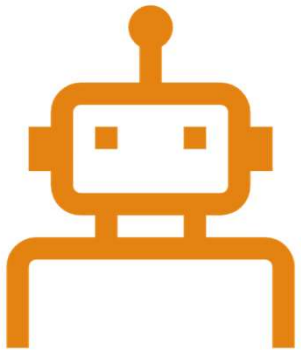


Machine Learning EN.601.475/675

DR. PHILIP GRAFF

Linear Regression

OUR FIRST ALGORITHM!

If she loves you more each and every day,
by linear regression she hated you before you met.

Why start here?

Well-known algorithm

Not *strictly* machine learning

Can learn about fundamentals with a simple example

Example

I have a large number of undergraduate applications for Johns Hopkins. I want to accept students who I think will get the highest GPA (0-4.0).

Goal: Predict an applicant's GPA

Data: Previous applications and resulting GPAs

How do I do this?

Data Model

Assume dependent variable (y) can be modeled by a linear function of the input variables (x)

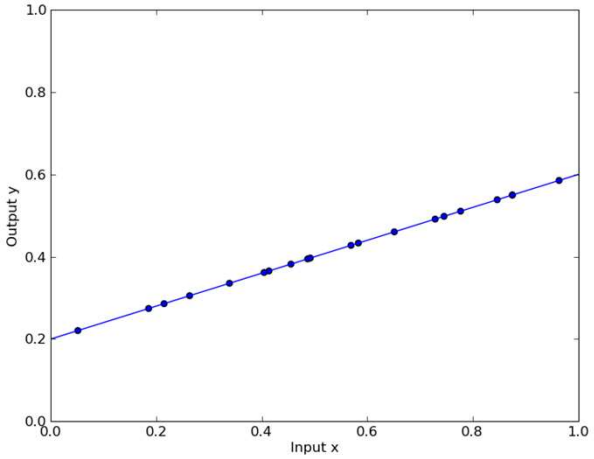
- $y = wx + b$

2 dimensions

- Compute w and b from two points

Solution?

- Given y and x, solve for w



Regression

Data:

Learn: a mapping from x to y

Examples:

- GPAs
- Stock Prices
- Miles per gallon
- Age of author

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$$

$$\mathbf{x}_i \in \Re^M \qquad y_i \in \Re$$

$$y = f(\mathbf{x})$$

Try it at home!

Linear regression demo

http://mste.illinois.edu/users/exner/java.f/leastsquares/

http://onlinestatbook.com/2/regression/linear_fit_demo.html

https://www.geogebra.org/m/FUe3HfRf

Recall our Definition

Fitting a function to data

Fitting: Solve for w given y and x

Function: linear function

Data: assume dependent variable linear combination of independent variables

Minimize a function

- What function are we minimizing?

What is the Goal?

You probably know linear regression from statistics

- "In statistics, linear regression is an approach for modeling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables) denoted X." (Wikipedia)

Our goal: predict correctly the next example

- Minimize: reduce prediction error
  - More to say about this later

Loss Functions

Machine learning algorithms minimize loss functions

- Or some substitute for a loss function
- The best solution minimizes the loss function (maybe)

Definition

- A function that maps between (true label, prediction) → non-negative number
- 0 = perfect prediction

Examples

Which of the following are valid loss functions for regression?

$$L(f(\mathbf{x})) = \sum_{i=1}^{N} l(y_i, f(x_i))$$

$$l(y_i, f(x_i)) = y_i - f(x_i)$$

$$l(y_i, f(x_i)) = \frac{1}{3}(y_i - f(x_i))^2$$

$$l(y_i, f(x_i)) = \begin{cases} 1, & f(x_i) = y_i \\ 0, & \text{else} \end{cases}$$

$$l(y_i, f(x_i)) = \begin{cases} 0, & f(x_i) = y_i \\ |f(x_i)|, & \text{else} \end{cases}$$

Examples

Which of the following are valid loss functions for regression?

$$L(f(\mathbf{x})) = \sum_{i=1}^{N} l(y_i, f(x_i))$$

$$l(y_i, f(x_i)) = y_i - f(x_i)$$

$$l(y_i, f(x_i)) = \frac{1}{3}(y_i - f(x_i))^2$$

$$l(y_i, f(x_i)) = \begin{cases} 1, & f(x_i) = y_i \\ 0, & \text{else} \end{cases}$$

$$l(y_i, f(x_i)) = \begin{cases} 0, & f(x_i) = y_i \\ |f(x_i)|, & \text{else} \end{cases}$$

Loss: What we Minimize

Loss measures the badness of our prediction

What's a good loss function?

