Learning Theory: VC Dimension

These slides were assembled by Byron Boots, with only minor modifications from Eric Eaton's slides and grateful acknowledgement to the many others who made their course materials freely available online. Feel free to reuse or adapt these slides for your own academic purposes, provided that you include proper attribution.

Last Time: Bias-Variance Tradeoff

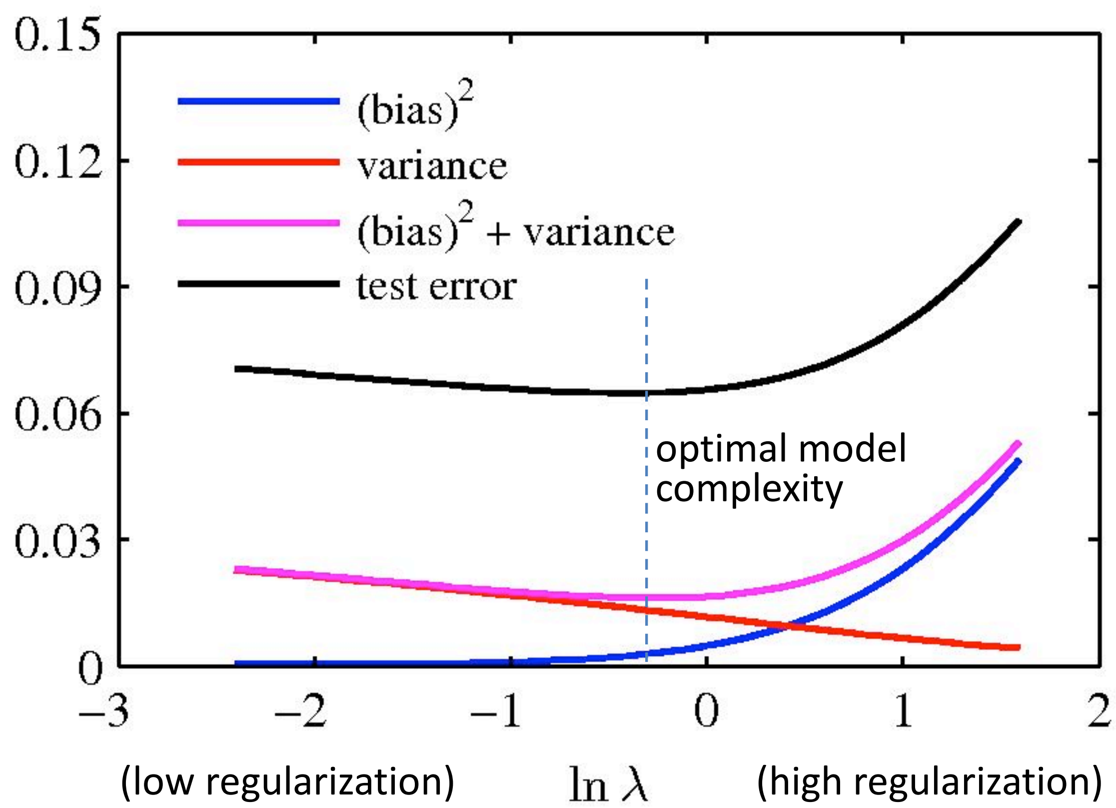


Figure provided by by Max Welling

A Way to Choose the Best Model

- It would be <u>really</u> helpful if we could get a guarantee of the following form:

$$\text{testingError} \leq \text{trainingError} + f(n, h, p)$$

n = size of training set

h = measure of the model complexity

p = the probability that this bound fails

We need p to allow for really unlucky training/test sets

- Then we could choose the model complexity that minimizes the bound on the test error

A Measure of Model Complexity

- Suppose that we pick n data points and assign labels of + or – to them at random
- If our model class (e.g., a decision tree, polynomial regression of a particular degree, etc.) can learn any association of labels with data, it is too powerful!

