

Generalization, Model Selection

Pradeep Ravikumar
Co-instructor: Aarti Singh

Machine Learning 10-701
Feb 20, 2017



MACHINE LEARNING DEPARTMENT



True vs. Empirical Risk

True Risk: Target performance measure

Classification – Probability of misclassification $P(f(X) \neq Y)$

Regression – Mean Squared Error $\mathbb{E}[(f(X) - Y)^2]$

Expected performance on a random test point (X,Y)

True vs. Empirical Risk

True Risk: Target performance measure

Classification – Probability of misclassification $P(f(X) \neq Y)$

Regression – Mean Squared Error $\mathbb{E}[(f(X) - Y)^2]$

Expected performance on a random test point (X,Y)

Empirical Risk: Performance on training data

Classification – Proportion of misclassified examples $\frac{1}{n} \sum_{i=1}^n 1_{f(X_i) \neq Y_i}$

Regression – Average Squared Error $\frac{1}{n} \sum_{i=1}^n (f(X_i) - Y_i)^2$

Some quick notation

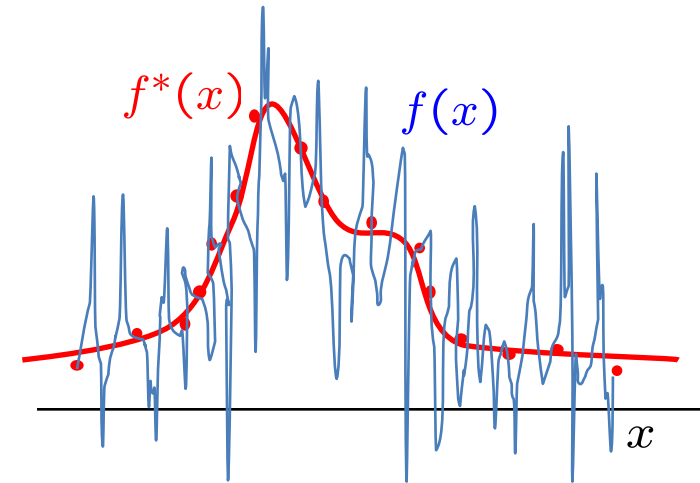
True Risk : $R(f) := \mathbb{E}(\ell(f(X), Y))$

Empirical Risk given data D : $\hat{R}_D(f) := \frac{1}{|D|} \sum_{i \in D} \ell(f(X_i), Y_i)$

Overfitting

Is the following predictor a good one?

$$f(x) = \begin{cases} Y_i, & x = X_i \text{ for } i = 1, \dots, n \\ \text{any value,} & \text{otherwise} \end{cases}$$



What is its empirical risk? (performance on training data)
zero !

What about true risk?
>> zero

Will predict very poorly on new random test point:
Large generalization error !

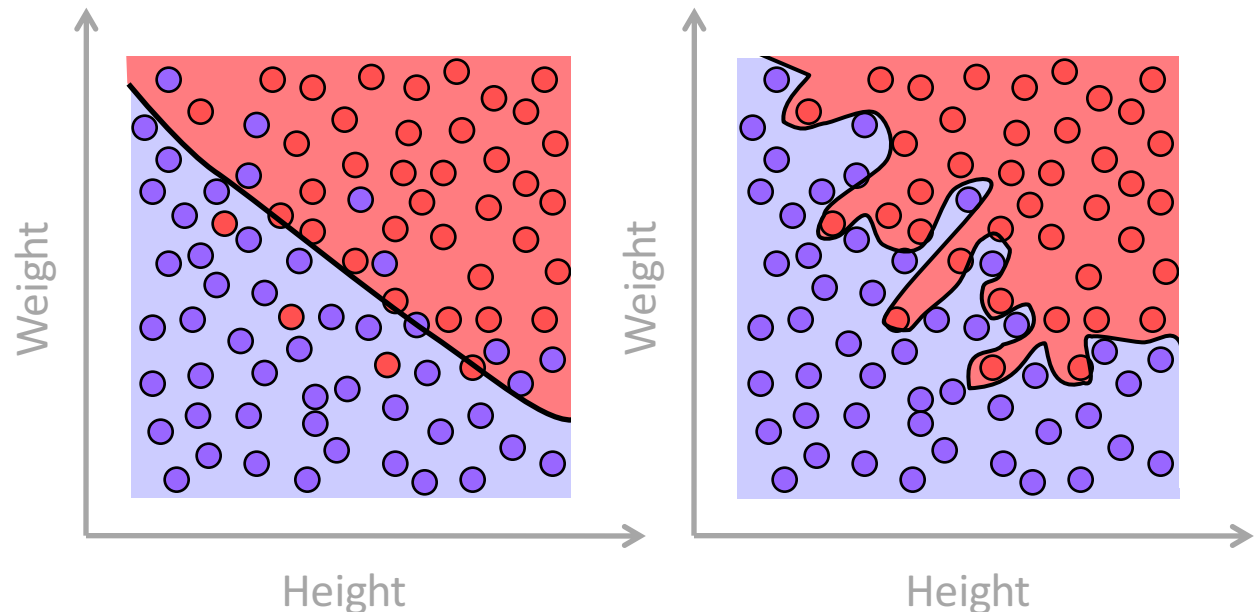
Overfitting

If we allow very complicated predictors, we could overfit the training data.

Examples: Classification (0-NN classifier)

Football player ?

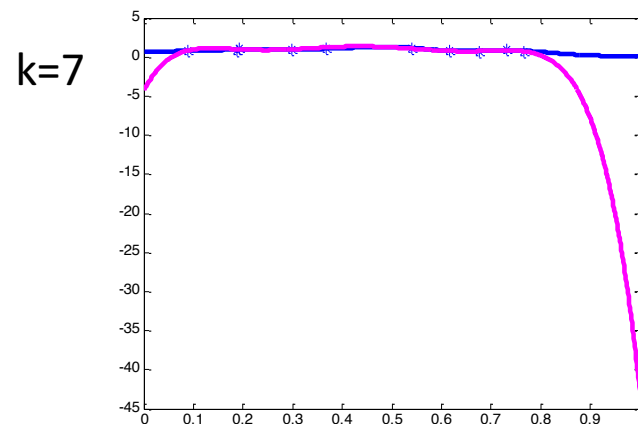
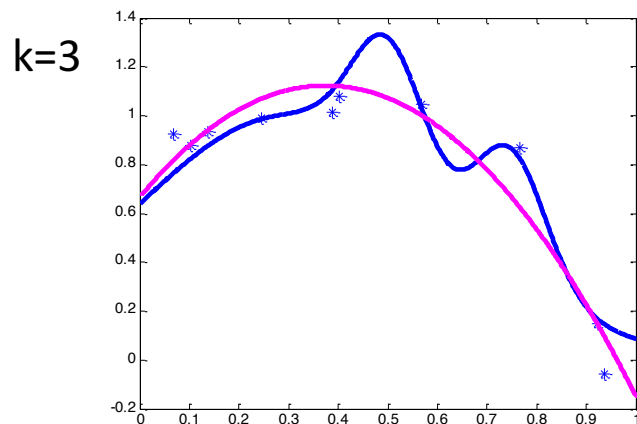
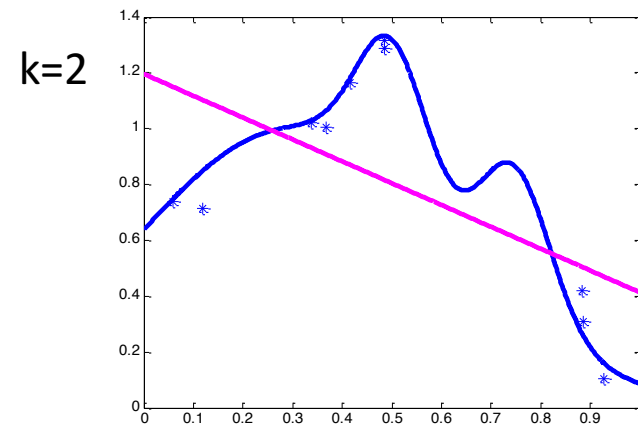
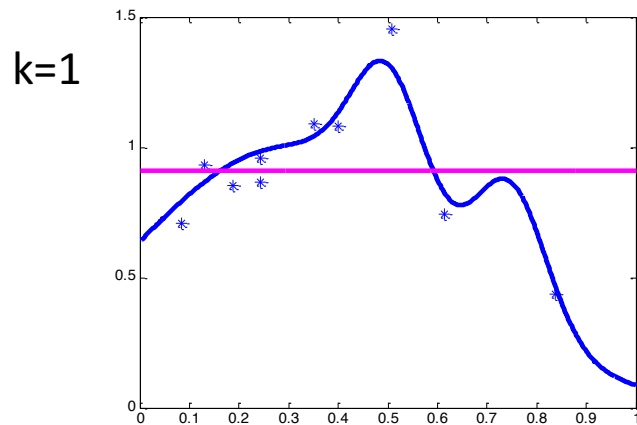
- No
- Yes



