Programming with Data Bootcamp: Lecture 5

http://dsg.csail.mit.edu/6.S079/

Slides courtesy of Sam Madden / Tim Kraska (6.S079)

Key ideas:

Traditional ML
- Clustering
- Dim. Reduction
- Classification
- Regression

MACHINE LEARNING PROBLEMS

	Supervised Learning	Unsupervised Learning
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction

Machine Learnin

Nightmare series

What if you model has an high error?

- Try getting more training examples
- Try smaller sets of features
- Try getting additional features
- Try creating features from existing features (kernels)
- Try decrease regularization
- Try increase regularization

What Error/Quality Metric to use?

Classification:

- **Accuracy**
- **F-score**
- **F1-micro**
- **F1-macro**
- ROC AUC (micro, macro)
- …

Regression

- Mean-Squared Error
- Root-Mean Squared Error
- Mean absolute Error
- R^2
- Cohen Kappa
- ..

Precision, Recall, Accuracy

	True	False
True	tp	fp
False	fn	tn

- **Precision**: correctly identified positive cases

  Precision $P = tp/(tp + fp)$

- **Recall**: correctly identified positive cases from all the actual positive cases.

  Recall $R = tp/(tp + fn)$

- **Accuracy:** measure of all the correctly identified cases

  Accuracy $R = (tp+tn)/(tp + fp + fn + tn)$

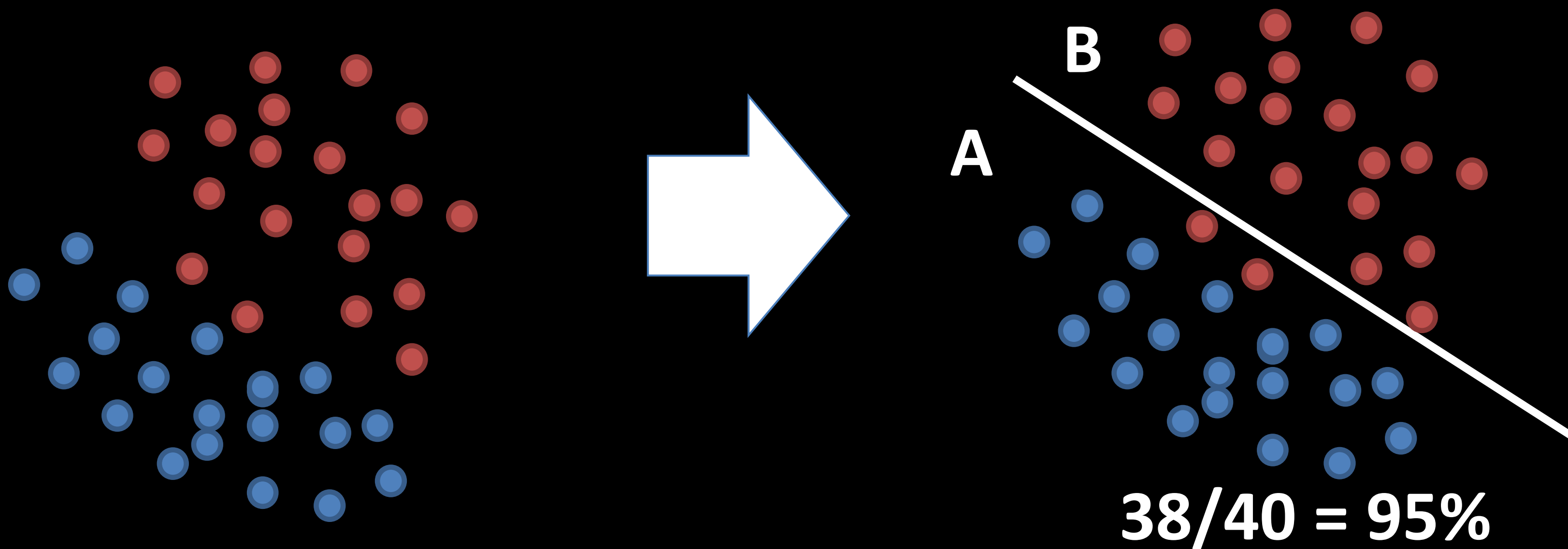
Evaluation:
Accuracy isn't always enough

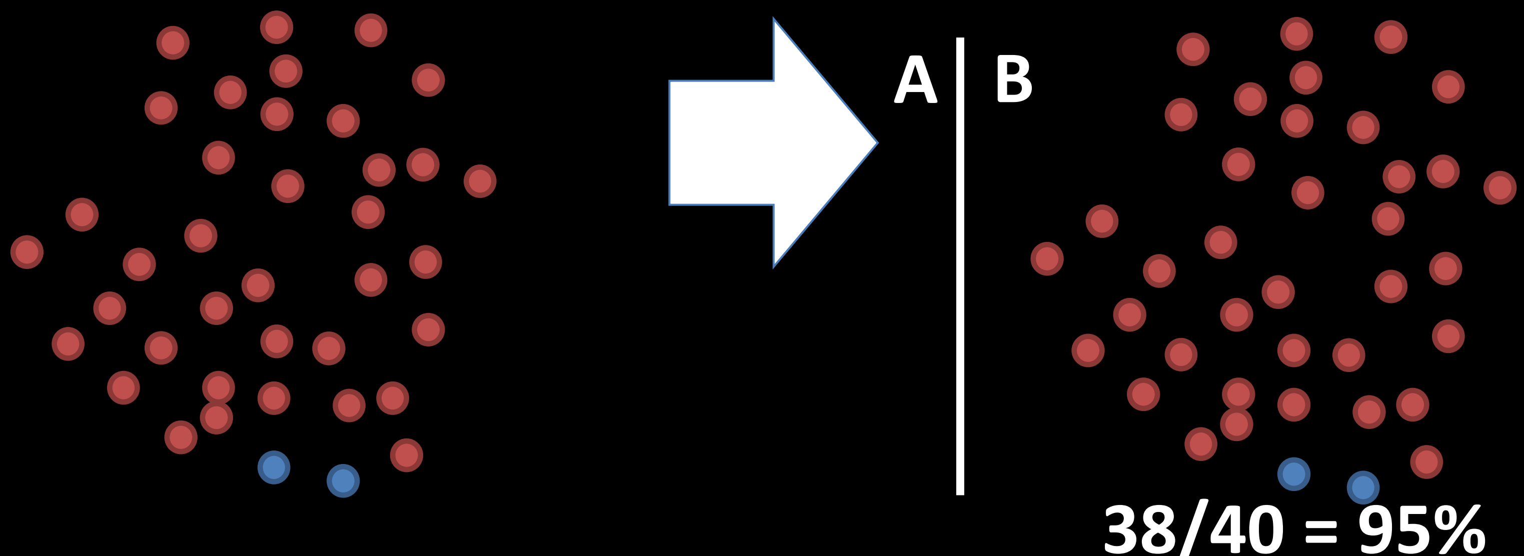
- How do you interpret 90% accuracy?

Evaluation: Accuracy isn't always enough

- How do you interpret 90% accuracy?
  - You can't; it depends on the problem
- Need a baseline:
  - Base Rate
    - Accuracy of trivially predicting the most-frequent class
  - Random Rate
    - Accuracy of making a random class assignment
    - Might apply prior knowledge to assign random distribution
  - Naïve Rate
    - Accuracy of some simple default or pre-existing model
    - Ex: "All females survived"

Why Optimize? Pitfalls





What Error/Quality Metric to use?

Classification:

- Accuracy
- **F-score**
- F1-micro
- F1-macro
- ROC AUC (micro, macro)
- …

Regression

- Mean-Squared Error
- Root-Mean Squared Error
- Mean absolute Error
- R^2
- Cohen Kappa
- ..

Precision, Recall, Accuracy

		True Label	
		True	False
Predicted Label	True	tp	fp
	False	fn	tn

- **Precision**: correctly identified positive cases

Precision P = tp/(tp + fp)

- **Recall**: correctly identified positive cases from all the actual positive cases.

Recall R = tp/(tp + fn)

- **F-Score:** is the harmonic mean of precision and recall

$$F = \frac{2}{\frac{1}{R}+\frac{1}{P}} = \frac{tp}{tp+\frac{1}{2}(fp+fn)}$$

F1 Micro

		True Label		
		L1	L2	L3
Predicted Label	L1	7	1	4
	L2	0	1	12
	L3	1	6	6

Precision micro: true positives for all the classes divided by the all positive predictions
Precision Score Micro = TP / (TP + FP)
TP = (7 + 1 + 6)
FP = 1 + 4 + 0 + 12 + 1 + 6

Recall micro: Sum of **true positives for all the classes** divided by the actual positives.
Recall Score Micro: TP / (TP + FN)

F1 Score: $\frac{tp}{tp+\frac{1}{2}(fp+fn)}$

Macro

F1 Macro

		True Label		
		L1	L2	L3
Predicted Label	L1	7	1	4
	L2	0	1	12
	L3	1	6	6

Precision micro: **arithmetic mean of all the precision scores** of different classes
Precision Score Macro = ((7/8) + (1/8) + (6/22))/3

Recall micro: **arithmetic mean of all the recall scores** .

