EECS 545 - Fall 2019 – 220 Chrysler

Machine Learning

Instructor: Alfred Hero

Lecture 1

Course introduction

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

Course information

Canvas website

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

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 next week (outside class)

- Linear algebra and probability (2 2hr sessions)
- Python (2 2hr sessions)

FAQs: Prerequisites

- Are the prerequisites important?
  - Yes. You need linear algebra and probability to understand ML
- Can't I pick them up along the way?
  - Maybe. But it will be easier if you have had previous exposure.
- Can I get an A or an A- if I have weak background in LA&P?
  - Yes. But it will take more work on your part.
- Will there be any review of background material?
  - Yes. We will be reviewing linear algebra and probability.
- When will I know if my background is adequate?
  - When you try the first homework, you will get a good idea.
- How can I get the necessary background?
  - Take a course in linear algebra and/or probability

FAQs: Enrollment and credit

- If I am enrolled in Sec 002 do I need also enroll in Sec 001?
  - Yes. Please use exercise your overrides asap. Sec. 002 will be closed.
- Can I audit/visit this class?
  - Yes, but only if there are enough seats. For-credit students have priority.
  - You will not be able to participate in hwks, exams, or projects.
  - Send me an email to get added as an observer (after Sept. 18)

CSE students who have taken or are taking EECS445 will not get credit for EECS545.

FAQs: Homeworks and exams

- How are homeworks turned in?
  - You must turn them in electronically to Gradescope.
- Are late homeworks accepted?
  - No. But unexpected things happen so we drop your lowest score.
- Can I work on homeworks with my friends in class?
  - Yes. But you must write up and turn-in your own individual work.
- What material will be covered in the midterm exam?
  - Everything up to the last homework you turned in before the exam
  - Materials covered in lectures and notes

FAQs: Projects

- How should I go about finding teammates for my projects?
  - Come to class and get to know your classmates.
  - Use Piazza and other social media to pitch an idea to the entire class and/or to respond to someone elses' pitch.
- What is the format of the project proposal and final project?
  - This will be posted on canvas in a couple of weeks. Will be ≤10pgs.
- How will my project be reviewed?
  - The way that a ML conference does it – you will submit and review proposals and final reports according to criteria to be posted on canvas.
- How will my project be graded?
  - Based on the quality of your proposal, final report, and reviewing.
  - Instructional staff will ultimately decide on quality.

FAQs: Communications

- How should I contact the course instructional staff?
  - Piazza
- What if I don't want to reveal my identity?
  - You can post to Piazza anonymously.
- What if I don't want the whole class to see my question?
  - You can indicate that your post is for instructors only.
- Can I just email the professor instead to answer questions?
  - No. Unless, it is for a personal matter that does not concern GSI's.

FAQs: In the classroom

- Is it important that I attend classes?
  - Yes. This will give you the opportunity to participate.
  - However, if you do miss class, a recording of lecture will be available on canvas.
- Can I use a laptop/tablet/phone during class?
  - Yes, for laptops and tablets.
  - No, for phones. Please turn them off while in class.

Any other questions?

If you sometimes feel lost...

Be patient and pay attention. We will need to develop some seemingly unrelated background material in order to build understanding of ML

"*Experience has shown, and a true philosophy will always show, that a vast, perhaps the larger portion of the truth arises from the seemingly irrelevant.*"

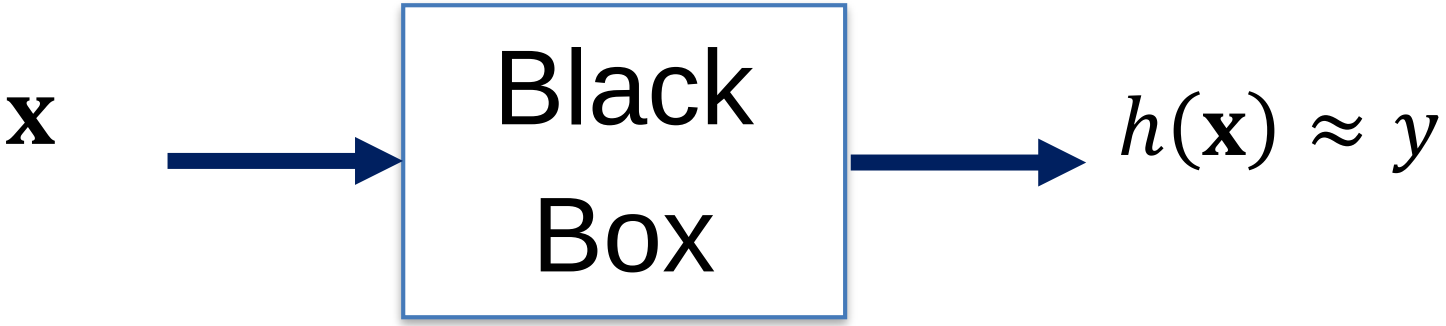
Edgar Allan Poe in *The Mystery of Marie Roget*

Overview of Machine Learning

- The black box paradigm of Machine Learning.
- What is Machine Learning?
- Machine learning pipelines and data ingestion.
- Some mathematical notation.
- Types of ML problems.
- Some nomenclature.
- An example: the kNN classifier

The black box paradigm

- A black box uses "ML" to process an observed input data sample $\mathbf{x}$, producing a prediction $h(\mathbf{x})$ of an unobserved response y



(An input/output map h)

- We sometimes call $\mathbf{x}$ the *predictor* and y the *predictee*

NB: *Explainable AI* & *Interpretable ML* are active research areas trying to break the black box paradigm

What goes into designing the black box?