Overfitting

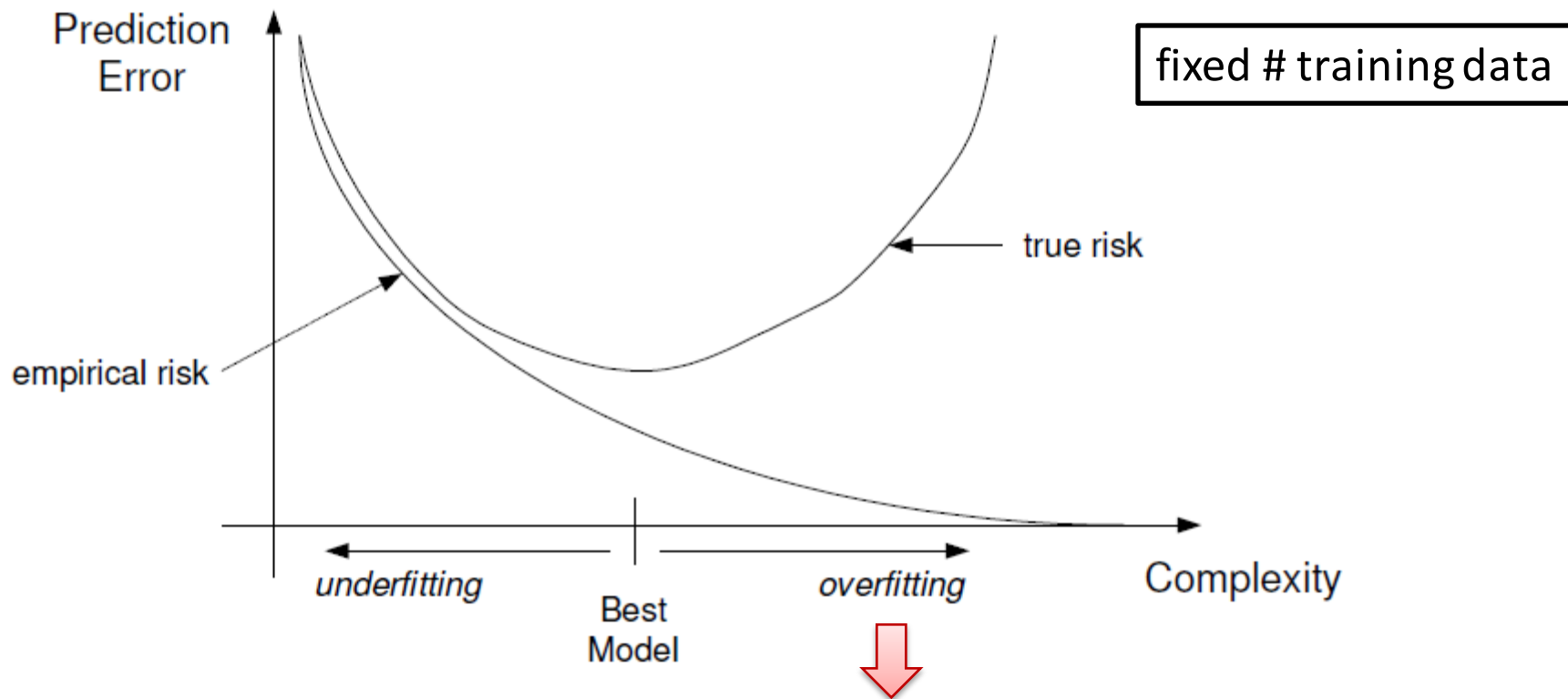
If we allow very complicated predictors, we could overfit the training data.

Examples: Regression (Polynomial of order k – degree up to $k-1$)



Overfitting: Effect of discrepancy between empirical and true risks

If we allow very complicated predictors, we could overfit the training data.



Empirical risk is no longer a good indicator of true risk

**WE COULD ASK HOW TO COME UP
WITH AN ESTIMATOR TO PREVENT
OVERFITTING?**

**EASIER QUESTION: GIVEN EXISTING
ESTIMATOR f , HOW SHOULD WE
ESTIMATE ITS “TRUE RISK”?**

Estimating True Risk

- Suppose we train an estimator \hat{f}_D on data D
- How do we estimate its true risk $R(\hat{f}_D)$?
- We could use the training data D itself i.e. use empirical risk on training data $\hat{R}_D(\hat{f}_D)$
- Not such a good idea
- If the midterm questions are comprised entirely of homework questions, would the midterm grade be an optimistic estimate of the “true” midterm grade?
- Similarly, using the empirical risk on training data would be an optimistic estimate of the true risk

Algorithmic and Closed Form Estimates of True Risk

- Algorithmic Estimates of True Risk:
 - Empirical Risk
 - Optimistic
 - Evaluating Risk on a holdout set
 - Cross-validation
- Closed form Estimates of True Risk
 - Structural Risk

Hold-out method

Can judge generalization error by using an independent sample of data.

Hold – out procedure:

n data points available

1) Split into two sets: $D \equiv \{X_i, Y_i\}_{i=1}^n$ Training dataset Holdout dataset

$$D_T = \{X_i, Y_i\}_{i=1}^m \qquad D_V = \{X_i, Y_i\}_{i=m+1}^n$$

2) Use D_T for training a predictor \hat{f}_{D_T}

3) Use D_V for evaluating the predictor

$$\hat{R}_{D_V}(\hat{f}_{D_T})$$

Hold-out method

Drawbacks:

- May not have enough data to afford setting one subset aside for getting a sense of generalization abilities
- Holdout error may be misleading (bad estimate of generalization error) if we get an “unfortunate” split

