EECS 545 - Fall 2019 – 220 Chrysler

Machine Learning

Instructor: Alfred Hero

Lecture 2

https://umich.instructure.com/courses/315575

Course information

Canvas website

https://umich.instructure.com/courses/315575


MICHIGAN
DATA SCIENCE TEAM
UNIVERSITY OF MICHIGAN
Fall Information Session
Thurs 9/26, 6-7PM
TBD
Learn about...
Data Science Competitions • Beginner-Friendly Tutorials
Public Service Projects • And more!
mdst.club/join

Important actions for you:

- Sign up for Piazza and Gradescope

  Piazza: EECS545's social media for communication
  https://piazza.com/class/jzz150krgh87bc?cid=6

  Gradescope: Course code 9KNG2E
  https://www.gradescope.com/courses/60834

Tutorial sessions this week

<u>Monday, Sept 9: (Linear Algebra & Probability 1)</u>

7 - 9 pm in Chrysler

<u>Tuesday, Sept 10: (Python 1)</u>
8 - 10 pm in Chrysler

<u>Wednesday, Sept 11: (Linear Algebra & Probability 2)</u>
8 - 10 pm in Chrysler

<u>Thursday, Sept 12: (Python 2)</u> 7 - 9 pm in 1571 GG Brown

Mathematical notation

- Predictor and predictee variables are respectively mapped to vector $\mathbf{x} \in \mathbb{R}^d$ and scalar $y \in \mathbb{R}$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} \in \mathbb{R}^d, \qquad y \in \mathbb{R}$$

- $\mathbf{x}$ is called an input, pattern, signal, instance, example, or feature vector.
- y is called an output, response or label.

Basic mathematical framework for ML

- $y \in \mathcal{Y}$: output variable, response variable, label variable
- $\mathbf{x} \in \mathcal{X}$: input variable, feature variable, covariate
- $h \in \mathcal{H}$: set of predictor functions $h : \mathcal{X} \to \mathcal{Y}$.
- $l(h(\mathbf{x}), y)$: loss or error function, characterizing goodness of fit of h
- $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$: a training sample, the available data.

A concise definition of ML

The objective of Machine Learning is to design a prediction function h using training data $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ in such a way that it can be applied in the future to accurately predict the unobserved label y of an observation $\mathbf{x}$.

In particular, given a loss function $l(h, y)$, the prediction function h should produce a prediction $h(\mathbf{x})$ that incurs low loss

$$l(h(\mathbf{x}), y)$$

for most y.

Nomenclature

Some adjectives are used to describe ML algorithms. Recall that ML uses a training set $S = \{(\mathbf{x}_i, y_i\}_{i=1}^n$ to produce a prediction function h for future application to a novel sample $\mathbf{x}$.

- **Distributional assumptions**: a machine learning algorithm is called generative if it is based on the full probabilistic model for the data S. It is discriminative if it assumes only a partial or no probabilistic model.
- **Computational form**: A machine learning algorithm is linear if it produces a linear/affine function h, otherwise it is non-linear.
- **Model complexity**: A learning algorithm has growing complexity in n if evaluation of $h(\mathbf{x})$ requires access to the entire sample S. It has fixed complexity in n if evaluation of $h(\mathbf{x})$ only requires access to a low dimensional summarization of S, with dimension not growing with n.

The k Nearest Neighbor (kNN) classifier

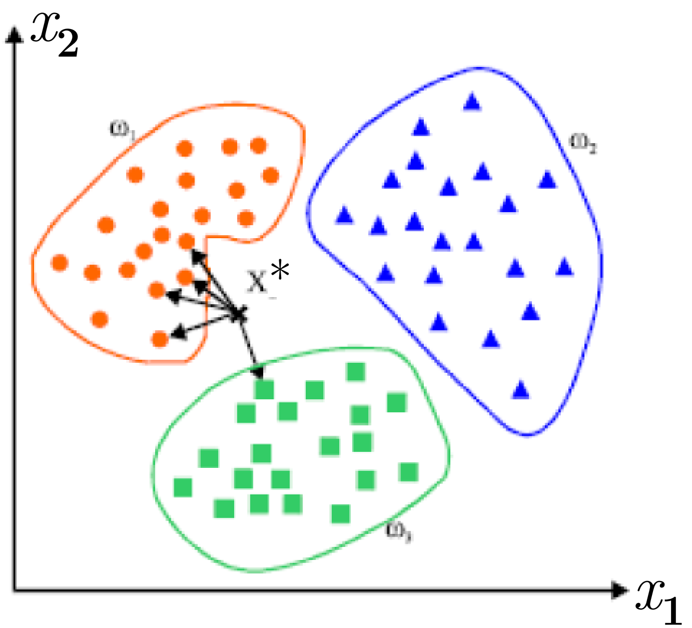
- Given labeled training data $S = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$
- For any out-of-sample data point $\mathbf{x}^* \notin S$

