Statistical Learning Theory (SLT)

CS 6316 – Machine Learning
Fall 2017

OUTLINE

- Overview
- Inductive Learning Problem Setting
- Keep-It-Direct Principle
- ERM
- VC-dimension

Overview

Objectives and Overview

Problems with philosophical approaches

- Lack quantitative description/ characterization of ideas
- Often no real predictive power (as in Natural Sciences)
- Often no agreement on basic definitions/ concepts (as in Natural Sciences)

- Goal:
  - To introduce Predictive Learning as a scientific discipline

History and Overview

- SLT is tied closely with **VC-theory** (Vapnik-Chervonenkis)
- Theory for estimating dependences from finite samples (predictive learning setting)
- Based on the *risk minimization* approach
- All main results originally developed in 1970s
- Recent renewed interest due to practical success of Support Vector Machines (SVM)

History and Overview

Main Conceptual Contributions:

- Distinction between problem setting, inductive principle, and learning algorithms
- Direct approach to estimation with finite data ("Keep It Direct" principle)
- Math analysis of ERM (standard inductive setting)
- Two factors responsible for generalization:
  - Empirical risk (fitting error)
  - Complexity (capacity) of approximating functions

Importance of VC-theory

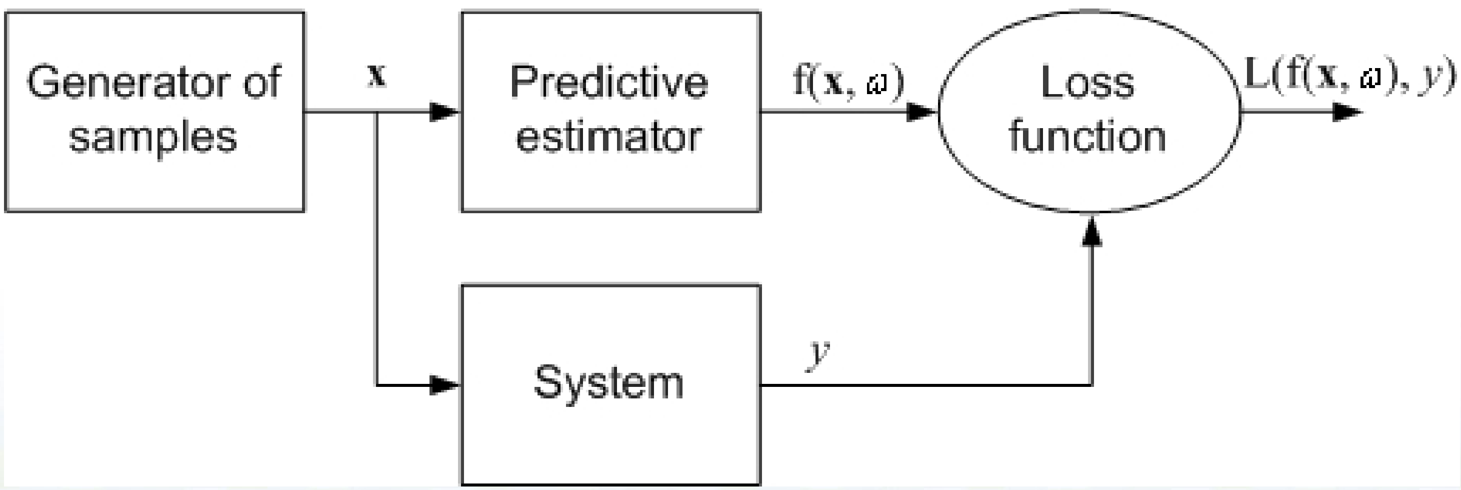
- Math: Under what general conditions the ERM approach leads to (good) generalization
- New approach to induction:
  - Predictive vs. generative modeling (in classical statistics)
- Connection to philosophy of science:
  - VC-theory developed for binary classification (pattern recognition) ~ the simplest generalization problem
  - Natural sciences: from observations to scientific law → VC-theoretical results can be interpreted using general philosophical principles of **induction**, and vice versa

Inductive Learning Problem Setting

Recall

Inductive Learning Setting

- The learning machine observes samples $(\boldsymbol{x}, y)$, and returns an estimated response $\hat{y} = f(\boldsymbol{x}, w)$
- Two modes of inference: identification vs imitation
- Risk: $\int Loss\big(y, f(\boldsymbol{x}, w)\big)\, dP(\boldsymbol{x}, y) \rightarrow min$



Recall

The Problem of Inductive Learning

- *Given:* finite training samples $\mathbf{Z}=\{(\mathbf{x}_i, y_i), i=1,2,\ldots n\}$ choose from a given set of functions $f(\mathbf{x}, \mathbf{w})$ the one that *approximates best* the true output (in the sense of risk minimization)

Concepts and Terminology

- approximating functions $f(\mathbf{x}, \mathbf{w})$
- (non-negative) loss function $L(f(\mathbf{x}, \mathbf{w}),\mathbf{y})$
- expected risk functional $R(\mathbf{Z},\mathbf{w})$

Goal: find the function $f(\mathbf{x}, \mathbf{w}_o)$ *minimizing* $R(\mathbf{Z},\mathbf{w})$ when the joint distribution $P(\mathbf{x},y)$ is *unknown*

Empirical Risk Minimization

- ERM principle in model-based learning
  - Model parameterization: $f(x, w)$
  - Loss function: $L(f(x, w), y)$
  - Estimate risk from data: $R_{emp}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} L(f(\mathbf{x}_i, \mathbf{w}), y_i)$
  - Choose $\mathbf{w}^*$ that minimizes R_{emp}

- Statistical Learning Theory developed from the theoretical analysis of ERM principle under finite sample settings

Probabilistic Modeling vs ERM

Given training examples $(\mathbf{x},y)$ sampled from unknown $P(\mathbf{x},y)$

Probabilistic Modeling

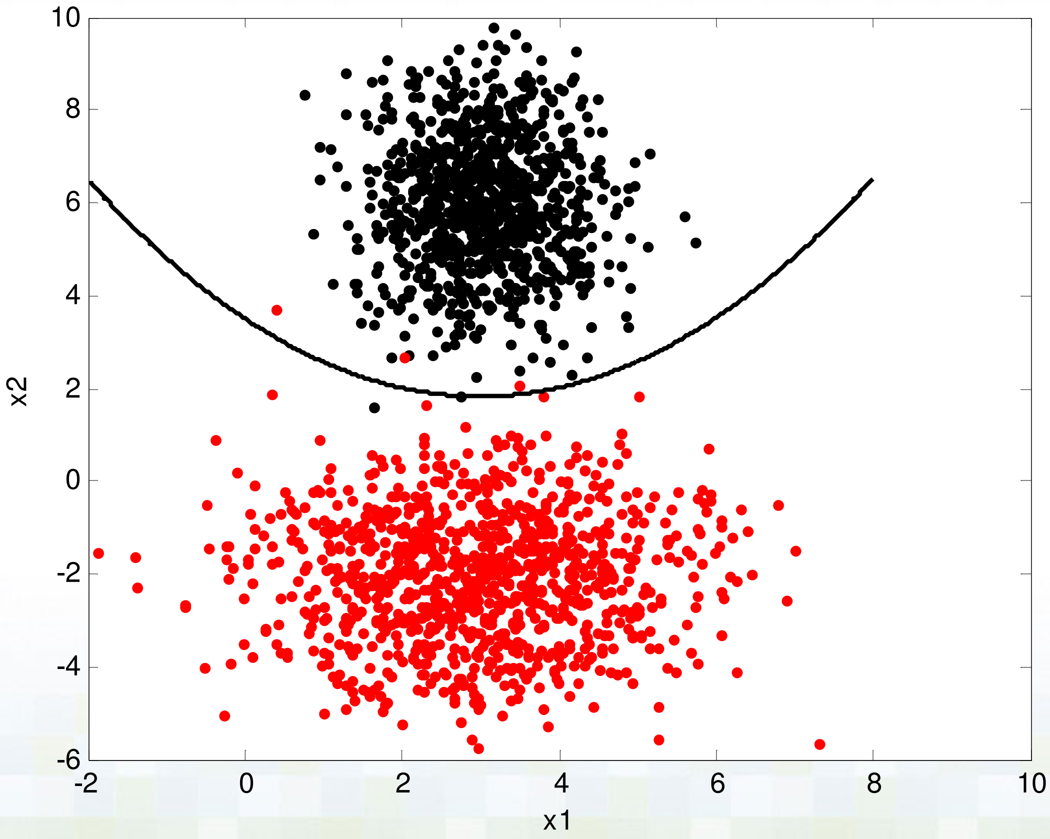
1. Make assumptions about the parametric form of $P(\mathbf{x},y)$.
2. Estimate the parameters $P(\mathbf{x},y)$ of from the training data.
3. Construct optimal decision rule from estimated probabilistic model and given misclassification costs.