- It depends on task and goals

Regression loss function?

- Proposal: How far are you from the correct answer

Sum of Squares Loss

Loss: $f(\mathbf{x}) - y$

Squared loss: $(f(\mathbf{x}) - y)^2$

Sum squared loss: $\sum_{i=1}^{N} (f(\mathbf{x}_i) - y_i)^2$

$$\sum_{i=1}^{N} (\mathbf{w} \cdot \mathbf{x}_i - y_i)^2$$

Over all answers

True answer

Predicted answer

Recall our Definition

Fitting a function to data

Fitting: Solve for w given y and x

Function: linear function

Data: assume dependent variable linear combination of independent variables

Minimize a function

- What function are we minimizing?

Hypothesis Class

Learning algorithm selects hypothesis from hypothesis class

Hypothesis class

- A set of possible hypotheses (functions) that can be used to label the data
- Can be finite or infinite

Each learning algorithm has a hypothesis class

- Fitting selects the best hypothesis using observed data

What is best hypothesis?

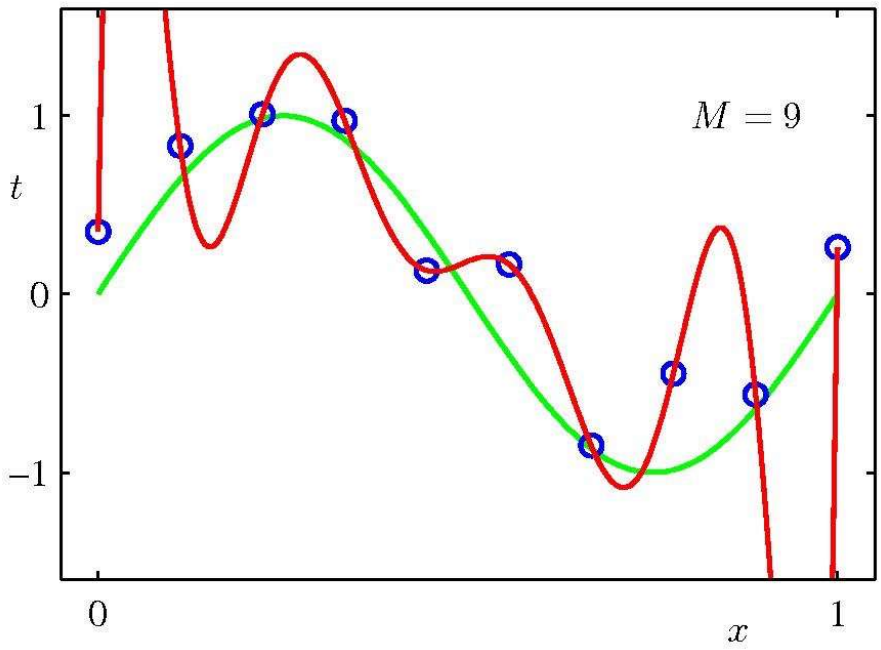
Choosing Hypothesis Class

What sort of hypothesis class do we want?

1. The hypothesis class should contain the optimal hypothesis
2. The hypothesis class for which our algorithm will find the best performing hypothesis

What is the difference?

