MLRF Lecture 06

J. Chazalon, LRDE/EPITA, 2019

🎉 with **Nicolas Boutry** 🎊
as guest lecturer 🎤

More theory on classification

Lecture 05 part 03

What is our goal?

*Given **samples** (described by features) and **true labels**,*
*find a **good** function*
*which will correctly **predict labels***
*given **new data samples***

Problems:

- Which family for our function?
- What is “good”?
- How to train / find such function?

Let us step back a little bit.

What are the sources of error?

Figures from Bradski & Kaehler, *Learning OpenCV,* O'Reilly 2008

Noise

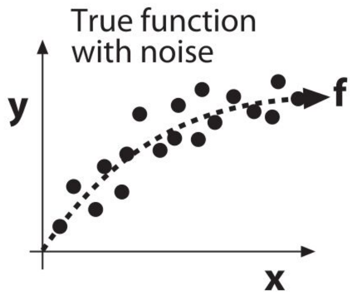
- Your data is not perfect. *(or "Every model is wrong.")*
- Even if there exist an optimal underlying model, the observations are corrupted by noise.

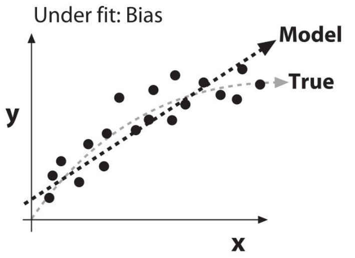
Bias

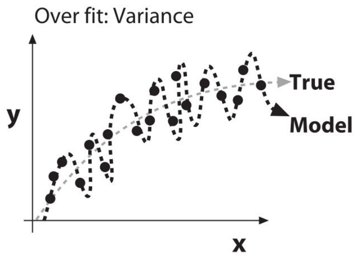
- You need to simply to generalize.
- You classifier needs to drop some information about the training set to have generalization power.

Variance

- You have many ways to explain your training dataset.
- It is hard to find an optimal solution among those many possibilities.







Two big issues

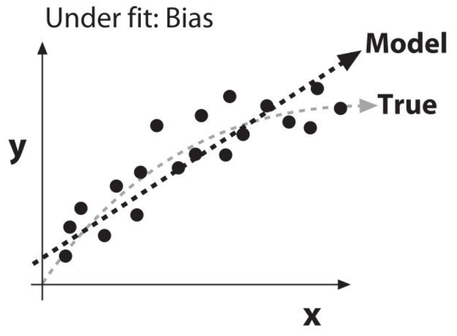
Figures from Bradski & Kaehler, *Learning OpenCV*, O'Reilly 2008

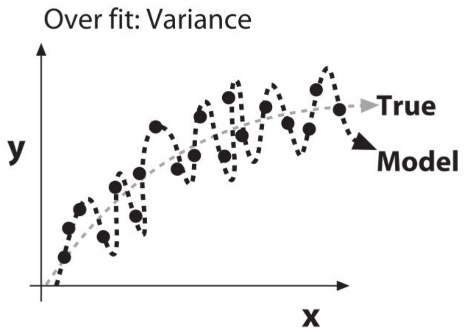
Under-fitting

- Caused by bias
- Your model assumptions are too strong for the data, so the model won't fit well.

Over-fitting

- Caused by variance
- Your algorithm has memorized the data *including* the noise, so it can't generalize.





The theory

Expected Risk

Let D_n be a training set of examples z_i drawn independently from an unknown distribution ***p(z)***

We need a set of functions ***F***. Example: linear functions $f(x) = a \cdot x + b$

We need a loss function ***L(z, f)***. Example: $L((x, y), f) = (f(x) - y)^2$

The <u>**Expected Risk**</u>, ie the generalization error, is:

$$R(f) = E_Z[L(z, f)] = \int_Z L(z, f)p(z)dz$$

But we do not know ***p(z)***, and we cannot test all ***z***!

