Introduction to Computational Social Science

Session 4: Machine learning

Maximilian Haag   Constantin Kaplaner

Geschwister-Scholl-Institute for Political Science
LMU Munich

28.11.2022
Room B U103, Tue 14:00–18:00 (bi-weekly)

I Today’s session

Lecture

1. What is machine learning?
2. The supervised learning problem
3. How can machines learn? — The concept of loss
4. Machine learning error
5. Assess performance for classification
6. Quick example

II Today's session

Lab

1. ML for classification
2. ML for regression
3. Text classification: Measuring polarization

What is machine learning?

Definition

> “Ask five researchers what machine learning is and you will likely get five different answers. Most agree that certain methods—for example, deep neural networks—are machine learning, but other techniques—for example, linear regression and the least absolute shrinkage and selection operator (LASSO)—originate in statistics even if they are taught in nearly all machine learning courses. We argue that machine learning is as much a culture defined by a distinct set of values and tools as it is a set of algorithms.”

Source: Grimmer, J., Roberts, M. E., & Stewart, B. M. (2021). Machine learning for social science: An agnostic approach. Annual Review of Political Science, 24, 395-419.

Data versus theory

In political science context:

- **Substantive theory** informs model specification
- **Statistical theory** informs parametric inference

Data versus theory

In machine learning:

- Focus on outcome y instead of focus on predictor/explanatory variable β
- Interest in getting the most accurate prediction of y, not necessarily interested in what explains y

Lingo

- Outcomes/Label = dependent or response variable
- Features = independent variables or predictors
- Model = defines relationship between features and outcome
- Example = particular instance of our data

The supvervised learning problem

“Model”

$$y = f(x, \theta)$$

- y labeled outcome
- x vector of features
- $f(.)$ function we need to learn from the data, $D = (y, x)$
- θ vector of parameters

Types of outcome

- **Classification task**: outcome is a class with 2 or more levels
- **Regression task:** outcome numeric variable

Types of parameters

- **Statistical parameters** parameters that can be estimated from the data (think of regression coefficients)
- **Tuning parameters** influence the behavior of the algorithm, but are not estimated from the data

Rough outline of a supervised machine learning approach

1. Generate **training**, **validation**, and **test** set
2. In the training set, *learn* $f(.)$ and the associated (statistical) parameters
3. In validation set, obtain prediction of $f(x, \theta)$ use those to find optimal values of the tuning paramters of θ
4. After establishing θ make predictions for the test set to evaluate performance on unseen data

How can machines learn? — The concept of Loss

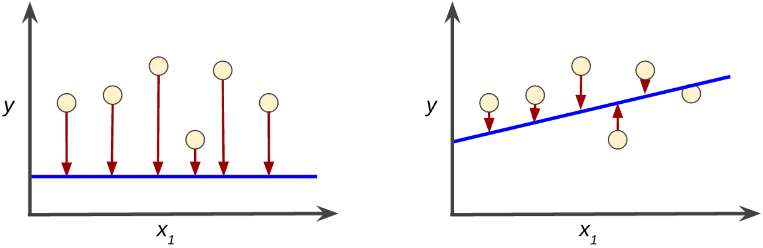
What is loss?

- Loss = penalty for a bad prediction
- Role in machine learning:

Training a model means finding $f(x)$ that **minimizes** loss.

Example for loss: Regression

- Blue: Prediction
- Arrows: Loss



Example loss function (regression)

A loss function is the mathematical representation of loss. For example the L2 loss or squared loss function for a single example is:

$$= (y_{actual} - y_{predicted})^2$$

MSE (mean squared error) is the average squared loss per example over the whole dataset. To calculate MSE, sum up all the squared losses for individual examples and then divide by the number of examples:

$$MSE = \frac{1}{N} \sum_{x,y \in D} (y_{actual} - y_{predicted})^2$$

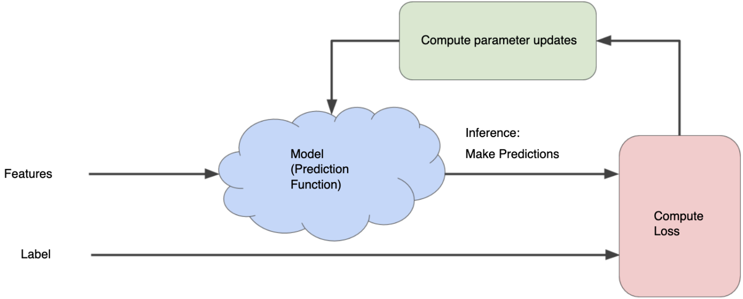
Reducing loss

Challenge:

- How do we reduce loss to generate a good model?

