Probabilistic Machine Learning:
4. Data modeling and prediction

Tomasz Kajdanowicz, Piotr Bielak, Maciej Falkiewicz, Kacper Kania, Piotr Zieliński

Department of Computational Intelligence
Wroclaw University of Science and Technology

The presentation has been inspired and in some parts totally based on

1. https://towardsdatascience.com/bias-variance-dilemma-74e5f1f52b12
2. https://towardsdatascience.com/measuring-the-power-of-a-classifier-c765a7446c1c
3. Vapnik, Vladimir, Esther Levin, and Yann Le Cun. "Measuring the VC-dimension of a learning machine." Neural computation 6.5 (1994): 851-876.
4. Gardner, Paul, Charles Lord, and Robert J. Barthorpe. "An Evaluation of Validation Metrics for Probabilistic Model Outputs." ASME 2018 Verification and Validation Symposium. American Society of Mechanical Engineers Digital Collection, 2018.

Table of Contents

Pre-reading

1. there was nothing to
2. read

Table of Contents

The dilemma of bias-variance

- relevant for supervised machine learning
- a way to diagnose an algorithm performance by breaking down its prediction error
- three types of prediction errors: bias, variance, and irreducible error

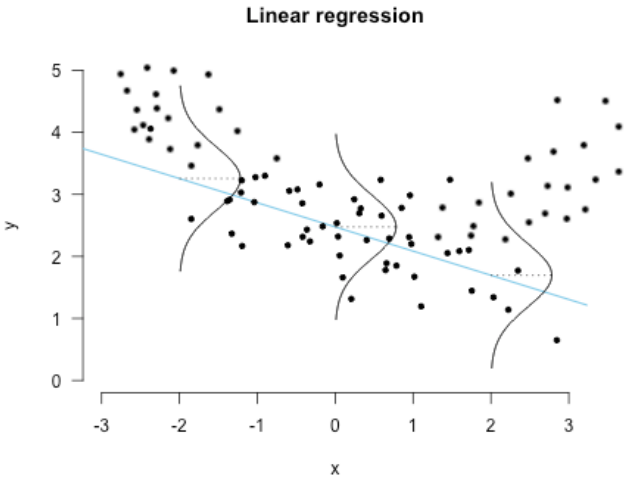
Bias error

Bias error

This is an error due to the difference between the expected prediction of the model and the true value which is trying to predict.

REMIND: bayesian model results with parameter distribution. One can consider the bias from the parameters or multiple realisations of the model point of view.

Bias error



Bayesian linear regression won't be able to model the data! This is known as **underfitting**.

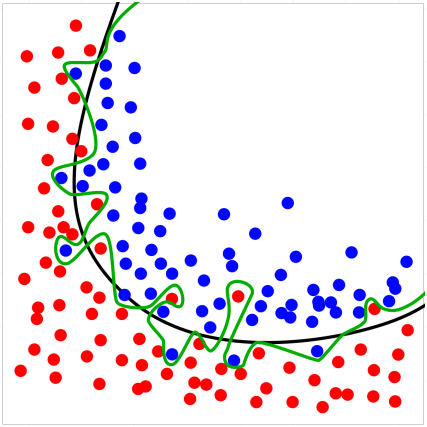
Variance error

Variance error

This is an error due to the the variability of a model prediction for a given data point.

REMIND: bayesian model results with parameter distribution. One can consider the bias from the parameters or multiple realisations of the model point of view.

Variance error



The model has basically memorized the training set, including all the noise. This is known as **overfitting**.