1. Compute the n distances $d_{i,*} = \|\mathbf{x}^* - \mathbf{x}_i\|$
2. Rank order $d_{i,*}$'s and keep track of rank indices

$$d_{i_1,*} < d_{i_2,*} < \quad < d_{i_n,*}$$

3. Select top k indices in this rank ordering
4. $h_{kNN}(\mathbf{x}^*)$:=most common label in $y_{i_1}, , y_{i_k}$ (majority vote assignment rule)

C=3 classes



kNN classifier

$\mathbf{x}^* = (x_1, x_1)$

kNN algorithm is specified by one parameter k

Training the kNN has computational complexity of order dn^2

Illustration: kNN for k=1 (NN)

- NN classifier: assigns to $\mathbf{x}$ the same label as that of the closest $\mathbf{x_i}$

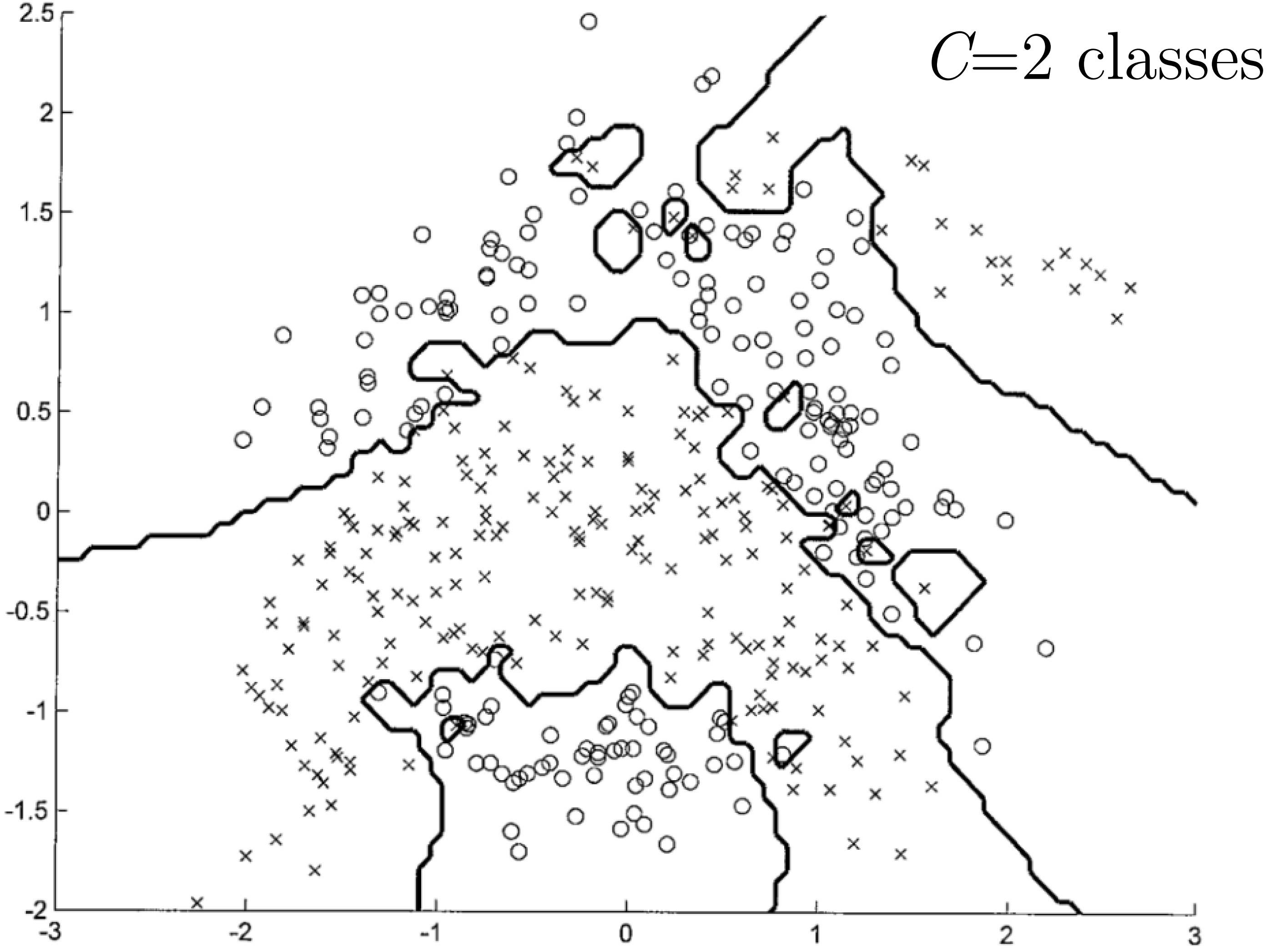
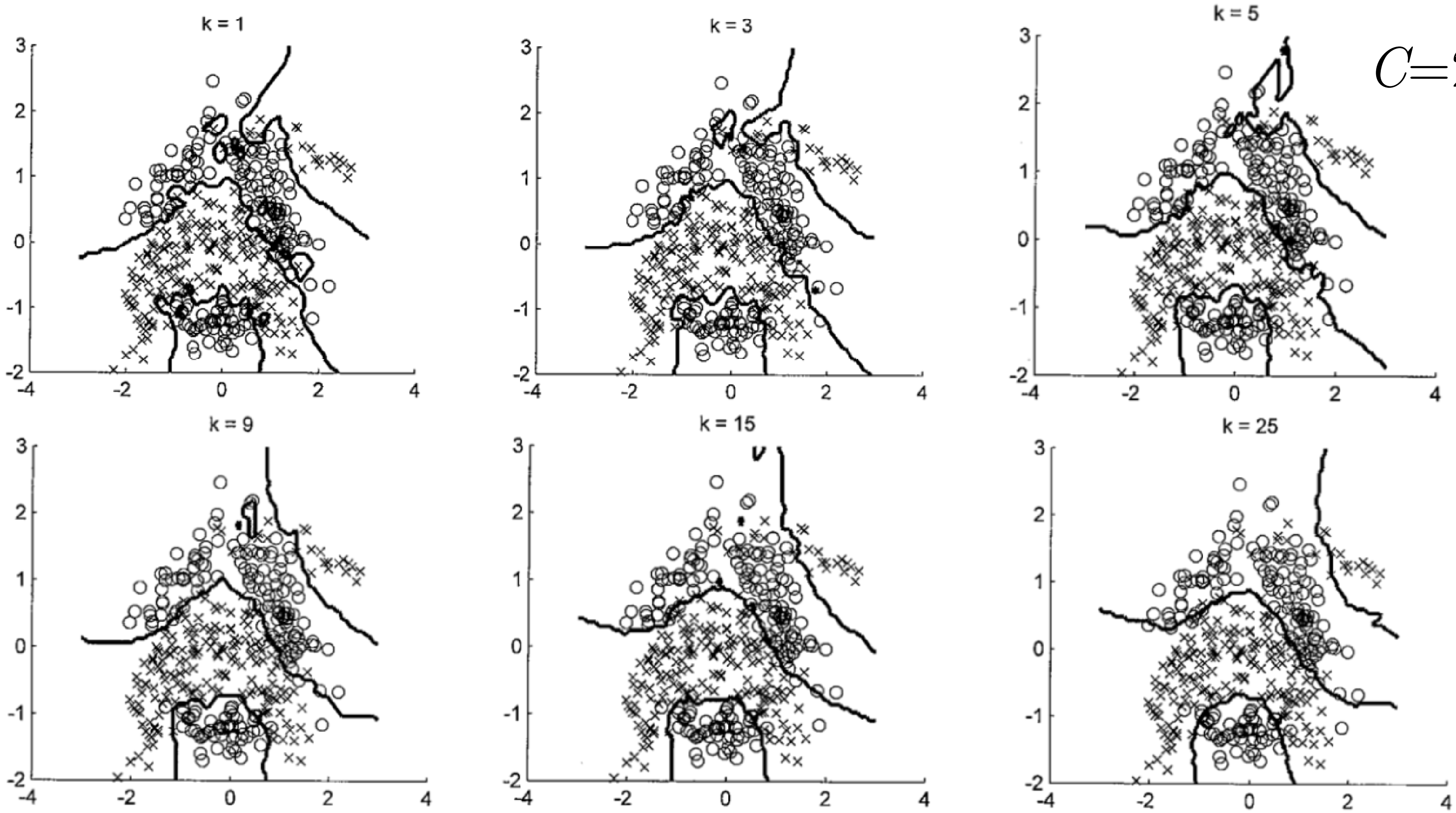


Illustration: kNN for k>1

- kNN classifier: $\mathbf{x}$ gets the majority label of the k closest $\mathbf{x_i}$ in S