Concept of iteration for reducing loss

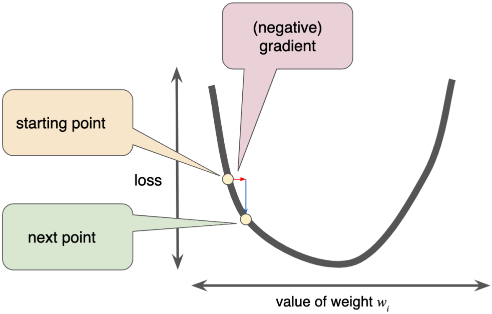
One way of minimizing loss is repeatably trying different parameters until we get the lowest value of loss:



Source: https://developers.google.com/machine-learning/crash-course/reducing-loss/an-iterative-approach?hl=en

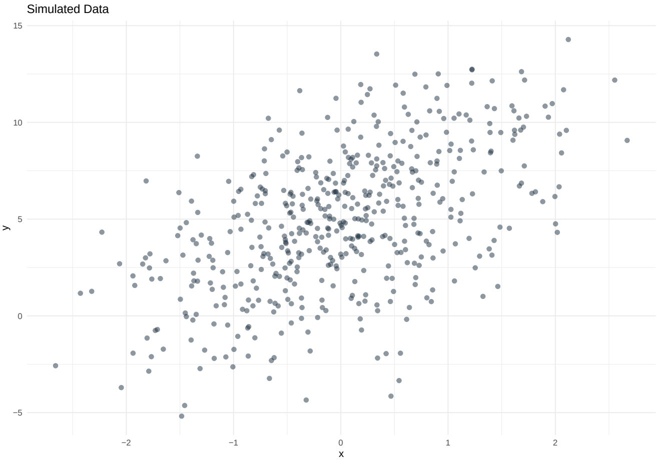
Parameter updates?

But how do we find out what values we should try?

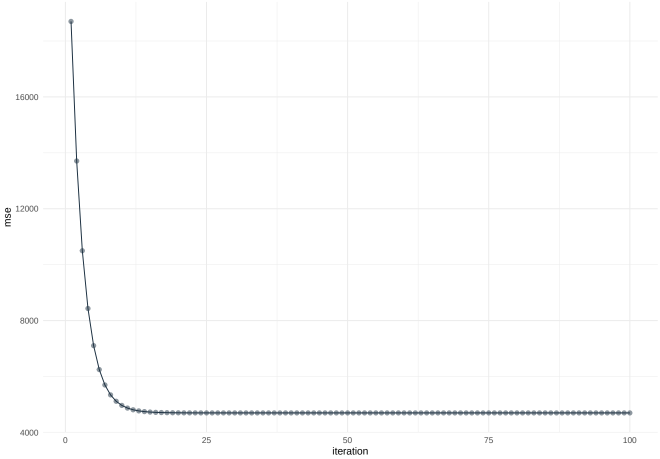


Source: https://developers.google.com/machine-learning/crash-course/reducing-loss/gradient-descent?hl=en