More power: can model more complex functions, but may overfit

Less power: won't overfit, but limited in what it can represent

- **Idea**: characterize the power of a model class by asking how many data points it can perfectly learn all possible assignments of labels
  - This number of data points is called the Vapnik-Chervonenkis (VC) dimension

Based on slides by Geoff Hinton

VC Dimension

- A measure of the power of a particular class of models
  - It does not depend on the choice of training set

- The VC dimension of a model class is the maximum number of points that can be arranged so that the class of models can shatter those points

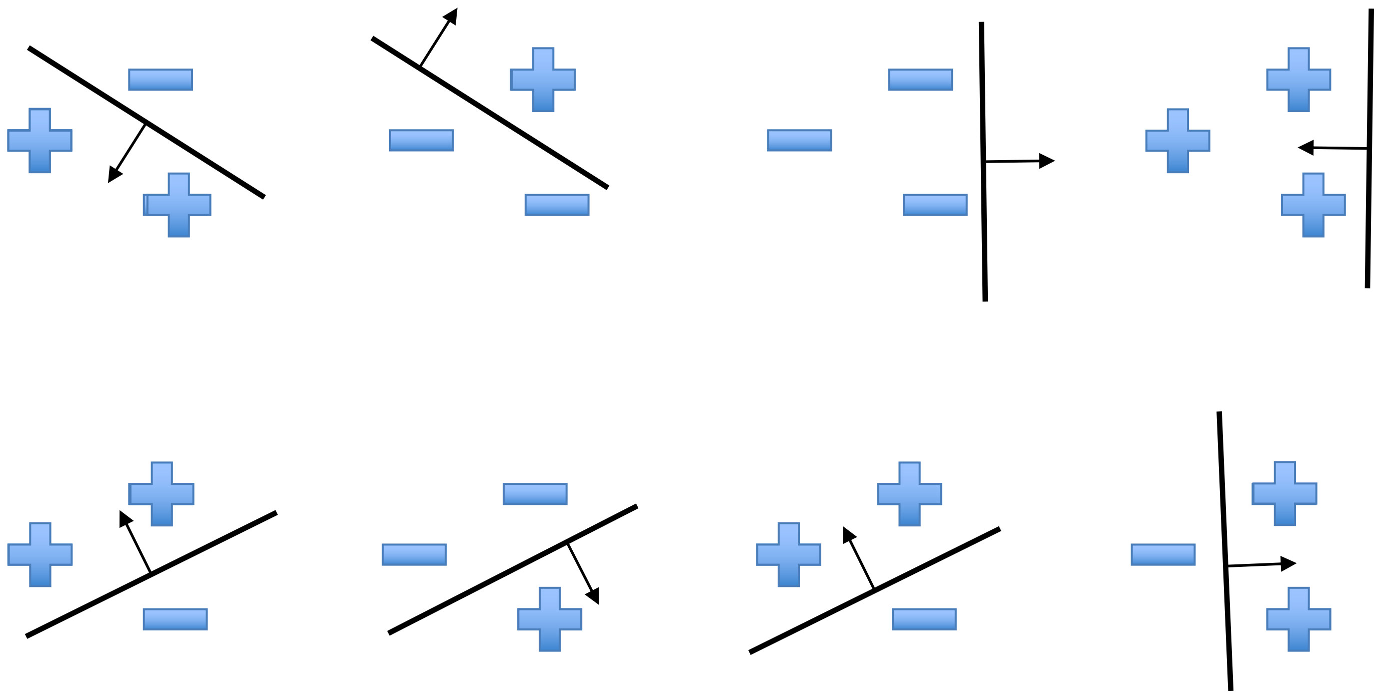
Definition: a model class can shatter a set of points

$$\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(r)}$$

if for <u>every</u> possible labeling over those points, there exists a model in that class that obtains zero training error

An Example of VC Dimension

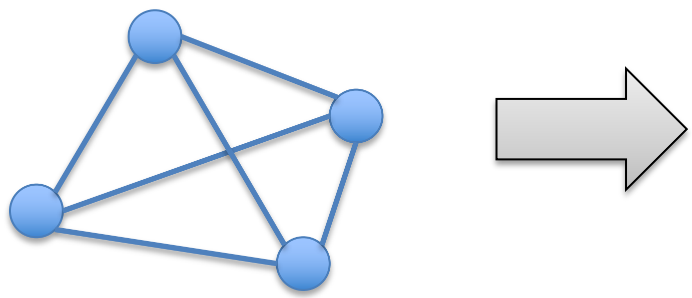
- Suppose our model class is a hyperplane
- Consider all labelings over three points in $\mathbb{R}^2$