When to use F1 Micro and when to use F1 Macro?

F1 Macro

		True Label		
		L1	L2	L3
Predicted Label	L1	7	1	4
	L2	0	1	12
	L3	1	6	6

Precision micro: **arithmetic mean of all the precision scores** of different classes
Precision Score Macro = ((7/8) + (1/8) + (6/22))/3

Recall micro: **arithmetic mean of all the recall scores** .

When to use F1 Micro and when to use F1 Macro?

- Micro weights each instance or prediction equally.
- Macro weights each class equally (better for imbalance of labels)
- Use weighted macro-averaging score in case of class imbalances (different number of instances related to different class labels).

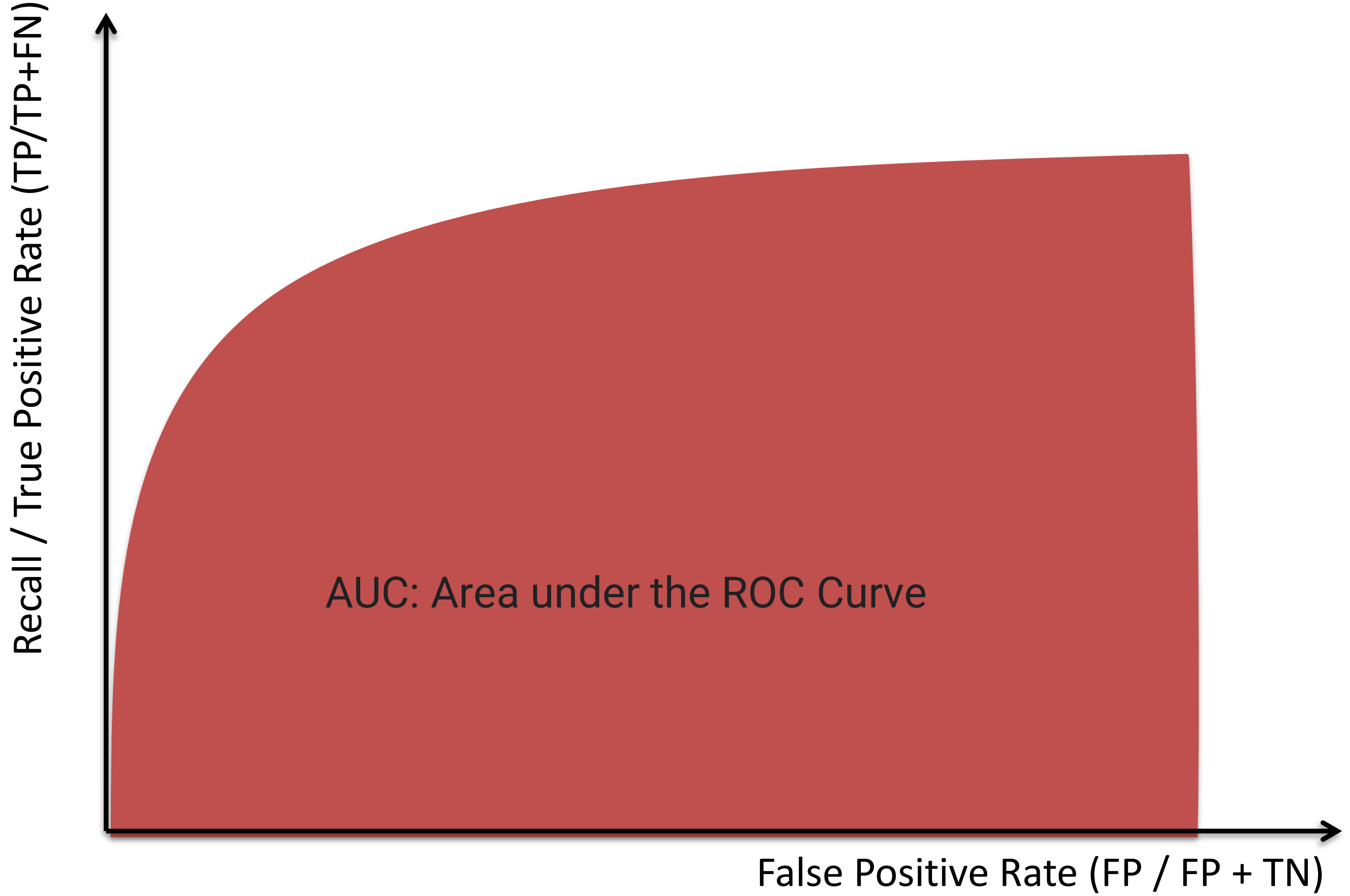
ROC AUC

(usually used for models with a threshold)

What would be the ideal ROC curve?
How would a random guess look like

ROC AUC

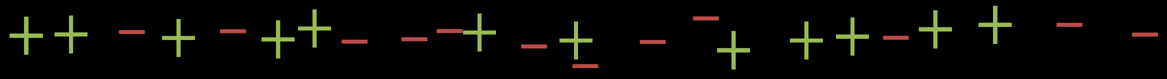
(usually used for models with a threshold)



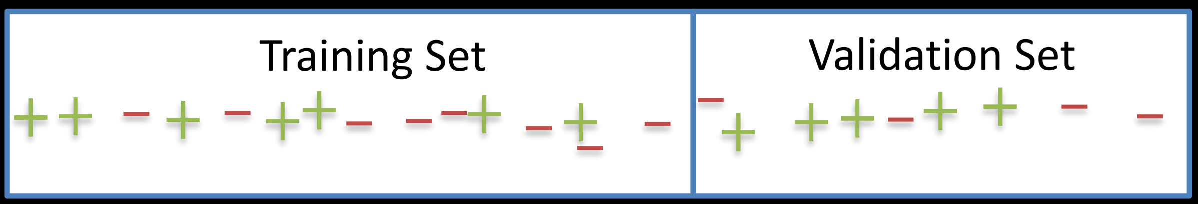
What if your model has a high error?

- Try getting more training examples
- Try smaller sets of features
- Try getting additional features
- Try creating features from existing features (kernels)
- Try decrease regularization
- Try increase regularization