Artificial Intelligence (CS)

PAC learning theory (CS)

Information theory (EE)

Signal processing (EE)

Algorithms (CS)

Databases (CS)

Reproducibility - workflows (SI)

ML

Probabilistic modeling (Statistics)

Statistical inference (Statistics)

Decision theory (Economics)

Linear algebra (Mathematics)

Optimization (Mathematics)

Learning theory (Psychology)

Neural modeling (Neuroscience)

What is machine learning?

"The term machine learning refers to the automated detection of meaningful patterns in data."

-Shalev-Shwartz and Ben David, 2014 [SSBD]

"Machine Learning is the study of data-driven methods capable of mimicking, understanding and aiding human and biological information processing tasks."

-Barber, 2012 [B]

"...machine learning, a philosophically atheistic approach to statistical inference."

- Efron and Hastie, 2016 [EH]

These articulate 3 basic ML philosophies

Minimalist modeling

L. Valiant

V. Vapnik

“One should solve the classification problem directly and never solve a more general problem as an intermediate step”

-V. Vapnik

Objectivist modeling

G. Box

R. A. Fisher

“All models are wrong, but some are useful.”

-George E. P. Box

Subjectivist modeling

I.J. Good

J. Bayes

“The subjectivist (i.e. Bayesian) states his judgements, whereas the objectivist sweeps them under the carpet….”

-Irving John Good

Probability model: Unspecified | Frequentist principles | Bayesian principles

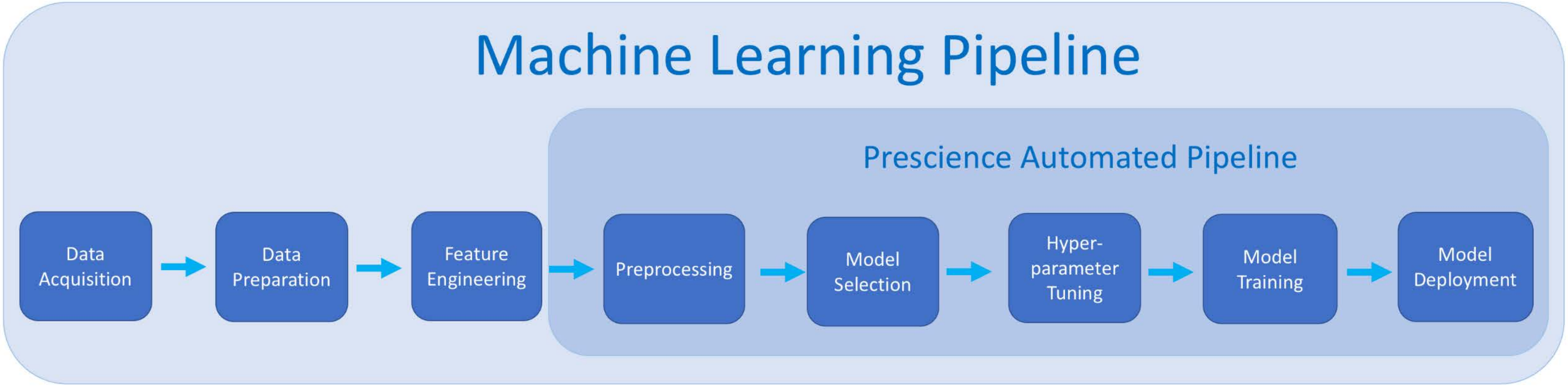
Philosophical approaches to ML

Minimalist	Objectivist	Subjectivist
No model for data distribution **Strong assumptions on h** • [SSBD] Shai Shalev-Shwartz and Shai Ben-David, Understanding Machine Learning: from Theory to Algorithms, Cambridge 2014. • [MAA] Mehryar Mohri, Ameet Talwalkar, Afshin Rostamizadeh, Foundations of Machine Learniing, Oxford 2012.	**Parametric distribution model w/o priors** **Loose assumptions on h** [HTF] Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning, Springer, 2009. • [EH] Brad Efron and Trevor Hastie, Computer Age Statistical Inference, Cambridge, 2016.	**Impose Bayesian prior on model parameters** **No assumptions on learning algorithms** • [B] David Barber, Bayesian Reasoning with Machine Learning, Cambridge, 2012. • [M] Kevin Murphy, Machine Learning, a Probabilistic Perspective. MIT, 2012.