C=2 classes

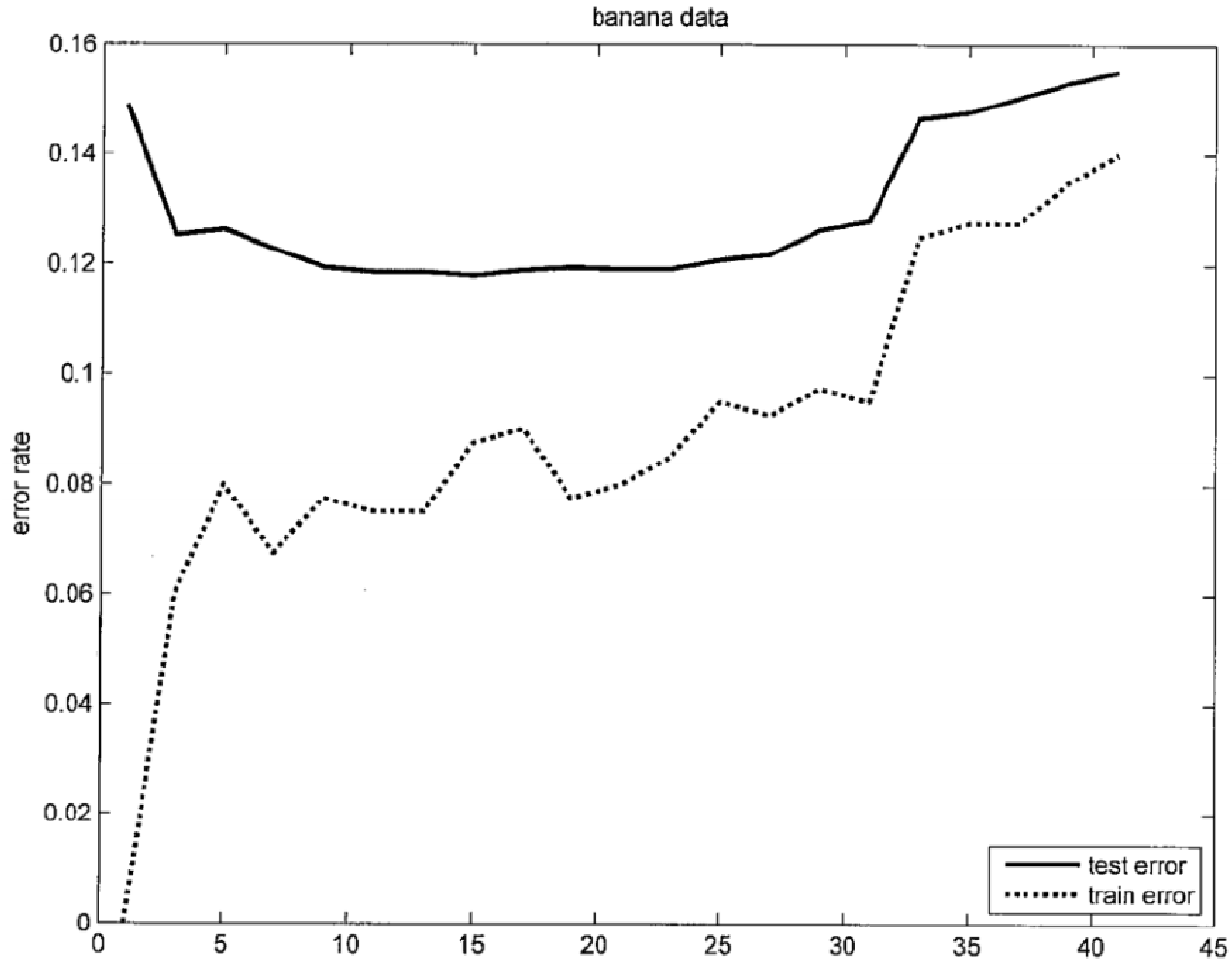
Group exercise

Introduce yourself to the people around you. Form groups of about 3 or 4 people. I may call on groups at random so be prepared.

- Is k-NN discriminative or generative? Fixed complexity or growing complexity? Linear or non-linear?
- How would you expect the k-NN classifier to scale (large d/large n)?
- For what value of k does the k-NN classifier minimize the observed classification error on the training set?
- How do you think k might be chosen for k-NN to do well on a novel $\mathbf{x}$?

The error observed on training data is optimistic

- k is a parameter that affect smoothness of the classifier. Larger k means more smoothness. k controls the tradeoff between underfitting&overfitting.



Matrix representation of the data

- It will be convenient to work with the feature samples as a matrix
- Most aspects of ML are best understood in terms of matrices
  - Feature transformations: standardization, PCA, dimensionality reduction
  - ML procedures: LDA, QDA, SVM, probabilistic graphical models, …
- This is where linear matrix algebra enters the scene!