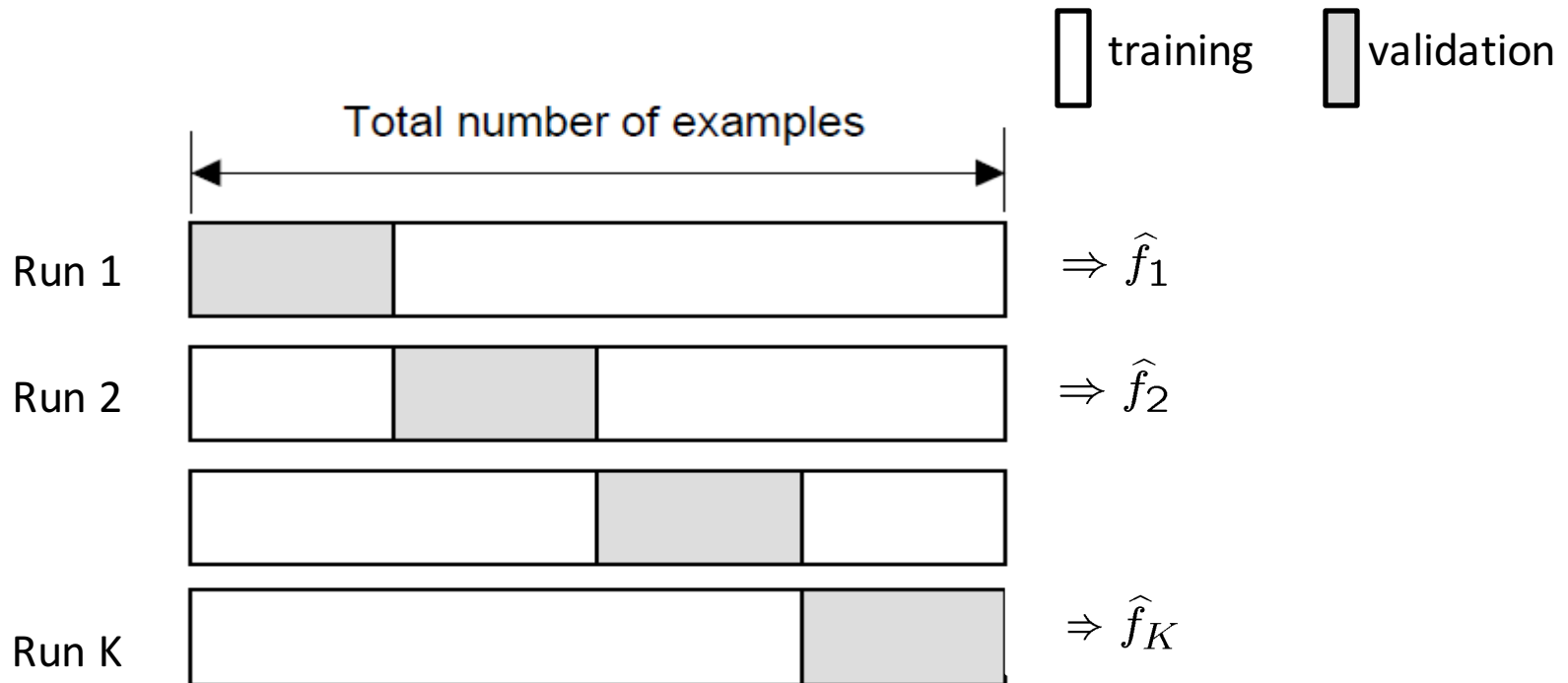
Cross-validation

K-fold cross-validation

Create K-fold partition of the dataset.

Form K hold-out predictors, each time using one partition as validation and rest K-1 as training datasets.

Final predictor is average/majority vote over the K hold-out estimates.

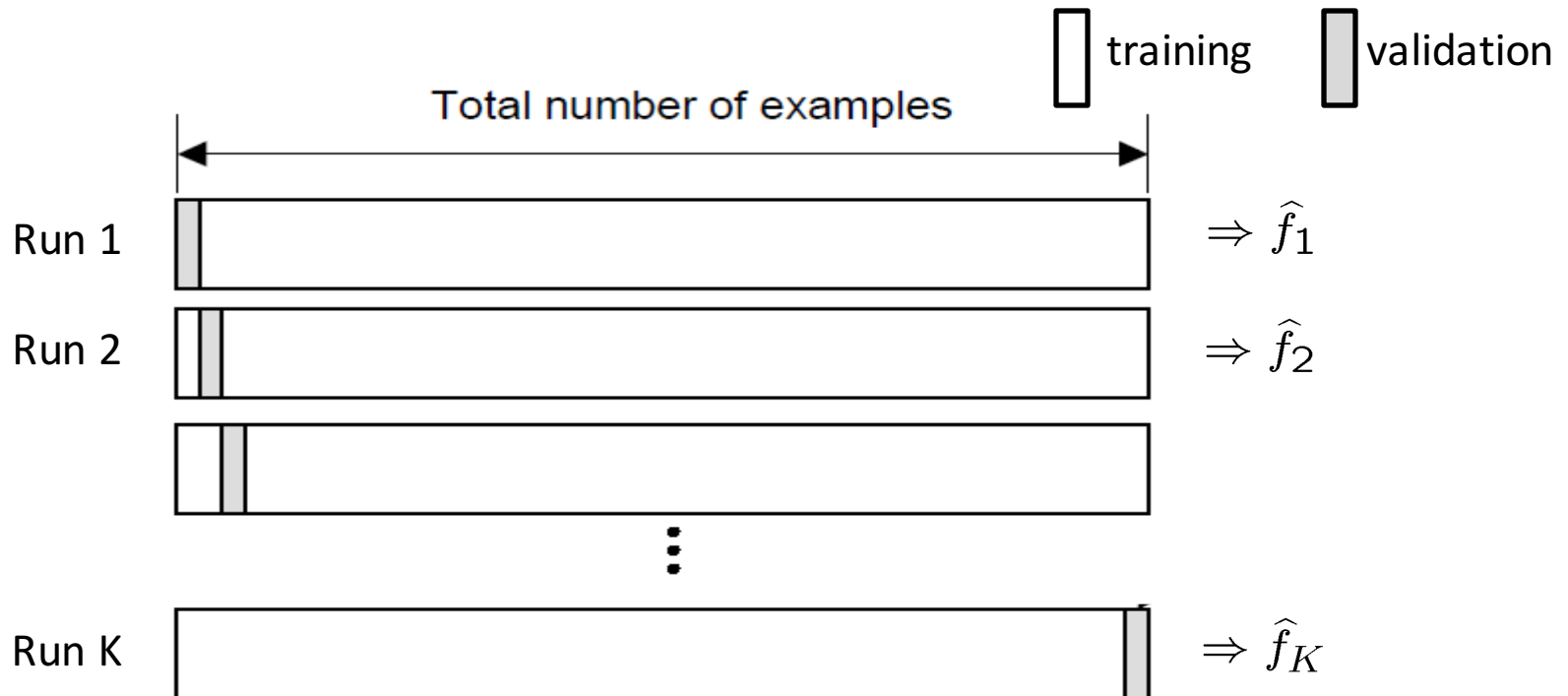


Cross-validation

Leave-one-out (LOO) cross-validation

Special case of K-fold with $K=n$ partitions

Equivalently, train on $n-1$ samples and validate on only one sample per run for n runs