Under/Over Fitting



Hypothesis Tradeoff

Rich hypothesis class → Over-fitting

Easier to search hypothesis class → Under-fitting

Realization: we are unlikely to ever find the hypothesis that exactly explains the data

Simplifying assumptions help find a reasonable hypothesis

Select hypothesis class based on knowledge of data

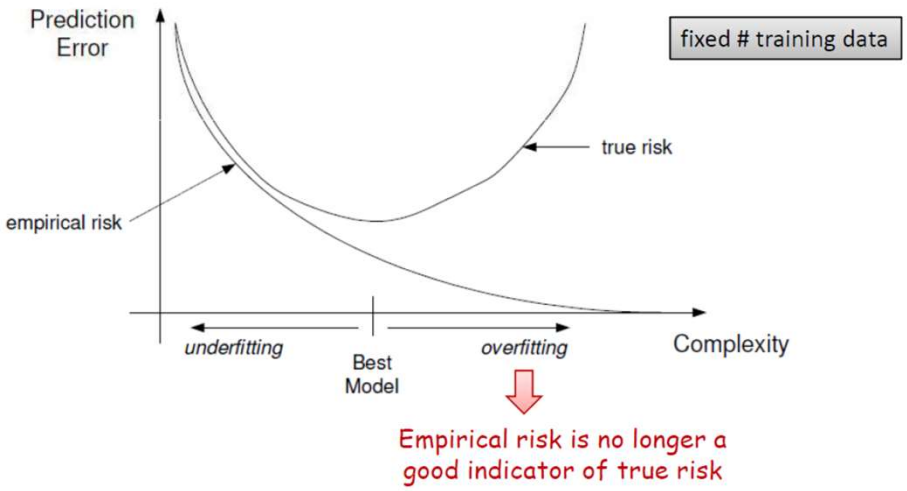


Figure Credit: Bin Zhao

True risk

True risk is a combination of estimation error and approximation error.

- Approximation error comes from the inadequacy of the algorithm.
- Estimation error comes from randomness of the training data.

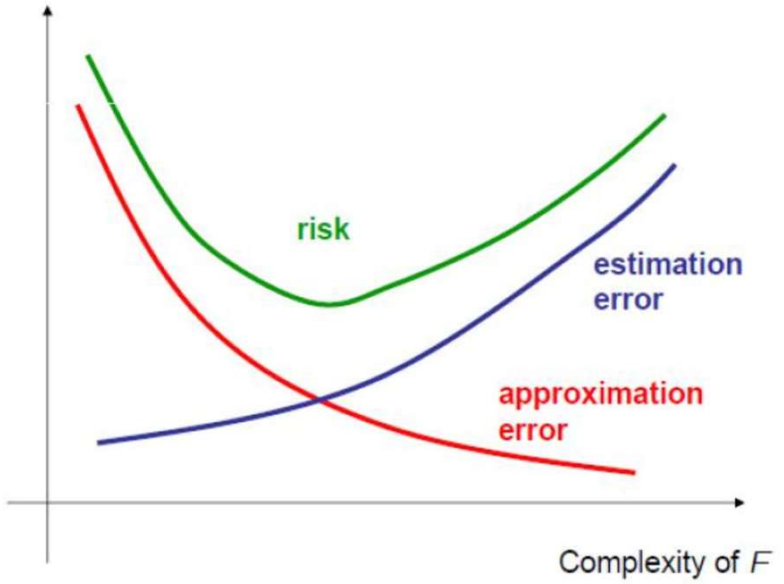


Figure Credit: Bin Zhao

Hypothesis Class for Linear Regression

What is the hypothesis class of linear regression?

Linear functions

- All possible linear functions encoded by parameters **w**
- Hypothesis chosen by setting parameter values **w**

How large is the hypothesis class?

- Infinite

What is Learning? Another View

Select a hypothesis from the hypothesis class

- Model parameters correspond to hypotheses
- Learn parameters of model based on data

How do we write learning algorithms?

- Theory: Objective driven
  - Write an objective that you want to minimize
  - Develop a procedure to minimize the objective
    - This is called the learning algorithm
- Empirical: intuition driven
  - Many algorithms based on motivation and heuristics
  - Post hoc analysis of objective

Learning and Error

Recall our Definition

Fitting a function to data

Fitting: Solve for w given y and x

Function: linear function

Data: assume dependent variable linear combination of independent variables

Minimize a function

- **What function are we minimizing?**

Goal of Learning

Our goal is to correctly predict the next example

- Minimize: prediction error
- On what?

