Foundations of Data Science, Fall 2020

5. Basis Expansion, Learning Curves, Overfitting

Prof. Dan Olteanu

DaST Data•(Systems+Theory)

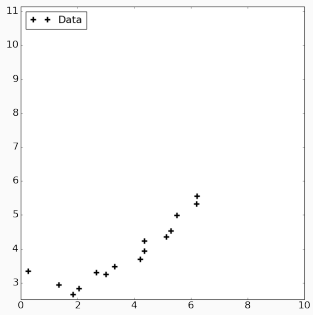
University of Zurich[UZH]

Oct 6, 2020

https://lms.uzh.ch/url/RepositoryEntry/16830890400

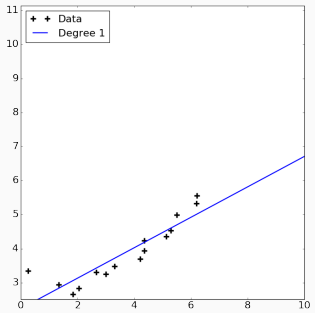
https://uzh.zoom.us/j/96690150974?pwd=cnZmMTduWUtCeWoxYW85Z3RMYnpTZz09

Linear Regression : Polynomial Basis Expansion



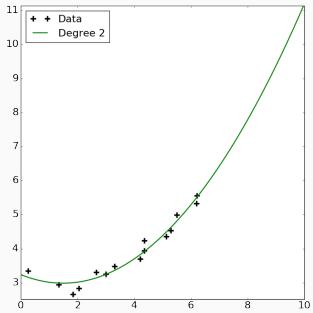
Linear Regression : Polynomial Basis Expansion

$\psi(x) = [1, x]^\top \qquad \mathbf{w} = [w_0, w_1]^\top \qquad \mathbf{w} \cdot \psi(x) = w_0 + w_1 x$



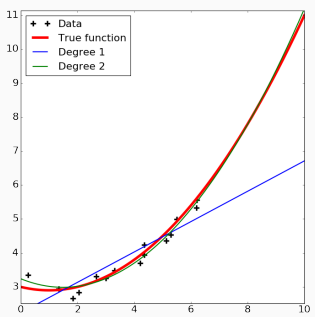
Linear Regression : Polynomial Basis Expansion

$\psi(x) = [1, x, x^2]^\top \qquad \mathbf{w} = [w_0, w_1, w_2]^\top \qquad \mathbf{w} \cdot \psi(x) = w_0 + w_1 x + w_2 x^2$



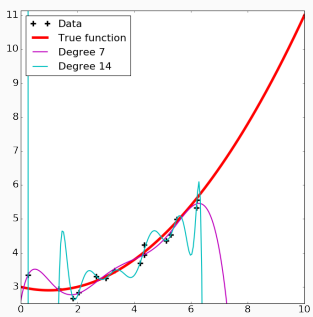
Linear Regression : Polynomial Basis Expansion

$\psi(x) = [1, x, x^2]^\top \qquad \mathbf{w} = [w_0, w_1, w_2]^\top \qquad \mathbf{w} \cdot \psi(x) = w_0 + w_1 x + w_2 x^2$



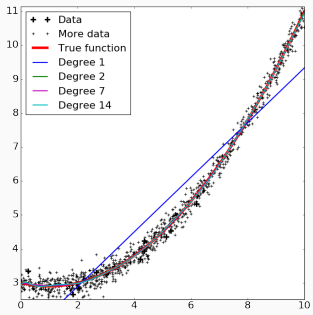
Linear Regression : Polynomial Basis Expansion

$\psi(x) = [1, x, x^2, \cdots, x^d]^\top \qquad \mathbf{w} = [w_0, \ldots, w_d]^\top \qquad \mathbf{w} \cdot \psi(x) = \sum_{i=0}^{d} w_i x^i$



Linear Regression : Polynomial Basis Expansion

Getting more data can avoid overfitting!