Distribution: **Unspecified** | **Frequentist principles** | **Bayesian principles**

Machine Learning Pipeline

- 1st stage of the ML pipeline: data ingestion=acquisition+preparation
- Data ingestion maps data into real valued variables.
- These variables are then processed to train a mathematical model, designed using methods of optimization, linear algebra and probability.
- A pipeline is shown below (source: Prescience, Inc)



labs.ovh.com

Data ingestion illustration - health data

- **Raw data** *is mapped to real-valued* **Variables**

Clinical adjudication $\in$ {sick, not sick} $\longrightarrow$ $y \in \{0,1\}$

Gene expression $\in$ {RNA abundances} $\longrightarrow$ $\mathbf{x} \in \{\text{vectors in } \mathbb{R}^{d}\}$

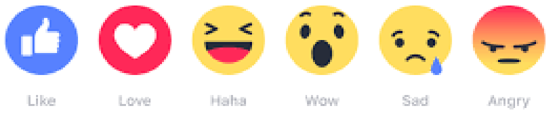
Patients corpus = {  ...  } $\longrightarrow$ $\mathbb{X} \in \{\text{matrices in } \mathbb{R}^{d\times n}\}$

$\mathbf{y} \in \{\text{vectors in } \mathbb{R}^{n}\}$

- d = # genes
- n = # patients

Data ingestion illustration - texting data

- **Raw data** *is mapped to real-valued* **Variables**

Emoticon ∈ {  } $\rightarrow$ $y \in \{1,2,3,4,5,6\}$

Post ∈ { words in text } $\rightarrow$ $\mathbf{x} \in \{\text{word histograms in } \mathbb{R}^{d}\}$

Posts corpus = {  ...  } $\rightarrow$ $\mathbb{X} \in \{\text{matrices in } \mathbb{R}^{d\times n}\}$

$\mathbf{y} \in \{\text{vectors in } \mathbb{R}^{n}\}$

- d = vocabulary size
- n = # posts

Data ingestion illustration - scientific data

- **Raw data** *is mapped to real-valued* **Variables**

Experimental outcome $\in$ {reaction yield} $\longrightarrow$ $y \in$ {scalars in $\mathbb{R}$}

Design parameters $\in$ {concentrations} $\longrightarrow$ $\mathbf{x} \in$ {vectors in $\mathbb{R}^2$ }

Experiments corpus = {  …  } $\longrightarrow$ $\mathbb{X} \in$ { matrices in $\mathbb{R}^{2\times n}$ }

$\mathbf{y} \in$ {vectors in $\mathbb{R}^n$ }

- d = # chemical compounds,
- n = # experiments

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.

A formal model 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 more 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.

Supervised learning

In supervised learning, the learner/user is given labeled training data

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$$

- Thus a fully labeled set $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ is available for training
- Objective: train h to predict output $\boldsymbol{y}$ given a novel input $\mathbf{x}$
- Examples: Classification and regression

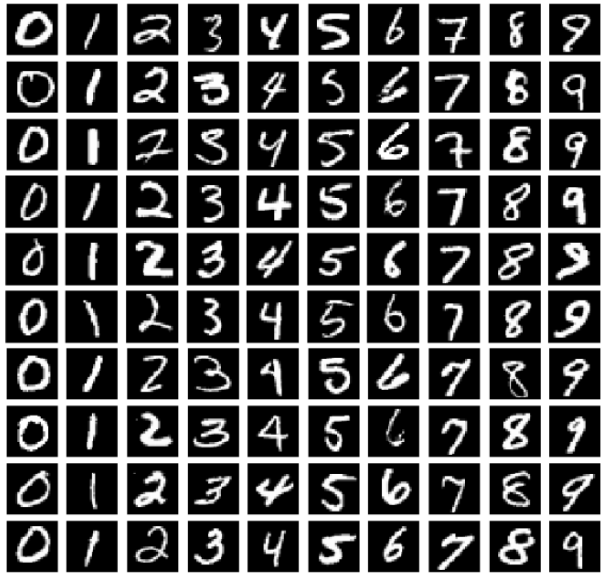
Classification

- Outputs are called labels, which belong to finite set

$$y \in \{1, 2, \ldots, C\}$$

Where C denotes the number of classes.

- Example: handwritten digit recognition:

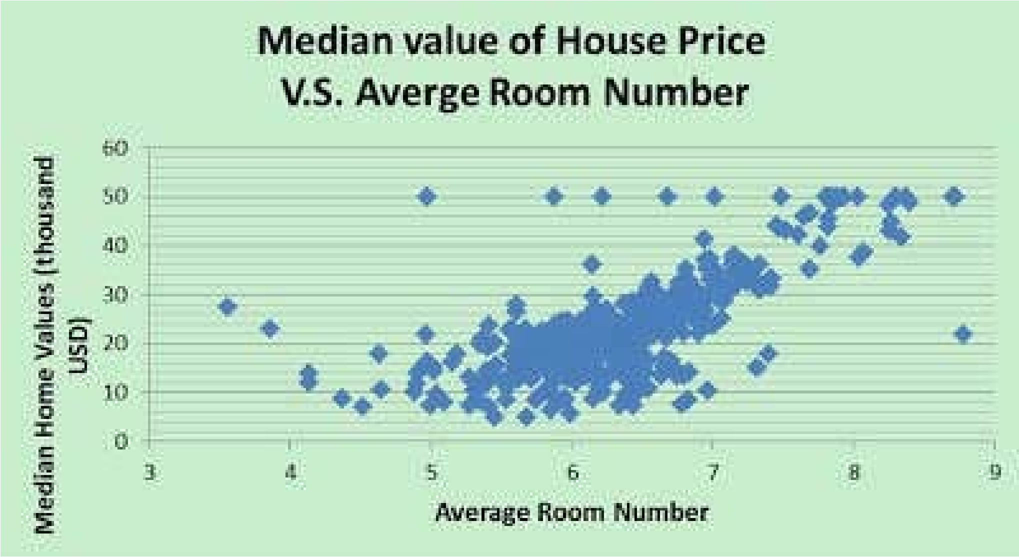


Regression

- Outputs are called responses and are continuous valued

$$y \in (-\infty, \infty) = \mathbb{R}$$

- Example: prediction of home value from its number of rooms



Unsupervised learning

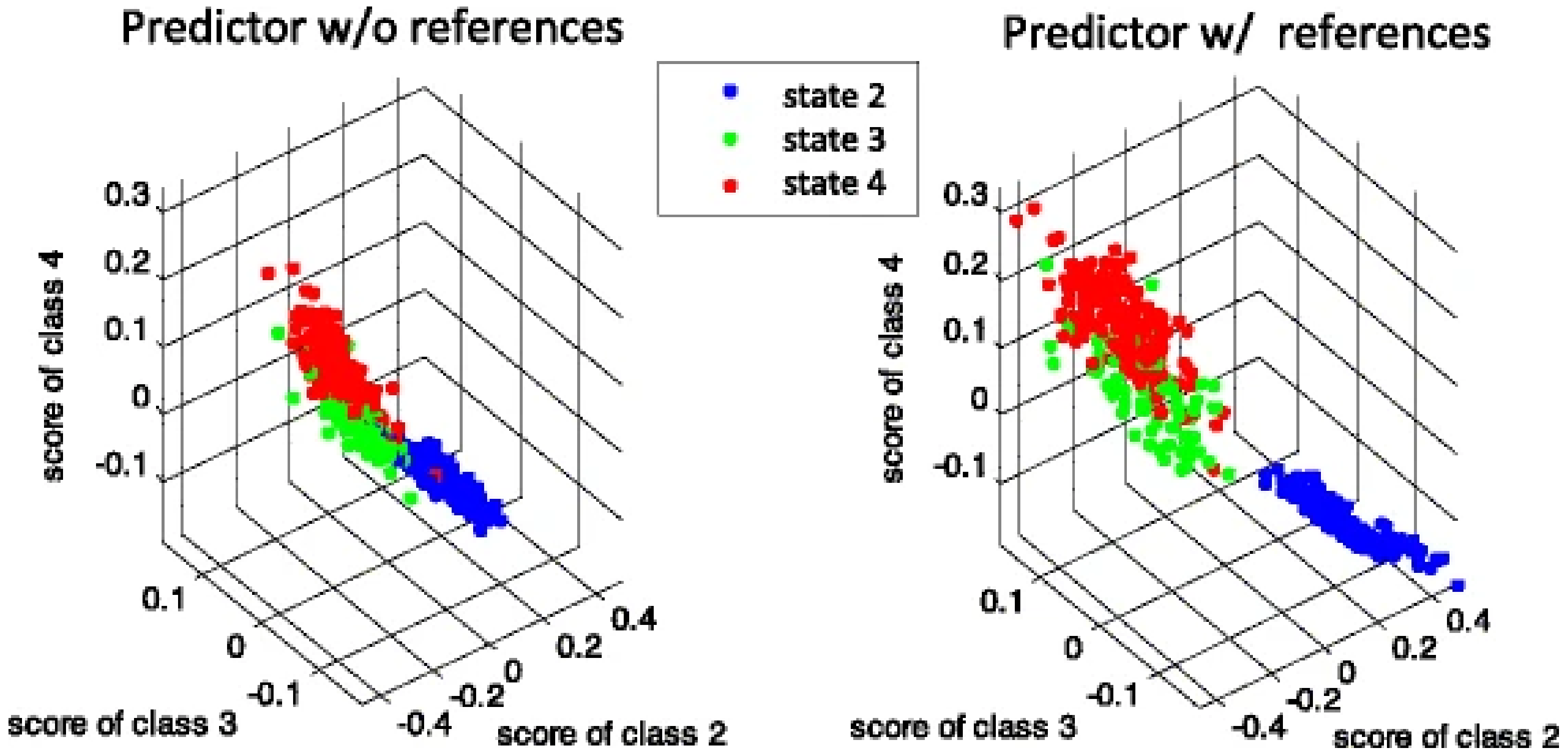
In unsupervised learning the learner is given unlabeled data (no y variables):

$$\mathbf{x}_1, \ldots, \mathbf{x}_n, \qquad \mathbf{x}_i \in \mathbb{R}^d$$

- Only an unlabeled dataset $S = \{\boldsymbol{x}_{\mathrm{i}}\}_{i=\mathbf{1}}^{n}$ is available during training
- Objective: train h to extract properties of $\mathbf{x}$
- Examples:
  - Clustering: do instances in S fall into several distinct clusters?
  - Density estimation: what is the underlying probability density function?
  - Dimensionality reduction: does the data live in lower dimension than d?