Empirical Risk

Because we **cannot measure** the real **Expected Risk**, we have to **estimate it** using the **Empirical Risk**:

$$\hat{R}(f, D_n) = \frac{1}{n}\sum_{i=1}^{n} L(z_i, f)$$

(***D*** is the training set)

And our training procedure then relies on **Empirical Risk Minimization** (ERM):

$$f^\star(D_n) = \arg\min_{f\in\mathcal{F}} \hat{R}(f, D_n)$$

And the training error is given by:

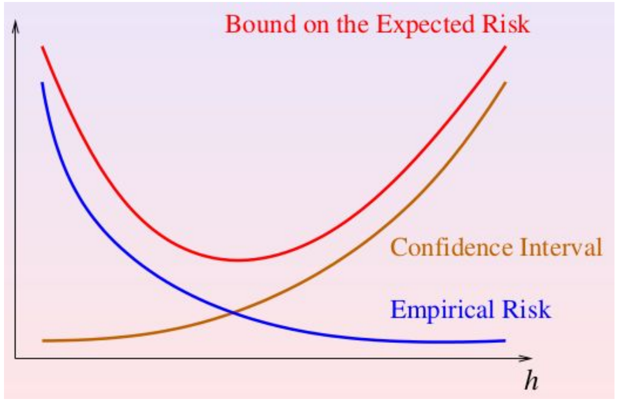
$$\hat{R}(f^\star(D_n), D_n)$$

From S. Bengio

Estimate the Expected Risk with the Empirical Risk

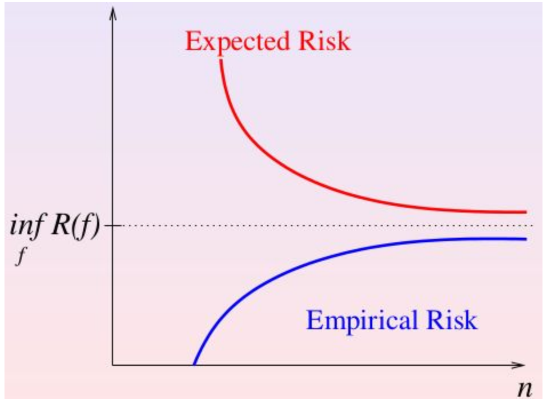
The difference between Expected Risk and Empirical Risk is **bounded** but depends on the **capacity** of ***F*** (set of possible functions).

There is an **optimal** capacity for a given number of training samples ***n***.



Fixed ***n***

For a given capacity, using more samples to train and evaluate your predictor **should** make your Empirical Risk converge toward the best possible Expected Risk, if the ERM is consistent for ***F***, given your training set D_n.



Fixed ***h***

From S. Bengio

Capacity

The capacity ***h(F)*** is a measure of its size, or complexity.

For classification, the capacity of ***F*** is defined by Vapnik & Chervonenkis as