Polynomial Basis Expansion in Higher Dimensions

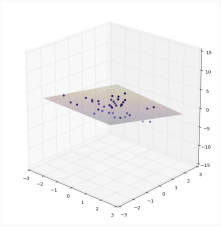
Basis expansion can be performed in higher dimensions

We're still fitting linear models, but using more features

$$y = \mathbf{w} \cdot \psi(\mathbf{x}) + \epsilon$$

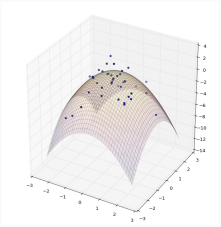
Linear Model

$\psi(\mathbf{x}) = [1, x_1, x_2]^\top$



Quadratic Model

$\psi(\mathbf{x}) = [1, x_1, x_2, x_1^2, x_2^2, x_1 x_2]^\top$



Using degree d polynomials in k dimensions results in $O(k^d)$ features!

Basis Expansion Using Kernels

We can use kernels as features [Sec. 5.7.2 in GBC; Ch. 15 in M]

For some expansion ϕ, a kernel computes the dot product

$$\kappa(\mathbf{x}', \mathbf{x}) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}, \qquad \kappa(\mathbf{x}', \mathbf{x}) = \phi(\mathbf{x}') \cdot \phi(\mathbf{x})$$

Examples of kernels:

- Polynomial kernel

$$\kappa_{\text{poly}}(\mathbf{x}', \mathbf{x}) = (\mathbf{x} \cdot \mathbf{x}' + \theta)^d$$

- Radial Basis Function kernel

$$\kappa_{\text{RBF}}(\mathbf{x}', \mathbf{x}) = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$$

- Further kernels: String kernels, Graph kernels (Schölkopf and Smola, 2002)

Side Note: Expansion Function for Polynomial Kernel

Q: Find expansion function for kernel $\kappa_{\text{poly}}(\mathbf{x}', \mathbf{x}) = (\mathbf{x} \cdot \mathbf{x}')^d$, where $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^k$

A: We use $(z_1 + \cdots + z_k)^d = \sum_{n_i \geq 0, \sum_i n_i = d} \underbrace{\frac{d!}{n_1! \cdots n_k!}}_{\text{multinomial coeff. } C(d; n_1, \ldots, n_k)} z_1^{n_1} \cdots z_k^{n_k}$

C = # of ways to distribute d balls into k bins, where the j-th bin holds $n_j \geq 0$ balls

Now assume $z_l = x_i x_i'$ in the above formula. Then,

$$(\mathbf{x} \cdot \mathbf{x}')^d = \underbrace{\sum_{n_i \geq 0, \sum_i n_i = d}}_{\text{dim of } \phi_{\text{poly}}} \underbrace{\sqrt{C(d; n_1, \ldots, n_k)} x_1^{n_1} \cdots x_k^{n_k}}_{\text{one row in } \phi_{\text{poly}}(\mathbf{x})} \underbrace{\sqrt{C(d; n_1, \ldots, n_k)} (x_1')^{n_1} \cdots (x_k')^{n_k}}_{\text{one row in } \phi_{\text{poly}}(\mathbf{x}')}$$

The dimension of the vectors $\phi_{\text{poly}}(\mathbf{x})$ and $\phi_{\text{poly}}(\mathbf{x}')$ is $O(k^d)$.

Complexity of computing $\kappa(\mathbf{x}', \mathbf{x})$: $O(k^d)$ using ϕ_{poly} vs. $O(k \log d)$ using $(\mathbf{x} \cdot \mathbf{x}')^d$!