Clustering and dimensionality reduction

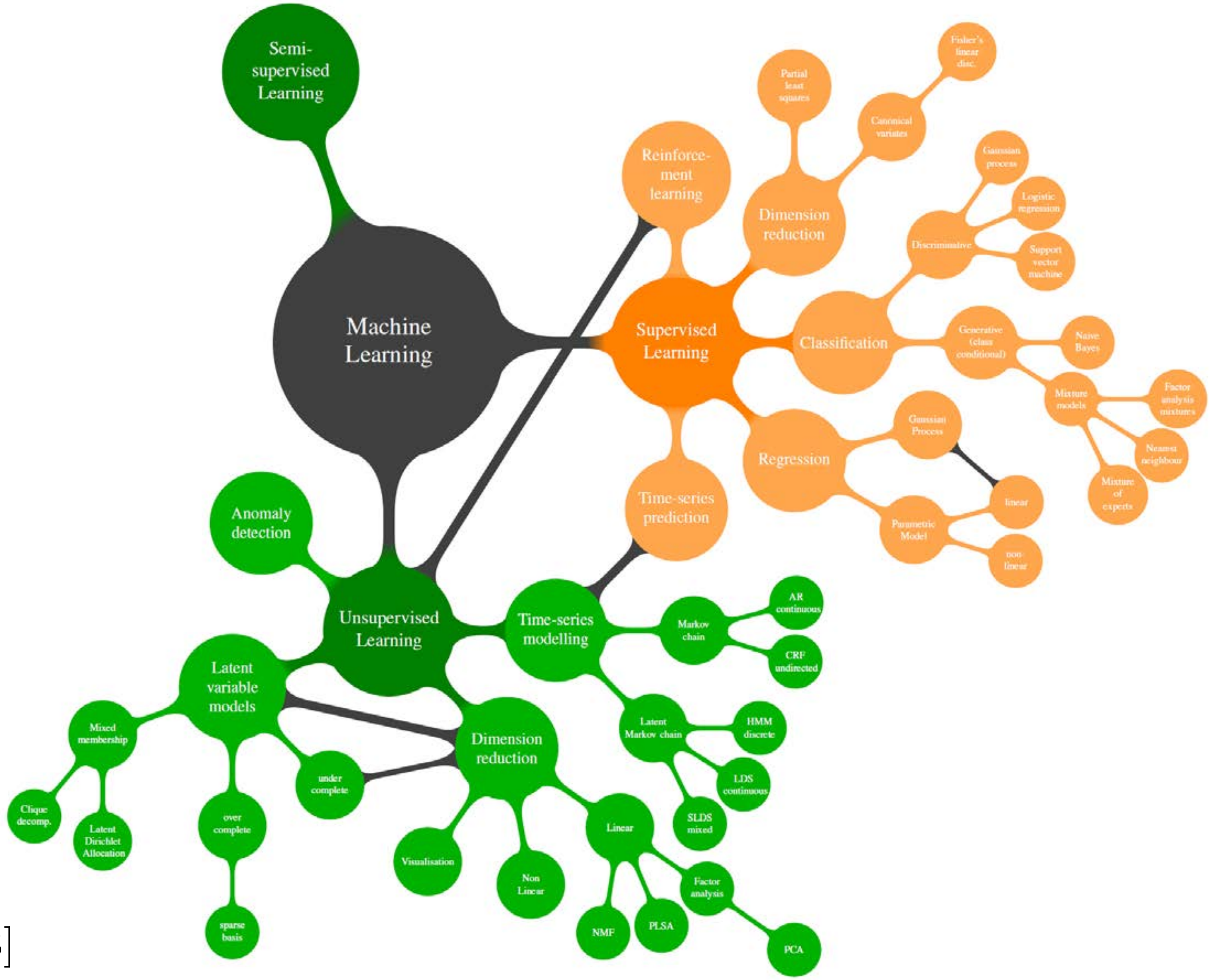
- Projection of samples of blood RNA from d=10,000 to d=3
- Clustering of samples into 3 categories of health outcome (RGB)