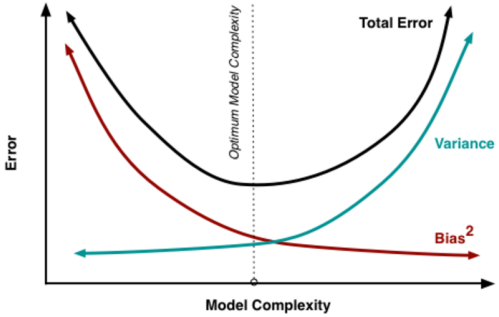
Irreducible error

Irreducible error

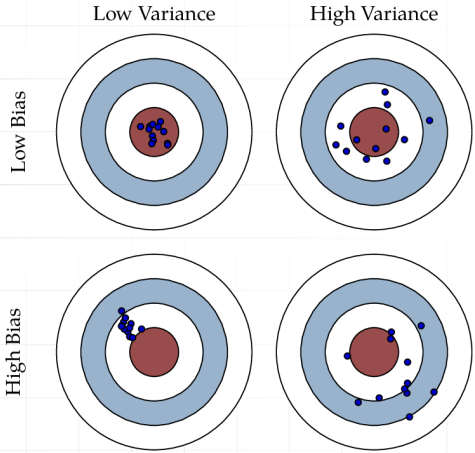
It is the noise term in the true relationship that cannot fundamentally be reduced by any model. It typically comes from inherent randomness or an incomplete feature set

Bias-variance

- bias is reduced and variance is increased in relation to model complexity
- e.g. more polynomial terms are added to a linear regression, the greater the resulting model's complexity will be
- bias has a negative first-order derivative in response to model complexity while variance has a positive slope

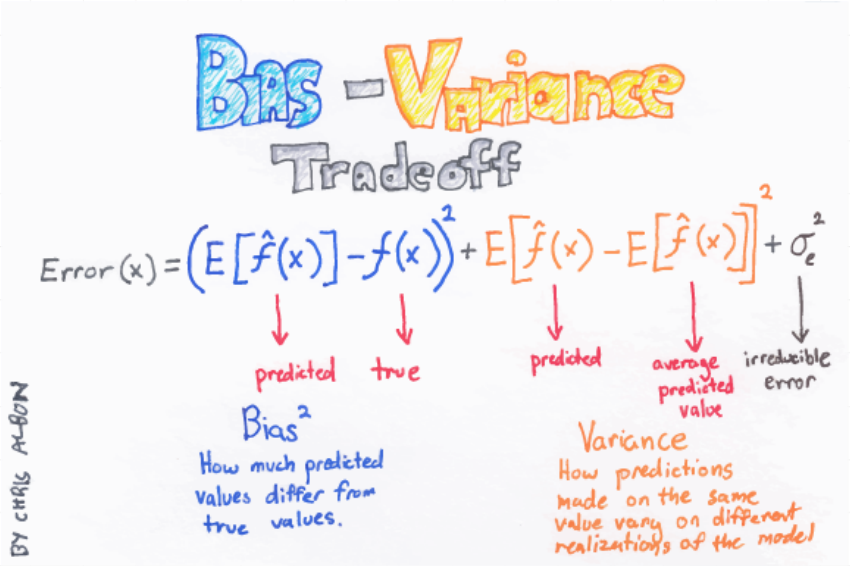


Why bias-variance trade-off?

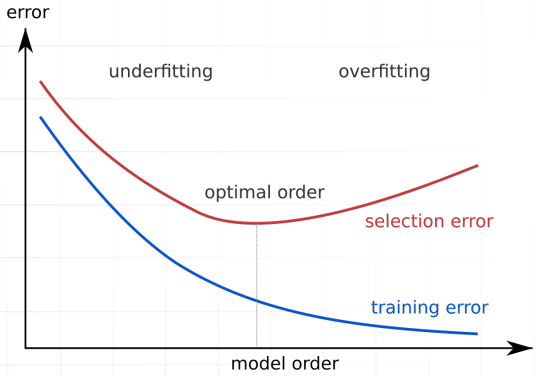


- low variance (high bias) algorithms turn to be less complex, with simple or rigid underlying structure e.g. linear or parametric algorithms such as Bayesian regression and naive Bayes
- low bias (high variance) algorithms turn to be more complex, with a flexible underlying structure e.g. non-linear or non-parametric Bayesian algorithms such as graphical models

What is the total error?



