Lecture 4: Learning Theory

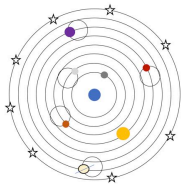
COSC470

Lech Szymanski
Department of Computer Science, University of Otago

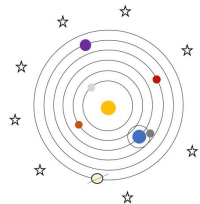
July 31, 2018

Modelling the movements of planets

Ptolemy's geocentric model



Copernicus' heliocentric model



Which model is better?
Which model is more *correct*?

In this lecture...

Broadly speaking, machine learning (ML) is about finding patterns in data.

Typically:

- these patterns are not known;
- these patterns are not obvious;
- these patterns are noisy.

How do we know that our ML methods find the *correct* patterns?

Mathematical framework

- x - input/sensory data (given)
- y - desired output (given in supervised learning)
- $f(x, \beta)$ - model/hypothesis (needs to be chosen appropriately for the problem)
- β - parameters (need to derive through the learning process)

Terminology:
Hypothesis - specific $f(x, \beta^*)$ for given choice of β^*
Hypothesis space - all possible $f(x, \beta)$ for different choices of β

Measuring the learner's performance

Supervised learning

The task in supervised learning is to find $f(x, \beta)$ that models the relationship between given input x and output y.

Loss function $\mathcal{L}(f(x, \beta), y)$ gives learner a score for given set of values β :

- Classification

$$\mathcal{L}(f(x, \beta), y) = \begin{cases} 0 & f(x, \beta) = y \\ 1 & f(x, \beta) \neq y \end{cases}$$

- Regression

$$\mathcal{L}(f(x, \beta), y) = \big(f(x, \beta) - y\big)^2$$

- Cross-entropy

$$\mathcal{L}(f(x, \beta), y) = -y \ln f(x, \beta) - (1 - y) \ln \big(1 - f(x, \beta)\big)$$

Training

The process of training modifies β so as to minimise the loss $\mathcal{L}(f(x, \beta), y)$. It will give a set of parameters β^*.

Risk

True risk

True risk is the expectation of the loss (performance of the hypotheis on all possible data):

$$R(\beta^*) = \int p(x,y)\mathcal{L}\Big(f(x,\beta^*),y\Big)dxdy$$

Not computable!!!

Empirical risk

Empirical risk is the average of the loss computed from available data (performance of the hypotheis on the data we have):

$$R_{\text{emp}}(\beta^*) = \frac{1}{N}\sum_{n=1}^{N}\mathcal{L}\Big(f(x_n,\beta^*),y_n\Big)$$

Generalisation

Consistency

A hypothesis that gives small loss on training data is said to be *consistent*.

Generalisation

A hypothesis where $R_{\text{emp}}(\beta^*) \approx R(\beta^*)$ is said to *generalise well*.

Since $R(\beta^*)$ is not computable, it's impossible to guarantee good generalisation. The best we can do is to examine **guarantees in probability** of a good generalisation?

Generalisation guarantees in probability: the principle

Hoeffding's inequality

This inequality give an upper bound in probability of an average of m samples being different from its expectation by more than ϵ.

$$P(A_m \leq E[A_m] - \epsilon) \leq e^{-2m\epsilon^2}$$

Subbing $R_{\text{emp}}(\beta^*)$ for A_m we have:

- $A_m = R_{\text{emp}}(\beta^*)$
- $E[A_m] = E[R_{\text{emp}}(\beta^*)] = R(\beta^*)$

and thus the probability of $R_{\text{emp}}(\beta^*)$ being more than ϵ outside of $R(\beta^*)$.

Generalisation guarantees in probability: single hypothesis

Using Hoeffding's inequality:

$$P\big(R_{\text{emp}}(\beta^*) \leq R(\beta^*) - \epsilon\big) \leq e^{-2m\epsilon^2}$$

or with probability at most $e^{-2m\epsilon^2}$ the empirical risk is more than ϵ outside of true risk.

Defining $q = P\big(R_{\text{emp}}(\beta^*) \leq R(\beta^*) - \epsilon\big)$, and after some rearranging we get the following expression:

$$R(\beta^*) < R_{\text{emp}}(\beta^*) + \sqrt{\frac{\ln(1/q)}{2m}}$$

with probability $1 - q$.

Generalisation guarantees in probability: finite hypothesis space

What about assurances of generalisation for the choice of model $f(x, \beta)$ before we start training (we don't know β^*)? Assuming there is a finite number of choices for β, we have a finite hypothesis space. Let's denote this set of hypotheses $\mathcal{H}$ and the number of hypotheses $|\mathcal{H}|$.