Many types of ML tasks

- **Classification**: learn to classify the label of a variable: $X \in \mathbb{R}^{\mathrm{d}}$, $Y \in \{1, \ldots, C\}$
- **Regression**: learn to estimate the value of a unseen covariate: $X \in \mathbb{R}^{\mathrm{d}}, Y \in \mathbb{R}$
- **Forecasting**: learn to predict future values of a time sequence: *ibid*
- **Ranking**: learn to compare variables, e.g., preferences: $X \in \mathbb{R}^{\mathrm{d}} \times \mathbb{R}^{\mathrm{d}}$, $Y \in \{0,1\}$
- **Clustering**: learn to separate subpopulations in data: $X \in \mathbb{R}^{\mathrm{d}}$, $Y \in \{1,2, \ldots C\}$
- **Selection**: learn to select the most important variables: $X \in \mathbb{R}^{\mathrm{d}}$, $Y \in \{0,1\}^d$
- **Anomaly Detection**: learn to detect strange sample values: $X \in \mathbb{R}^{\mathrm{d}}$, $Y \in \{0,1\}$
- **Unmixing**: learn to unmix multiple signals, e.g.,cocktail party: $X \in \mathbb{R}^{\mathrm{d}}, Y \in \mathbb{R}^{2d}$
- **Imputation**: learn to fill-in missing information in a table: $X \in \mathbb{R}^{\mathrm{d}}, Y \in \mathbb{R}^d$
- **Denoising**: learn to remove noise, e.g., from an image: $X \in \mathbb{R}^{\mathrm{d}}, Y \in \mathbb{R}^d$
- **Bounding**: learn to estimate best achievable ML performance: $X \in \mathbb{R}^{\mathrm{d}}, Y \in \mathbb{R}$
- …

Types of ML visualized as an ecosystem



Source: Barber Part III [B]

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.

Coverage of this course

- Fundamentals of Machine Learning
- Derivation of algorithms from first principles
  - Linear algebra, computation, optimization, probability
- Discussion of pervasive phenomena in ML
  - Overfitting and generalization error
  - Regularization against inadequate number of samples
  - Slow convergence
- Exposure to modern challenges and applications

The k Nearest Neighbor (kNN) classifier

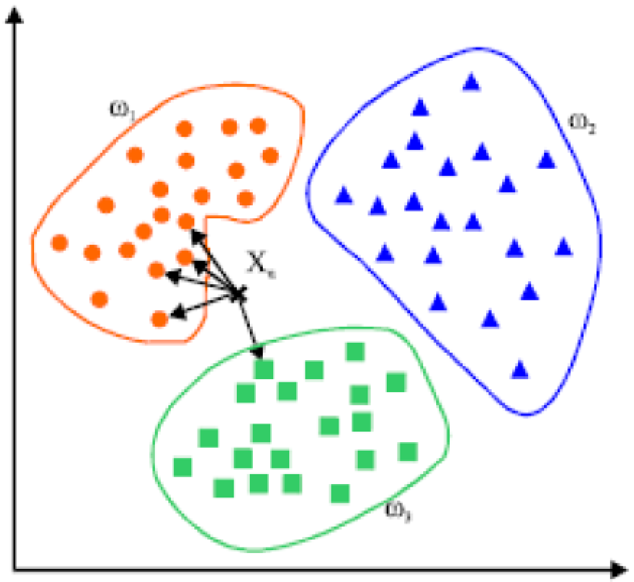
- Given labeled training data $S = \{\mathbf{x}_i, y_i\}_{i=1}^n$
- For any out-of-sample data point $\mathbf{x}_* \notin S$

C=3 classes

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)