True error:

$$\text{error}_D(h) = P_{x \in D}[l(h(x_i), y_i) \neq 0]$$

- Requires infinite data to measure

It's tough to make predictions, especially about the future. – Yogi Berra

Goal of Learning

If we can't measure true error, how do we judge learning success?

Should an algorithm maximize performance on observed data?

Proposal: measure error on the given data

- Call this the "training data"
- Is this a good idea?

Measuring Error

Very bad idea

Machine learning cares about prediction (the future)

- How well will the system do once deployed?

Memorizing training data is easy

- Most hypothesis classes are rich enough to exactly learn the training data

Can be Difficult to Understand

People routinely make this mistake!

Actual true stories

- Project accepted to predict hot real estate markets
  - Promising because very high accuracy
  - Problem: measured error on training data
- Paper submission to major conference claimed to have solved problem with 100% accuracy
  - Trained on the test data
- Paper submission showed high accuracy on classification task
  - Data *written* by researchers with task in mind

Bayes Optimal Rule

Ideal goal: Construct a prediction rule

$$f^* : \mathcal{X} \to \mathcal{Y}$$

Bayes optimal rule:

$$f^* = \arg\min_f \mathbb{E}_{XY}[\text{loss}(f(X), Y)]$$

Best possible performance (Bayes risk)

$$R(f^*) < R(f) \quad \forall f$$

But the optimal rule is not computable!

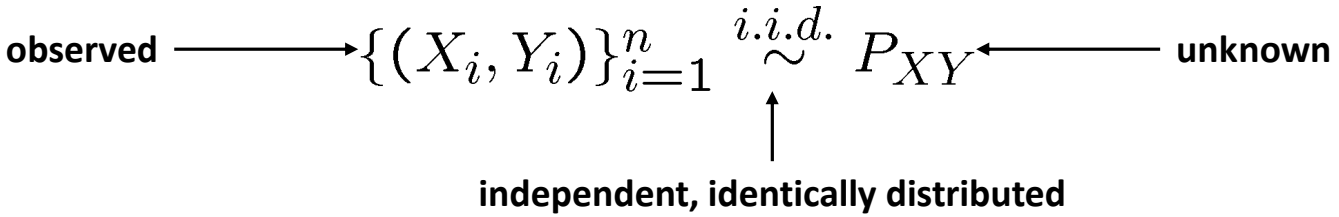
- It depends on an unknown distribution of X and Y

Training Data as Experience

Can't minimize the risk since P_{XY} is unknown in reality

Training data gives us a glimpse of P_{XY} through a sample

- Learning from experience
- Data can come from an expert, a measuring device, or some sort of experiment



Performance Revisited

Performance of a learning algorithm:
How well does the algorithm do on average?

For a single test input drawn at random?

For a set of data points drawn at random?

Expected Risk (a.k.a. Generalization Error): $D_n = \{(X_i, Y_i)\}_{i=1}^n$

$$\mathbb{E}_{D_n}\left[R(\widehat{f}_n)\right] \equiv \mathbb{E}_{D_n}\left[\mathbb{E}_{XY}\left[\text{loss}(Y, \widehat{f}_n(X))\right]\right]$$

Generalization

Generalization is the ability of an algorithm to generalize knowledge learned from observed data to new data

Simple example: memory-based classifier

- Binary classification task
- Train: remember each sample
- Test: if seen before, report label, else guess randomly
- Yields train error of 0% and test error of 50%
  - Over-fitting!

Bias vs. Variance

How do we achieve good generalization?