Empirical Risk Minimization Modeling

1. Make assumptions about parameterization of admissible decision functions $f(\mathbf{x},\omega)$.
2. For each admissible model, estimate empirical risk (classification error) for the training data.
3. Select the classifier (decision function) providing smallest empirical risk.

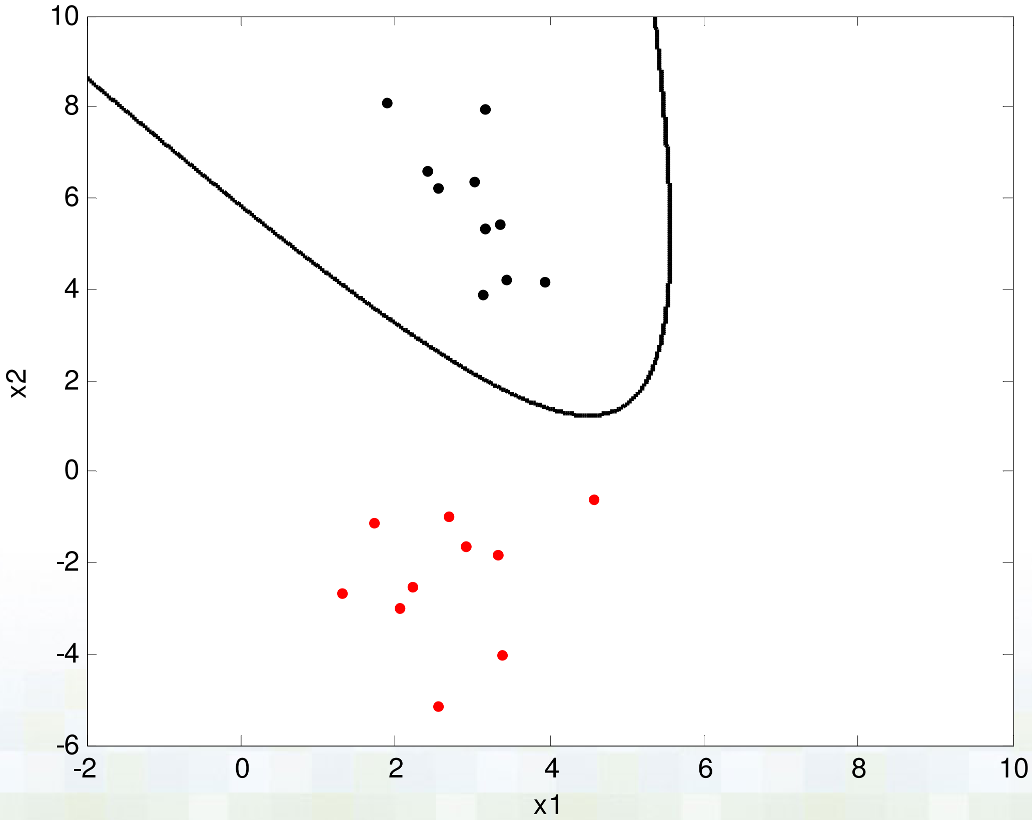
Probabilistic Modeling vs ERM: Example

- If we knew the class distribution → optimal decision boundary



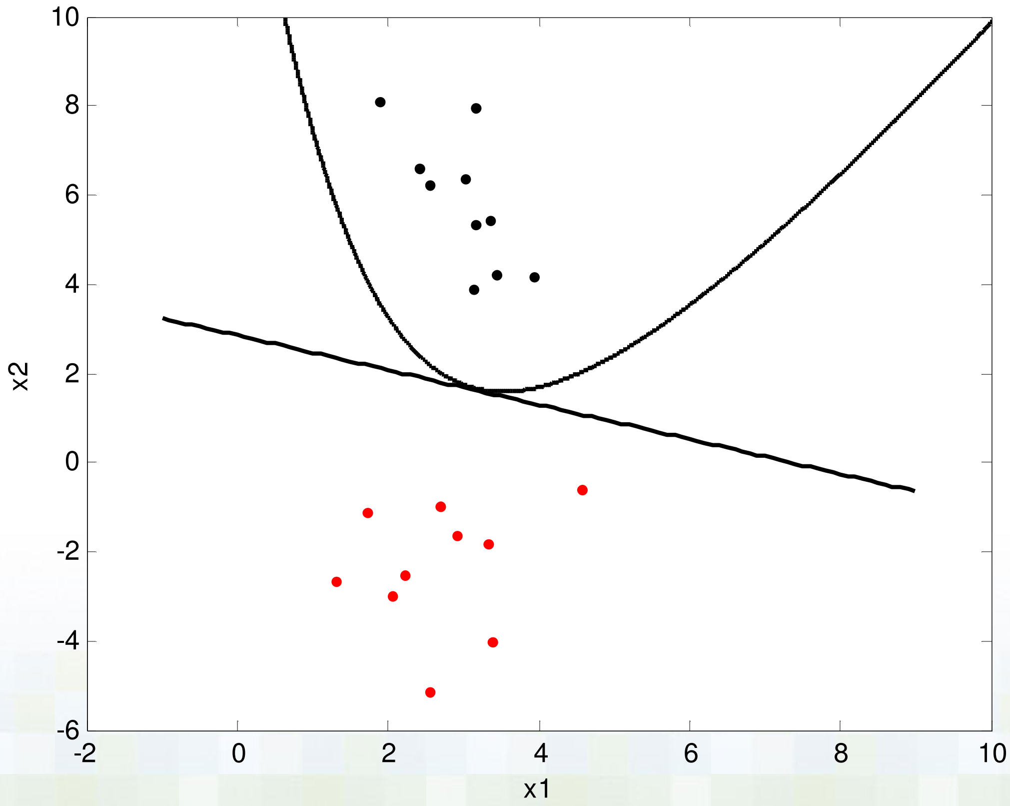
Probabilistic Approach

- Estimate parameters of Gaussian class distributions, and plug them into quadratic decision boundary



ERM Approach

- Quadratic and linear decision boundary estimated via minimization of squared loss



Estimation of Multivariate Functions

- Is it possible to estimate a function from finite data?