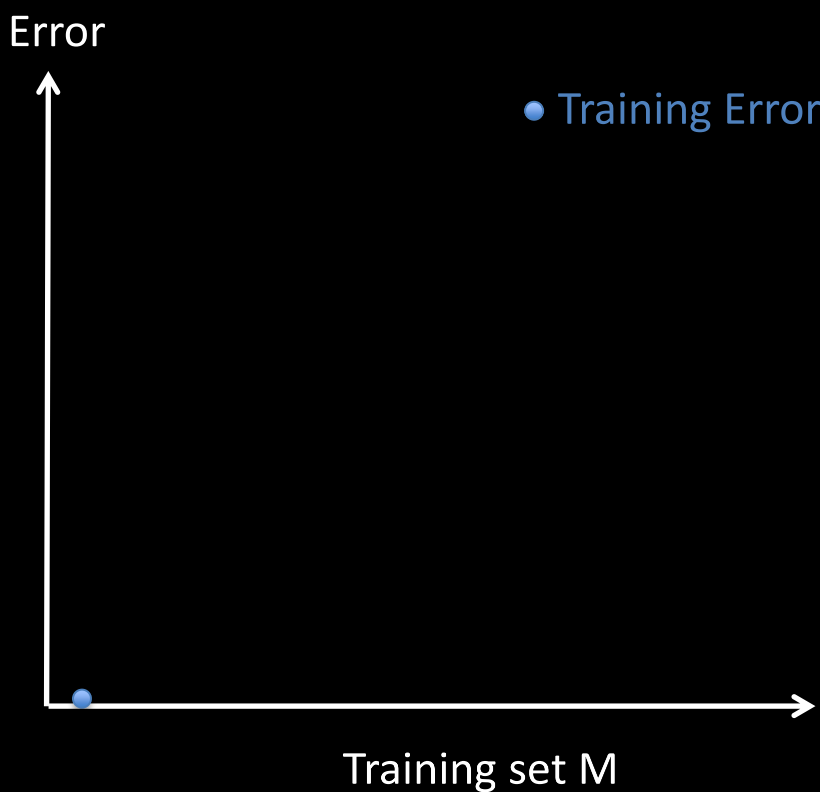
Bias and Variance

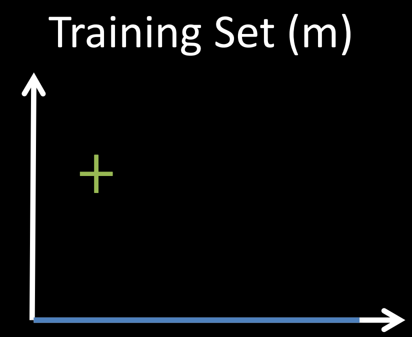


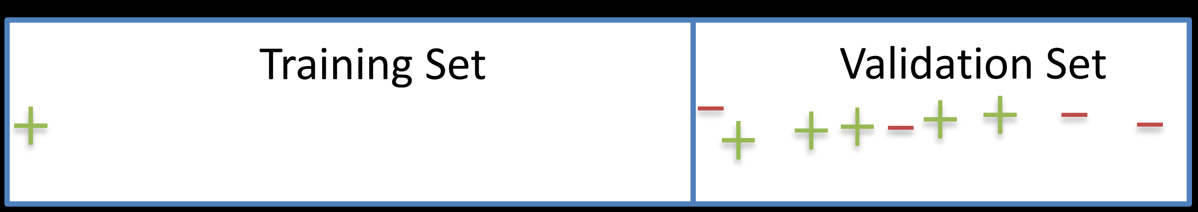
Bias and Variance



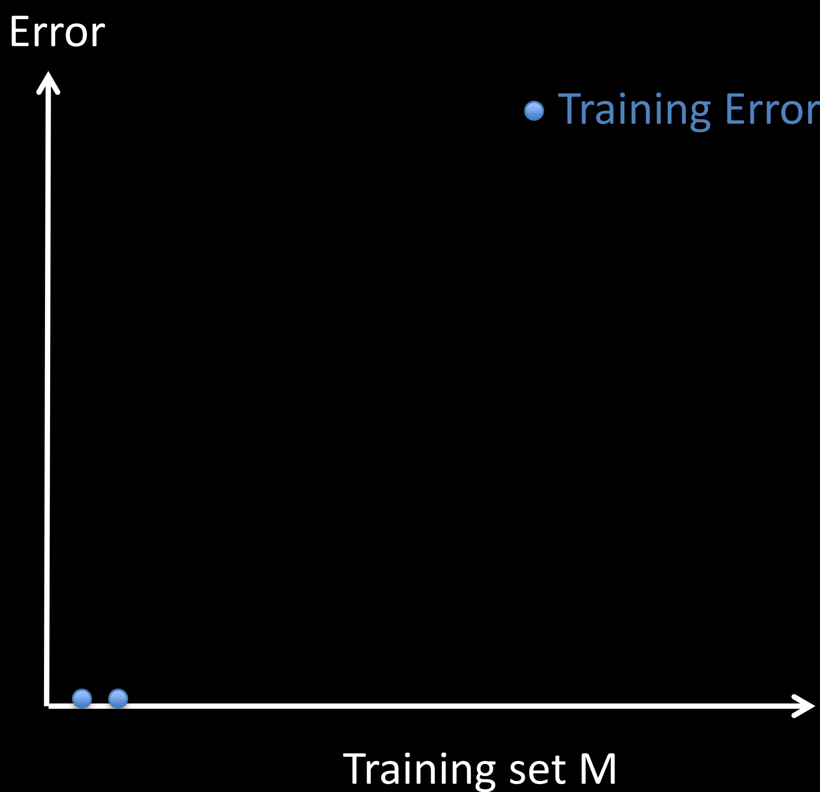
Bias and Variance

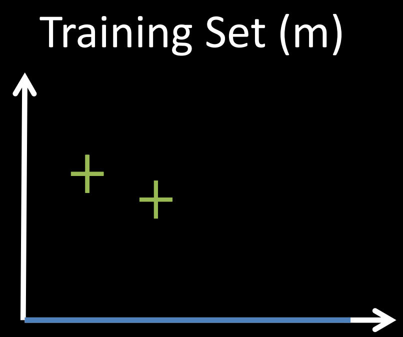


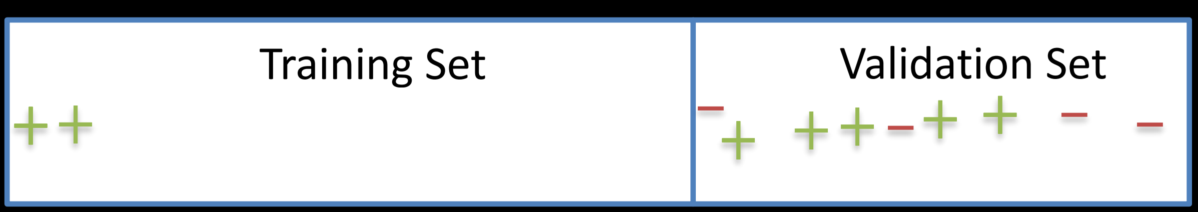




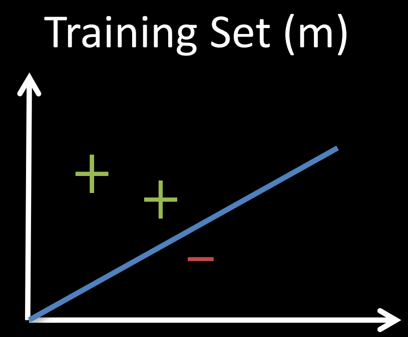
Bias and Variance



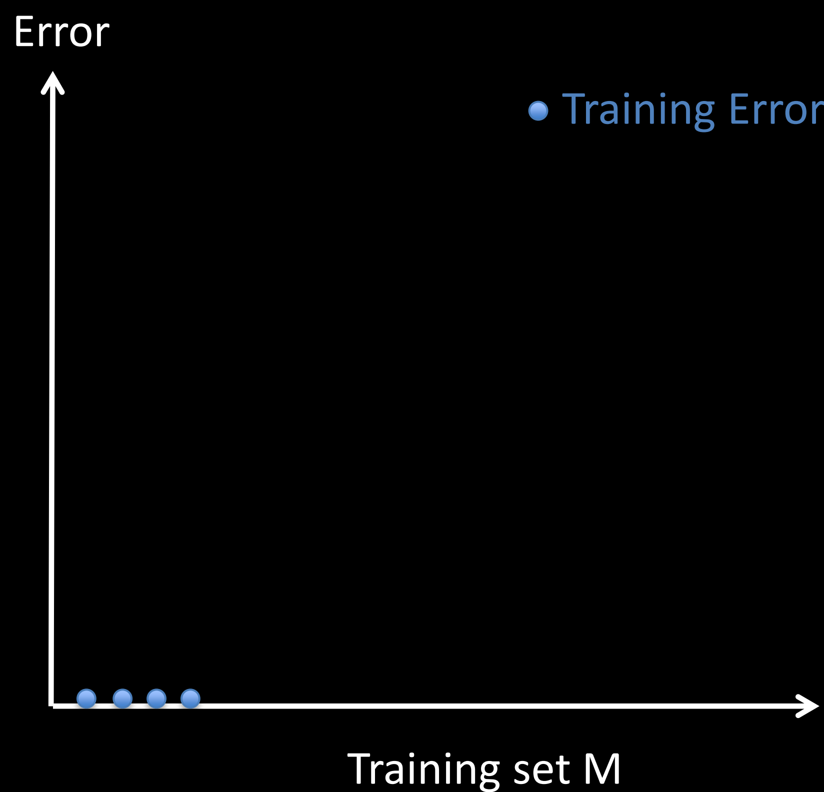


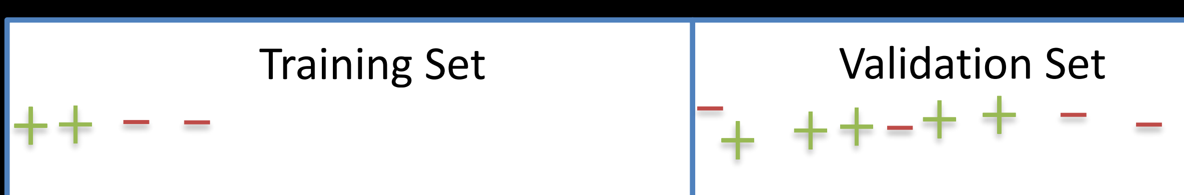


