Mathematics for Machine Learning

Prof Willie Brink

Applied Mathematics, Stellenbosch University

Lecture 7: When Models Meet Data

Contents of the module

8.1 Data, models, and learning

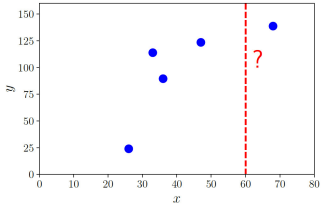
We will think of data as N vectors: each input $\boldsymbol{x}_n$ is a D-dimensional vector of real numbers (features, attributes, covariates).

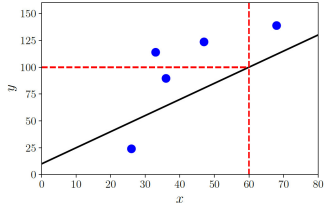
In the supervised learning setting, we have a label (target, response, annotation) y_n associated with each $\boldsymbol{x}_n$.

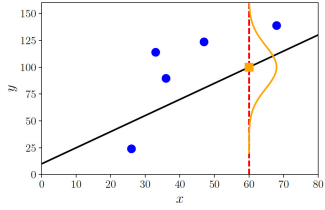
A guiding principle of machine learning is that good models should generalise, that is, perform well (predict labels accurately) on data not seen during training.

The model might be a function: $f : \mathbb{R}^D \to \mathbb{R}$
We often consider linear functions $f(\boldsymbol{x}) = \boldsymbol{\theta}^\top \boldsymbol{x} + \theta_0$, with unknown $\boldsymbol{\theta}$ and θ_0.

Or the model might be a probability distribution, describing the distribution of possible functions and able to express uncertainty in predictions.







Prediction is when we use a trained model on new input data. For probabilistic models, prediction is called inference.

Training is when we estimate the parameters of the model, based on training data, by means of empirical risk minimisation or the principle of maximum likelihood.

Choosing an appropriate model is called model selection, and picking an appropriate structure (e.g. number of parameters, or distribution class) is hyperparameter tuning.

8.2 Empirical risk minimisation

Consider the case of a predictor that is a function, in the supervised learning setting where we have data $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)$.

Want to estimate the parameters $\boldsymbol{\theta}$ of a predictor $f(\cdot, \boldsymbol{\theta})$, such that we fit the data well:

$$f(\boldsymbol{x}_n, \boldsymbol{\theta}^*) = \hat{y}_n \approx y_n \quad \text{for all } n = 1, \ldots, N$$

For some loss (or error) $\ell(y_n, \hat{y}_n)$, we may attempt to minimise the empirical risk:

$$\mathrm{R}_{\mathrm{emp}}(f, \mathcal{X}, \mathcal{Y}) = \frac{1}{N} \sum_{n=1}^{N} \ell(y_n, \hat{y}_n)$$

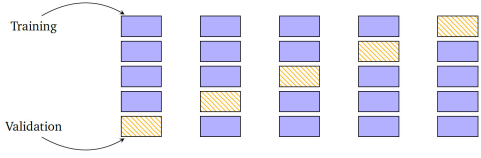
We would actually want a predictor that minimises the expected risk over the infinite set of all possible data and labels, but that's not directly possible.

Empirical risk minimisation can lead to overfitting on the training set (the empirical risk underestimates the expected risk).

Regularisation: introduce a penalty term to make it harder for the optimiser to return an overly flexible predictor.

Can use K-fold cross-validation to assess the generalisation error:

- divide the training data into K chunks; train on $K-1$ of them, use the remaining one for validation to measure the risk on unseen data
- repeat for all K choices of the validation set, and average the performance of the model from the K runs



8.3 Parameter estimation

Maximum likelihood and priors are analogous to loss functions and regularisation.

Maximum likelihood estimation

The likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$ models the uncertainty of the data for a given parameter setting.

Maximum likelihood estimation: find parameter setting $\boldsymbol{\theta}^*$ that "most likely" generated a given dataset $\boldsymbol{x}$.

In supervised learning, we want a predictor that takes input vector $\boldsymbol{x}_n$ and produces a probability distribution of the label y_n.