Side Note: Expansion Function for Polynomial Kernel (Examples)

For $\mathbf{x} = [x_1 \; x_2]^\top$ and $\mathbf{x}' = [x_1' \; x_2']^\top$ find ϕ such that $\kappa(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x}) \cdot \phi(\mathbf{x}')$

$$\kappa(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}')^2 = (x_1 x_1' + x_2 x_2')^2 = x_1^2 (x_1')^2 + 2 x_1 x_1' x_2 x_2' + x_2^2 (x_2')^2 = \phi(\mathbf{x}) \cdot \phi(\mathbf{x}')$$

$$\phi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2} x_1 x_2 \end{bmatrix} \qquad \phi(\mathbf{x}') = \begin{bmatrix} (x_1')^2 \\ (x_2')^2 \\ \sqrt{2} x_1' x_2' \end{bmatrix}$$

$$\begin{aligned} \kappa(\mathbf{x}, \mathbf{x}') &= (\mathbf{x} \cdot \mathbf{x}' + \theta)^2 \\ &= (x_1 x_1' + x_2 x_2' + \theta)^2 = x_1^2 (x_1')^2 + 2 x_1 x_1' x_2 x_2' + x_2^2 (x_2')^2 + 2 x_1 x_1' \theta + 2 x_2 x_2' \theta + \theta^2 \end{aligned}$$

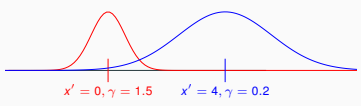
$$\phi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2} x_1 x_2 \\ \sqrt{2\theta} x_1 \\ \sqrt{2\theta} x_2 \\ \theta \end{bmatrix} \qquad \phi(\mathbf{x}') = \begin{bmatrix} (x_1')^2 \\ (x_2')^2 \\ \sqrt{2} x_1' x_2' \\ \sqrt{2\theta} x_1' \\ \sqrt{2\theta} x_2' \\ \theta \end{bmatrix}$$

Radial Basis Function Kernel

A Radial Basis Function (RBF) kernel is defined as

$$\kappa_{\text{RBF}}(\mathbf{x}', \mathbf{x}) = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2) = \frac{1}{e^{\gamma \|\mathbf{x} - \mathbf{x}'\|^2}}$$

Hyperparameters: width γ and centres $\mathbf{x}'$



RBF acts as a similarity measure between $\mathbf{x}$ and $\mathbf{x}'$: $\kappa(\mathbf{x}', \mathbf{x}) \in (0, 1]$

- $\kappa_{\text{RBF}}(\mathbf{x}', \mathbf{x}) \to 0$ when $\gamma \|\mathbf{x} - \mathbf{x}'\|^2 \to \infty$

  $\mathbf{x}$ and $\mathbf{x}'$ are far apart (large squared Euclidian distance) or γ is very large

- $\kappa_{\text{RBF}}(\mathbf{x}', \mathbf{x}) \to 1$ when $\gamma \|\mathbf{x} - \mathbf{x}'\|^2 \to 0$

  $\mathbf{x}$ and $\mathbf{x}'$ are very close (small squared Euclidian distance) or γ is very small

Side Note: Expansion Function for RBF Kernel

What is ϕ_{RBF} such that $\kappa_{\text{RBF}}(\mathbf{x}', \mathbf{x}) = \exp(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2) = \phi_{\text{RBF}}(\mathbf{x}) \cdot \phi_{\text{RBF}}(\mathbf{x}')$?

We use the following:

$$\|\mathbf{x} - \mathbf{x}'\|^2 = \langle \mathbf{x} - \mathbf{x}', \mathbf{x} - \mathbf{x}' \rangle = \langle \mathbf{x}, \mathbf{x} \rangle + \langle \mathbf{x}', \mathbf{x}' \rangle - 2\langle \mathbf{x}, \mathbf{x}' \rangle = \|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2 - 2\mathbf{x} \cdot \mathbf{x}'$$

$$\exp(\mathbf{x} \cdot \mathbf{x}') = \sum_{k=0}^{\infty} \frac{1}{k!} \underbrace{(\mathbf{x} \cdot \mathbf{x}')^k}_{\phi_{poly}(\mathbf{x}) \cdot \phi_{poly}(\mathbf{x}')} \qquad \texttt{Taylor expansion:} \quad \exp(z) = \sum_{k=0}^{\infty} \frac{z^k}{k!}$$