Bias and Variance

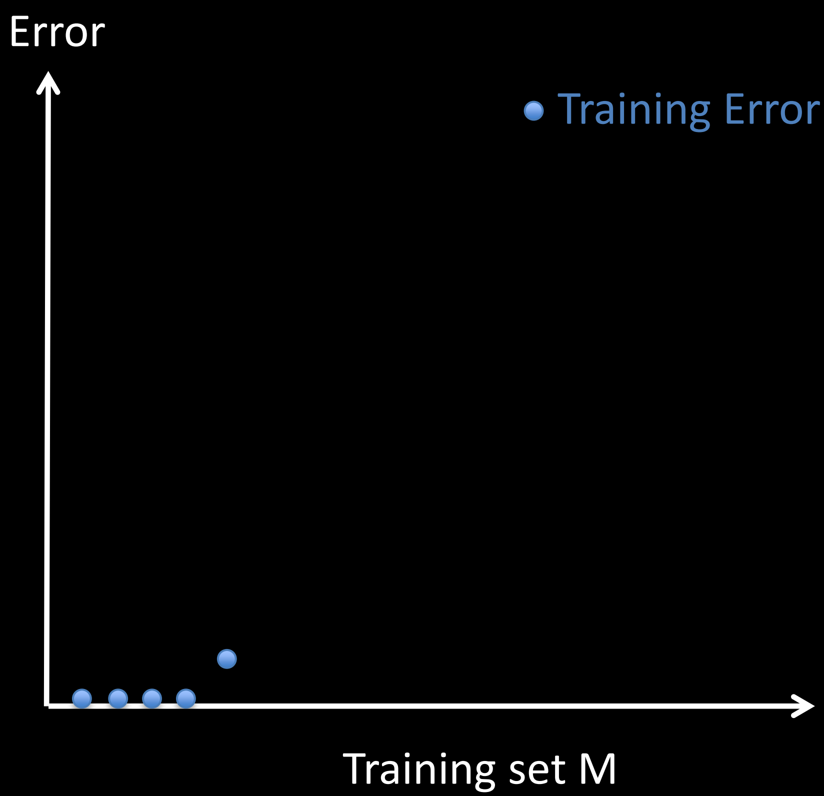


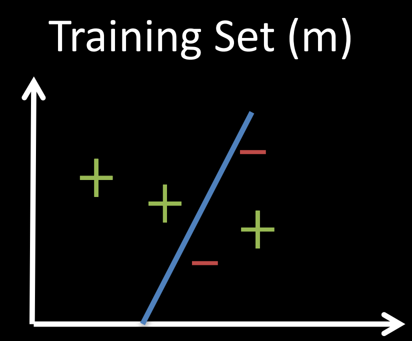
Bias and Variance



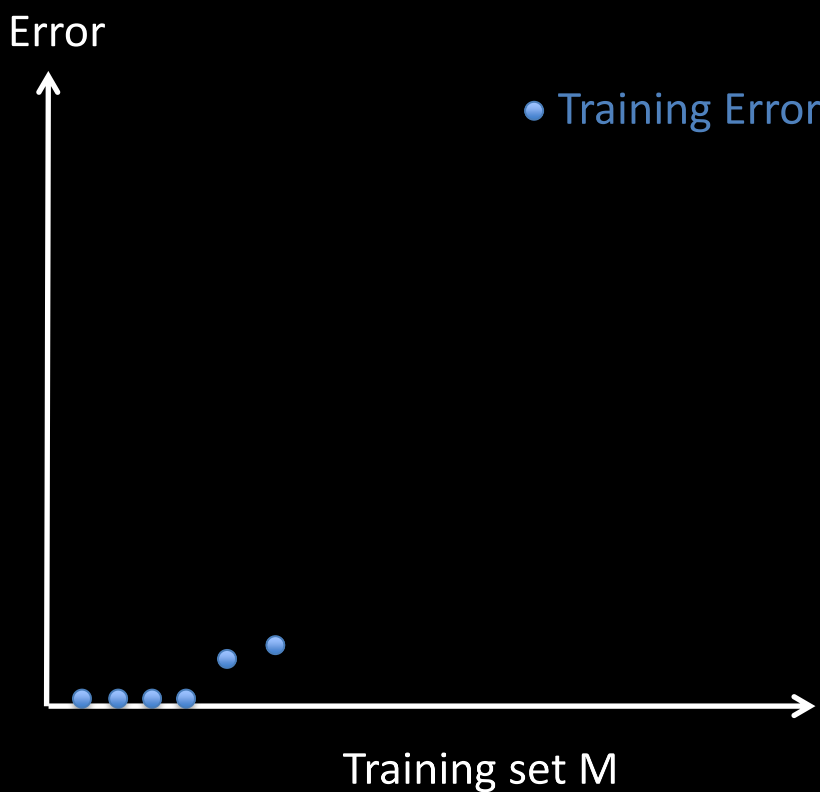


Bias and Variance

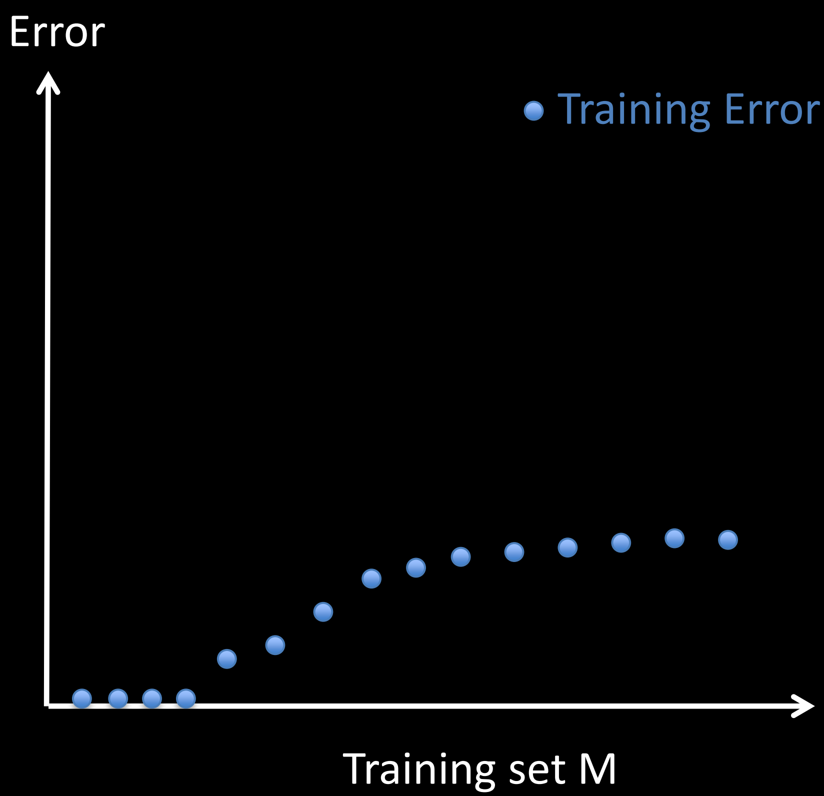


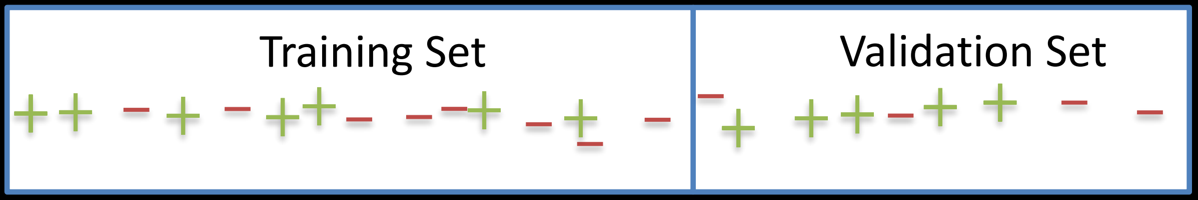


Bias and Variance

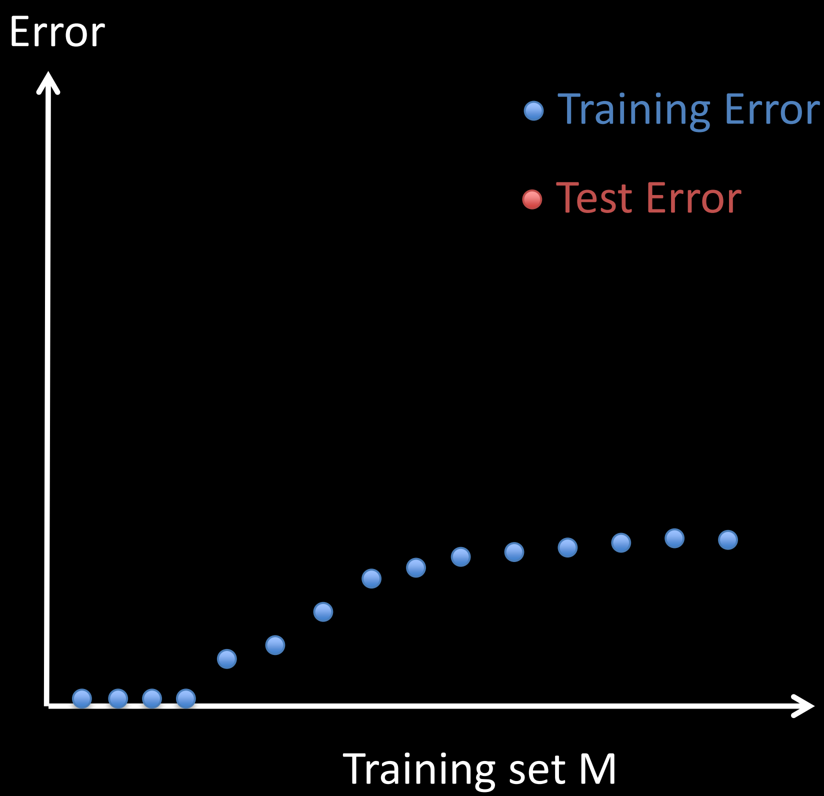


Bias and Variance





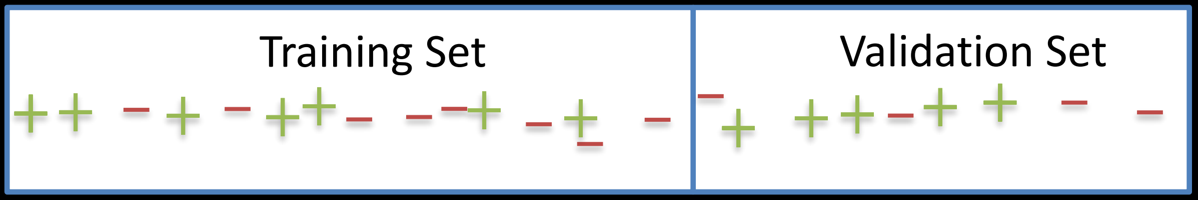
Bias and Variance



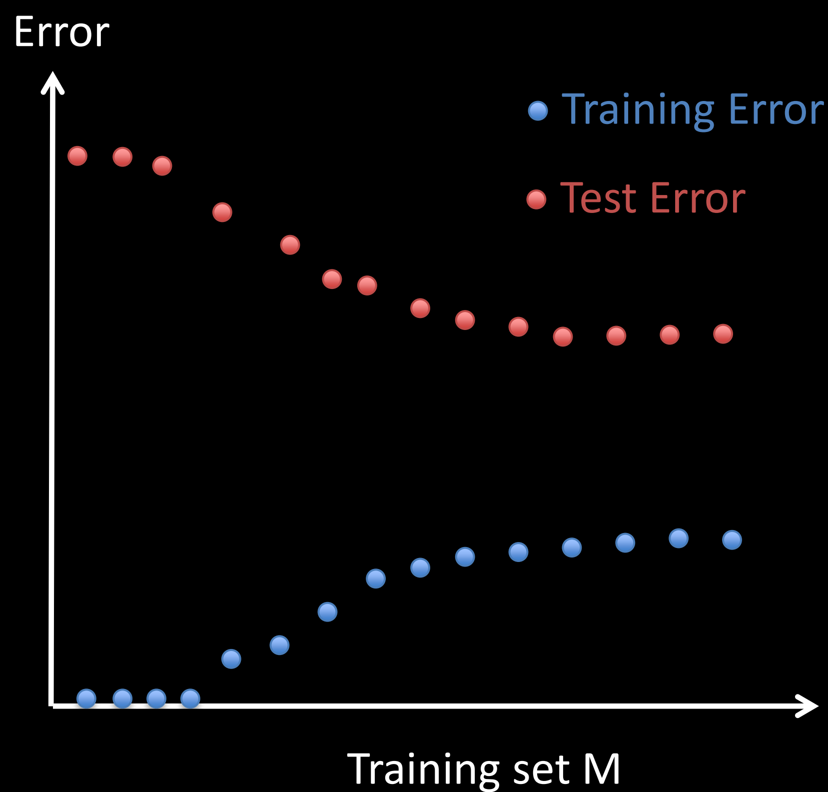
Clicker:

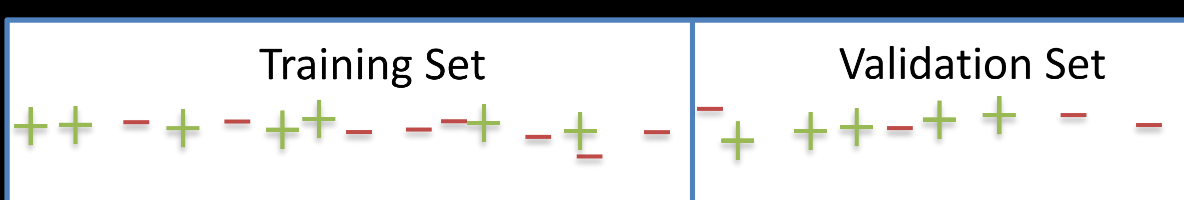
Test error

a) decreases with M
b) increases with M
c) stays constant

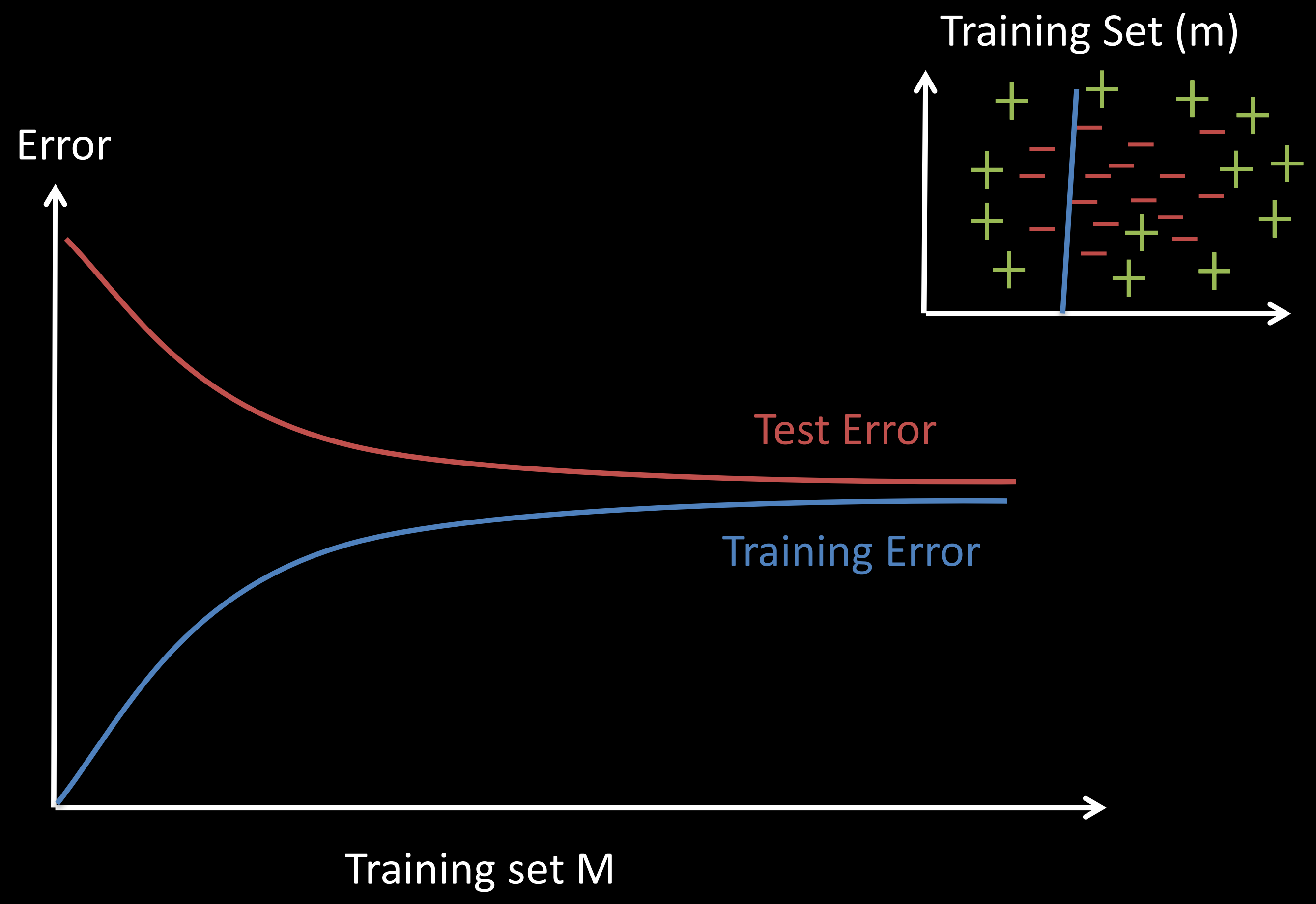


Bias and Variance

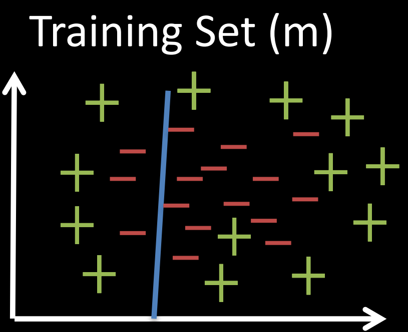




High Bias


Training Set (m)
Error
Test Error
Training Error
Training set M

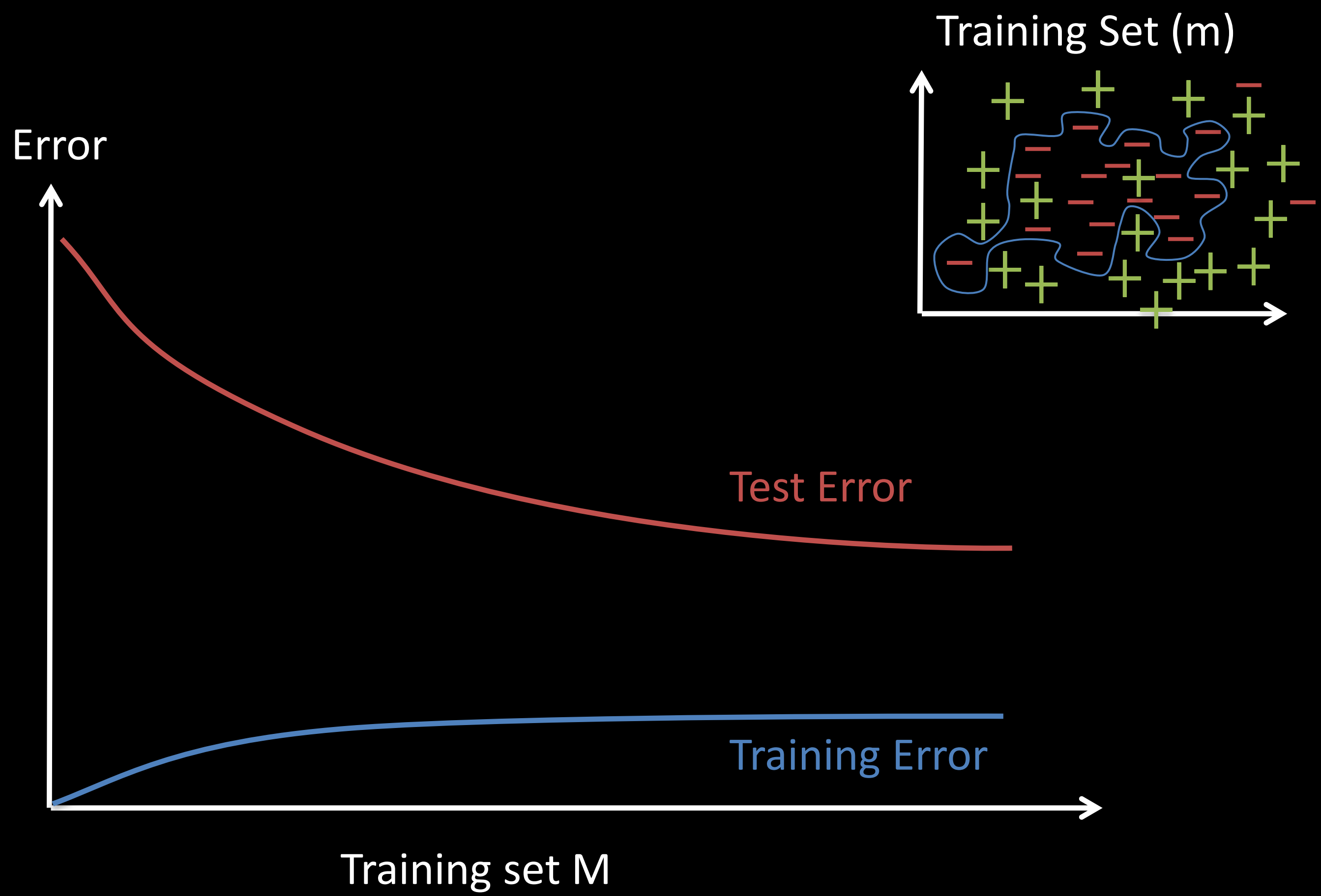
High Bias



Clicker: If you have high-bias, does more data help?

a) No
b) Yes

High Variance



Clicker: If you have high-variance, does more data help?

a) No
b) Yes

Clicker

1. **Get more training examples**
2. Try smaller sets of features
3. Try getting additional features
4. Try adding polynomial features (kernels)
5. Try increase regularization
6. Try decrease regularization

Helps with

A. High Variance
B. High Bias
C. Both
D. None

Clicker

1. Get more training examples
2. **Try smaller sets of features**
3. Try getting additional features
4. Try adding polynomial features (kernels)
5. Try increase regularization
6. Try decrease regularization

Helps with

A. High Variance
B. High Bias
C. Both
D. None

Clicker

1. Get more training examples
2. Try smaller sets of features
3. **Try getting additional features**