- Simplified problem:
  - Estimation of unknown continuous function from noise-free samples

Estimation of Multivariate Functions

- Many results from function approximation theory:
  - To estimate accurately a d-dimensional function one needs $O(n^d)$ data points
  - For example, if **3** points are needed to estimate 2-nd order polynomial for d=1, then 3^{10} points are needed to estimate 2-nd order polynomial in 10-dimensional space
  - Similar results in signal processing

Estimation of Multivariate Functions

- NEVER ENOUGH data points to estimate multivariate functions in most practical applications (image recognition, genomics, etc…)

- For multivariate function estimation, the number of free parameters increases *exponentially* with problem *dimensionality*
(the Curse of Dimensionality)

Curse of Dimensionality in ML

- In machine learning problems that involve learning a "state-of-nature" (maybe an *infinite distribution*) from a finite number of data samples in a high-dimensional feature space with each feature having a number of possible values, an **enormous** amount of training data is required to ensure that *there are several samples with each combination of values*
- With a *fixed* number of **training** samples, ***the predictive power reduces as the dimensionality increases***

Keep It Direct Principle

Keep-It-Direct Principle

- The goal of learning is generalization rather than estimation of true function (*system identification*)

$$\int Loss\big(y, f(\boldsymbol{x}, w)\big)\, dP(\boldsymbol{x}, y) \rightarrow min$$

- **Keep-It-Direct** Principle (Vapnik, 1995)
  - Do not solve an estimation problem of interest by solving a more general (harder) problem as an intermediate step

Keep-It-Direct Principle

- Good predictive model reflects some properties of unknown distribution $P(\mathbf{x},y)$
- Since model estimation with **finite data** is ill-posed, one should never try to solve a more general problem than required by given application

  → Importance of formalizing application requirements via *appropriate learning formulation*

- The goal of prediction (1) is different (less demanding) than the goal of estimating the true target function (2) everywhere in the input space.
- The curse of dimensionality applies to system identification setting (2), but may not hold under predictive setting (1)
- Both settings coincide if the input distribution is uniform (i.e., in signal and image denoising applications)

Philosophical Interpretation of KID

- Interpretation of predictive models
  - Realism ~ objective truth (hidden in Nature)
  - Instrumentalism ~ creation of human mind (imposed on the data) – favored by KID
  - Objective Evaluation still possible (via prediction risk reflecting application needs) → Natural Science

Methodological implications

- Importance of good learning formulations (asking the 'right question')
- Accounts for 80% of success in applications

Analysis of ERM

Empirical Risk Minimization

VC-theory has 4 parts:

1. Analysis of consistency/convergence of ERM

$$R_{emp}(\omega) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i, \omega)) \rightarrow \min$$

2. Generalization bounds
3. Inductive principles (for finite samples)
4. Constructive methods (learning algorithms) for implementing (3)

NOTE: (1) → (2) → (3) → (4)

Consistency/Convergence of ERM

- Empirical Risk known but Expected Risk unknown
- **Asymptotic consistency requirement**:

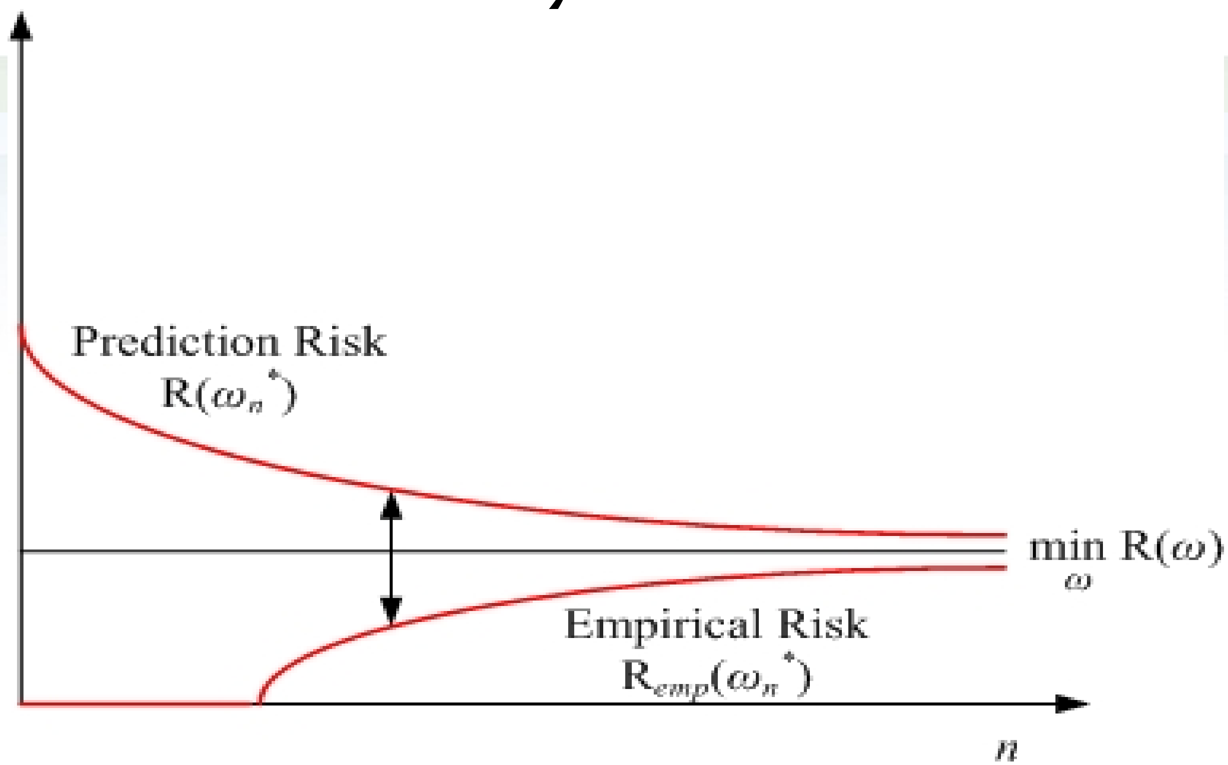
  under what (general) conditions models providing *min Empirical Risk* will also provide *min Prediction Risk*, when the number of samples grows large?

- **Why asymptotic analysis** is needed?

  - helps to develop useful concepts

  - necessary and sufficient conditions ensure that VC-theory is general and can not be improved

Consistency of ERM



- Convergence of empirical risk $R_{emp}(\omega)$ to expected risk $R(\omega)$ *does not imply consistency* of ERM
- Models estimated via ERM (w*) are *always biased estimates* of the functions minimizing true risk:

$$R_{emp}(\omega_n^*) < R(\omega_n^*)$$

Conditions for Consistency of ERM

- **Main insight**: consistency is not possible without restricting the set of possible models

- Consider binary decision functions (~ classification)
- How to measure their flexibility ~ ability to 'explain'/fit available data (for binary classification)?

- This complexity index for indicator functions:
  - is independent of unknown data distribution;
  - measures the ***capacity*** of a set of possible models, rather than characteristics of the 'true model'

VC-dimension

The Vapnik-Chervonekis Dimension

Rules of the *Game*

- Let's start with the representation of a learning problem
- Assume we are looking at a binary classification task with 2 labels: "+" and "–"
- The data points are plotted in a n-dimensional space,
- Classification: finding a *surface* that has only points with the "+" label on one side of it, and points with "–" labels on the other side
- It is known which side has which label

Looking for the Classifier

Why do we need this arrangement?

- When a **new data point is introduced**, the task is to:
  - Find out which side of this surface it falls on
  - Declare the label of the new data point to be the label for this side

How would you look for this separating surface or *classifier*?

- Try *every* possible surface available … ?

Is there a scientific manner in which you can narrow down the search?