Tradeoff between bias and variance

- BIAS: favor certain predictions
- VARIANCE: diversity of predictions

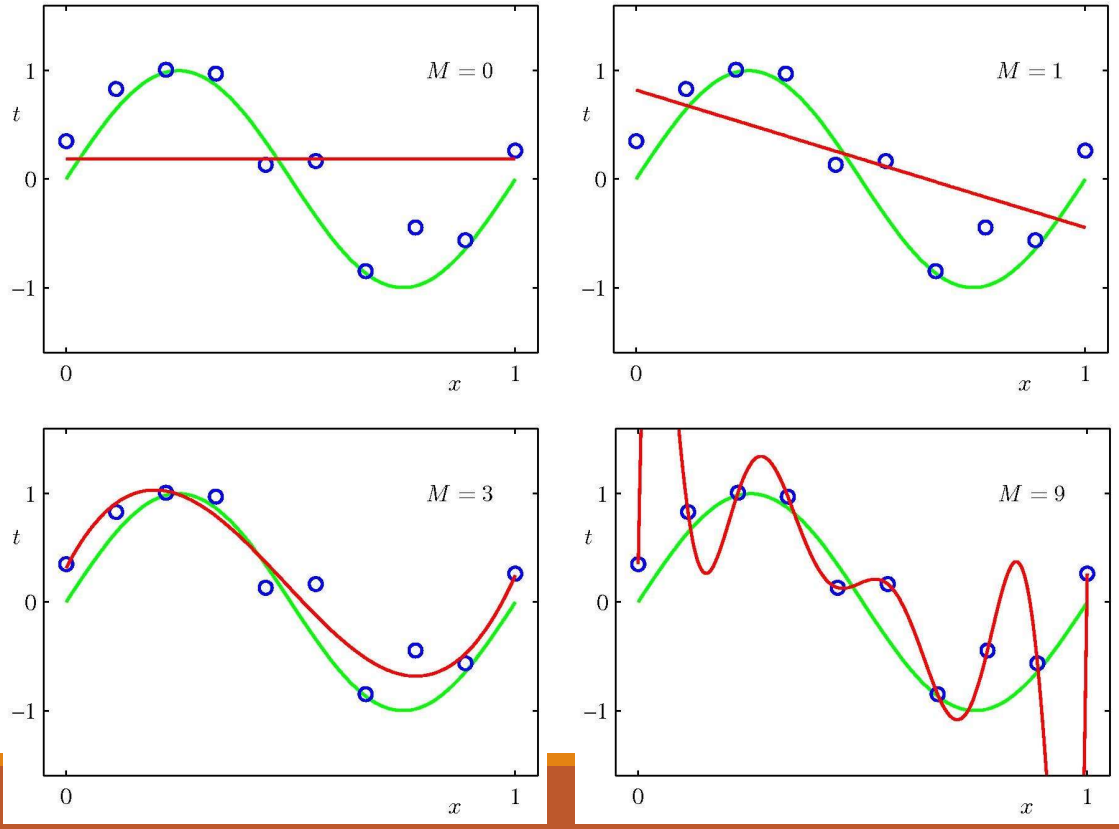
Under-fit observed data

- Favors **bias**: large changes to input have same output

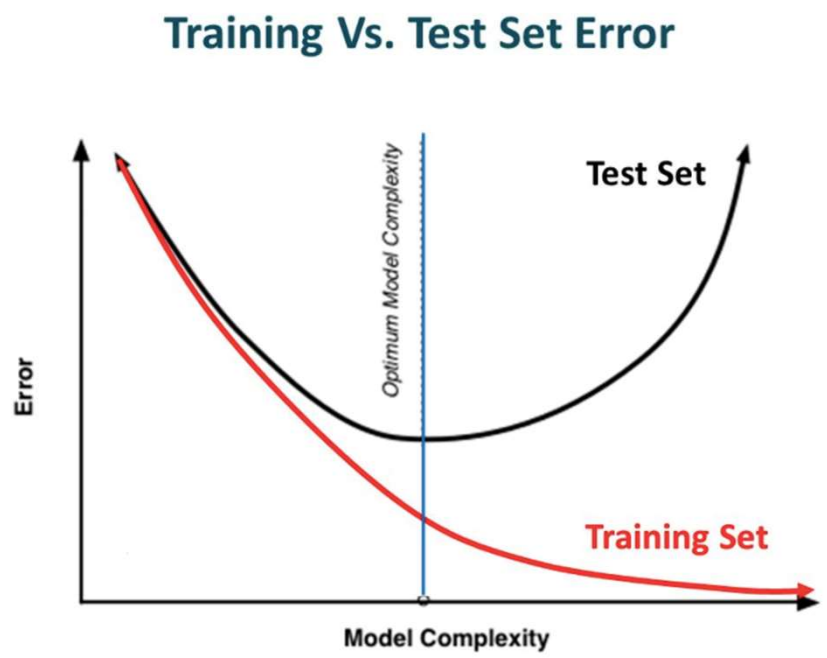
Over-fit observed data

- Favors **variance**: small changes to input have large changes in output

Under/Over-Fitting



Typical Learning Curves



No Free Lunch Theorem

Models make assumptions about the world

- An assumption-less model — allows every hypothesis — would be easily distracted by unnecessary detail and have difficulty learning
- We make assumptions to focus the models on what's important
- These assumptions limit the model if we are wrong

There is no single model that works best for every problem

Measuring Error the Right Way

Collect 2 sets of data

- Training data – used for fitting the model with an algorithm
- Test data – only used for evaluation of model performance
  - Only good if never seen before, not if you continuously tune on it

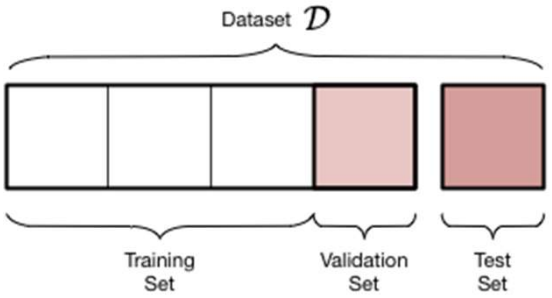
How do we balance bias/variance?