Detecting overfitting and underfitting



- overfitting results in low training error and high test error, while underfitting results in high errors in both the training and test set
- good choice to use is a technique called cross-validation
- simple model and high bias? pick a more complex model
- high variance? use more data, use regularization (injects bias into the model by telling it not to become too complex)

Table of Contents

Curse of dimensionality

Example

- kids like to eat cookies
- let us assume that we have a whole truck with cookies having a different colour, a different shape, a different taste, a different price ...
- if the kid has to choose but only take into account one characteristic e.g. the taste, then it has four possibilities: sweet, salt, sour, bitter, so the kid only has to try four cookies to find what (s)he likes most
- if the kid likes combinations of taste and colour, and there are 4 (I am rather optimistic here :-)) different colours, then he already has to choose among 4x4 different types;
- if he wants, in addition, to take into account the shape of the cookies and there are 5 different shapes then he will have to try 4x4x5=80 cookies
  We could go on, but it can get really difficult to remember the differences in the taste of each cookie.

Curse of dimensionality

Definition

The **curse of dimensionality** refers to the problem of finding structure in data embedded in a highly dimensional space. The more features we have, the more data points we need in order to fill space.

Example

Take for example a hypercube with side length equal to 1, in an n-dimensional space.

- the volume of the hypercube is 1
- f we want to allocate that volume among N smaller cubes (each containing a data point) distributed more or less homogeneously in the n-dimensional hypercube, each small cube will have a volume equal to $\frac{1}{N}$
- their side length d would be

$$d = \left(\frac{1}{N}\right)^{\frac{1}{n}}$$

- for a finite N, d converges to 1 when n goes to infinity
- **that is, the new smaller cubes have each "almost" the same volume as the bigger cube**
- in infinite dimensional space you can put N cubes of volume 1 inside a cube of volume 1!

The curse of dimensionality

The curse of dimensionality:

- this term was first used by R. Bellman in the introduction of his book “Dynamic programming” in 1957:
  All [problems due to high dimension] may be subsumed under the heading "the curse of dimensionality". Since this is a curse, [...], there is no need to feel discouraged about the possibility of obtaining significant results despite it.
- he used this term to talk about the difficulties to find an optimum in a high-dimensional space using an exhaustive search, in order to promote dynamic approaches in programming

Points in high dimensional spaces are isolated

Vapnik–Chervonenkis dimension

Definition

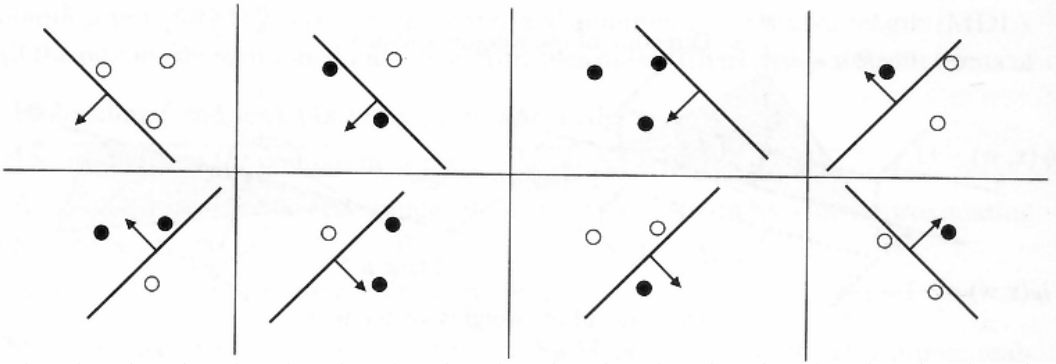
Vapnik–Chervonenkis (VC) dimension is a measure of the complexity of a space of functions that can be learned by a statistical classification algorithm. It is defined as the cardinality of the largest set of points that the algorithm can shatter.

Shattering a set of points

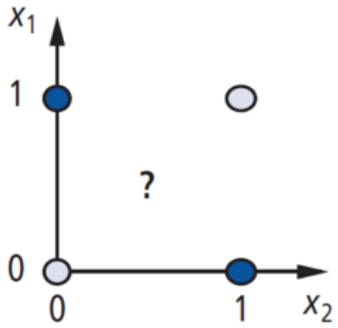
- consider binary classification
- a configuration of N points on the plane is just any placement of N points
- to have a VC dimension of at least N, a classifier must be able to shatter a single configuration of N points
- in order to shatter a configuration of points, the classifier must be able to, for every possible assignment of positive and negative class for the points, perfectly partition the plane such that the positive points are separated from the negative points
- for a configuration of N points, there are 2^N possible assignments of positive or negative

Shattering example

Configuration of 3 points consist of 8 class assignments. VC dimension for a linear classifier is at least 3, since it can shatter this configuration of 3 points. The classifier is able to perfectly separate the two classes.