the largest ***n***

such that there exist a set of examples D_n

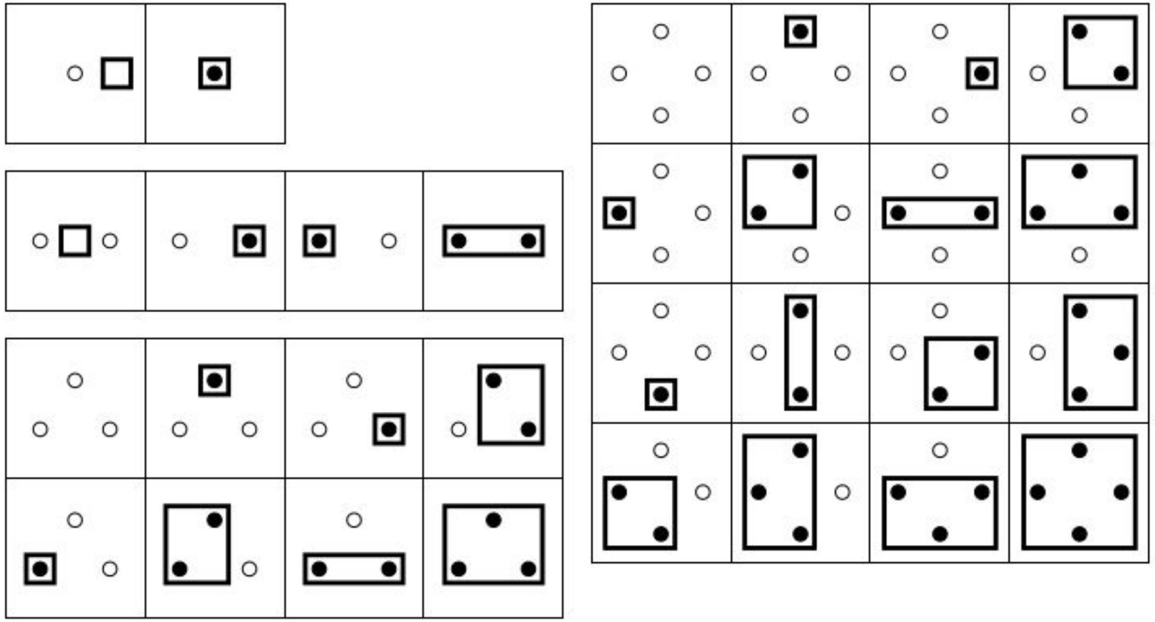
such that one can always find an $f \in F$

which gives the correct answer for all examples in D_n,

for any possible labeling.

Capacity

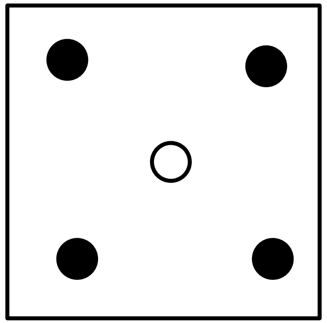
Consider for ***F*** the characteristic functions of rectangles. We can find families of 1, 2, 3 or 4 points which can be labelled arbitrarily:



From F. Fleuret

Capacity

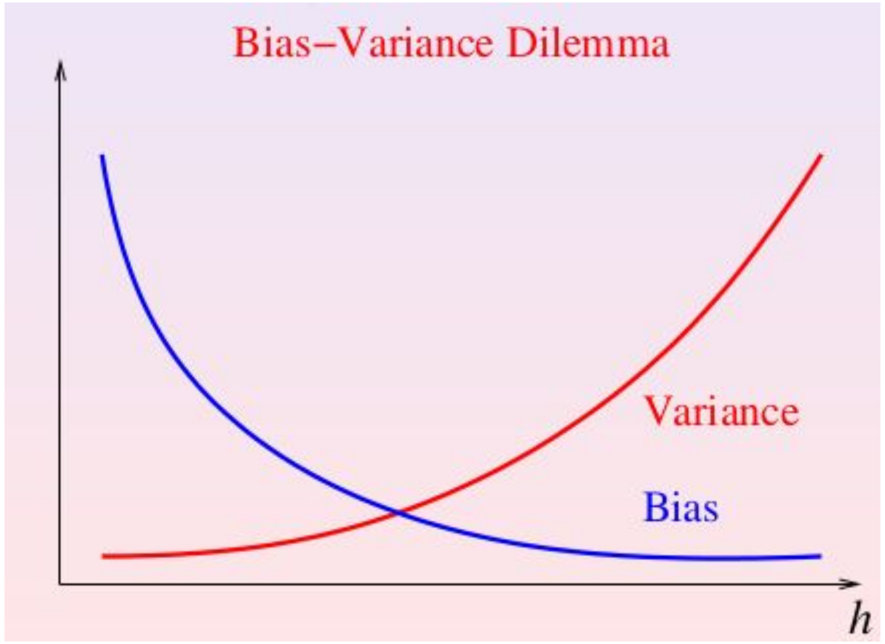
However, given a family of 5 points, if the four external points are labelled 1 and the center point labelled 0, than no function from ***F*** can predict that labelling. Hence here D = 4.