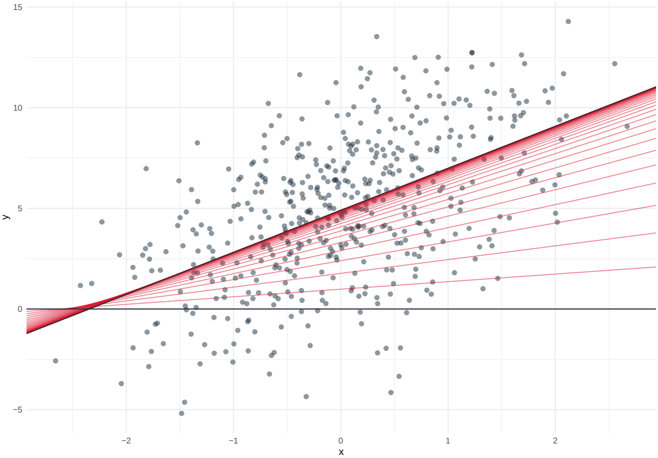
Example I



Example II



Example III



Machine Learning Error

Error

- While loss is optimized within the training portion, error occurs when applying the model to new data
- It can be defined as:

$$Prediction \neq Outcome$$

Sources of Error

- Irreducible error
- Bias error
- Variance error

Irreducible error

Cannot be reduced!

Causes:

- Missing features
- Measurement error in features

Solution:

- Collect more and better data

Bias error

Predictions are systematically off

Causes:

- Poor data prep
- Insufficient features included
- Complex relationships not captured
- Wrong algorithm for the task

Solution:

- Better data prep
- Adjustment of hyperparameters
- Increased model complexity
- Different algorithm

Variance error

Error arises when algorithm is applied to new data

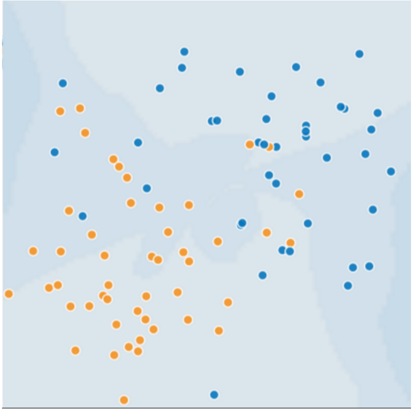
Causes:

- Overfitting

Solution:

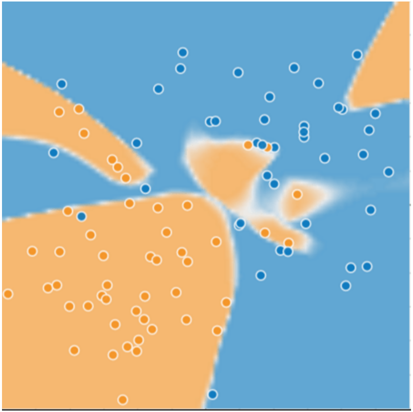
- Dropout layers
- Regularization
- Validation sets

Overfitting I



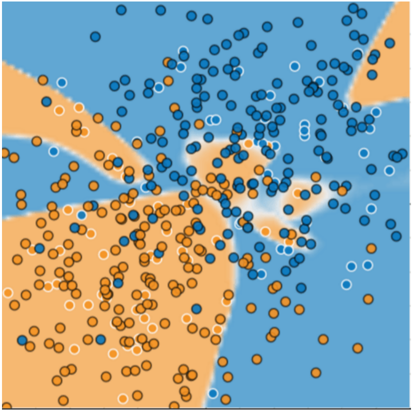
Source: https://developers.google.com/machine-learning/crash-course/generalization/peril-of-overfitting?hl=en

Overfitting II



Source: https://developers.google.com/machine-learning/crash-course/generalization/peril-of-overfitting?hl=en

Overfitting III



Source: https://developers.google.com/machine-learning/crash-course/generalization/peril-of-overfitting?hl=en

Bias variance tradeoff

Problem:

- To reduce bias error, we increase model complexity
- Increased complexity might induce variance error