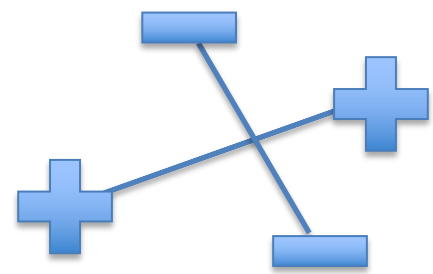
- In $\mathbb{R}^2$, we can find a hyperplane (i.e., a line) to capture any labeling of 3 points. A 2D hyperplane shatters 3 points

An Example of VC Dimension

- But, a 2D hyperplane cannot deal with some labelings of four points:



Connect all pairs of points; two lines will always cross



Can’t separate points if the pairs that cross are the same class

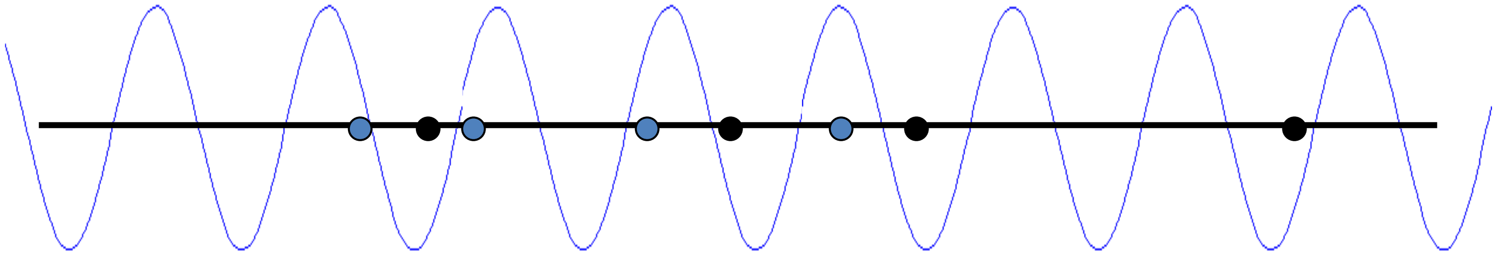
- Therefore, a 2D hyperplane cannot shatter 4 points

Some Examples of VC Dimension

- The VC dimension of a 2D hyperplane is 3.
  - In d dimensions it is d+1
    - It's just a coincidence that the VC dimension of a hyperplane is almost identical to the # parameters needed to define a hyperplane

- A sine wave has infinite VC dimension and only 2 parameters!
  - By choosing the phase & period carefully we can shatter any random set of 1D data points (except for nasty special cases)

$$h(x) = a\sin(bx)$$



Assumptions

- Given some model class (which defines the hypothesis space H)
- Assume all training points were drawn i.i.d from distribution $\mathcal{D}$
- Assume all future test points will be drawn from $\mathcal{D}$

Definitions:

$$\underbrace{R(\boldsymbol{\theta})}_{\text{"official" notation}} = \underbrace{\text{testError}(\boldsymbol{\theta})}_{\text{notation we'll use}} = \underbrace{E\left[\frac{1}{2}|y - h_{\boldsymbol{\theta}}(\boldsymbol{x})|\right]}_{\text{probability of misclassification}}$$

$$R^{\text{emp}}(\boldsymbol{\theta}) = \text{trainError}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\left|y^{(i)} - h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)})\right|$$

A Probabilistic Guarantee of Generalization Performance

Vapnik showed that with probability (1 –η):

$$\mathrm{testError}(\boldsymbol{\theta}) \leq \mathrm{trainError}(\boldsymbol{\theta}) + \sqrt{\frac{h(\log(2n/h)+1) - \log(\eta/4)}{n}}$$

n = size of training set

h = VC dimension of model class

η= the probability that this bound fails

- So, we should pick the model with the complexity that minimizes this bound
  - Actually, this is only sensible if we think the bound is fairly tight, which it usually isn't
  - The theory provides insight, but in practice we still need some witchcraft

Take Away Lesson

Suppose we find a model with a low training error...

- If hypothesis space H is very big (relative to the size of the training data n), then we most likely overfit

- If the following holds:
  - H is sufficiently constrained in size (low VC dimension)
  - and/or the size of the training data set n is large,

  then low training error is likely to be evidence of low generalization error