From F. Fleuret

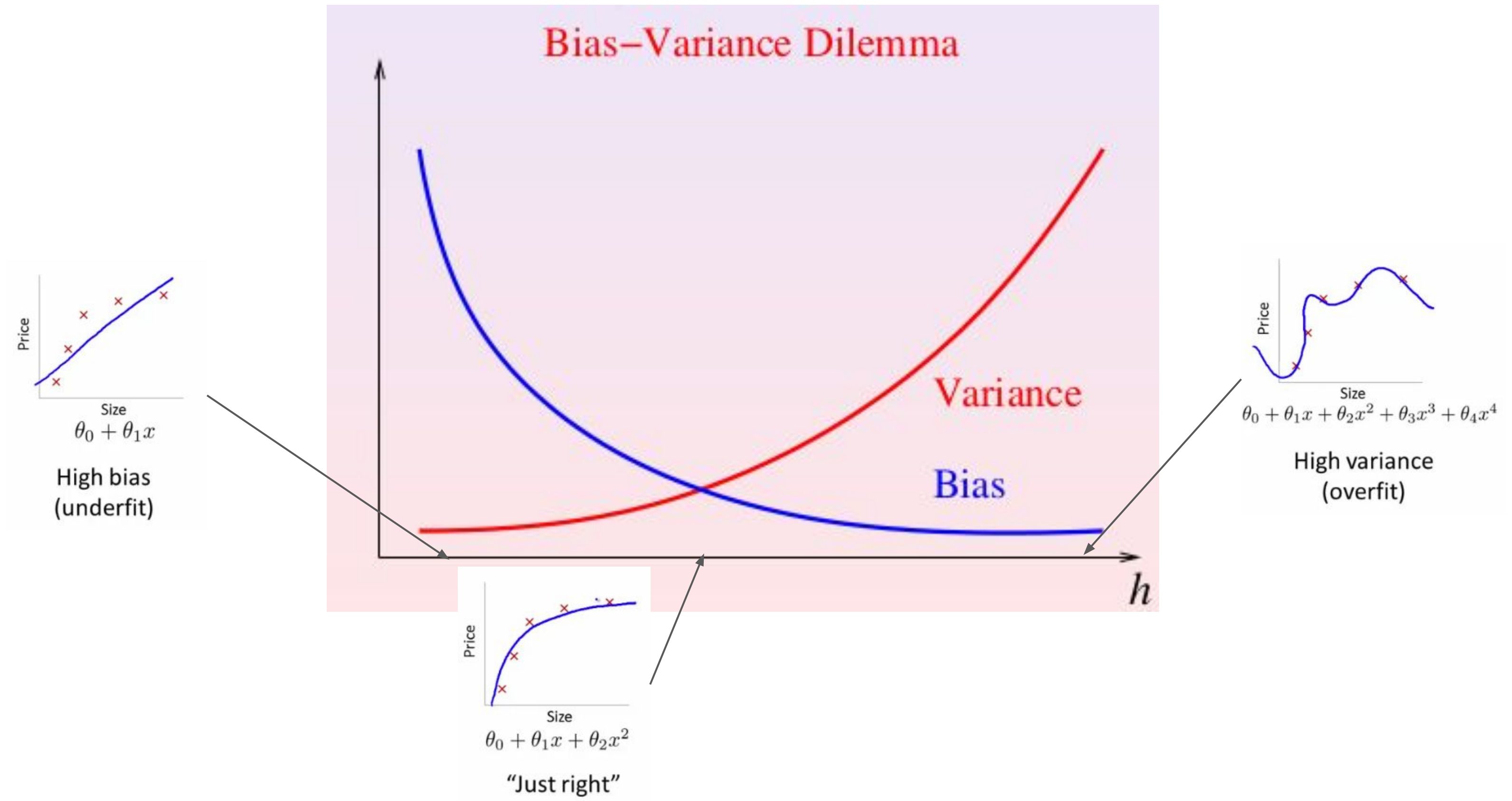
The Bias-Variance Dilemma

Intrinsic dilemma: when the capacity ***h(F)*** grows, the bias goes down, but the variance goes up!