Vice versa

Assess performance for classification

True postive, true negative

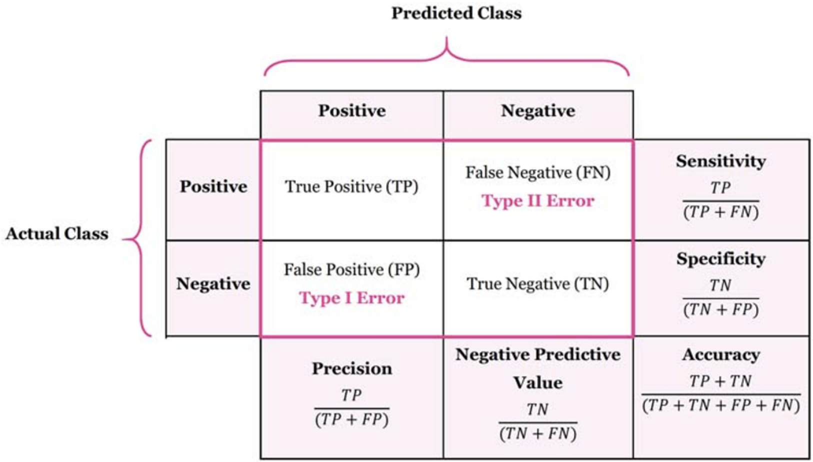
- In classification task we can assess how well our generated model performs by assessing different metrics
- For this we look at true/false positives and true/false negatives

What a true/false postives/negatives

Think of a COVID test:

- If its positive and you really have COVID that would be a true positive
- If its positive but you do not have COVID that would be a false positive
- If its negative and you really do not have COVID that would be a true negative
- If its negative and you really have COVID that would be a false negative

Measures



		Predicted Class		
		Positive	**Negative**	
Actual Class	**Positive**	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN + FP)}$
		Precision $\frac{TP}{(TP + FP)}$	**Negative Predictive Value** $\frac{TN}{(TN + FN)}$	**Accuracy** $\frac{TP + TN}{(TP + TN + FP + FN)}$

Source: https://medium.com/@m.virk1/classification-metrics-65b79bfdd776

Example for machine learning algorithm: Support vector machines

Idea

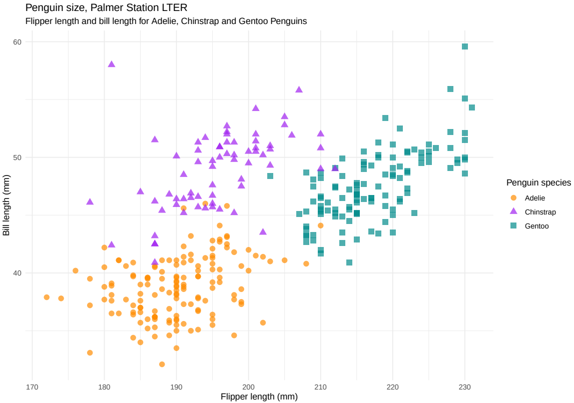
- Find the optimal *hyperplane* which maximes the margins between two classes
- Hyperplane = plane that spans more than 3 dimensions (in 2d line, in 3d plane)
- Margins = Distance between the hyperplane and the closest point

Support vectors

- Support vectors are the data points closest to the decision surface or hyperplane
- They are the points hardest to classify
- They directly influence where the optimal hyperplane lays

2 dimensional case I

Given following data:

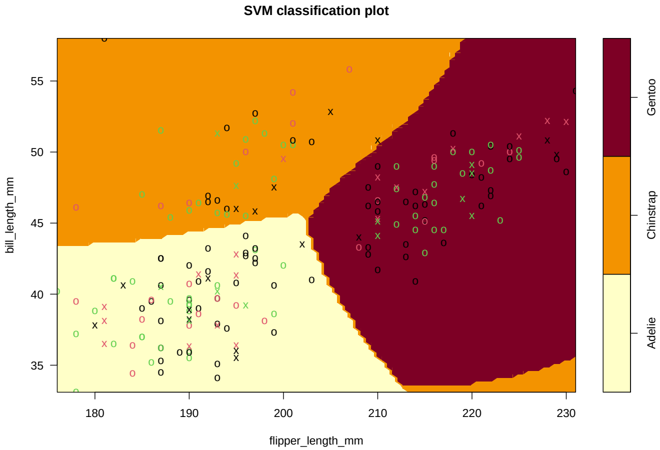


2 dimensional case II

Challenge:

Find the lines that maximize the margins!

II Example: Classifying penguins


SVM classification plot
bill_length_mm
flipper_length_mm
Gentoo
Chinstrap
Adelie