Matrix algebra: cast of characters

- Actor: generic vector (a feature vector, a single feature over time)
- Actor: Indicator vector
- Actor: Ones vector
- Scene 1: Linear vector spaces and subspaces
- Scene 2: Inner product and outer product of vectors
- Scene 3: Sums of vector outer products=matrices
- Scene 4: Projections and projection matrices
- Scene 5: The feature matrix and response vector for ML

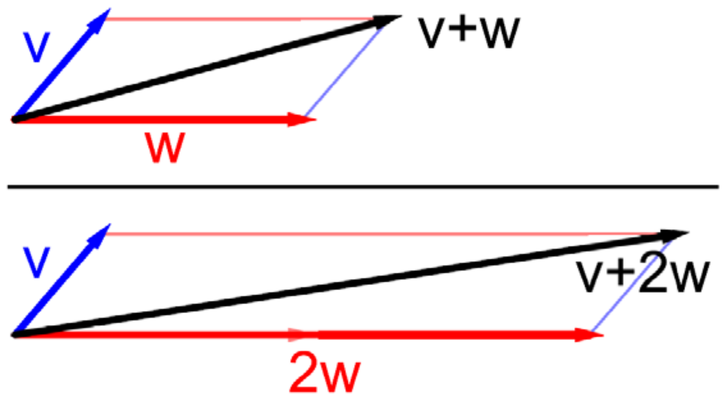
Vectors and linear vector spaces

A linear vector space V is a set of objects $\{\mathbf{v}\}$ (vectors) that is closed under linear combinations over a field F of scalars $\{a\}$

$$\mathbf{v}, \mathbf{w} \in V \quad \Rightarrow \quad a\mathbf{v} + b\mathbf{w} \in V \qquad \text{for any} \quad a, b \in F$$

Two vectors $\mathbf{v}, \mathbf{w}$ are linearly independent if

$$a\mathbf{v} + b\mathbf{w} = 0 \;\; \Rightarrow \;\; a, b = 0$$



Ex: Real Euclidean space of dimension n:

- $V = \mathbb{R}^n$
- $F = \mathbb{R}$

Linear subspaces

- A linear subspace U of V satisfies:
  1. $U \subset V$
  2. U is itself a linear vector space (closed under linear combinations)

- Generating a subspace U from a set of vectors (called a *basis* for U):

Let $\mathbf{u}_1, \ldots \mathbf{u}_d \in V$ be linearly independent $(d \leq n)$. The linear span is

$$\text{span}\{\mathbf{u}_1, \ldots \mathbf{u}_d\} = \text{the set of all linear combination of } \mathbf{u}_1, \ldots \mathbf{u}_d$$

This subspace of V has *dimension d*

Vector inner product = scalar

$\mathbf{w}, \mathbf{x} \in \mathbb{R}^n$ have inner (dot) product, denoted $< \mathbf{w}, \mathbf{x} >$ or $\mathbf{w} \cdot \mathbf{x}$:

$$\mathbf{w}^T\mathbf{x} \quad = \quad [w_1, \ldots, w_n] \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad = \quad \sum_{i=1}^{n} w_i x_i$$

Norm, angle and orthogonality

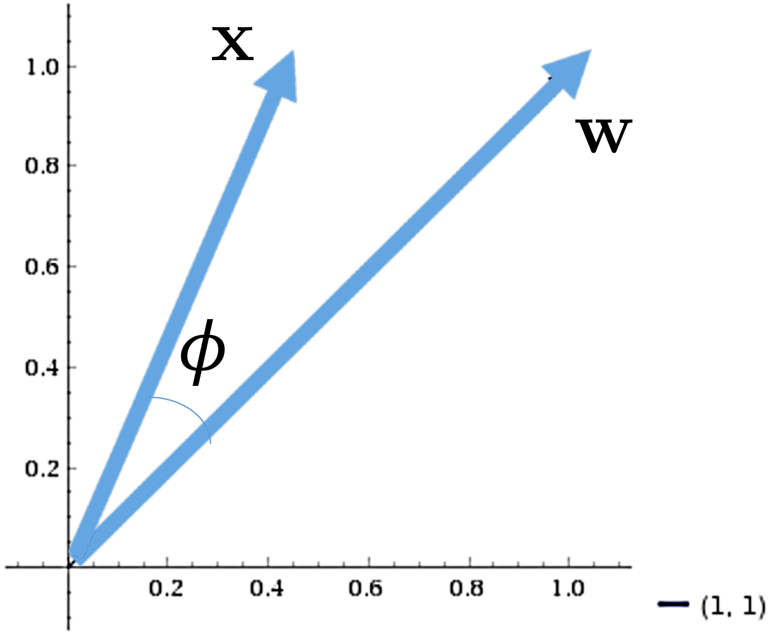
- Euclidean norm of $\mathbf{x}$

$$\|\mathbf{x}\| = \sqrt{x_1^2 \ldots + x_n^2}$$

- Angle between $\mathbf{x}$ and $\mathbf{w}$:

$$\cos(\phi) = \frac{\mathbf{x}^T\mathbf{w}}{\|\mathbf{x}\|\|\mathbf{w}\|}$$

- If $\mathbf{x}^T\mathbf{w} = 0$ then $\mathbf{x} \perp \mathbf{w}$ (orthogonal vectors)