Without loss of generality, assume $\gamma = 1$. Then,

$$\exp(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2) = \sum_{k=0}^{\infty} \underbrace{\left(\sqrt{\frac{1}{k!}} \exp(-\|\mathbf{x}\|^2)\phi_{poly}(\mathbf{x})\right)}_{\text{row } k \text{ in } \phi_{\text{RBF}}(\mathbf{x})} \cdot \underbrace{\left(\sqrt{\frac{1}{k!}} \exp(-\|\mathbf{x}'\|^2)\phi_{poly}(\mathbf{x}')\right)}_{\text{row } k \text{ in } \phi_{\text{RBF}}(\mathbf{x}')}$$

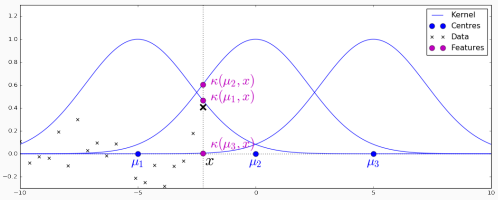
$\phi_{\text{RBF}} : \mathbb{R}^d \to \mathbb{R}^\infty$ projects vectors into an infinite dimensional space!

- Not feasible to compute $\kappa_{\text{RBF}}(\mathbf{x}', \mathbf{x})$ using ϕ_{RBF}!

Linear Model with RBF Kernel

- RBF kernel: $\kappa_{\text{RBF}}(\mathbf{x}', \mathbf{x}) = \exp(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2)$
- Choose centres $\mu_1, \mu_2, \ldots, \mu_M$
- Feature map: $\psi_{\text{RBF}}(\mathbf{x}) = [1, \kappa_{\text{RBF}}(\mu_1, \mathbf{x}), \ldots, \kappa_{\text{RBF}}(\mu_M, \mathbf{x})]^\intercal$

$$y = w_0 + w_1\kappa_{\text{RBF}}(\mu_1, \mathbf{x}) + \cdots + w_M\kappa_{\text{RBF}}(\mu_M, \mathbf{x}) + \epsilon = \mathbf{w} \cdot \psi(\mathbf{x}) + \epsilon$$



How to Choose RBF Hyperparameters?

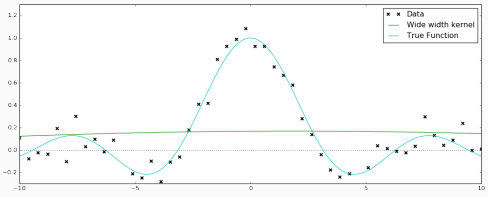
Reasonable choice for centres: The data points themselves can be centres

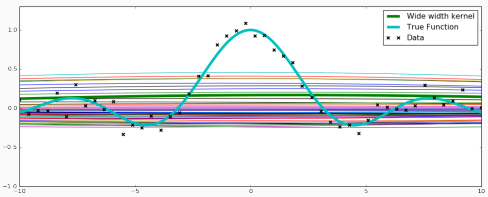
Choice for width parameter: overfitting or underfitting may occur

- Overfitting occurs if the kernel is too narrow
  i.e., γ very large
- Underfitting occurs if the kernel is too wide
  i.e., γ very small

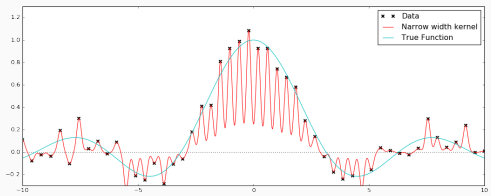
Similar situation with the choice for degree in polynomial basis expansion

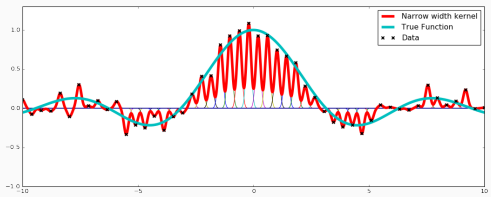
When the kernel is too wide (γ very small)