From S. Bengio

Look for an optimal balance



In practice

In practice: Empirical Risk and Expected Risk

Measure train and test error.

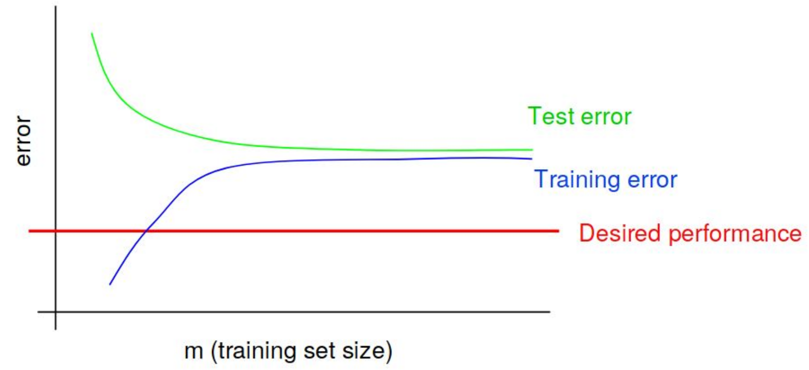
Use hold-out sets, cross-validations, etc. to get a test error.

Train error: **Empirical** Risk.

Test error: Coarse estimate of the **Expected** Risk.

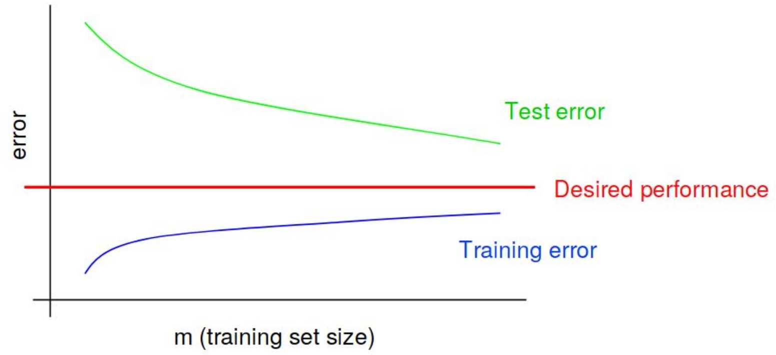
Detect under-fitting and over-fitting

High bias: This learning curve shows high error on both the training and test sets, so the algorithm is suffering from high bias.



Even training error is unacceptably high.
Small gap between training and test error.

High variance: This learning curve shows a large gap between training and test set errors, so the algorithm is suffering from high variance.



Test error still decreasing as m increases.

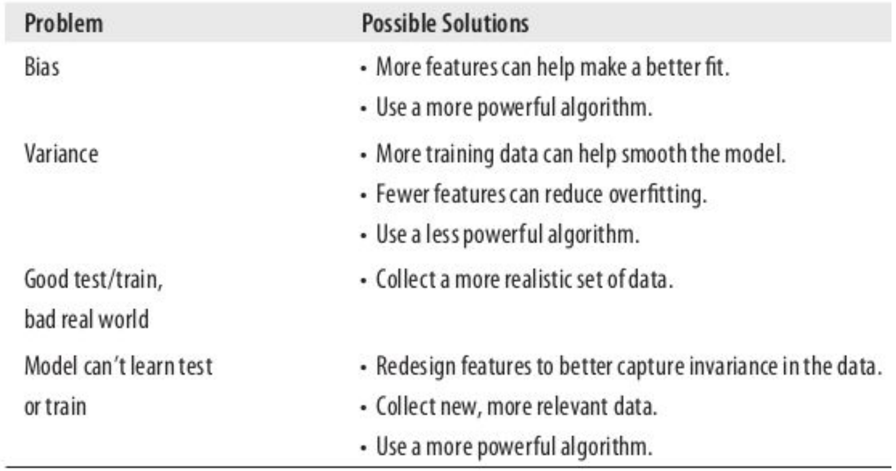
Suggests larger training set will help.