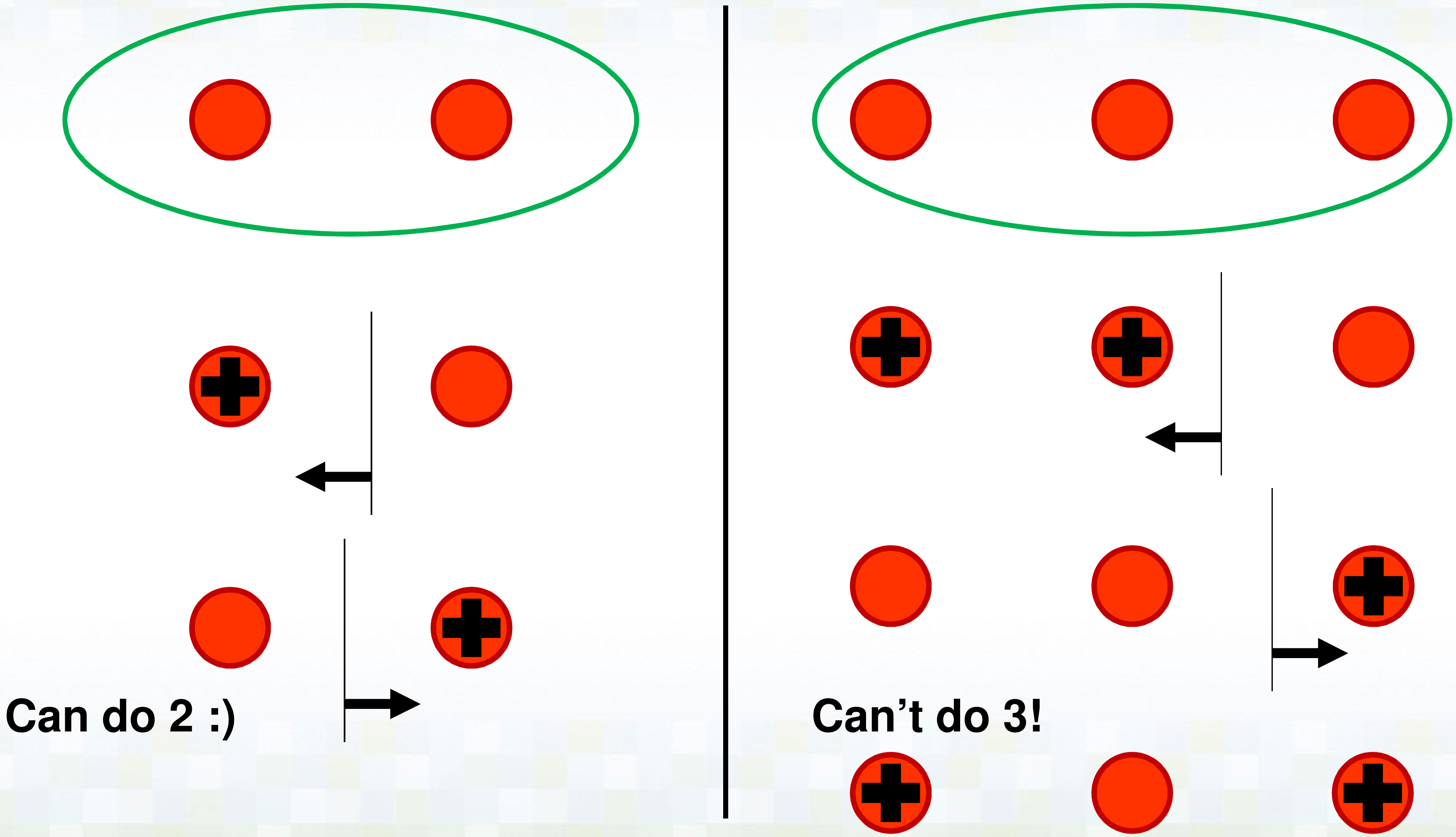
Before VC dimension definition, let's discuss…

Continuous hypothesis Spaces

- $|H|$ = infinity
- Infinite variance??

- As with decision trees (we'll study these soon), only care about the maximum number of points that can be classified exactly!

How many points can a linear boundary classify exactly? (1-D)



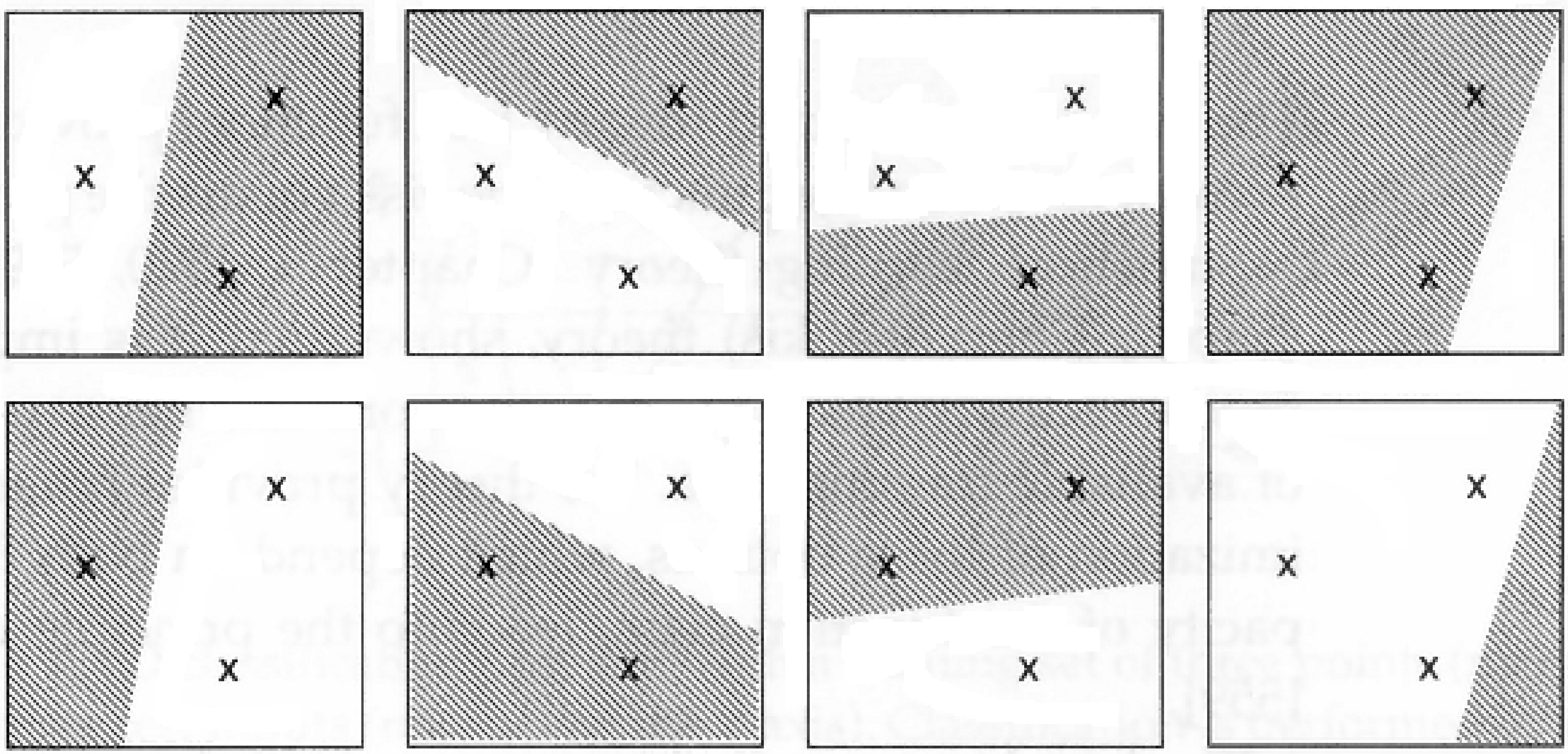
SORRY! :(

How many points can a linear boundary classify exactly? (2-D)

- We can do 2 (see previous)
- **Can we do 3?**

How many points can a linear boundary classify exactly? (2-D)