The upper bound on risk being out of empirical risk by ϵ is the sum of probabilities of every hypothesis having empirical error ϵ. And thus:

$$R(\beta) \leq R_{\mathsf{emp}}(\beta) + \sqrt{\frac{\ln |\mathcal{H}| + \ln(1/q)}{2m}}$$

with probability $1 - q$.

Generalisation guarantees in probability: complexity of $\mathcal{H}$

The expression $\ln |\mathcal{H}|$ is a rough measure of complexity of $\mathcal{H}$. A more accurate measure of complexity of a hypothesis space is VC-dimension, denoted as d. With VC-dimension complexity measure we have:

$$R(\beta) \leq R_{\mathsf{emp}}(\beta) + O\left(\sqrt{\frac{d \ln m/d + \ln(1/q)}{m}}\right)$$

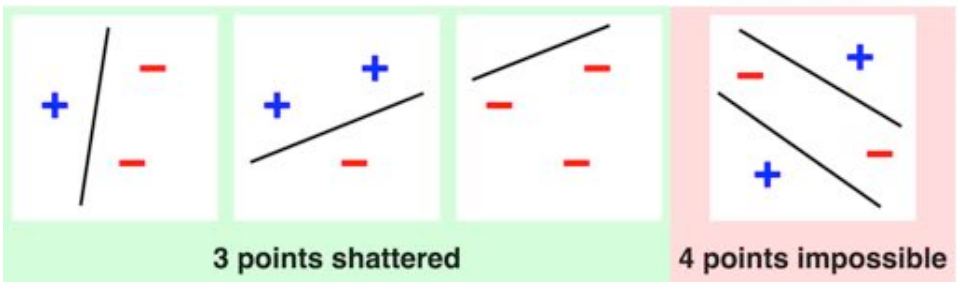
with probability $1 - q$.

Generalisation principle

Choosing a hypothesis space with smaller VC-dimension guarantees (in probability) better generalisation. In other words, the simpler the model, the better chance of generalisation.

VC-dimension

VC-dimension is the maximum number of points that a hypothesis can *shatter*. It measures the complexity of a hypothesis space in terms of its representational power.



https://en.wikipedia.org/wiki/VC_dimension

Exercise: computing the VC-dimension

- Linear classifier?
- Axis-alined rectangles?
- A sinusoid?
- A neural network?

VC-dimension and margin

Theorem 5.1 Vapnik's "The Nature of Statistical Learning Theory" [1]

VC-dimension of hyperplane with margin M is;

$$d \leq \min\left(\frac{1}{M^2}, n\right) + 1$$

Increasing the margin of separation between classes reduces VC-dimension of a hyperplane classifier.

VC-dimension and Support Vector Machines

Recall from previous lecture that SVMs maximises the margin subject to constraints:

$$\max_{\beta,\beta_0} M$$
$$\text{subject to } \frac{1}{||\beta||} y_i(\beta^T \mathbf{X_i} + \beta_0) \geq M.$$

Support Vector Machine minimises the VC-dimension of the separating hyperplane with constraints that ensure the hyperplane separates the data as desired.

VC-dimension and deep learning

- A single hidden layer (shallow) neural network is a universal function approximator.
- Since both shallow and deep networks can do anything, why bother then with deep?
- For certain types of functions (i.e. types of problems), when approximated to the same accuracy by a shallow and deep network, the deeper network has a lower VC-dimension.
- Deep network generally generalise much better than the VC dimension bound would suggest.

Maximum margin and boosting

- Generalisation in boosting improves the more weak classifiers are used (is this at odds with the generalisation principle?)
- Adding weak classifiers in boosting is equivalent to increasing the margin of separation [2]
- It turns out the distribution of the points around the margin play a role in generalisation too - the more point lying on the margin the better generalisation [3]

Maximum margin and neural networks

- Is maximising margin at the penultimate layer of a neural network is meaningless in deep architectures?
- Does, as is the case with boosting, the distribution of points around the margin (or equivalently maximisation of a normalised margin) improve generalisation?

Other complexity measures

VC-dimension is not the only complexity measure of $\mathcal{H}$ [4]:

- Rademacher complexity - measures the ability of the model to fit random noise
- Covering number - measures the size of the hypothesis space.

References

[1] Vladimir N. Vapnik. *The nature of statistical learning theory* Springer-Verlag, Brerlin, Heidelberg, 1995.

[2] Robert E. Schapire, and Yoav Freund. *Boosting: foundations and algorithms*. The MIT Press, Cambridge, Massachusetts, 2012.

[3] Wei Gao, and Zhi-Hua Zhou. *On the doubt about margin explanation of boosting*. Artificial Intelligence, 203:1–18, 2013.

[4] Mohri Mehryar, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, Cambridge, Massachusetts, 2012.