- Tune model hyper-parameters on development/validation data (a 3rd set)

Two Common Methods

Train/Dev/Test

- Use held-out sets for evaluation
- Good when you have a lot of data
- Make sure these are random/representative samples!



Cross-validation

- Create many randomly sampled train/test splits
- Divide data into folds and use each fold for testing once, rest as training
- Good when you don't have as much data
- Gets statistics on variation of model fit quality

Bayesian Model Selection

How do we determine the best model out of many?

A more efficient approach than cross-validation *may be* **Bayesian model selection**

$$\Pr(m|\mathcal{D}) = \frac{\Pr(\mathcal{D}|m)\Pr(m)}{\Pr(\mathcal{D})} \Rightarrow \frac{\Pr(m_0|\mathcal{D})}{\Pr(m_1|\mathcal{D})} = \frac{\Pr(\mathcal{D}|m_0)}{\Pr(\mathcal{D}|m_1)} \times \frac{\Pr(m_0)}{\Pr(m_1)}$$

$$\Pr(\mathcal{D}) = \sum_{m \in \mathcal{M}} \Pr(m, \mathcal{D}) = \sum_{m \in \mathcal{M}} \Pr(\mathcal{D}|m)\Pr(m)$$

Bayesian Occam's Razor

Won't this always favor the model with the most parameters?

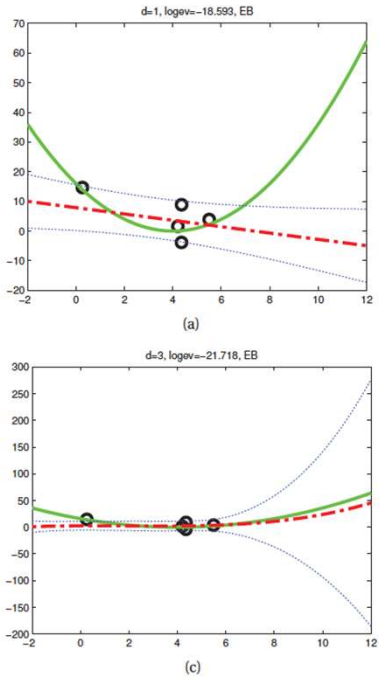
- True if we use the MLE or MAP estimate