Vector outer product=rank 1 matrix

Outer product of $\mathbf{w} \in \mathbb{R}^d$ and $\mathbf{x} \in \mathbb{R}^n$ is rank one matrix in $\mathbb{R}^{d \times n}$

$$\mathbf{w}\mathbf{x}^T = \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix} [x_1, \ldots, x_n] = [x_1\mathbf{w}, \ldots, x_n\mathbf{w}] = \begin{bmatrix} w_1x_1 & \cdots & w_1x_n \\ \vdots & \ddots & \vdots \\ w_dx_1 & \cdots & w_dx_n \end{bmatrix}$$

- colspan($\mathbf{w}\mathbf{x}^T$)=span$\{\mathbf{w}\} = \{\mathbf{u} : \mathbf{u} = a\mathbf{w}, a \in \mathbb{R}\}$: 1 dimensional space
- Column rank of $\mathbf{w}\mathbf{x}^T$ equals 1

Outer product decomposition of matrices

Summing d linearly independent outer products gives $\mathbf{A} \in \mathbb{R}^{d\times n}$ of rank d

$$\mathbf{A} = \sum_{i=1}^{d} \sigma_i \mathbf{w}_i \mathbf{x}_i^T, \qquad \sigma_i \neq 0$$

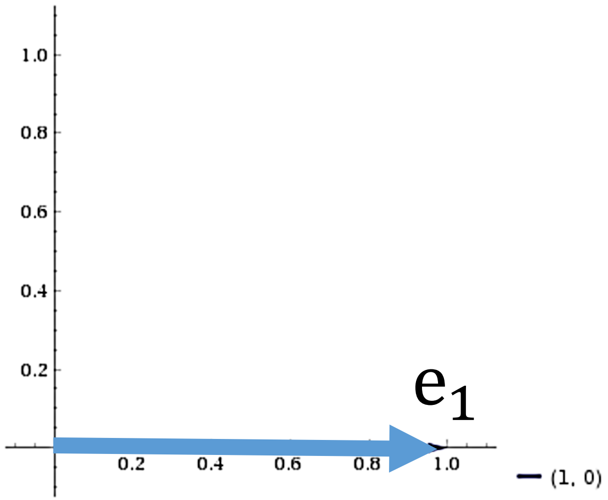
- colspan($\mathbf{A}$) =span$\{\mathbf{w}_1, \ldots, \mathbf{w}_d\}$
- This form is called the singular value decomposition (SVD) of $\mathbf{A}$ when
  - σ_i are singular values: $\sigma_i > 0$
  - $\mathbf{w}_i = \mathbf{u}_i$ are left singular vectors: $\mathbf{u}_i^T \mathbf{u}_j = \delta_{ij}$ and $\|\mathbf{u}_i\| = 1$
  - $\mathbf{x}_i = \mathbf{v}_i$ are right singular vectors: $\mathbf{v}_i^T \mathbf{v}_j = \delta_{ij}$ and $\|\mathbf{v}_i\| = 1$

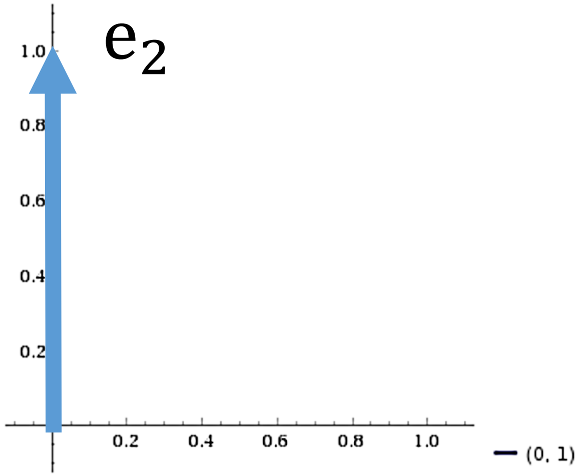
NB: $\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$

Indicator vector

- aka spike vector, selection vector, standard basis vector

$$\mathbf{e}_k = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \qquad \mathbf{e}_k^T = [0, \cdots, 0, 1, 0, \cdots, 0]$$



