CSC411
Tutorial #2
Sept. 2013

- A few of the fundamental ML concepts
- Linear regression demo (Matlab)

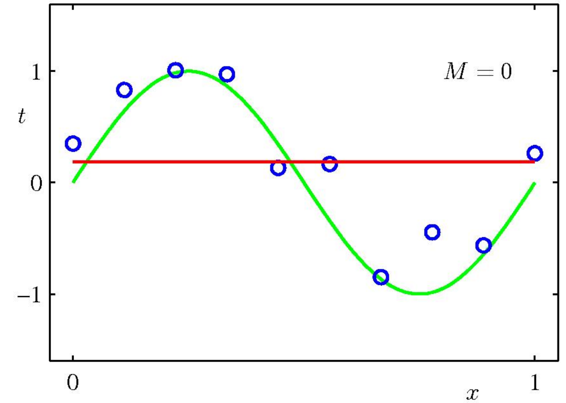
Tutor: Nitish Srivastava

Generalization

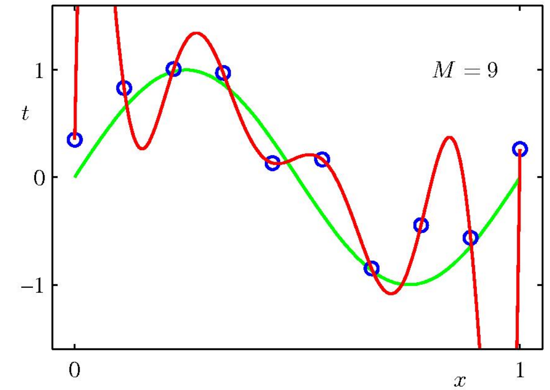
- The **generalization** of a machine learning algorithm is the performance (classification, regression, density estimation) on test data, not used for training, but drawn from the same distribution as the training data.
- Generally, our real goal is to get good generalization!

Overfitting and Underfitting

•When our model is not complex enough, it cannot capture the structure in our data no matter how much data we give it (underfitting/model bias)



•When our model is too complex for the amount of training data we have, it memorizes parts of the noise as well as learning the true problem structure (overfitting/model variance)



Model Selection & Performance Estimation

- Model Selection: out of a set of models (or continuum of model complexity), choose the model which will perform the best on future test data

- Model Assessment: for the selected model, **estimate** its generalization error on new data

- How do we go about selecting and assessing our models?

If we have lots of data

- Do model selection and assessment by dividing data into 3 parts:
  - –**Training Data**: used to train each model
  - –**Validation Data**: used to measure performance of each trained model in order to select best
  - –**Assessment (Test) Data**: used only once! On the final selected model, estimate performance on future test data.
- Typical split: 60%, 20%, 20%

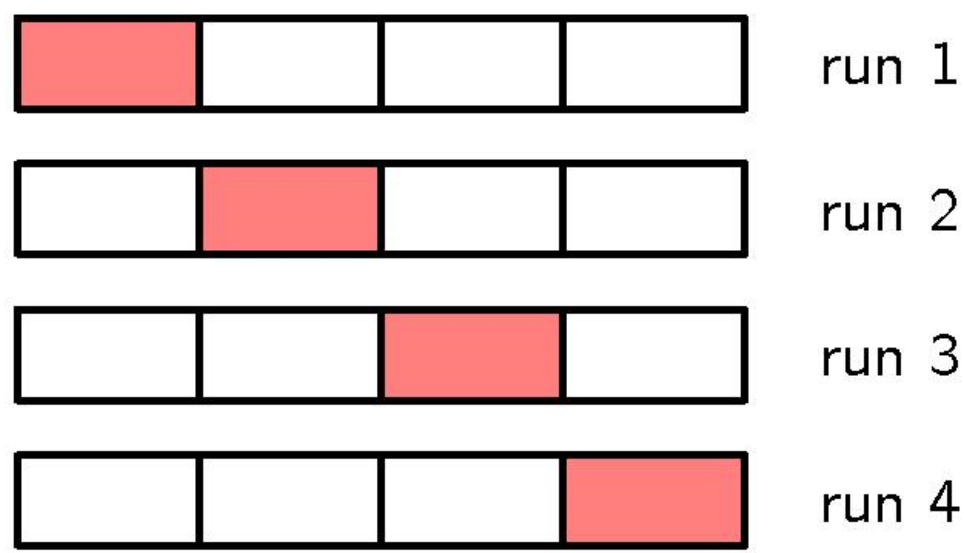
If we don't have lots of data?