Shattering example



- VC dim of a linear classifier is lower than 4
- in this configuration of 4 points, the classifier is unable to segment the positive and negative classes in at least one assignment
- two lines would be necessary to separate the two classes in this situation
- in principle: one must prove that there does not exist a 4 point configuration that can be shattered

Applications of VC dimension

- in most cases, the exact VC dimension of a classifier is not so important
- it is used to distinguish different types of algorithms by their complexities
- e.g. the class of simple classifiers could include basic shapes like lines, circles, or rectangles, whereas a class of complex classifiers could include classifiers such as multilayer perceptrons, boosted trees, or other nonlinear classifiers
- complexity of a classification algorithm and therefore its VC dimension, is related to the trade-off between bias and variance
- generally, a model with a higher VC dimension will require more training data to properly train, but will be able to identify more complex relationships in the data

Usage of VC dimension

The **VC dimension** can predict a probabilistic upper bound on the test error of a classification model

$$\Pr\left(\text{test error} \leqslant \text{training error} + \sqrt{\frac{1}{N}\left[D\left(\log\left(\frac{2N}{D}\right)+1\right) - \log\left(\frac{\eta}{4}\right)\right]}\right) = 1-\eta,$$

where D is the VC dimension of the classification model, $0 \leqslant \eta \leqslant 1$ and N is the size of the training set (restriction: this formula is valid when $D \ll N$. When D is larger, the test-error may be much higher than the training-error. This is due to overfitting).

Table of Contents

Evaluation of non bayesian models

- Confusion Matrix
- F1 Score
- Gain and Lift Charts
- Kolmogorov Smirnov Chart
- Log Loss
- Gini Coefficient
- Concordant – Discordant Ratio
- Root Mean Squared Error
- etc.

All of them might be used for evaluation of bayesian methods but will either:

- measure the empirical error of **one realization of bayesian model** (single parameters)
- measure the empirical error of **multiple realization of bayesian model** (multiple parameters drawn from estimated distributions):
  - for statistic of the result (averaging the predictions)
  - as a statistic of evaluation measure (averagin the result of evaluation measure)

Measuring performance of Probabilistic models

- AIM: **validating output distributions**
- ideal validation intuitively provides information on key divergences between the output and validation distributions

Criteria for validation metrics

- it should quantify the difference between the model predictions and observational data
- it should be interpretable and aid identifying improvements
- it should provide objective information and be consistent when applied to different probabilistic models or applications
- it should account for the complete form of the distributions (and not just statistical moments) - if the underlying distribution of the observational data is unknown it should have a non-parametric estimator
- it should be computationally efficient and where applicable, have appropriate convergence properties

Evaluation of distributions

Evaluation metric refers to quantifying the dissimilarities between predictions and observational data.
The most widely used validation metrics are:

- f-divergence
- integral probability metrics

f-divergence

- the class of distances/divergences D that depend on a ratio between probability measures P and Q

$$D_\phi(P, Q) = \int_M \phi\left(\frac{dP}{dQ}\right) dP$$

- for Kullback-Liebler (KL) divergence, $\phi(t) = t\log(t)$
- for Hellinger distance $\phi(t) = (\sqrt{t} - 1)^2$
- for total variation distance $\phi(t) = |t - 1|$

Integral probability metrics

- IPMs differ from f -divergences as they depend on the difference rather than ratio of probability measures

$$D_F(P,Q) = \sup_{f \in F} \left(\int_M f dP - \int_M f dQ \right)$$

where M is a class of functions on M

- Total Variation and Kolmogorov Distances

$$D_{TV}(P,Q) = \sqrt{\frac{1}{2} \int |p(x) - q(x)| dx}$$

- Maximum Mean Discrepancy

$$D_{MMD}(P,Q) = \sup_{f \in F} |E_x[f(x)] - E_y[f(y)]|$$

where x and y are samples from P and Q

Example: Kolmogorov Distances