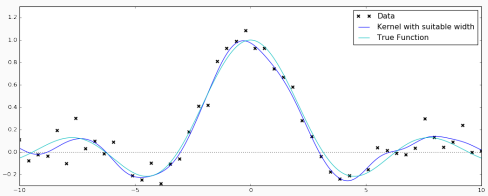


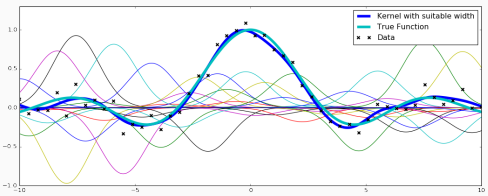
When the kernel is too narrow (γ very large)



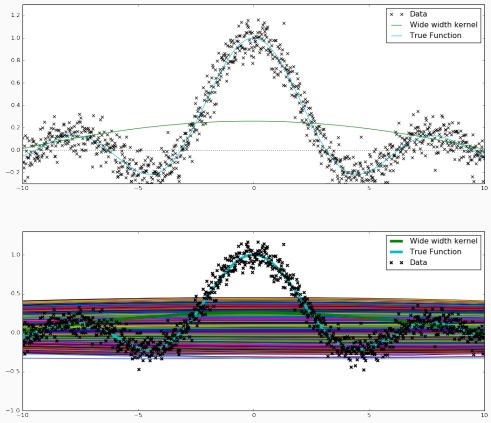


When the kernel is chosen suitably

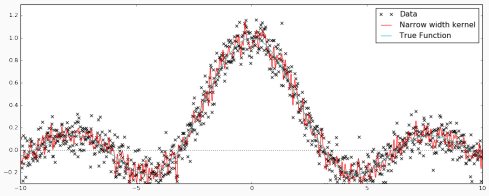


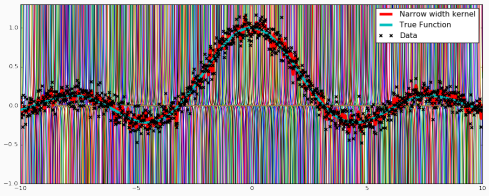


Big Data: When the kernel is too wide (γ very small)

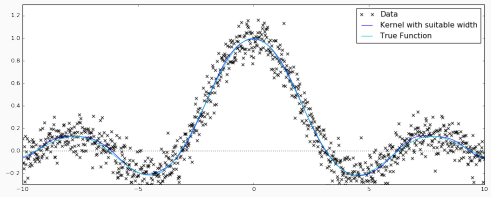


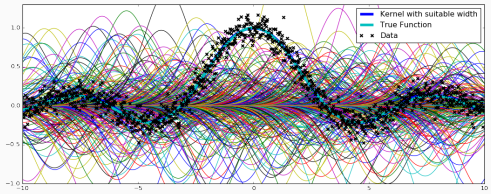
Big Data: When the kernel is too narrow (γ very large)





Big Data: When the kernel is chosen suitably





Basis Expansion using Kernels

- Overfitting occurs if the kernel is too narrow, *i.e.*, γ very large
  - Having more data can help reduce overfitting!
- Underfitting occurs if the kernel is too wide, *i.e.*, γ very small
  - Extra data does not help at all in this case!
- Curse of dimensionality: Data lies in a high-dimensional space
  - We may easily underfit
  - Might need exponentially large (in the dimension) dataset
- Radial basis functions are universal
  - as we increase the number of centres, the resulting model can approximate any continuous function
  - too powerful and computationally expensive

Fundamental Goal of Machine Learning

Generalise beyond the examples in the training set

- Test data is highly important
- Data alone is not enough
  - Every learner must embody knowledge/assumptions beyond the data
  - Wolpert's "no free lunch": no learner can beat random guessing over all possible functions to be learned.
  - Hope: Functions to learn in the real world are not drawn uniformly from the set of all mathematically possible functions
  - Reasonable assumptions behind ML's success: similar examples have similar classes; limited dependences; or limited complexity