4. Try adding polynomial features (kernels)
5. Try increase regularization
6. Try decrease regularization

Helps with

A. High Variance
B. High Bias
C. Both
D. None

Clicker

1. Get more training examples
2. Try smaller sets of features
3. Try getting additional features
4. **Try adding polynomial features (kernels)**
5. Try increase regularization
6. Try decrease regularization

Helps with

A. High Variance
B. High Bias
C. Both
D. None

Clicker

1. Get more training examples
2. Try smaller sets of features
3. Try getting additional features
4. Try adding polynomial features (kernels)
5. **Try increase regularization**
6. Try decrease regularization

Helps with

A. High Variance
B. High Bias
C. Both
D. None

Clicker

1. Get more training examples
2. Try smaller sets of features
3. Try getting additional features
4. Try adding polynomial features (kernels)
5. Try increase regularization
6. **Try decrease regularization**

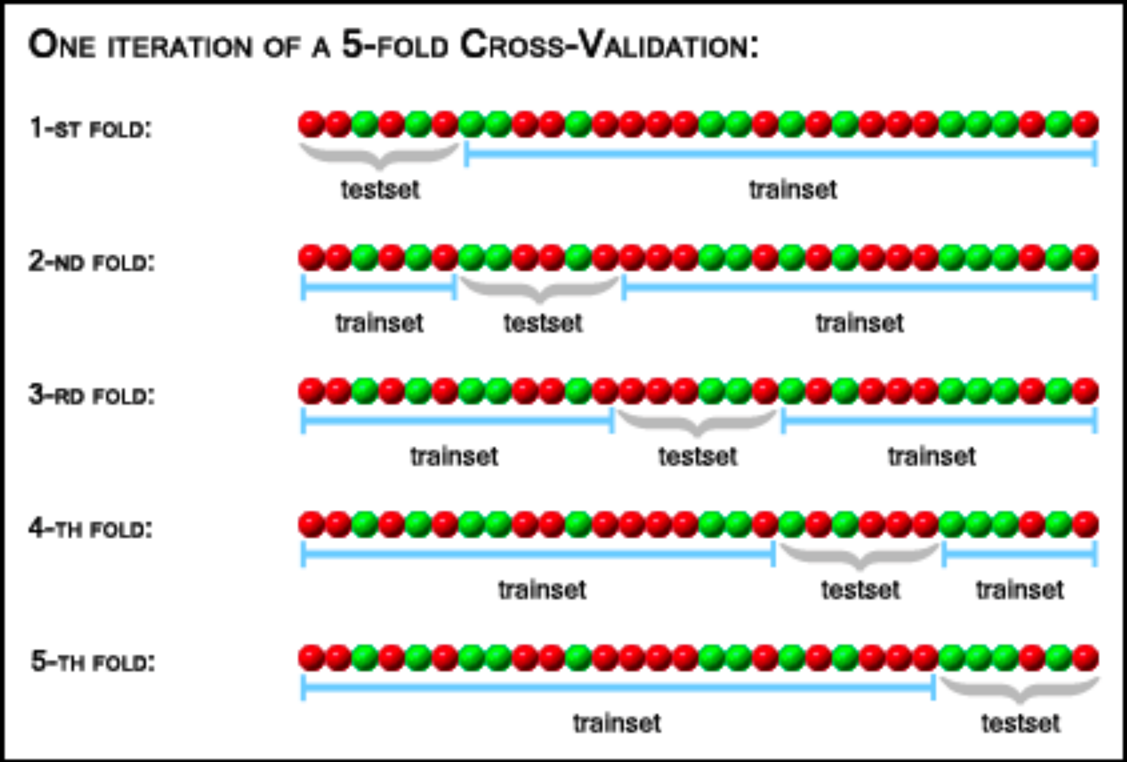
Helps with

A. High Variance
B. High Bias
C. Both
D. None

Cross-validation

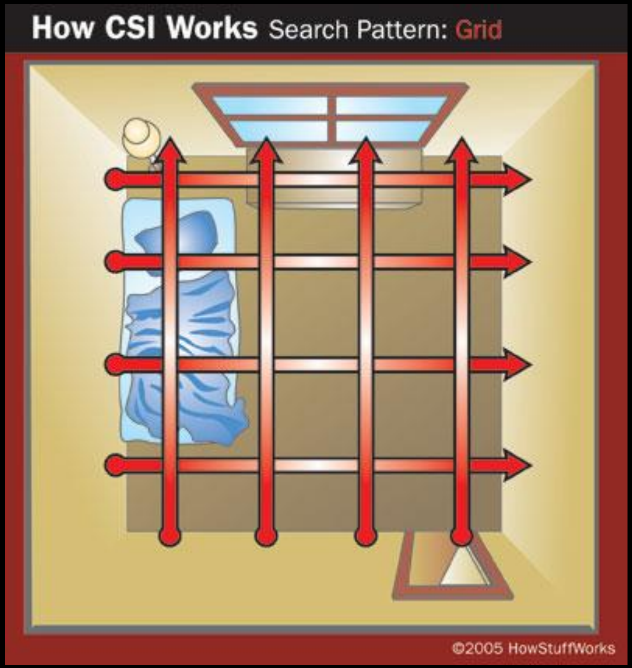
k-fold: split the data into k groups, train on every group except for one, which you test on.