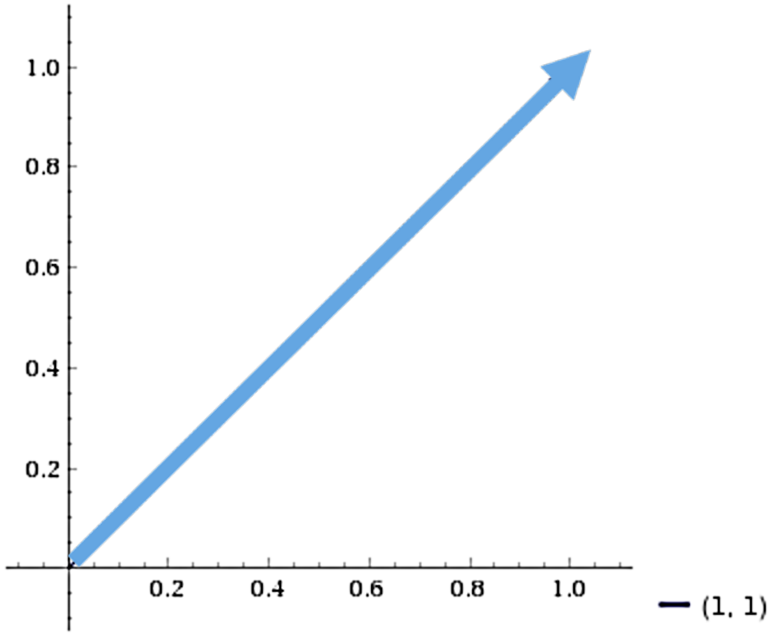


- Norm is 1: $\|\mathbf{e}_k\| = 1$
- Identity matrix: $\mathbf{I} = \sum_{i=1}^{n} \mathbf{e}_i \mathbf{e}_i^T$
- Indicator property: $\mathbf{e}_k^T \mathbf{x} = \mathbf{x}^T \mathbf{e}_k = x_k$

Ones vector

- Aka constant vector, replication vector, aggregation vector

$$\mathbf{1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \qquad \mathbf{1}^T = [1, \cdots, 1, 1, 1, \cdots, 1]$$



- Norm of $\mathbf{1} \in \mathbb{R}^n$ is $\|\mathbf{1}\| = \sqrt{n}$.

- When used in inner product, $\mathbf{1}$ acts as an *aggregator*

$$\mathbf{1}^T\mathbf{x} = \mathbf{x}^T\mathbf{1} = [x_1, \ldots, x_n] \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = \sum_{i=1}^{n} x_i$$

- When used in outer product, $\mathbf{1}$ acts as a *replicator*

$$\mathbf{1}\mathbf{x}^T = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} [x_1, \ldots, x_n] = \begin{bmatrix} x_1 & \cdots & x_n \\ \vdots & \ddots & \vdots \\ x_1 & \cdots & x_n \end{bmatrix} = \begin{bmatrix} \mathbf{x}^T \\ \vdots \\ \mathbf{x}^T \end{bmatrix}$$

$(d \times n)$

$$\mathbf{x}\mathbf{1}^T = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} [1, \ldots, 1] = \begin{bmatrix} x_1 & \cdots & x_1 \\ \vdots & \ddots & \vdots \\ x_n & \cdots & x_n \end{bmatrix} = [\mathbf{x}, \ldots, \mathbf{x}]$$

$(n \times d)$

- When right multiply matrix $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_n] \in \mathbb{R}^{d\times n}$ with $\frac{1}{n}\mathbf{1}$

$$\mathbf{A}\mathbf{1}\tfrac{1}{n} = \tfrac{1}{n}\sum_{j=1}^{n} \mathbf{a}_i \quad (\mathbf{A}\text{'s column average})$$

- Matrix $\mathbf{\Pi} = \frac{1}{n}\mathbf{1}\mathbf{1}^T$ is a matrix of constants with colspace($\mathbf{\Pi}$)=span$\{\mathbf{1}\}$

$$\tfrac{1}{n}\mathbf{1}\mathbf{1}^T = \frac{1}{n}\begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}$$