kNN classifier

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 $\mathbf{x_i}$

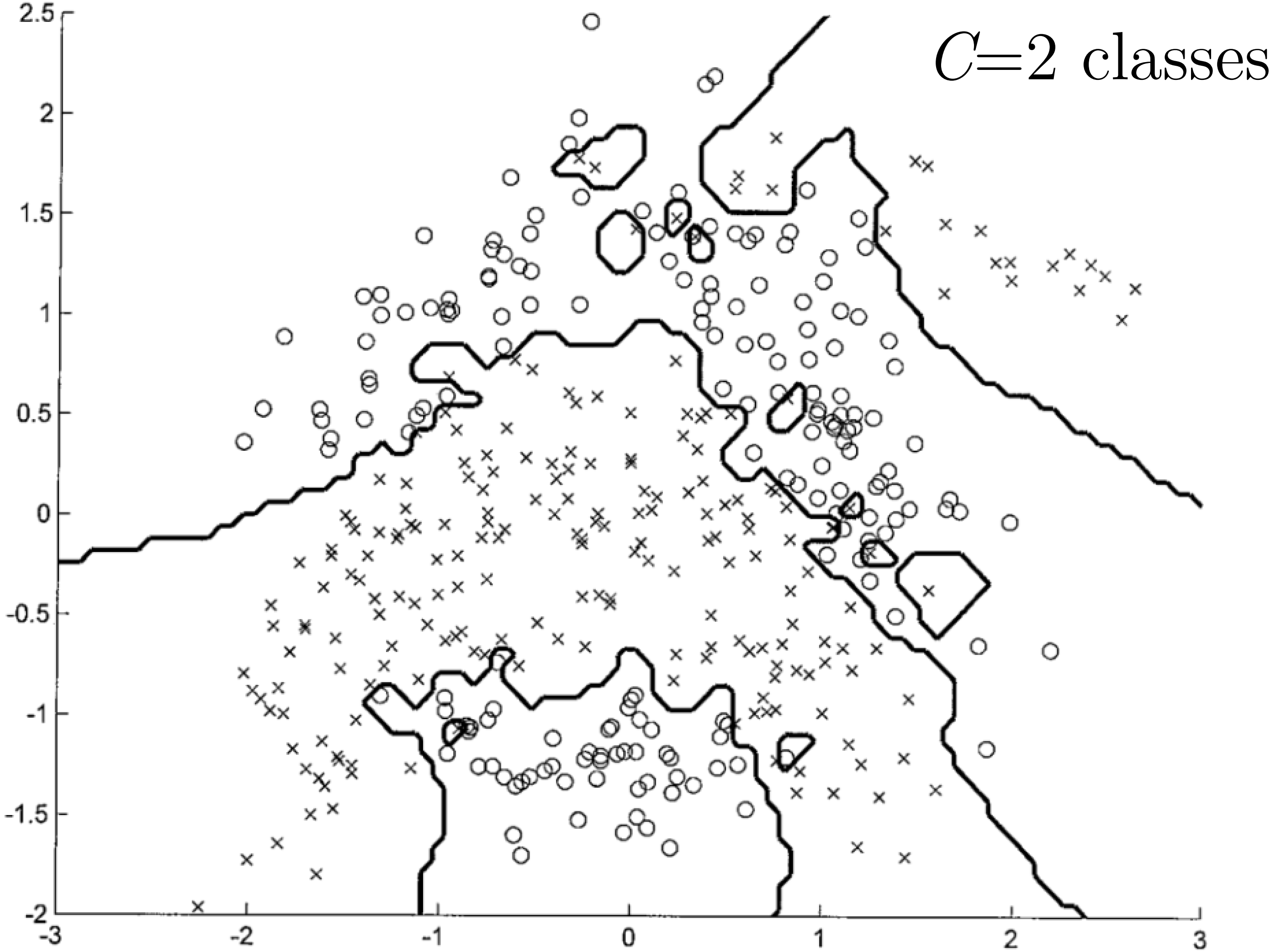
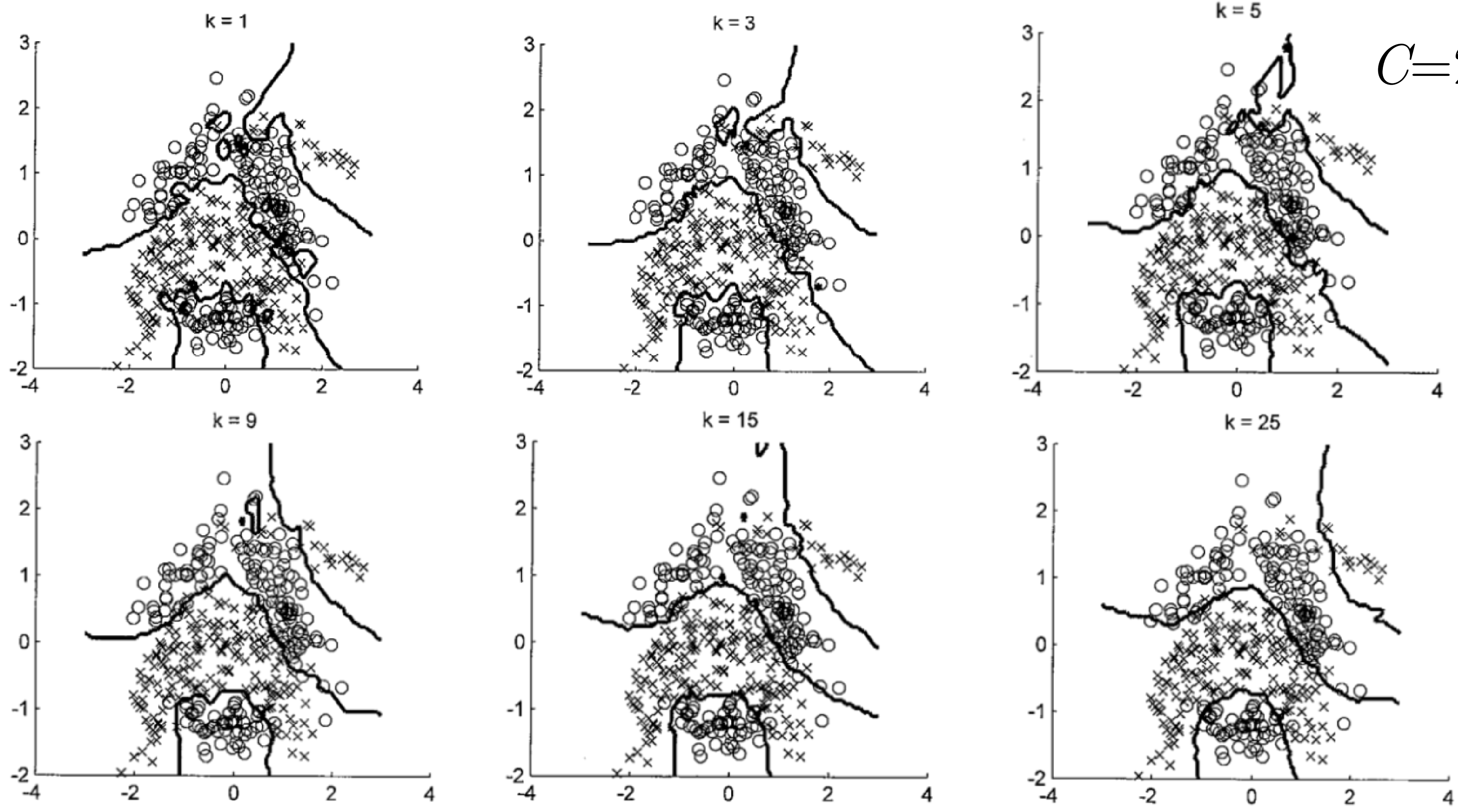


