Matrices can have many eigenvector/eigenvalue pairs.
- If A $(d \times d)$ is symmetric, positive definite, there is a set of d mutually orthogonal eigenvectors.

Recap

- **Goal**: Find unit vector $\vec{u}$ maximizing $\vec{u}^T C \vec{u}$.
  - I.e., find unit vector in direction of maximum variance.

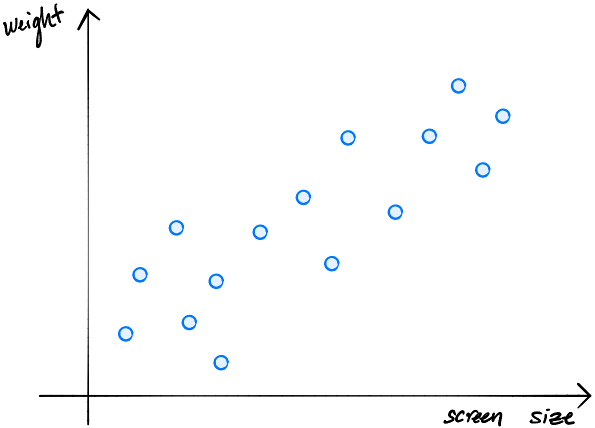
- Any solution must satisfy $2C\vec{u} = \lambda\vec{u}$
  - I.e., it must be an eigenvector of C.

- The **top eigenvector** of the covariance matrix points in direction of maximum variance.

Principal Components

- The **top eigenvector** of the covariance matrix is called the **principal component**.
- It points in the direction of maximum variance.
- Idea: it is the “most interesting” direction.

Principal Component Projections



Principal Component Analysis

- **Given**: data $\vec{x}^{(1)}, \dots, \vec{x}^{(n)} \in \mathbb{R}^d$
- **Map**: each data point $\vec{x}^{(i)}$ to a single feature, z_i.
- **PCA**: Let $z_i = \vec{x}^{(i)} \cdot \vec{u}$, where $\vec{u}$ is top eigenvector of covariance matrix.

Next Time

- **Given**: data $\vec{x}^{(1)}, \ldots, \vec{x}^{(n)} \in \mathbb{R}^d$
- **Map**: each data point $\vec{x}^{(i)}$ to a lower-dimensional vector, $\vec{z}^{(i)} \in \mathbb{R}^{d'}, d' \leq d$.