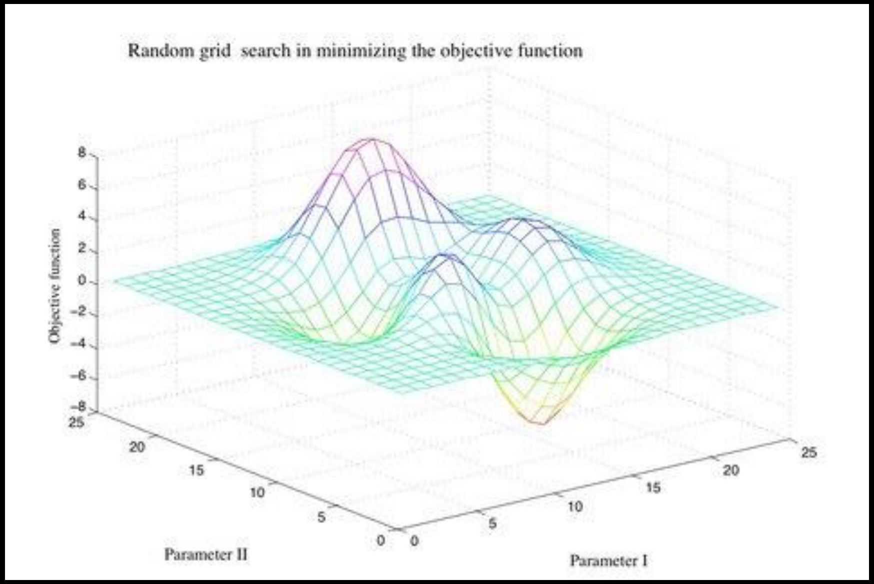
Repeat for all groups



Parameter Tuning

Grid Search





How to speed-up tuning?

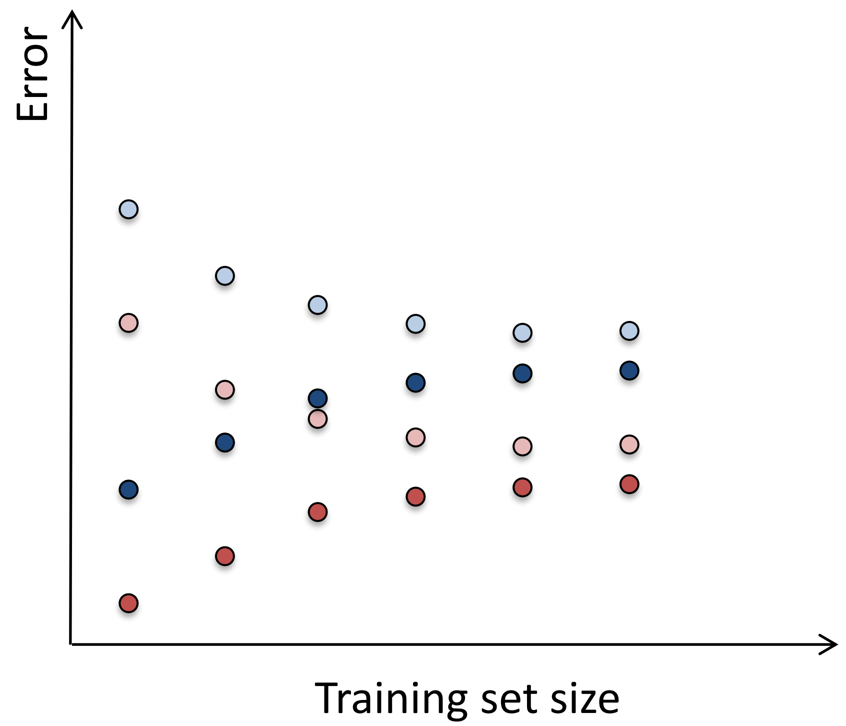
Can we use sampling?

Algorithm 1:

- Training
- Validation

Algorithm 2:

- Training
- Validation



How to speed-up tuning?

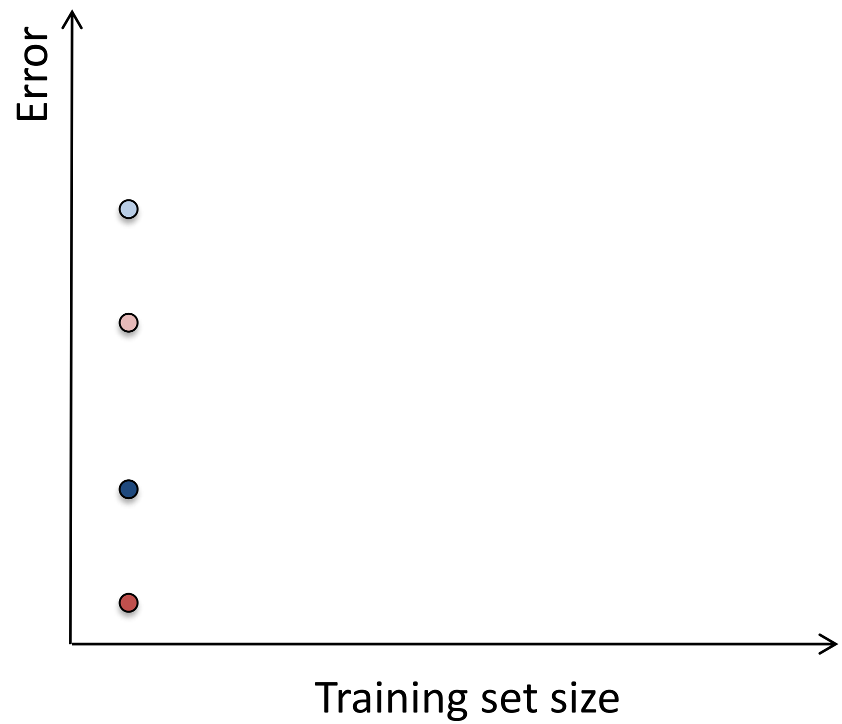
Can we use sampling?

Algorithm 1:

Training   Validation

Algorithm 2:

Training   Validation



How to speed-up tuning?

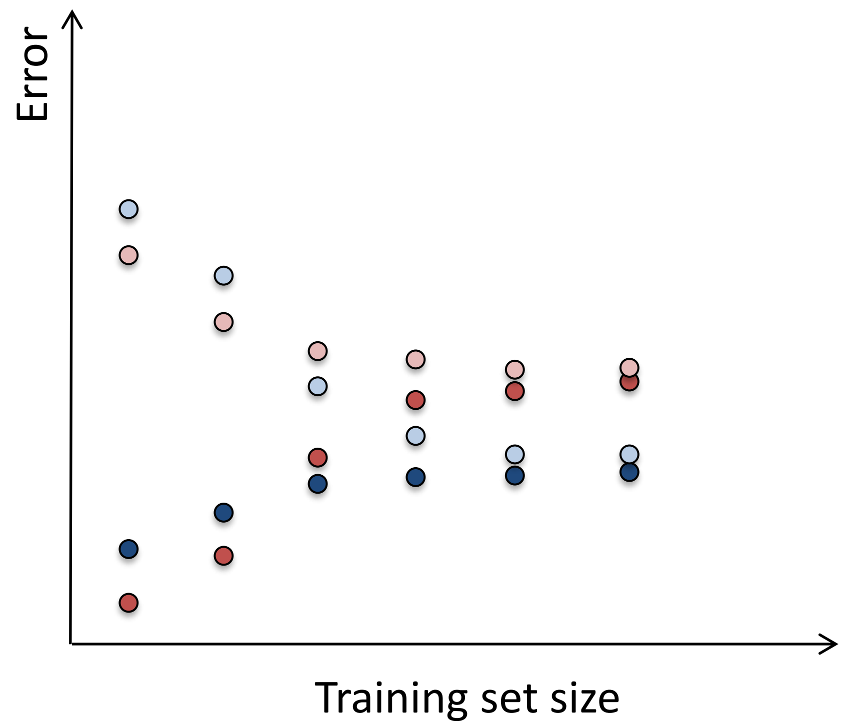
Can we use sampling?