- We can do 2 (see previous)
- Can we do 3? **– YES!**

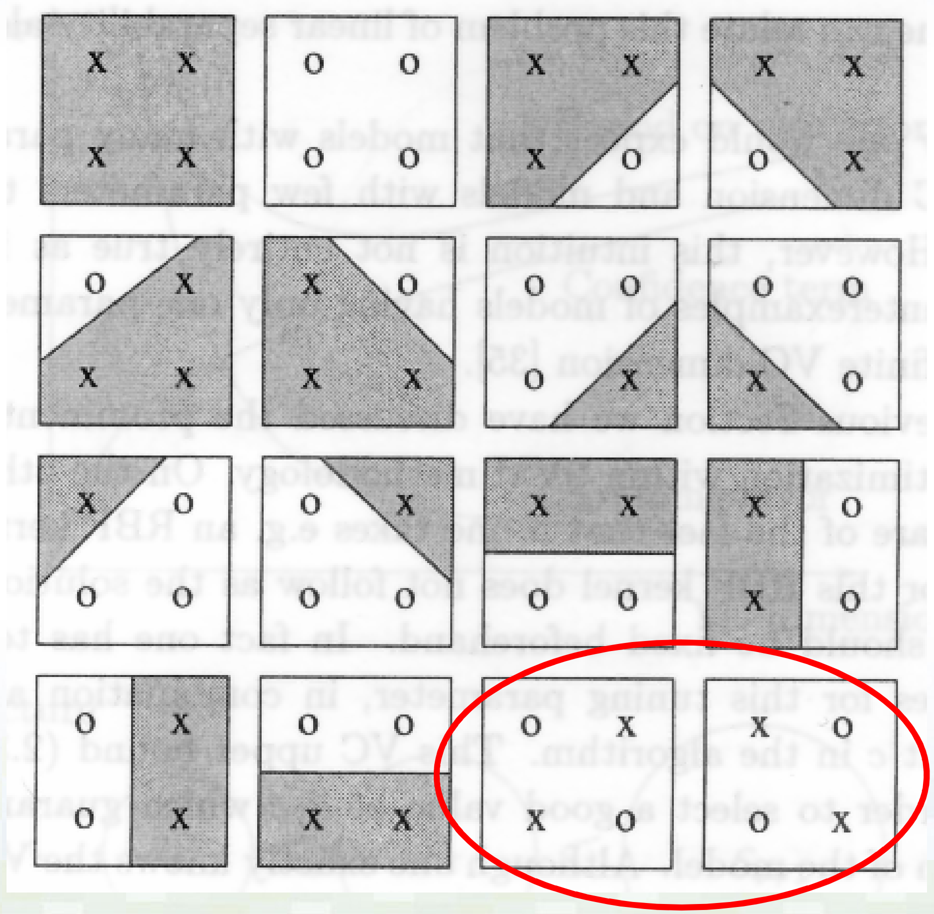


How many points can a linear boundary classify exactly? (2-D)

- We can do 2 and 3. **Can we do 4?**

How many points can a linear boundary classify exactly? (2-D)

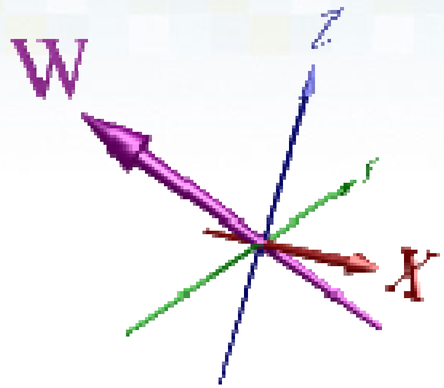
- We can do 2 and 3. Can we do 4? – **<u>NO</u>!**



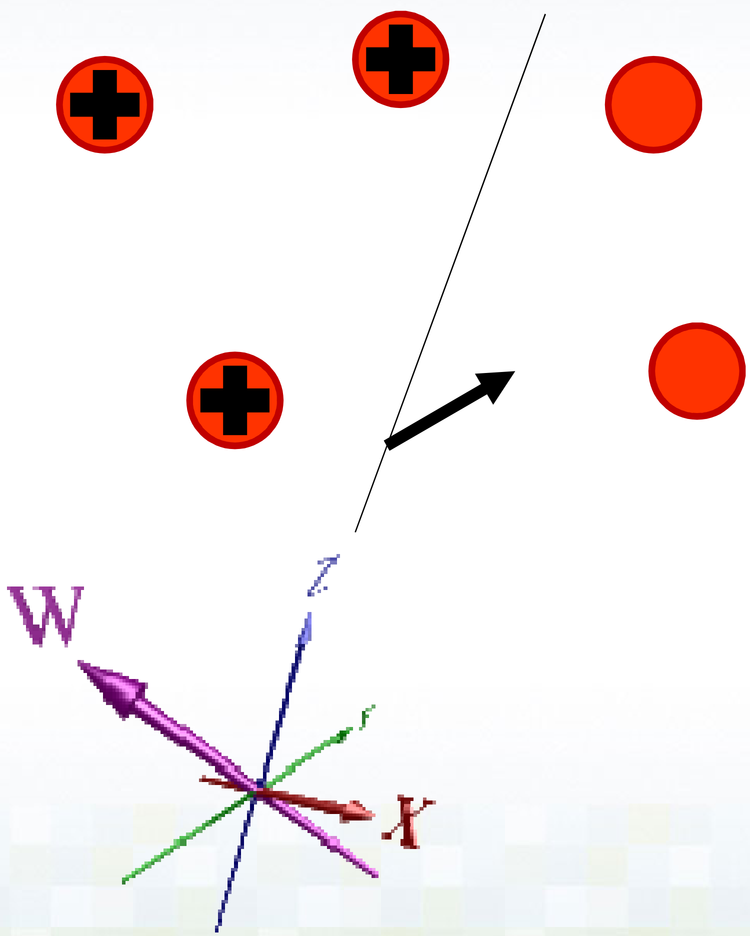
Can't do 4

Sorry!

How many points can a linear boundary classify exactly? (d-D)



How many points can a linear boundary classify exactly? (d-D)



Can do d + 1 points

How many parameters in a linear classifier in d-dimensions?

$$w_0 + \sum_{i=1}^{d} w_i x_i$$

Shattering a set of points

- Number of training points that can be classified exactly is VC dimension!

- *Definition*: a **dichotomy** of a set *S* is a partition of S into two disjoint subsets

- *Definition*: a set of instances *S* is **shattered** by hypothesis space *H* if and only if for every dichotomy of *S* there exists some hypothesis in *H* consistent with this dichotomy

Shattering a set of points

- Number of training points that can be classified exactly is VC dimension!

- *Definition*: a **dichotomy** of a set *S* is a partition of S into two disjoint subsets

$$S = \{x_1, x_2, \dots, x_n\}$$

$$S^+ = \{x_1, x_7, x_{12}, \dots\}$$

$$S^- = S - S^+ = \{x_2, x_3, x_4, x_5, x_6, x_8, \dots\}$$

Shattering a set of points

- *Definition*: a set of instances S is **shattered** by hypothesis space H if and only if for every dichotomy of S there exists some hypothesis in H consistent with this dichotomy