- **Approximate** the results of validation and assessment

- Two approaches:
  - –Analytic: derive algebraic expressions which try to approximate the test errors (BIC, MDL, VC-dim)
  - –Empirical: (Sample-recycling methods) try to estimate the test error computationally, using the same data that we trained on (cross validation, boot-straping)

Cross-Validation

- Instead of setting aside a separate validation set, we can leave out part of our data, train on the rest, measure errors on the part we left out, then repeat, leaving out a different bunch of data
- K-fold CV
- LOO (K=N)



Some Issues with CV

- Intensive use of CV can over fit if you explore too many models, by finding a model that accidentally predicts the whole training set well (and thus every LOO sample).
- Time consuming (always/if done naively)
  - There are efficient tricks that can save work over brute force

Regularization

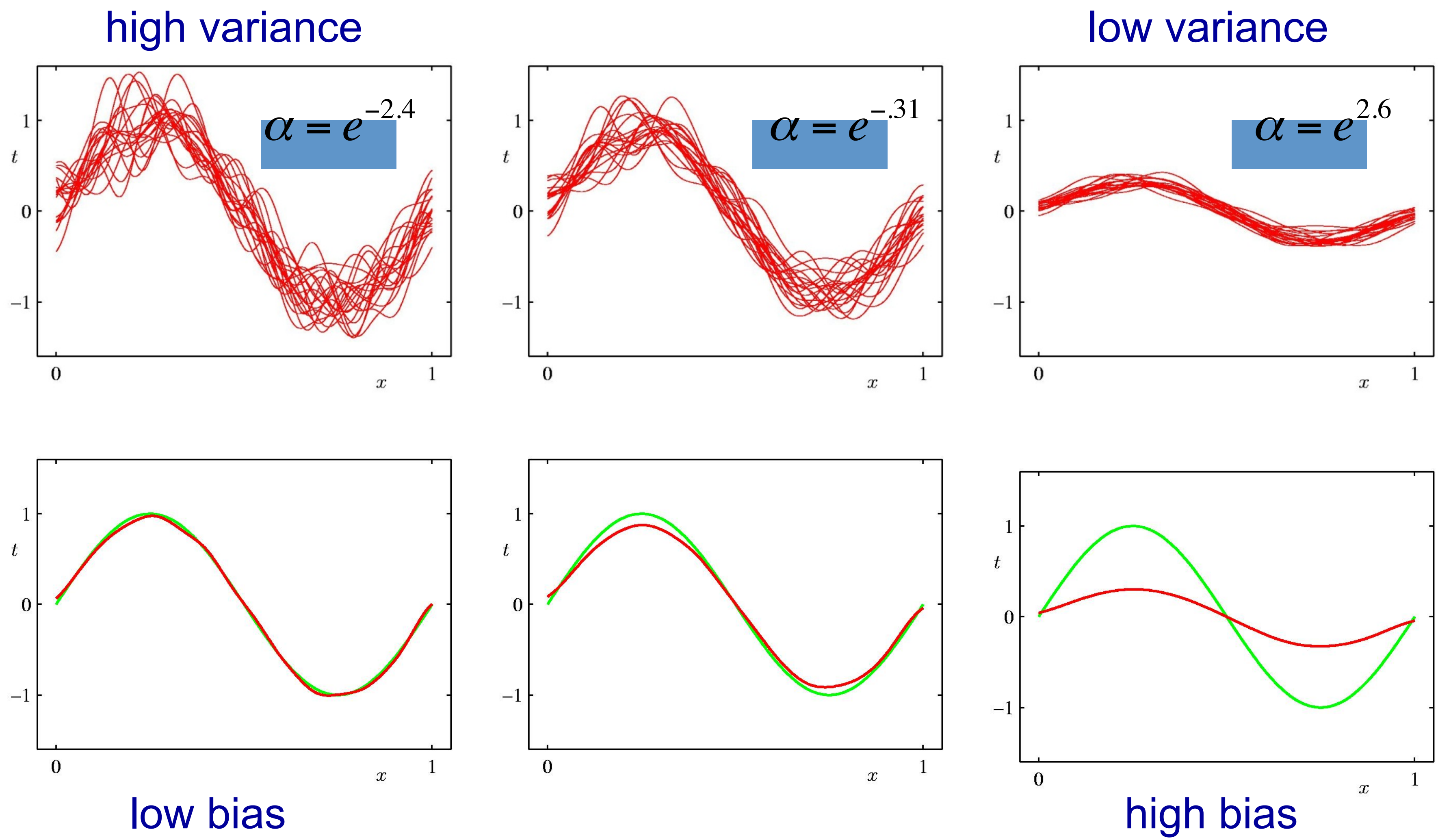
- You saw in class the formal Bias-Variance decomposition
- Roughly speaking, for some test point x
  - <e(x)> = Unavoidable error + Bias2 + Variance
- How to improve generalization? Reduce variance (simpler models) while not increasing bias too much (not too simple a model).
- We need a knob to control this tradeoff
  - Discretely constraining model structure/continuously regularizing model complexity or smoothness
- We need a way to set this knob
  - Decide on the right tradeoff

Regularization

$$\tilde{J}(\mathbf{w}) = \sum_{n=1}^{N} \{y(\mathbf{x}^{(n)}, \mathbf{w}) - t^{(n)}\}^2 \; + \; \alpha \mathbf{w}^T \mathbf{w}$$

- Alpha is a knob to control this tradeoff
  - Discretely constraining model structure/ continuously regularizing model complexity or smoothness

How the regularization parameter affects the bias and variance terms



An example of the bias-variance trade-off