Generalisation Error Decomposes into Bias and Variance

Generalisation error = bias + variance

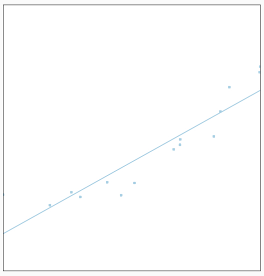
- Bias: tendency to consistently learn the same wrong thing
  - Linear learner has high bias: when the frontier between two classes is not a hyperplane the learner is unable to induce it.
  - Decision trees do not have this problem because they can represent any Boolean function.
- Variance: tendency to learn random things irrespective of the real signal
  - Decision trees suffer from high variance: when learned on different training sets generated by the same phenomenon they are often very different, when in fact they should be the same.

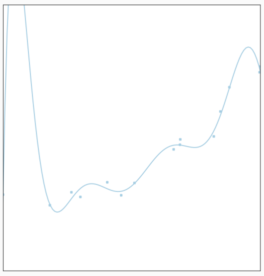
More powerful learners are not necessarily better than less powerful ones!

The Bias-Variance Tradeoff

Experiment: 14 1D points generated by a quadratic function

Linear model vs Higher-degree polynomial model





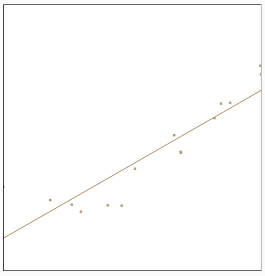
High Bias

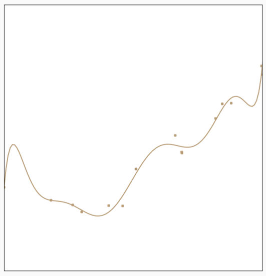
High Variance

The Bias-Variance Tradeoff

Experiment: 14 1D points generated by a quadratic function

Linear model vs Higher-degree polynomial model





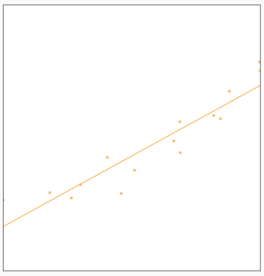
High Bias

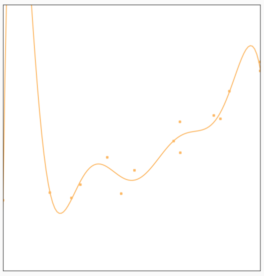
High Variance

The Bias-Variance Tradeoff

Experiment: 14 1D points generated by a quadratic function

Linear model vs Higher-degree polynomial model





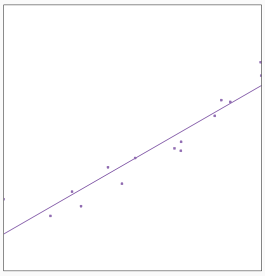
High Bias

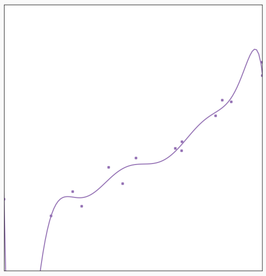
High Variance

The Bias-Variance Tradeoff

Experiment: 14 1D points generated by a quadratic function

Linear model vs Higher-degree polynomial model





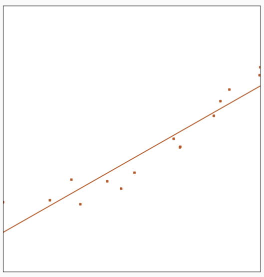
High Bias

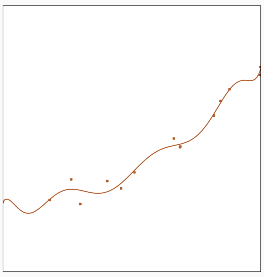
High Variance

The Bias-Variance Tradeoff

Experiment: 14 1D points generated by a quadratic function

Linear model vs Higher-degree polynomial model





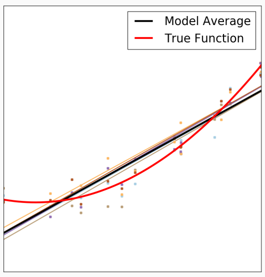
High Bias

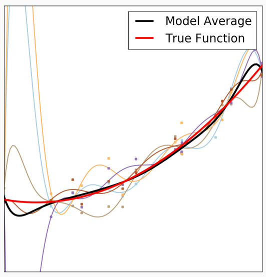
High Variance

The Bias-Variance Tradeoff

Experiment: 14 1D points generated by a quadratic function

Linear model vs Higher-degree polynomial model





High Bias

High Variance

The Bias-Variance Tradeoff

- Having high bias means that we are underfitting
- Having high variance means that we are overfitting