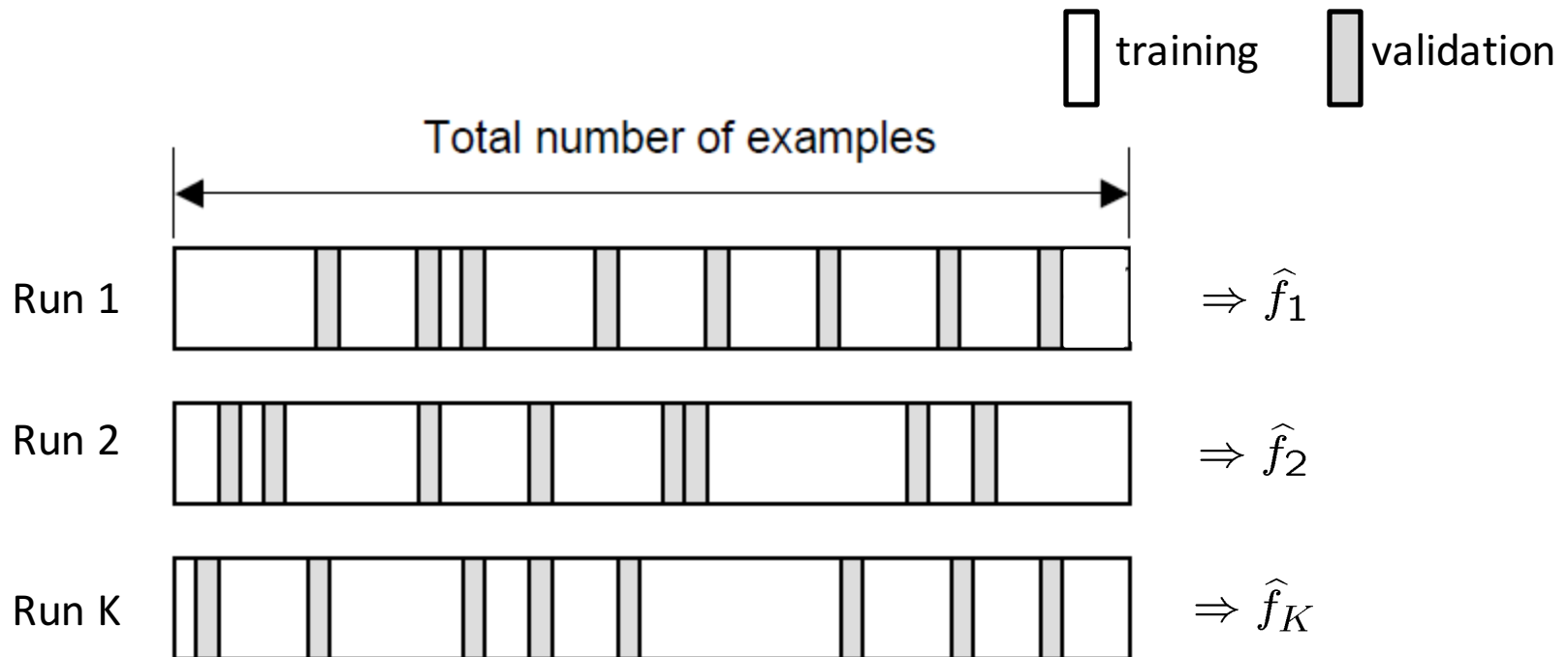
Cross-validation

Random subsampling

Randomly subsample a fixed fraction αn ($0 < \alpha < 1$) of the dataset for validation.
Form hold-out predictor with remaining data as training data.

Repeat K times

Final predictor is average/majority vote over the K hold-out estimates.



Estimating true risk

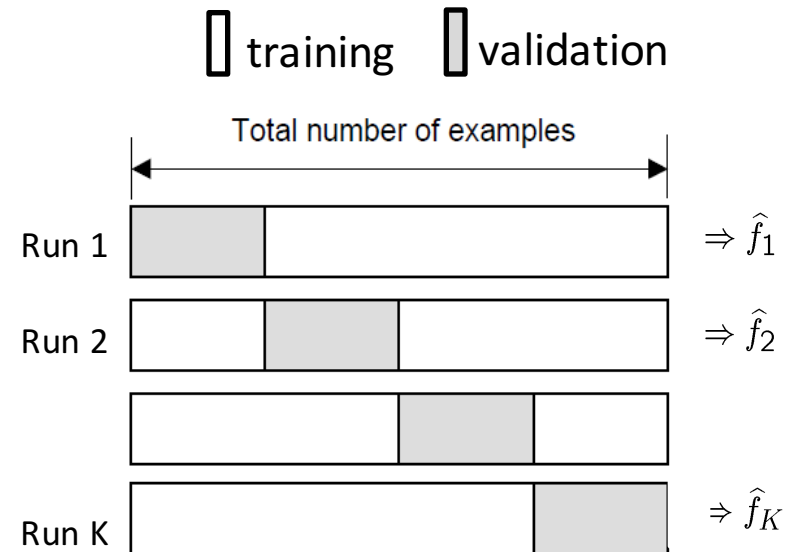
K-fold/LOO/random
sub-sampling:

$$\text{Error estimate} = \frac{1}{K} \sum_{k=1}^K \hat{R}_{V_k}(\hat{f}_{T_k})$$

We want to estimate the error of a predictor based on n data points.

If K is large (close to n), bias of error estimate is small since each training set has close to n data points.

However, variance of error estimate is high since each validation set has fewer data points and \hat{R}_{V_k} might deviate a lot from the mean.



Practical Issues in Cross-validation

How to decide the values for K and α ?

- Large K
 - + The bias of the error estimate will be small
 - The variance of the error estimate will be large (few validation pts)
 - The computational time will be very large as well (many experiments)
- Small K
 - + The # experiments and, therefore, computation time are reduced
 - + The variance of the error estimate will be small (many validation pts)
 - The bias of the error estimate will be large

Common choice: $K = 10$, $\alpha = 0.1$ 😊

Structural Risk

Add a penalty based on deviation of true and empirical risks:

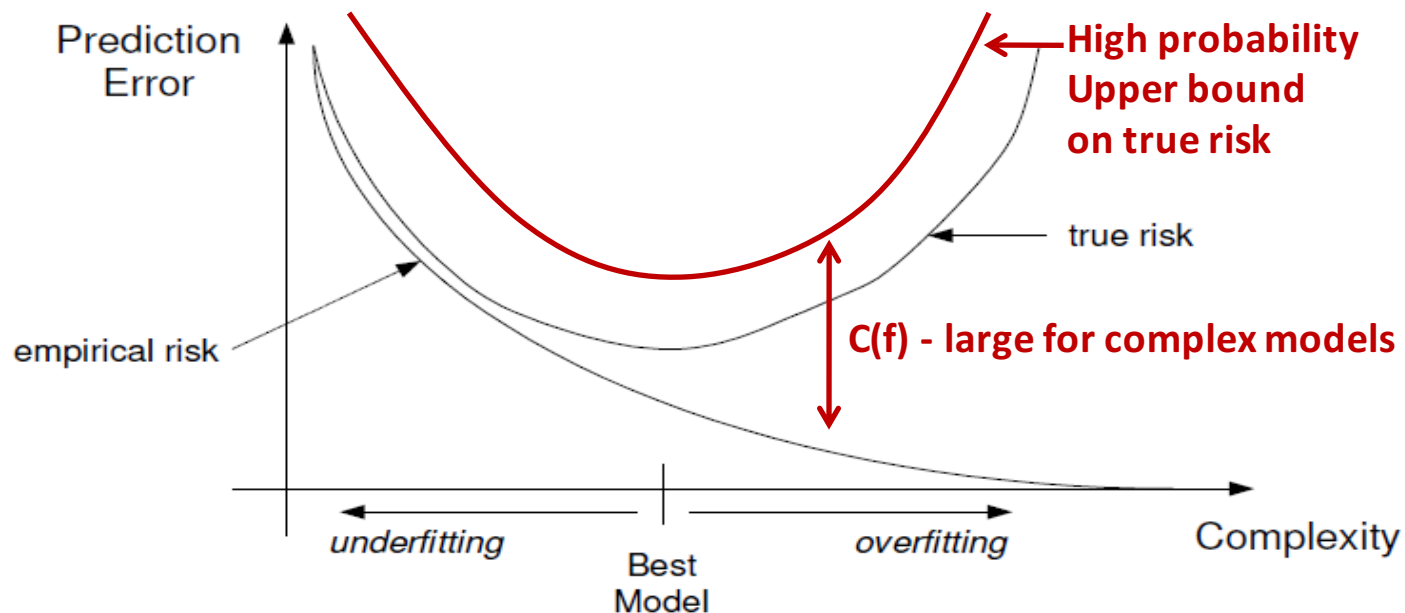
Suppose we have a bound, that with high probability:

$$|R(f) - \hat{R}_n(f)| \leq C(f) \quad \forall f \in \mathcal{F}$$

Concentration bounds
(later)

$$R(f) \leq \hat{R}_n(f) + C(f), \quad \forall f \in \mathcal{F}$$

Use $\hat{R}_n(\hat{f}_n) + C(\hat{f}_n)$ as a *pessimistic* estimate of true risk!



Risk Minimization

- OK, so we know how to estimate the “true risk” well, of an existing estimator e.g. empirical risk minimizer
- So now we can go back to solving the problem that empirical risk minimization might not be the right thing to do when the model complexity is high
 - i.e. it might overfit
 - with our great measures of "true risk", we might even get an indicator that hey it is overfitting
 - but what do we do with this information?
- We should not just estimate the true risk of an existing estimator, but our existing estimator should minimize the true risk!