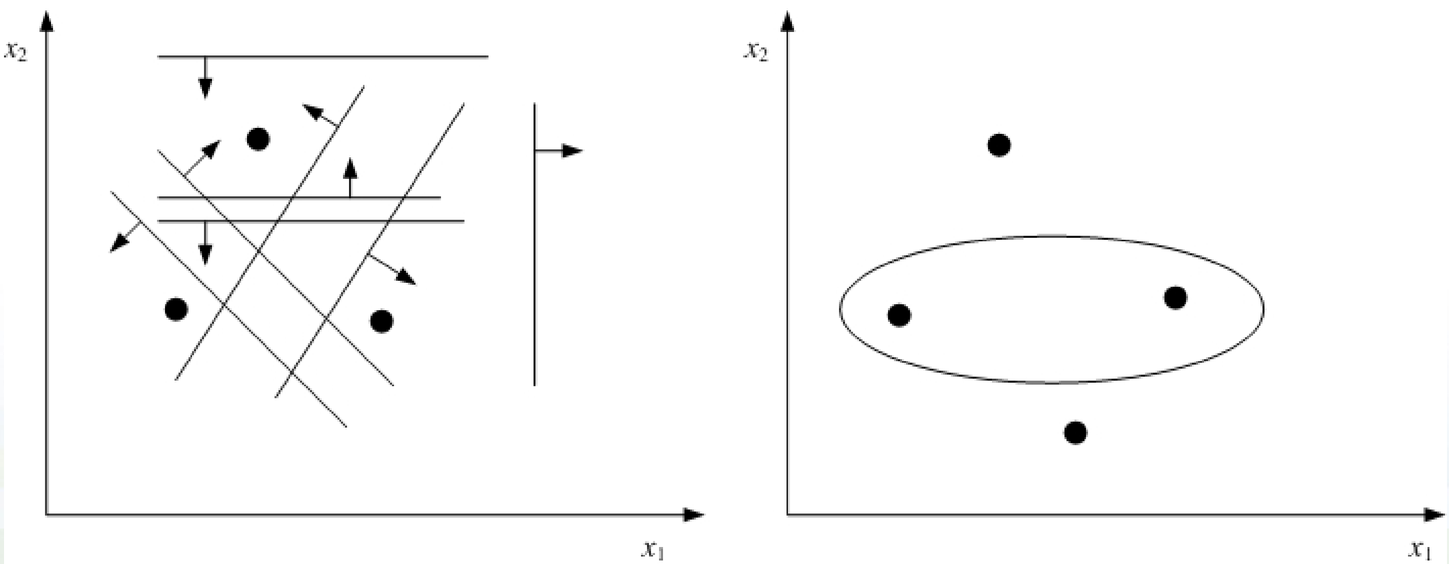
$$\forall\ partitions\ of\ S,$$

$$\exists\ h \in H, classifies\ all\ \begin{matrix} S+as\ positive \\ S-as\ negative \end{matrix}$$

- If a set of n samples can be separated by a set of functions in all 2^n possible ways, the sample is said to be shattered *(by the set of functions)*
- Shattering ~ a set of models can explain a given sample of size n *(for all possible labelings)*

VC Dimension *(finally! ☺)*

- *Definition*: The **Vapnik-Chervonenkis dimension**, $VC(H)$, of hypothesis space H defined over instance space X is the size of the largest finite subset of X **shattered** by H. If arbitrarily large finite sets of X can be shattered by H, then $VC(H) \equiv \infty$



VC Dimension (*finally!* ☺)

- *Definition*: The **Vapnik-Chervonenkis dimension**, $VC(H)$, of hypothesis space H defined over instance space X is the size of the largest finite subset of X **shattered** by H. If arbitrarily large finite sets of X can be shattered by H, then $VC(H) \equiv \infty$

- *For **linear classifiers** in 2D: Given a set of points, adversary labels them. E.g.* ***3 points*** *in 2D space.* $\forall$ labels $\exists h$ *(try it!)* ➔ *therefore CAN shatter 3 points!* ☺
- *Prove can't shatter 4 points:* $\forall$ *4 points, adversary can always pick XOR* ➔ $VC(H) < 4$
- *So,* $\boldsymbol{VC(H) = 3}$

Examples of VC Dimension

- Linear classifiers:
  - VC(H) = d+1, for d features plus constant term b
- Neural networks
  - VC(H) = # parameters
  - Local minima means NNs will probably not find best parameters
  - But if you find a NN with small # of parameters and training error is low → true error is also typically 'low'
- 1-Nearest neighbor?

Examples of VC Dimension

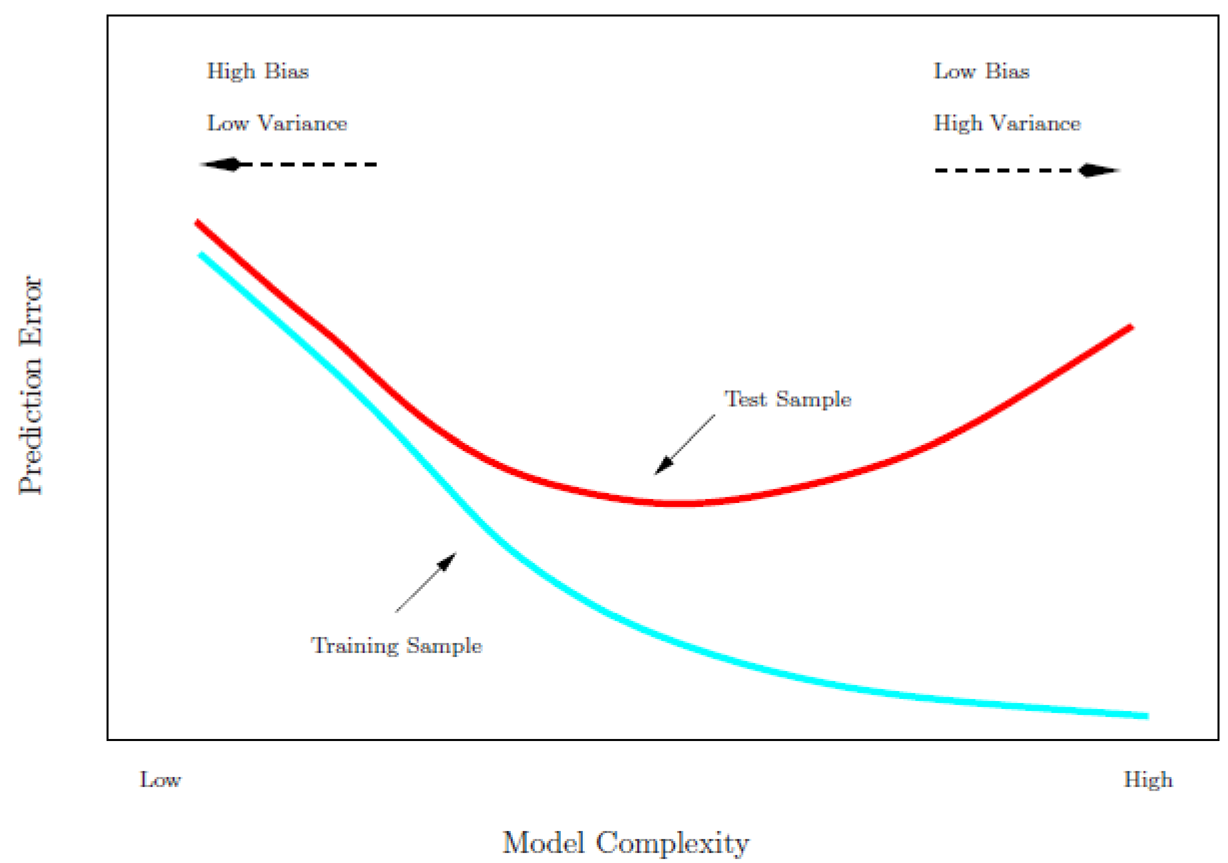
- Linear classifiers:
  - VC(H) = d+1, for d features plus constant term b
- Neural networks
  - VC(H) = # parameters
  - Local minima means NNs will probably not find best parameters
- 1-Nearest neighbor?
  - Given any set of points, it will *correctly classify 100%* of those points since each point is the closest point to itself → VC dimension of 1-NN classifier is ∞

Complexity of the classifier…

- … depends on the number of points that can be classified exactly
  - Finite case: decision trees
  - Infinite case: VC dimension
- Bias-Variance tradeoff in learning theory
- **VC-dimension is *infinite*** *if a sample of size n can be split in all 2^n possible ways*

  (in this case, no valid generalization is possible)
- Interpretation of the VC-dimension via **falsifiability**:
  - functions with *small* VC-dim can be easily *falsified*

Remember the Tradeoff



- Complexity found either by CV or by VC dimension, …

VC-dimension and Falsifiability

A set of functions has VC-dimension h if

- (a) It can explain (shatter) a set of x samples
  ~ there exists x samples that cannot falsify it

and

- (b) It can <u>not</u> shatter $x+1$ samples
  ~ any $x+1$ samples falsify this set

- Finiteness of VC-dim is necessary and sufficient condition for **generalization**
  (for *any* learning method based on Empirical Risk Minimization (ERM))