- The terms bias and variance are precisely defined statistical notions

  Detailed description: GBC book Sec. 5.4

Learning Curves

Suppose we've trained a model and used it to make predictions

But in reality, the predictions are often poor

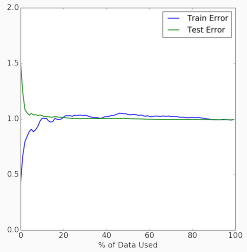
- How can we know whether we have high bias (underfitting) or high variance (overfitting) or neither?
  - Should we add more features (higher degree polynomials, narrower kernels, etc.) to make the model more expressive?
  - Should we simplify the model (lower degree polynomials, wider kernels, etc.) to reduce the number of parameters?
- Should we try and obtain more data?
  - Often there is a computational and monetary cost to using more data

Learning Curves

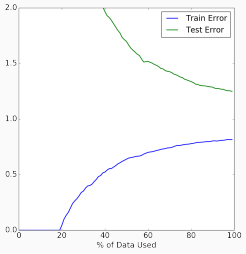
Split the data into a training set and testing set

Train on increasing sizes of data

Plot the training error and test error as a function of training data size



More data is not useful



More data would be useful

Overfitting: How does it occur?

When dealing with high-dimensional data (which may be caused by basis expansion) even for a linear model we have many parameters

With $D = 100$ input variables and using degree 10 polynomial basis expansion we have $\sim 10^{20}$ parameters!

Enrico Fermi to Freeman Dyson

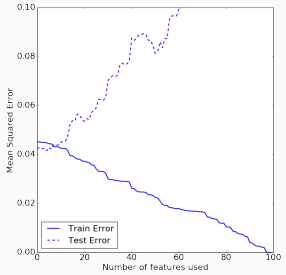
"I remember my friend Johnny von Neumann used to say, with four parameters I can fit an elephant, and with five I can make him wiggle his trunk."

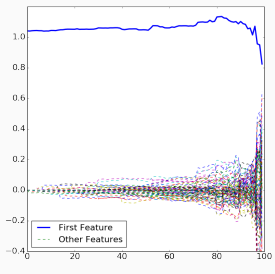
- Video with the elephant
- See paper and program in OLAT

Overfitting: How does it occur?

Suppose:

- $D = 100$ and $N = 100$ so that the data matrix $\mathbf{X} \in \mathbb{R}^{100 \times 100}$
- Every entry of $\mathbf{X}$ is drawn from $\mathcal{N}(0, 1)$
- Let $y_i = x_{i,1} + \mathcal{N}(0, \sigma^2)$, for $\sigma = 0.2$





How to Combat High Variance aka Overfitting

- Cross-validation
- Regularisation term to the evaluation function
  - Penalise models with more structure
  - Favour smaller models with less room to overfit
- Statistical significance test like chi-square before adding new structure
  - Decide whether the prediction is different w/o this structure

Avoiding overfitting may lead to high bias aka underfitting