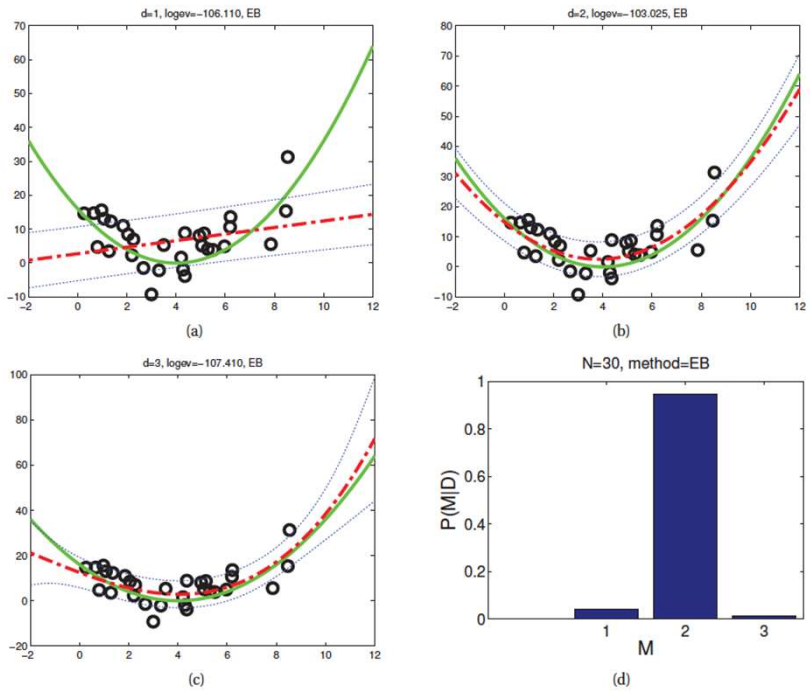
Integrating out the parameters – instead of maximizing – protects us from overfitting

- With more parameters, many options fit very poorly → correct solution is in a smaller prior volume

Bayesian Occam's Razor

N=5 data points





N=30 data points

Bayesian Information Criterion

Can we approximate a full Bayesian approach?

Yes!

$$\mathrm{BIC} = \mathrm{dof}(\hat{\boldsymbol{\theta}}) \log N - 2 \log \Pr(\mathcal{D}|\hat{\boldsymbol{\theta}})$$

Goal: minimize BIC

dof = degrees of freedom → penalizes more parameters

Larger penalty if more data so the fit must be better

Also AIC: $\mathrm{AIC} = 2\mathrm{dof}(\hat{\boldsymbol{\theta}}) - 2 \log \Pr(\mathcal{D}|\hat{\boldsymbol{\theta}})$

Where Does Error Come From?

Noise Error

- An example has an incorrect or inconsistent label
- Our data representation fails to encode necessary information

Parameter estimation error

- The model parameters are wrong

Model error

- Hypothesis class is deficient

Search error

- We made a mistake in scoring a prediction
- Common in tasks with complex output

Recall our Definition

Fitting a function to data

Fitting: Solve for w given y and x

Function: linear function

Data: assume dependent variable linear combination of independent variables

Minimize a function

- What function are we minimizing?

Linear Regression

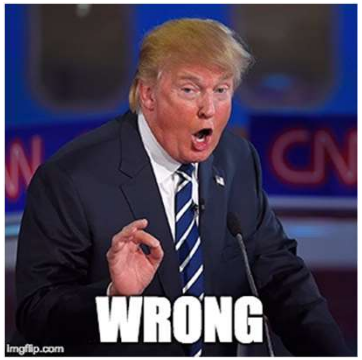
We assumed that the output (*y*) is a linear combination of the inputs (**x**)

This is wrong!

- Rarely is data actually linear

A realistic assumption may be too complex

Is there a reasonable middle ground?




NOISE!