Risk Minimization

- How do we improve empirical risk minimization with our newfound knowledge of how to better estimate true risk?
- Two-staged approach:
 - Given a model class, use better estimates of true risk to improve **empirical risk minimization**
 - Structural Risk Minimization
 - Regularize Empirical Risk with **Model Complexity**
 - Prior Information
 - Information-theoretic Criteria
 - We cannot use algorithmic risk estimates (e.g. cross-validation) because would be too expensive to train an estimator
 - Select model class (i.e. "tuning parameters") using better true risk estimates
 - We CAN use algorithmic risk estimates (e.g. cross-validation) since only finitely many options to choose from

IMPROVING EMPIRICAL RISK MINIMIZATION

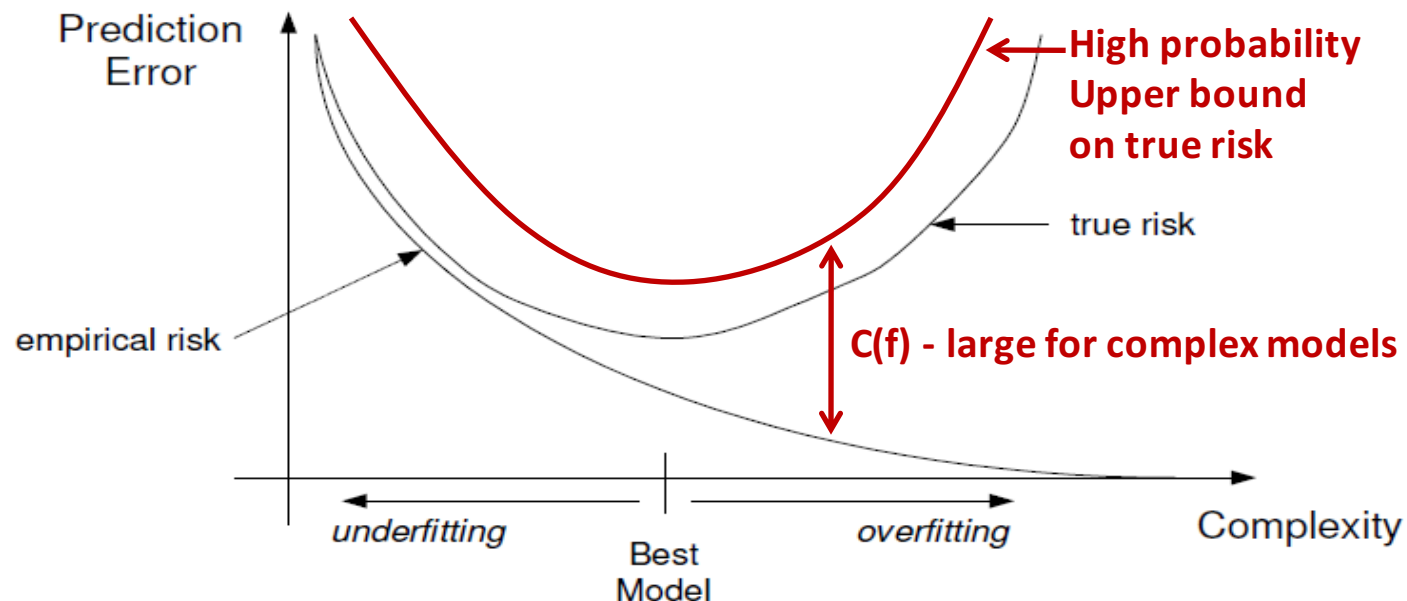
Structural Risk Minimization

Penalize models using bound on **deviation of true and empirical risks**.

$$\hat{f}_n = \arg \min_{f \in \mathcal{F}} \left\{ \hat{R}_n(f) + C(f) \right\}$$

Bound on deviation from true risk

With high probability, $|R(f) - \hat{R}_n(f)| \leq C(f) \quad \forall f \in \mathcal{F}$ Concentration bounds (later)



Structural Risk Minimization

Deviation bounds are typically pretty loose, for small sample sizes. In practice,

$$\hat{f}_n = \arg \min_{f \in \mathcal{F}} \{ \hat{R}_n(f) + \lambda C(f) \}$$

Choose by **model selection!**

Problem: Identify flood plain from noisy satellite images



Noiseless image



Noisy image



True Flood plain
(elevation level > x)

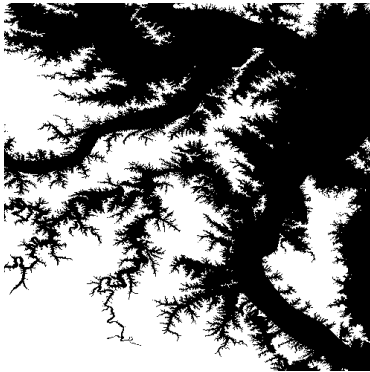
Structural Risk Minimization

Deviation bounds are typically pretty loose, for small sample sizes. In practice,

$$\hat{f}_n = \arg \min_{f \in \mathcal{F}} \{ \hat{R}_n(f) + \lambda C(f) \}$$

Choose by **model selection!**

Problem: Identify flood plain from noisy satellite images



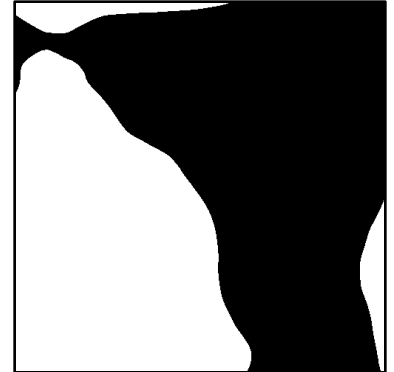
True Flood plain
(elevation level > x)



Zero penalty



CV penalty



Theoretical penalty

Occam's Razor

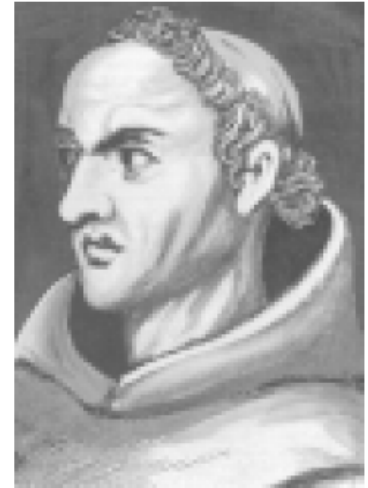
William of Ockham (1285-1349) *Principle of Parsimony*:

“One should not increase, beyond what is necessary, the number of entities required to explain anything.”

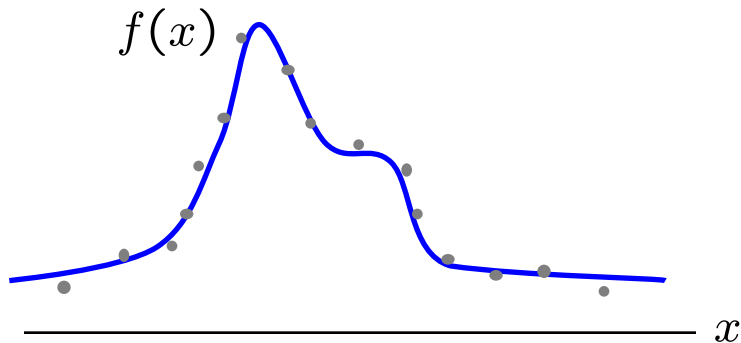
Alternatively, seek the simplest explanation.