Assuming i.i.d. (independent, identically distributed) data samples, the likelihood involving the dataset factorises into a product of likelihoods: $\prod_{n=1}^{N} p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta})$

In machine learning we often consider the negative log-likelihood:

$$\mathcal{L}(\boldsymbol{\theta}) = -\log \prod_{n=1}^{N} p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}) = -\sum_{n=1}^{N} \log p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta})$$

Minimising $\mathcal{L}(\boldsymbol{\theta})$ is the same as finding the maximum likelihood estimation of $\boldsymbol{\theta}$.

Maximum a posteriori estimation

If we have prior knowledge about the distribution of $\boldsymbol{\theta}$, we can multiply the likelihood by $p(\boldsymbol{\theta})$. Recall Bayes' theorem: $p(\boldsymbol{\theta}|\boldsymbol{x}) \propto p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})$.

Now we'll find a parameter setting $\boldsymbol{\theta}^*$ that minimises the negative log-posterior, leading to the maximum a posteriori estimate.

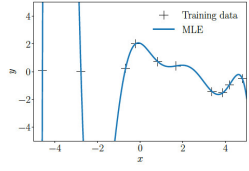
Here it might be convenient to choose a conjugate prior, if possible.

Model fitting

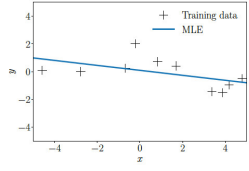
Overfitting refers to the situation where the parametrised model class is too rich to model the dataset, and fits to noise in the training set.

Underfitting is the opposite problem, where the model class is not rich enough, or not sufficiently flexible to model the data.

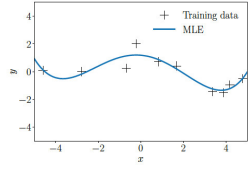
In practice we often define very rich model classes with many parameters (such as deep neural networks), and use regularisation or priors to mitigate overfitting.



(a) Overfitting



(b) Underfitting.



(c) Fitting well.

8.4 Probabilistic modelling and inference

In probabilistic modelling, the joint distribution $p(\boldsymbol{x}, \boldsymbol{\theta})$ of observed variables $\boldsymbol{x}$ and hidden parameters $\boldsymbol{\theta}$ is of central importance. It encapsulates the prior, the likelihood, marginals, and the posterior.

Unlike MLE and MAP that return a point estimate $\boldsymbol{\theta}^*$, Bayesian inference is about finding the posterior distribution. In doing this, uncertainty in parameters can be propagated to predictions, for more robust decision making.

Where MLE and MAP may require optimisation, Bayesian inference typically requires integration which can be practially challenging. May need to resort to

- stochastic approximations (e.g. Markov chain Monte Carlo),
- or deterministic approximations (e.g. variational inference).

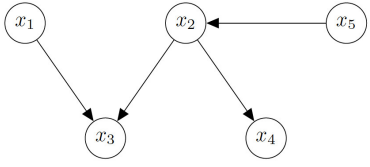
8.5 Directed graphical models

A graphical model visually captures how the joint distribution over random variables can be decomposed into factors depending on only a subset of the variables.

- nodes are random variables $x_1, \ldots, x_K$ (nodes of observed variables are shaded)
- directed links indicate conditional probabilities

The joint distribution $p(x_1, \ldots, x_K)$ corresponding to a graphical model would be

$$p(\boldsymbol{x}) = \prod_{i=1}^{K} p(x_k \,|\, \mathrm{Pa}_k)$$ with Pa_k the parents of x_k (nodes with arrows pointing to x_k)



$$p(x_1, \ldots, x_5) \;=\; p(x_1)\, p(x_2|x_5)\, p(x_3|x_1, x_2)\, p(x_4|x_2)\, p(x_5)$$

8.6 Model selection

Nested cross-validation can be implemented to for choosing a best model (with an inner loop that tests hyperparameters of a particular model).

Simpler models tend to be less prone to overfitting. An objective of model selection can be to find the simplest model that explains the data well (Occam's razor).

In Bayesian model selection, complex and very expressive models may automatically turn out to be a less probable choice for modelling a given dataset.