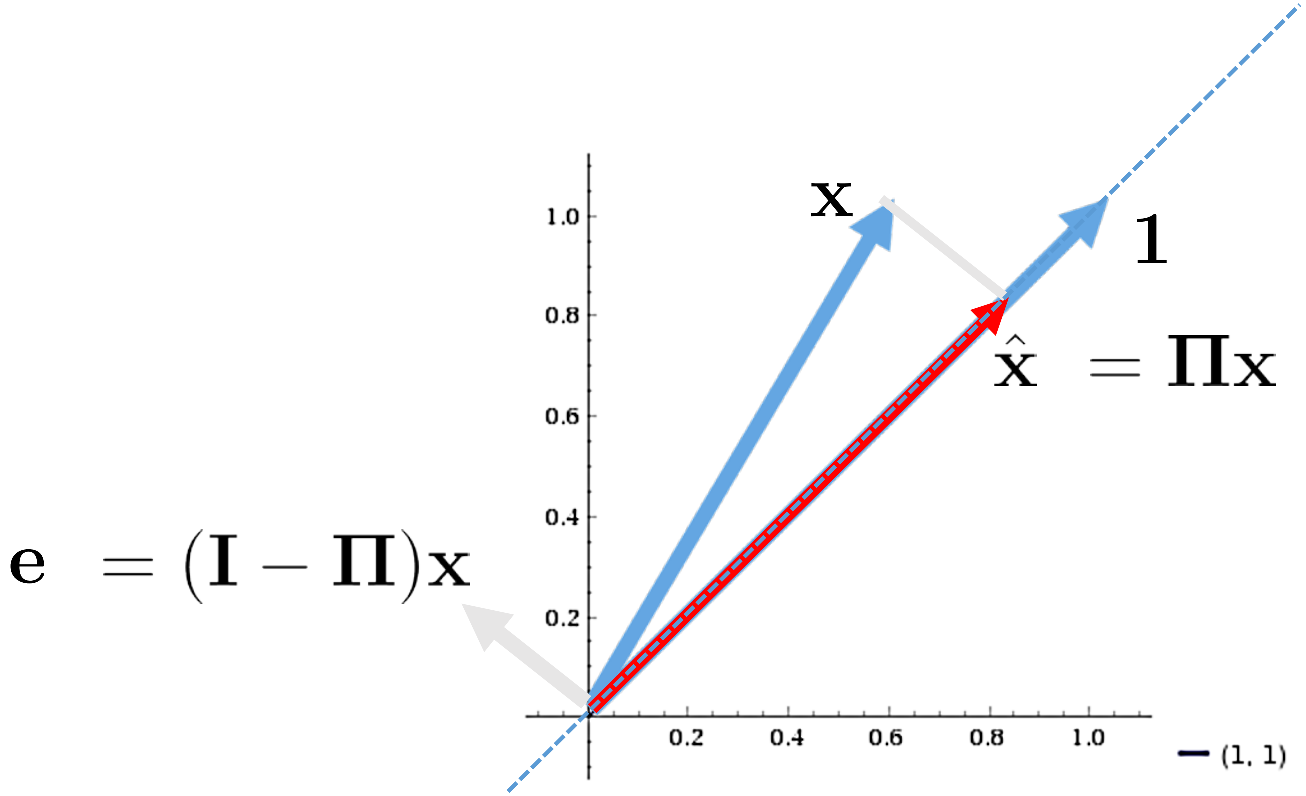
- Action of $\mathbf{\Pi}$ and $\mathbf{I} - \mathbf{\Pi}$ on constant vectors $[a, \ldots, a]^T = \mathbf{1}a$:

$$\mathbf{\Pi}(\mathbf{1}a) = \tfrac{1}{n}\mathbf{1}\underbrace{\mathbf{1}^T\mathbf{1}}_{\|\mathbf{1}\|^2=n} a = \mathbf{1}a, \qquad (\mathbf{I} - \mathbf{\Pi})(\mathbf{1}a) = \mathbf{0}$$

- $\mathbf{\Pi} = \frac{1}{n}\mathbf{1}\mathbf{1}^T$ orthogonally projects vectors $\mathbf{x}$ onto span$\{\mathbf{1}\}$

$$\mathbf{\Pi x} = \hat{\mathbf{x}} \quad \text{and} \qquad \hat{\mathbf{x}} \perp \mathbf{e}, \qquad \text{where} \qquad \mathbf{e} = (\mathbf{x} - \hat{\mathbf{x}}) = (\mathbf{I} - \mathbf{\Pi})\mathbf{x}$$

- $\mathbf{I} - \mathbf{\Pi}$ orthogonally projects onto space orthogonal to span$\{\mathbf{1}\}$.



General projection matrices

- $\mathbf{\Pi} \in \mathbb{R}^{n\times n}$ is a projection matrix that projects orthogonally onto colspan($\mathbf{\Pi}$) if it satisfies

$$(1)\ \mathbf{\Pi}^2 = \mathbf{\Pi}, \qquad (2)\ \mathbf{\Pi}^T = \mathbf{\Pi}$$

- Example: $\mathbf{\Pi x}$ projects $\mathbf{x} \in \mathbb{R}^n$ onto the one dimensional line: span$\{\mathbf{u}\}$ where $\mathbf{u} \in \mathbb{R}^n$:

$$\mathbf{\Pi} = \frac{1}{\|\mathbf{u}\|}\mathbf{u}\mathbf{u}^T$$

Additional reading

- For breezy ML introduction and overview – Murphy Ch 2.
- For slides 11-13 – "kNN classifier:" Sec. 2.3 of [HTF]
- For linear algebra – see the handouts in "files" on canvas
  - A. Hero, Machine Learning Notes– main_EECS545_F2019.pdf
  - Z. Kolter, Linear algebra review and reference – linalgreview.pdf
  - P. Olver, Inner products and norms – Olver_NumericalLinearAlgebra_Notes_inner….pdf
  - I. Savov, Linear algebra explained in four pages – linearAlgebra_4pgs.pdf

References

- [M] Murphy, Machine Learning, a Probabilistic Perspective. MIT, 2012
- [HTF] Hastie, Tibshirani, Friedman, The Elements of Statistical Learning, Springer, 2009.