Penalize complex models based on

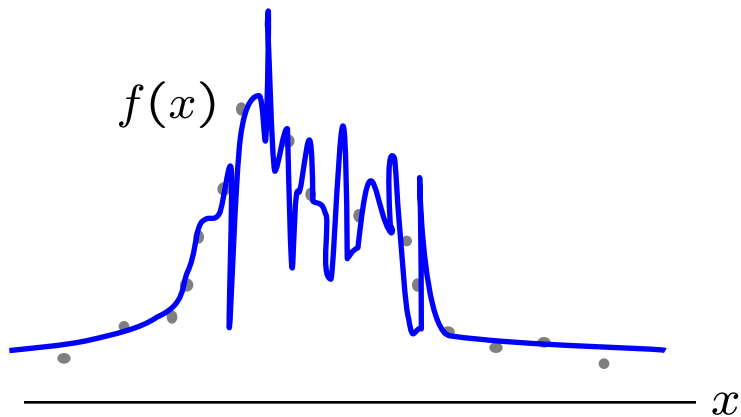
- Prior information (bias)
- Information Criterion (MDL, AIC, BIC)



Importance of Domain knowledge



Oil Spill Contamination




Distribution of photon arrivals



Compton Gamma-Ray Observatory Burst and Transient Source Experiment (**BATSE**)

Complexity Regularization

Penalize complex models using **prior knowledge**.

$$\hat{f}_n = \arg \min_{f \in \mathcal{F}} \left\{ \hat{R}_n(f) + C(f) \right\}$$


Cost of model
(log prior)

Bayesian viewpoint:

prior probability of f , $p(f) \equiv e^{-C(f)}$

cost is small if f is highly probable, cost is large if f is improbable

ERM (empirical risk minimization) over a restricted class F

\equiv uniform prior on $f \in F$, zero probability for other predictors

$$\hat{f}_n^L = \arg \min_{f \in \mathcal{F}_L} \hat{R}_n(f)$$

Complexity Regularization

Penalize complex models using **prior knowledge**.

$$\hat{f}_n = \arg \min_{f \in \mathcal{F}} \left\{ \hat{R}_n(f) + C(f) \right\}$$

Cost of model
(log prior)

Examples: MAP estimators

Regularized Linear Regression - Ridge Regression, Lasso

$$\hat{\theta}_{\text{MAP}} = \arg \max_{\theta} \log p(D|\theta) + \log p(\theta)$$

$$\hat{\beta}_{\text{MAP}} = \arg \min_{\beta} \sum_{i=1}^n (Y_i - X_i\beta)^2 + \lambda \|\beta\|$$

Penalize models based
on some norm of
regression coefficients

How to choose tuning parameter λ ? **Cross-validation**

Information Criteria – AIC, BIC

Penalize complex models based on their **information content**.

$$\hat{f}_n = \arg \min_{f \in \mathcal{F}} \left\{ \hat{R}_n(f) + C(f) \right\}$$

└─ # bits needed to describe f
(description length)

AIC (Akaike IC) $C(f) = \# \text{ parameters}$

Allows # parameters to be infinite as # training data n become large

BIC (Bayesian IC) $C(f) = \# \text{ parameters} * \log n$

Penalizes complex models more heavily – limits complexity of models as # training data n become large

MODEL SELECTION

Model Selection

- Model Classes with increasing complexity
- Regression with polynomials of order $k = 0, 1, 2, \dots$
 - Higher degree \Rightarrow Higher complexity
 - Question: How to select k ?
- Regularization parameter λ in previous estimators
 - Larger values of $\lambda \Rightarrow$ Lower complexity
 - Question: How to select λ ?
- The general setup:
 - Have a finite set of model classes (indexed by some "tuning parameter") with differing model complexities
 - Given any one model class, can use estimates of true risk to find optimal estimator
 - Model selection: find the optimal model class

Model Selection

Setup:

Model Classes $\{\mathcal{F}_\lambda\}_{\lambda \in \Lambda}$ of increasing complexity $\mathcal{F}_1 \prec \mathcal{F}_2 \prec \dots$

$$\min_{\lambda} \min_{f \in \mathcal{F}_\lambda} J(f, \lambda)$$

Stage I: Given lambda, pick estimator f_λ using

- empirical risk minimization
- structural risk minimization
- complexity regularized risk minimization

Stage II: Pick that lambda for which f_λ has least true risk estimated using

- cross-validation
- holdout
- information-theoretic risk estimates

A more theoretical understanding these two stages of the risk minimization story

Estimated Predictor : \hat{f}_n

Optimal Predictor : f^*

Risk of Estimated Predictor : $R(\hat{f}_n)$

Above is random due to samples in training data

Expectation of above wrt training data : $\mathbb{E}(R(\hat{f}_n))$

Risk of Optimal Predictor : $R(f^*)$

Players in the risk minimization story

Estimated Predictor : \hat{f}_n

Optimal Predictor : f^*

Risk of Estimated Predictor : $R(\hat{f}_n)$

Above is random due to samples in training data

Expectation of above wrt training data : $\mathbb{E}(R(\hat{f}_n))$

Risk of Optimal Predictor : $R(f^*)$

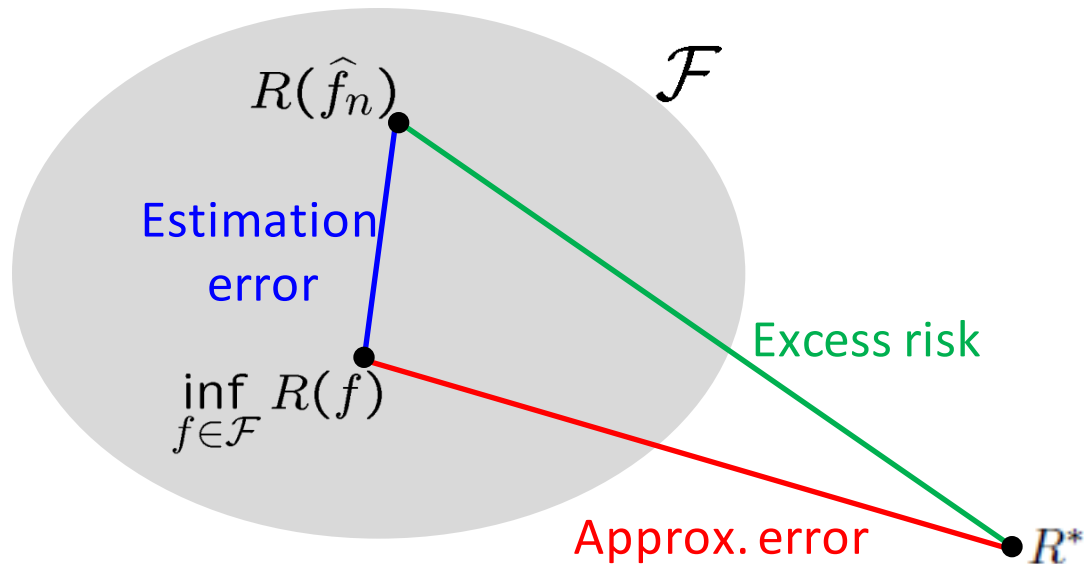
Interested in the excess risk: $\mathbb{E}(R(\hat{f}_n)) - R(f^*)$