Recall Occam's Razor

- Main problem in predictive learning
  - Complexity control (model selection)
  - How to measure complexity?
- **Interpretation of Occam's razor** (in Statistics):
  Entities ~ model parameters
  Complexity ~ degrees-of-freedom (DoF)
  Necessity ~ explaining (fitting) available data

→ Model complexity = number of parameters (DoF)

- Consistent with classical statistical view:
  learning = function approx. / density estimation

Philosophical Principle of VC-falsifiability

- **Occam's Razor**: Select the model that explains available data *and* has the *smallest* number of free parameters (entities)

- **VC theory**: Select the model that explains available data *and* has **low VC-dimension** (i.e. can be ***easily falsified***)

- → New principle of VC-falsifiability

Calculating the VC-dimension

- How to estimate the VC-dimension (for a given set of functions)?
- Apply definition (via shattering) to derive analytic estimates – works for 'simple' sets of functions
- Generally, such analytic estimates are not possible for complex nonlinear parameterizations (i.e., for practical machine learning and statistical methods)

Examples >>

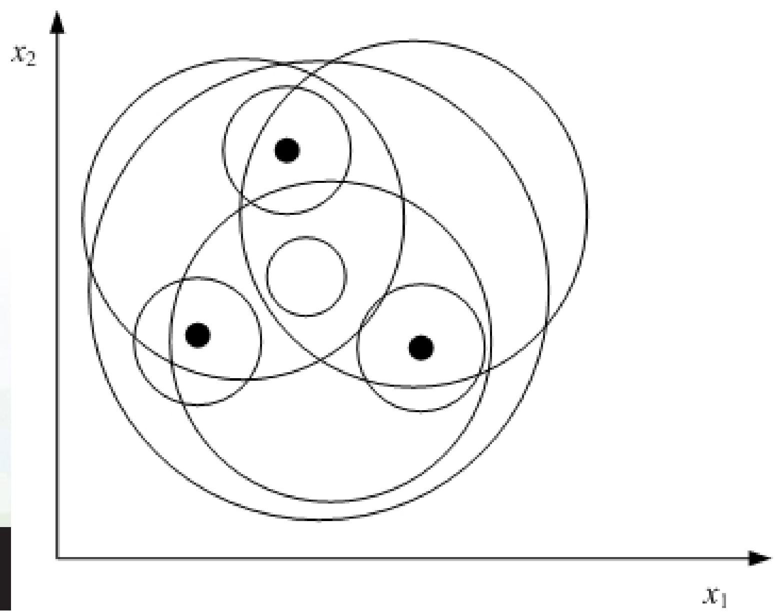
Example 1

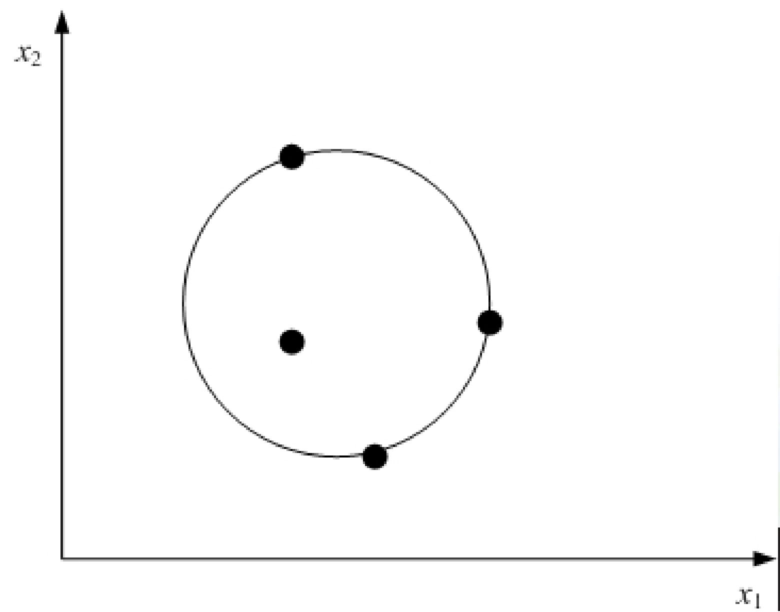
VC-dimension of spherical indicator functions

Consider spherical decision surfaces in a d-dimensional $\mathbf{x}$-space, parameterized by center $\mathbf{c}$ and radius r parameters:

$$f(\boldsymbol{x}, \boldsymbol{c}, r) = I((\boldsymbol{x} - \boldsymbol{c})^2 \leq r^2)$$

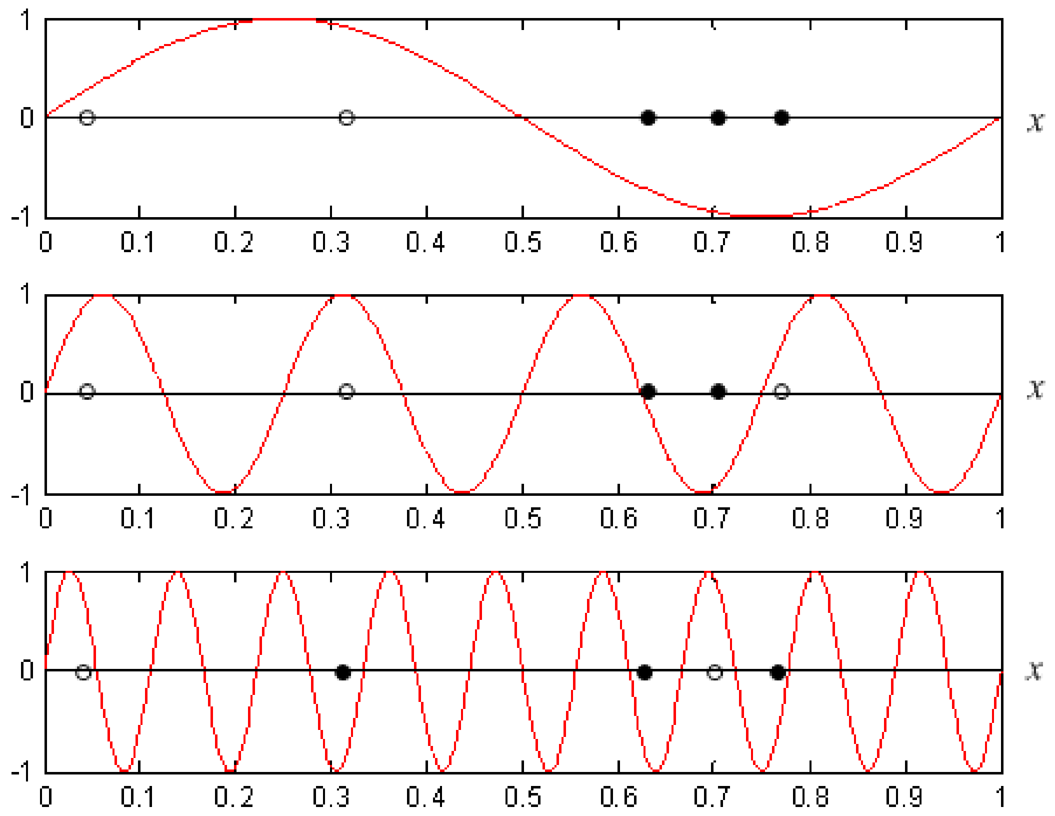
In a 2-dim space (d=2) there exists 3 points that can be shattered, but 4 points cannot be shattered so **VC(H) = 3**





- Example 2: VC-dimension of a linear combination of fixed basis functions (i.e. polynomials, Fourier expansion etc.) Assuming that basis functions are linearly independent, the VC-dim equals the number of basis functions (free parameters).
- Example 3: single parameter but infinite VC-dimension

$$f(x, w) = I(\sin wx > 0)$$



VC-dimension for Regression Problems

- VC-dimension was defined for indicator functions
- Can be extended to real-valued functions, i.e. third-order polynomial for univariate regression:

$$f(x, w, b) = w_3x^3 + w_2x^2 + w_1x + \mathrm{b}$$

linear parameterization → VC-dim = 4

- Qualitatively, the VC-dimension ~ the ability to fit (or explain) finite training data for regression