Illustration: kNN for k>1

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



C=2 classes

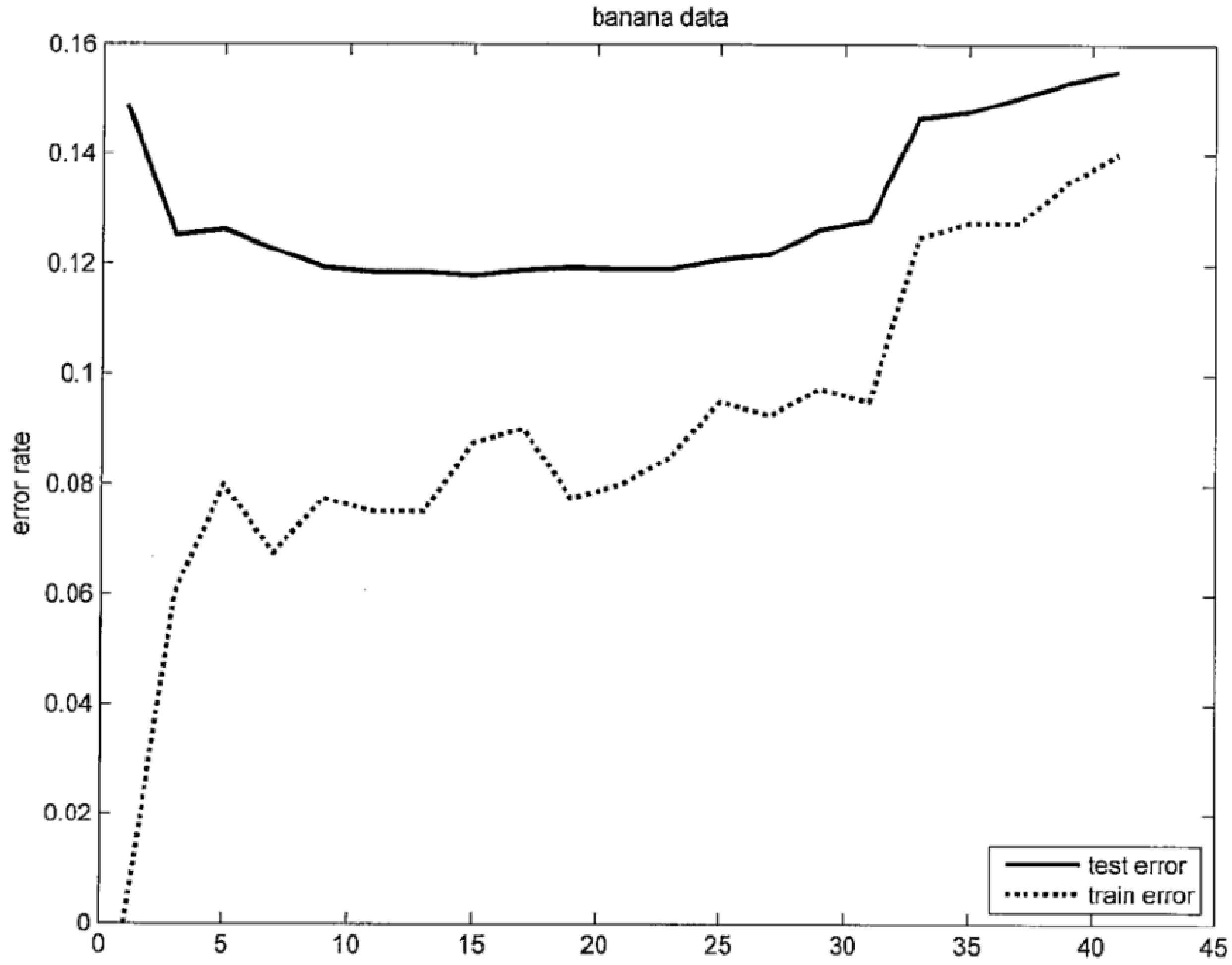
Group exercise

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.



Additional reading

- For breezy ML introduction and overview – Murphy Ch 2.
- For slides 25-26 – "Basic mathematical framework:" Ch 2.1 of [SSBD]
- For slides 37-39 – "kNN classifier:" Sec. 2.3 of [HTF]

References

- [M] Murphy, Machine Learning, a Probabilistic Perspective. MIT, 2012
- [HTF] Hastie, Tibshirani, Friedman, The Elements of Statistical Learning, Springer, 2009.
- [SSBD]Shalev-Shwartz and Ben-David, Understanding Machine Learning: from Theory to Algorithms, Cambridge 2014.