Algorithm 1:

Training Validation

Algorithm 2:

Training Validation



How to speed-up tuning?

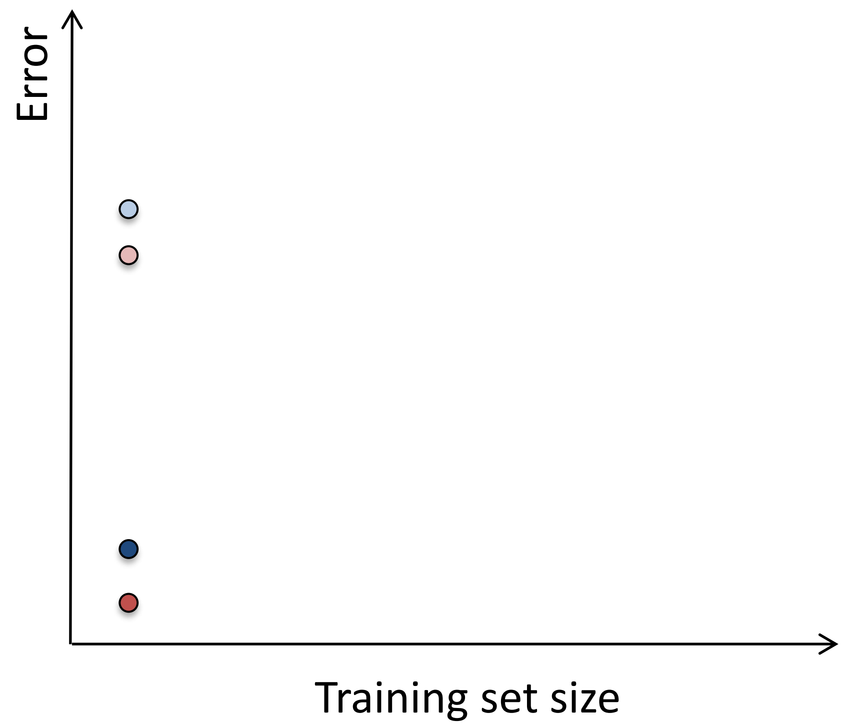
Can we use sampling?

Algorithm 1:

Training Validation

Algorithm 2:

Training Validation



How to speed-up tuning?

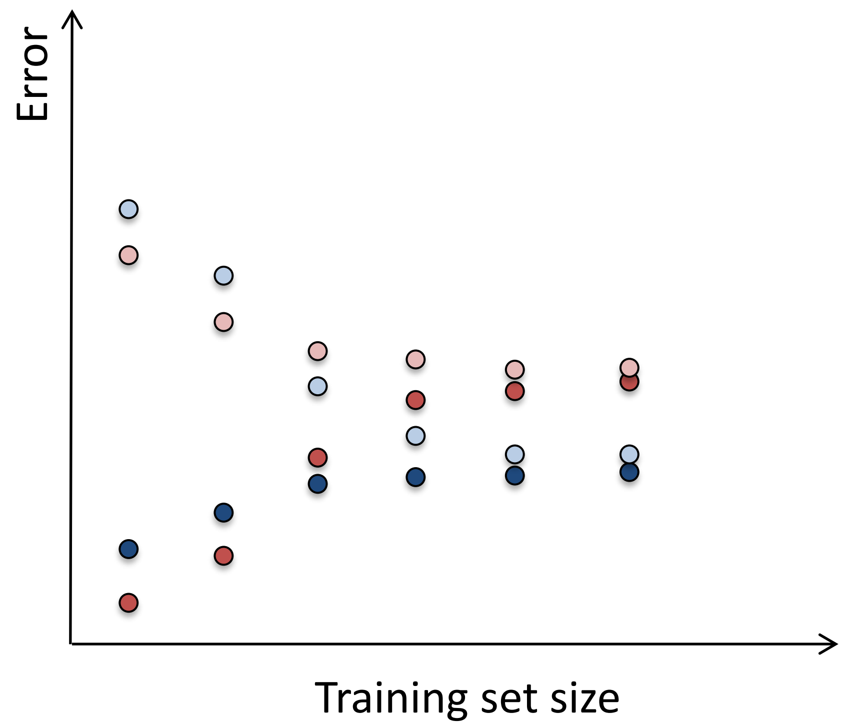
Can we use sampling?

Algorithm 1:

Training Validation

Algorithm 2:

Training Validation



How to speed-up tuning?

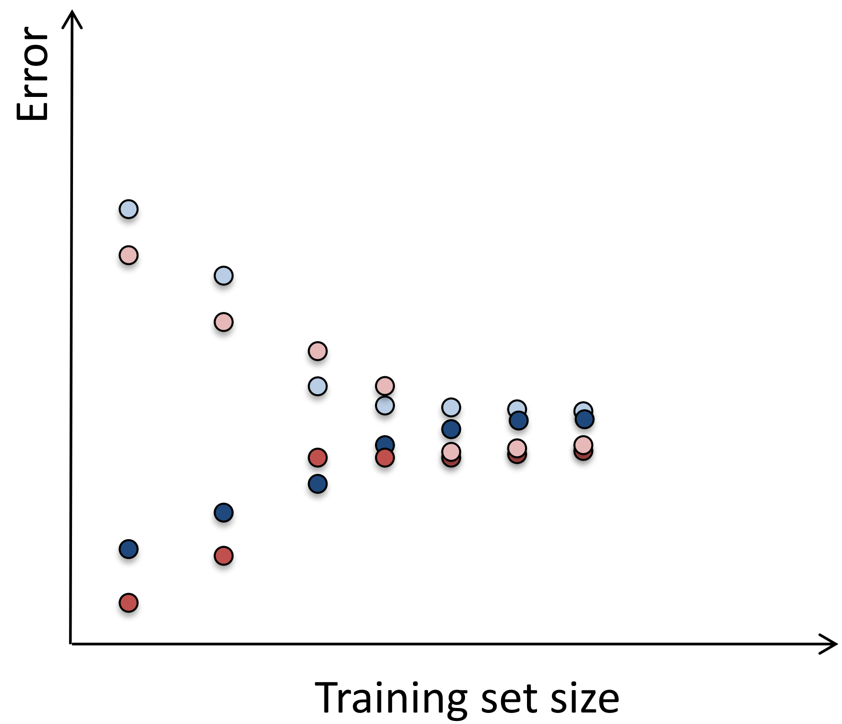
Can we use sampling?

Algorithm 1:

- Training
- Validation

Algorithm 2:

- Training
- Validation



Can we prune now?

How to speed-up tuning?

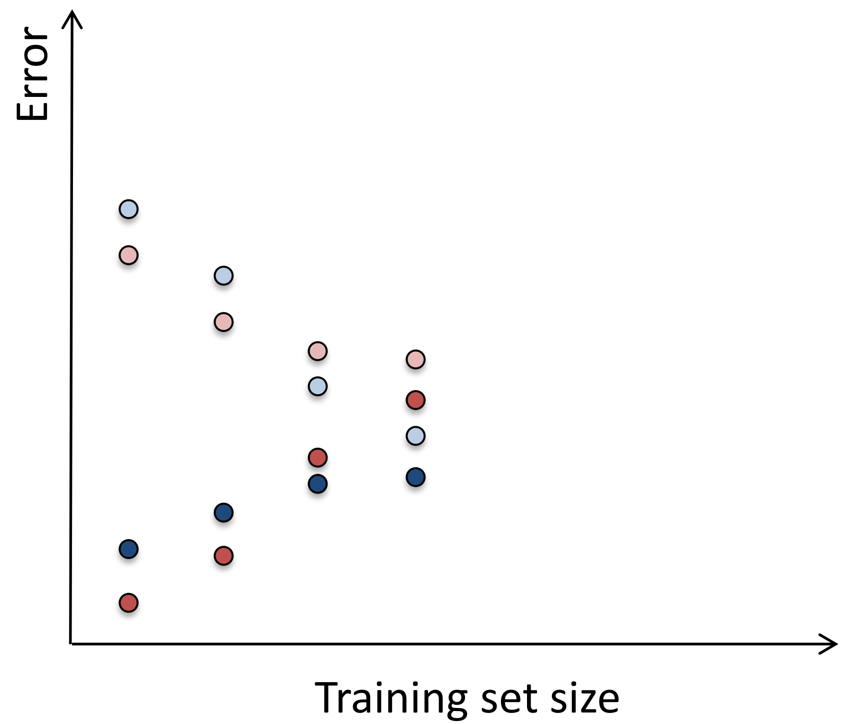
Can we use sampling?

Algorithm 1:

Training   Validation

Algorithm 2:

Training   Validation



Algorithm 1 training error > Algorithm 2 validation error