Behavior of True Risk

Want \hat{f}_n to be as good as optimal predictor f^*

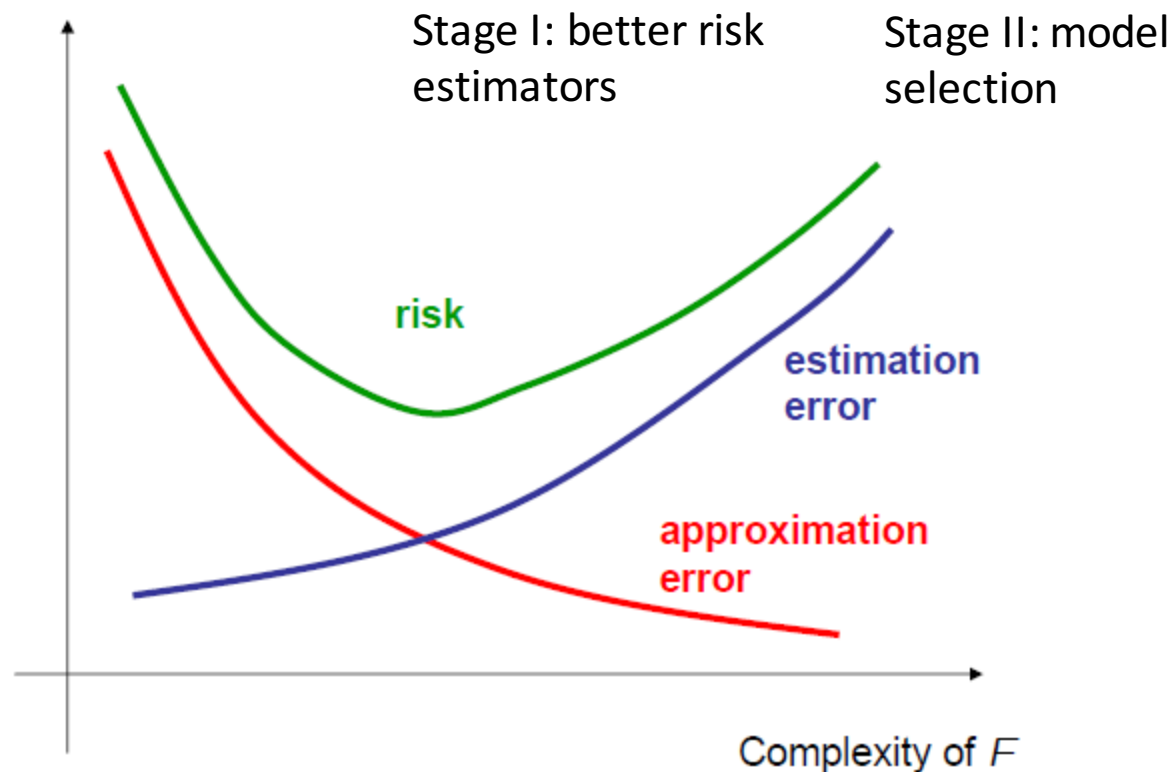
Excess Risk
$$E[R(\hat{f}_n)] - R^* = \underbrace{\left(E[R(\hat{f}_n)] - \inf_{f \in \mathcal{F}} R(f)\right)}_{\text{estimation error}} + \underbrace{\left(\inf_{f \in \mathcal{F}} R(f) - R^*\right)}_{\text{approximation error}}$$

finite sample size + noise ← Due to randomness of training data Due to restriction of model class



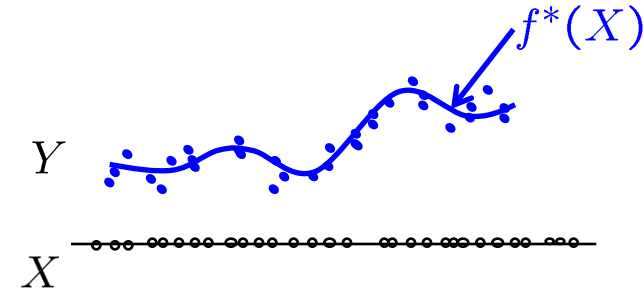
Behavior of True Risk

$$E[R(\hat{f}_n)] - R^* = \underbrace{\left(E[R(\hat{f}_n)] - \inf_{f \in \mathcal{F}} R(f)\right)}_{\text{estimation error}} + \underbrace{\left(\inf_{f \in \mathcal{F}} R(f) - R^*\right)}_{\text{approximation error}}$$



Bias – Variance Tradeoff

Regression: $Y = f^*(X) + \epsilon \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$



$$R^* = \mathbb{E}_{XY}[(f^*(X) - Y)^2] = \mathbb{E}[\epsilon^2] = \sigma^2$$

Notice: Optimal predictor does not have zero error

$$\mathbb{E}_{D_n}[R(\hat{f}_n)] = \mathbb{E}_{X,Y,D_n}[(\hat{f}_n(X) - Y)^2]$$

D_n - training data of size n

$$\begin{aligned} & \vdots \\ & \vdots \\ & \vdots \\ & = \underbrace{\mathbb{E}_{X,Y,D_n}[(\hat{f}_n(X) - \mathbb{E}_{D_n}[\hat{f}_n(X)])^2]}_{\text{variance}} + \underbrace{\mathbb{E}_{X,Y}[(\mathbb{E}_{D_n}[\hat{f}_n(X)] - f^*(X))^2]}_{\text{bias}^2} + \underbrace{\sigma^2}_{\text{Noise var}} \end{aligned}$$

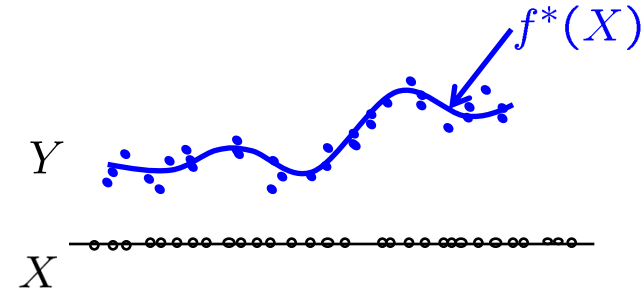
$$\text{Excess Risk} = \mathbb{E}_{D_n}[R(\hat{f}_n)] - R^* = \text{variance} + \text{bias}^2$$

Random component \equiv est err

\equiv approx err

Bias – Variance Tradeoff: Derivation

Regression: $Y = f^*(X) + \epsilon \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$



$$R^* = \mathbb{E}_{XY}[(f^*(X) - Y)^2] = \mathbb{E}[\epsilon^2] = \sigma^2$$

Notice: Optimal predictor
does not have zero error

$$\mathbb{E}_{D_n}[R(\hat{f}_n)] = \mathbb{E}_{X,Y,D_n}[(\hat{f}_n(X) - Y)^2] \quad D_n - \text{training data of size } n$$

$$= \mathbb{E}_{X,Y,D_n} [(\hat{f}_n(X) - \mathbb{E}_{D_n}[\hat{f}_n(X)] + \mathbb{E}_{D_n}[\hat{f}_n(X)] - Y)^2]$$

$$= \mathbb{E}_{X,Y,D_n} [(\hat{f}_n(X) - \mathbb{E}_{D_n}[\hat{f}_n(X)])^2 + (\mathbb{E}_{D_n}[\hat{f}_n(X)] - Y)^2 + 2(\hat{f}_n(X) - \mathbb{E}_{D_n}[\hat{f}_n(X)])(\mathbb{E}_{D_n}[\hat{f}_n(X)] - Y)]$$

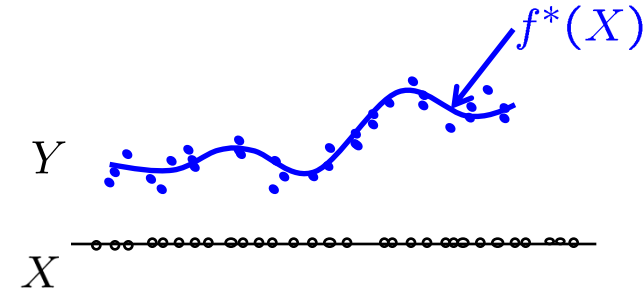
$$= \mathbb{E}_{X,Y,D_n} [(\hat{f}_n(X) - \mathbb{E}_{D_n}[\hat{f}_n(X)])^2] + \mathbb{E}_{X,Y,D_n} [(\mathbb{E}_{D_n}[\hat{f}_n(X)] - Y)^2]$$

$$+ \mathbb{E}_{X,Y} [2(\mathbb{E}_{D_n}[\hat{f}_n(X)] - \mathbb{E}_{D_n}[\hat{f}_n(X)])(\mathbb{E}_{D_n}[\hat{f}_n(X)] - Y)]$$