Large gap between training and test error.

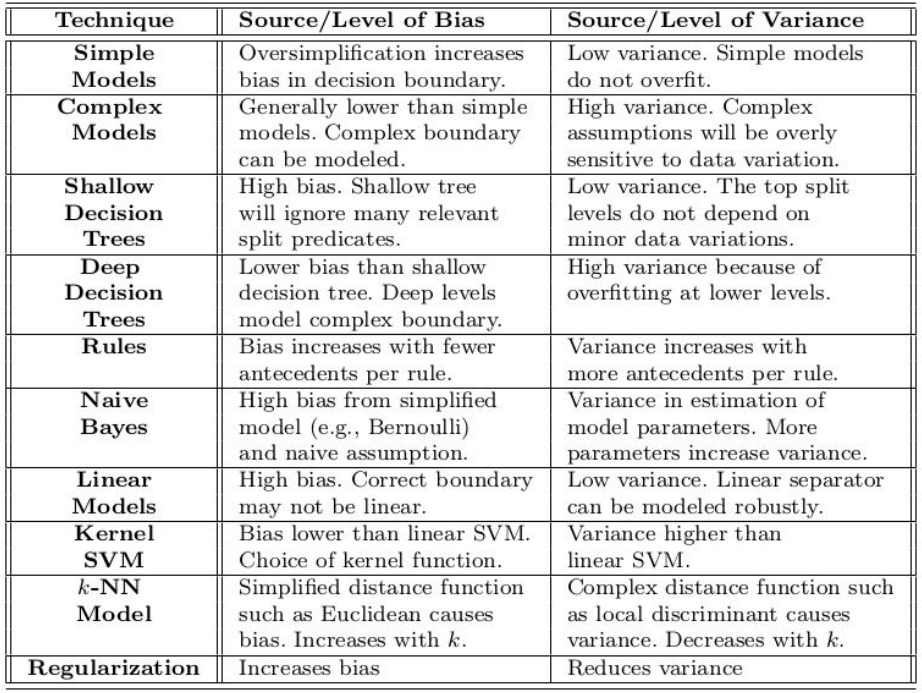
Illus. from Andrew Ng

Some solutions / hints

From Bradski & Kaehler, Learning OpenCV, O'Reilly 2008 ↓



Problem	Possible Solutions
Bias	• More features can help make a better fit. • Use a more powerful algorithm.
Variance	• More training data can help smooth the model. • Fewer features can reduce overfitting. • Use a less powerful algorithm.
Good test/train, bad real world	• Collect a more realistic set of data.
Model can't learn test or train	• Redesign features to better capture invariance in the data. • Collect new, more relevant data. • Use a more powerful algorithm.



Technique	Source/Level of Bias	Source/Level of Variance
Simple Models	Oversimplification increases bias in decision boundary.	Low variance. Simple models do not overfit.
Complex Models	Generally lower than simple models. Complex boundary can be modeled.	High variance. Complex assumptions will be overly sensitive to data variation.
Shallow Decision Trees	High bias. Shallow tree will ignore many relevant split predicates.	Low variance. The top split levels do not depend on minor data variations.
Deep Decision Trees	Lower bias than shallow decision tree. Deep levels model complex boundary.	High variance because of overfitting at lower levels.
Rules	Bias increases with fewer antecedents per rule.	Variance increases with more antecedents per rule.
Naive Bayes	High bias from simplified model (e.g., Bernoulli) and naive assumption.	Variance in estimation of model parameters. More parameters increase variance.
Linear Models	High bias. Correct boundary may not be linear.	Low variance. Linear separator can be modeled robustly.
Kernel SVM	Bias lower than linear SVM. Choice of kernel function.	Variance higher than linear SVM.
k-NN Model	Simplified distance function such as Euclidean causes bias. Increases with k.	Complex distance function such as local discriminant causes variance. Decreases with k.
Regularization	Increases bias	Reduces variance

From C. Aggarwal, Data Mining: The Textbook, Springer 2015 →