~Additional Material~

Keep-It-Direct Principle

-EXAMPLE-

Learning vs. System Identification

Learning vs System Identification

- Consider regression problem $y = g(\mathbf{x}) + \delta$
  where unknown target function $g(\mathbf{x}) = E(y/\mathbf{x})$
- Goal 1: Prediction $R(\mathbf{w}) = \int (y - f(\mathbf{x}, \mathbf{w}))^2 dP(\mathbf{x}, y) \to \min$
- Goal 2: Function Approximation (system identification)

  $$R(\mathbf{w}) = \int (f(\mathbf{x}, \mathbf{w}) - g(\mathbf{x}))^2 d\mathbf{x} \to \min$$

  or

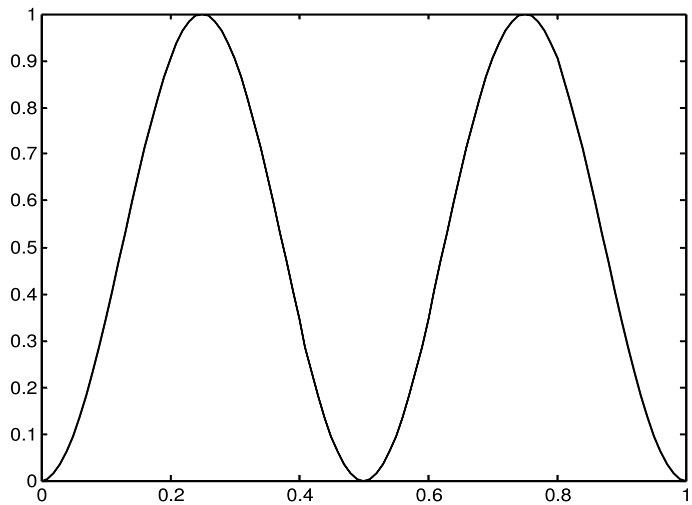
  $$\left\| f(\mathbf{x}, \mathbf{w}) - E(y/\mathbf{x}) \right\| \to \min$$
- Admissible models: algebraic polynomials
- Purpose of comparison: contrast goals (1) and (2)

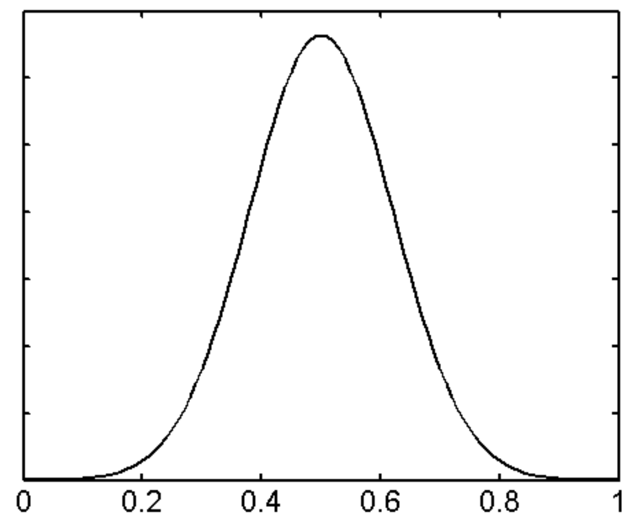
NOTE: most applications assume Goal 2, i.e.

Noisy Data ~ true signal + noise

Empirical Comparison

- Target function: sine-squared $g(x) = \sin^2(2\pi x) \quad x \in [0,1]$





- Input distribution: non-uniform Gaussian pdf
- Additive gaussian noise with st. deviation = 0.1

Empirical Comparison (cont'd)

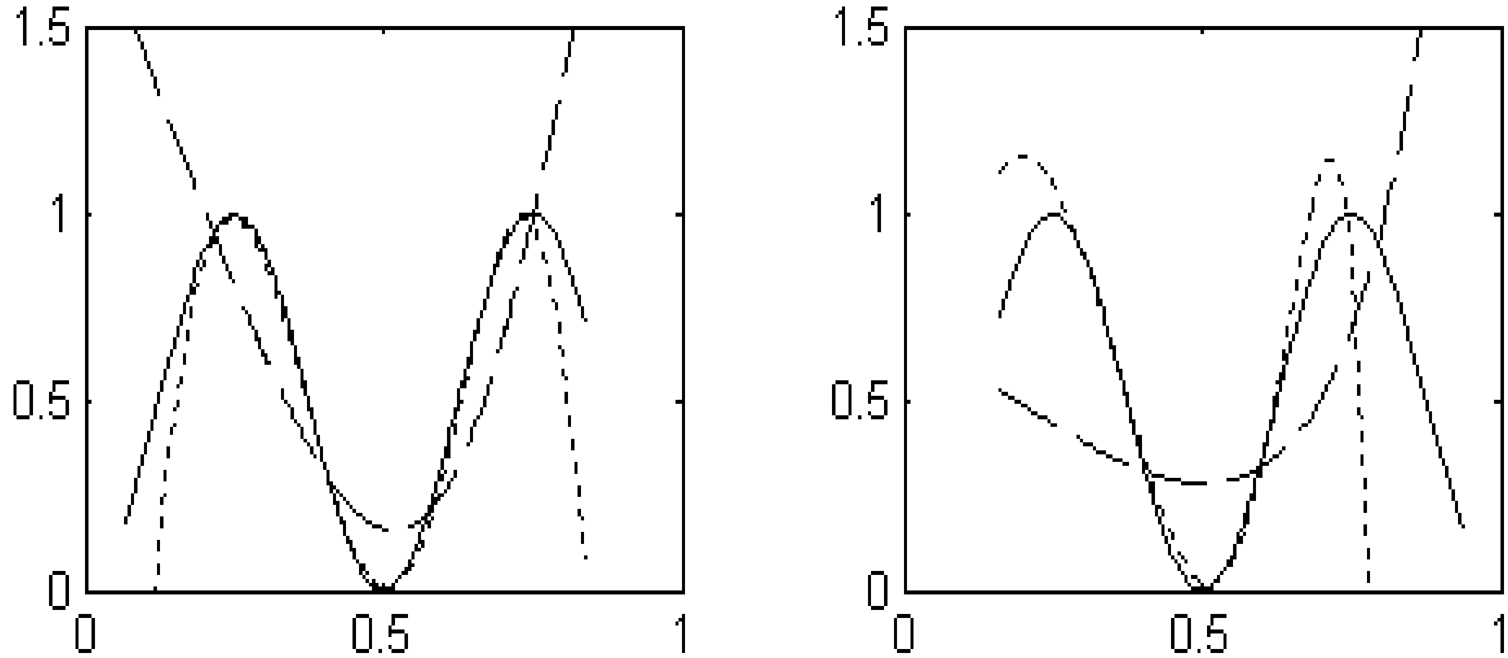
- Model selection: use separate data sets
  - *training* : for parameter estimation
  - *validation*: for selecting polynomial degree
  - *test*: for estimating prediction risk (MSE)

- Validation set generated differently to contrast (1)&(2)
  Predictive Learning (1) ~ Gaussian
  Funct. Approximation (2) ~ uniform fixed sampling

- Training + test data ~ Gaussian

- Training set size: 30 Validation set size : 30

- Regression estimates (2 typical realizations of data):



Dotted line ~ estimate obtained using predictive learning

Dashed line ~ estimate via function approximation setting

→ Estimated models are *too smooth* (under fct approx.)

Conclusion

- The goal of prediction (1) is different (less demanding) than the goal of estimating the true target function (2) everywhere in the input space.
- The curse of dimensionality applies to system identification setting (2), but may not hold under predictive setting (1).
- Both settings coincide if the input distribution is uniform (i.e., in signal and image denoising applications)