0

Bias – Variance Tradeoff: Derivation

Regression: $Y = f^*(X) + \epsilon \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$



$$R^* = \mathbb{E}_{XY}[(f^*(X) - Y)^2] = \mathbb{E}[\epsilon^2] = \sigma^2$$

Notice: Optimal predictor does not have zero error

$$\mathbb{E}_{D_n}[R(\hat{f}_n)] = \mathbb{E}_{X,Y,D_n}[(\hat{f}_n(X) - Y)^2]$$

D_n - training data of size n

$$= \underbrace{\mathbb{E}_{X,Y,D_n}[(\hat{f}_n(X) - \mathbb{E}_{D_n}[\hat{f}_n(X)])^2]}_{\text{variance}} + \mathbb{E}_{X,Y,D_n}[(\mathbb{E}_{D_n}[\hat{f}_n(X)] - Y)^2]$$

variance – how much does the predictor vary about its mean for different training datasets

Now, let's look at the second term:

$$\mathbb{E}_{X,Y,D_n}[(\mathbb{E}_{D_n}[\hat{f}_n(X)] - Y)^2] = \mathbb{E}_{X,Y}[(\mathbb{E}_{D_n}[\hat{f}_n(X)] - Y)^2]$$

Note: this term doesn't depend on D_n

Bias – Variance Tradeoff: Derivation

$$\begin{aligned}\mathbb{E}_{X,Y} \left[(\mathbb{E}_{D_n}[\hat{f}_n(X)] - Y)^2 \right] &= \mathbb{E}_{X,Y} \left[(\mathbb{E}_{D_n}[\hat{f}_n(X)] - f^*(X) - \epsilon)^2 \right] \\ &= \mathbb{E}_{X,Y} \left[(\mathbb{E}_{D_n}[\hat{f}_n(X)] - f^*(X))^2 + \epsilon^2 \right. \\ &\quad \left. - 2\epsilon(\mathbb{E}_{D_n}[\hat{f}_n(X)] - f^*(X)) \right] \\ &= \mathbb{E}_{X,Y} \left[(\mathbb{E}_{D_n}[\hat{f}_n(X)] - f^*(X))^2 \right] + \mathbb{E}_{X,Y} [\epsilon^2] \\ &\quad - 2\mathbb{E}_{X,Y} [\epsilon(\mathbb{E}_{D_n}[\hat{f}_n(X)] - f^*(X))] \\ &\quad \text{0 since noise is independent and zero mean}\end{aligned}$$

$$= \underbrace{\mathbb{E}_{X,Y} \left[(\mathbb{E}_{D_n}[\hat{f}_n(X)] - f^*(X))^2 \right]}_{\text{bias}^2} + \underbrace{\mathbb{E}_{X,Y} [\epsilon^2]}_{\text{noise variance}}$$

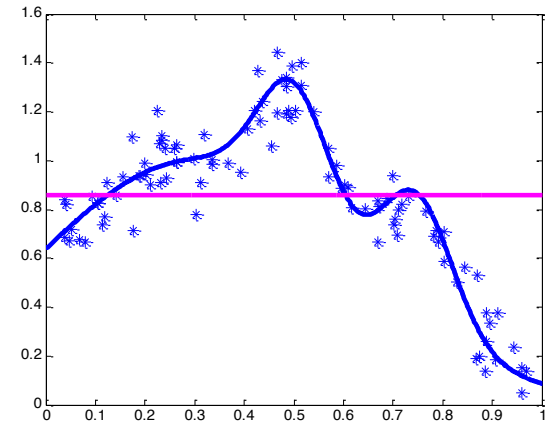
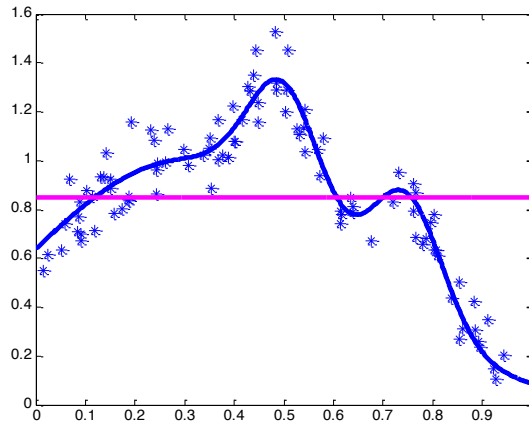
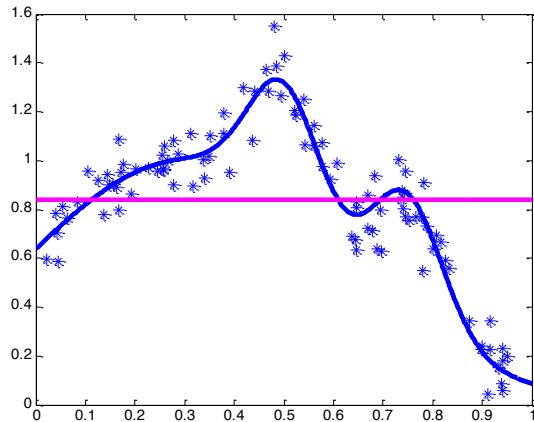
bias² – how much does the mean of the predictor differ from the optimal predictor

noise variance

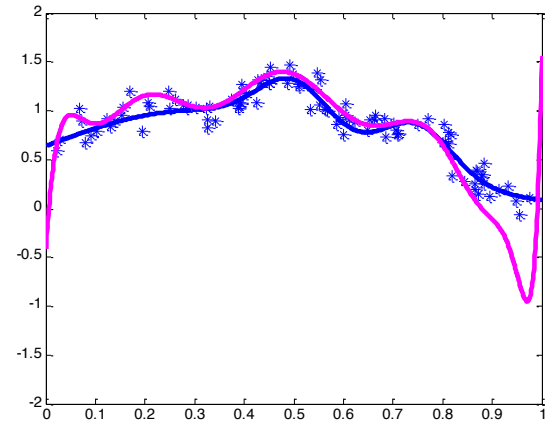
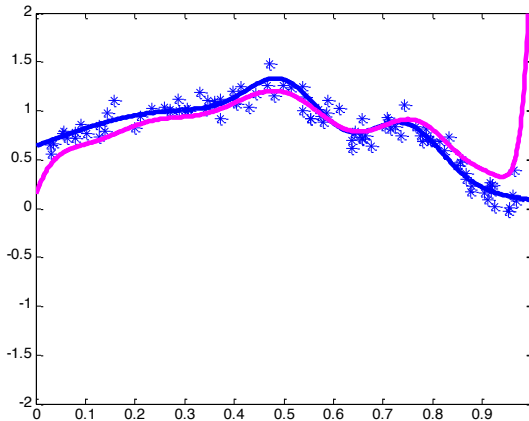
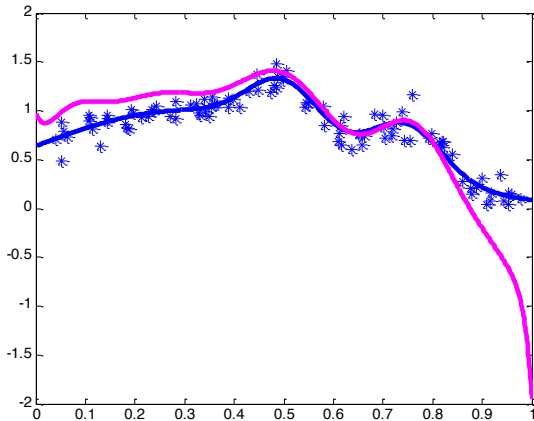
Bias – Variance Tradeoff

3 Independent training datasets

Large bias, Small variance – poor approximation but robust/stable



Small bias, Large variance – good approximation but unstable



Summary

True and Empirical Risk

Over-fitting

Model Selection, Estimating Generalization Error

- Hold-out, K-fold cross-validation
 - Structural Risk Minimization
 - Complexity Regularization
 - Information Criteria – AIC, BIC
-
- Approx err vs Estimation err, Bias vs Variance tradeoff