4F13: Probabilistic Machine Learning

Summarized from C. Rasmussen & D. Krueger lectures, Michaelmas 2021

Oussama Chaib

October 2021

Contents

1 Prelude

1.1 Machine learning: what and why?

- Approach of the book: the best way to make machines that can learn from data is to use the tools of probabilistic theory.
- Probability theory: applied to anything involving uncertainties.
  - What is the best prediction?
  - What is the best model?
  - What measurement should I perform next?
- This systematic approach of using probability theory is often referred to as the **Bayesian approach**.
- To avoid upsetting some audiences, we use the more "neutral" term **probabilistic approach** (some of the methods we use like *maximum likelihood* estimation are not Bayesian, but certainly fall under probabilistic methods).
- **Machine learning:** A set of methods that can automatically detect patterns in data, and then use them to predict future data or to perform other kinds of decision making using that data.

1.2 Types of machine learning

1. **Supervised (predictive):** Learn a mapping from inputs x to outputs y given a training set $D = (x_i, y_i)_{i=1}^N$ containing N samples.
   (a) **Classification:** When the output y_i is nominal (categorical) variable of a finite set (i.e: gender).
   (b) **Regression:** When the output y_i is real-valued.
2. **Unsupervised (descriptive):** No specified pattern, no obvious error metric to use (i.e: neural networks).
3. **Reinforcement learning:** Learning how to behave when given occasional reward or punishment signals.

1.3 Supervised learning

1.3.1 Classification

- Typical example: $y = f(x)$ with y is a **finite** number of points (x can be continuous, discrete, or a combination of both).
- We use the hat symbol to denote an estimate (i.e: $\hat{y}$ is an estimate of y).
- We would like to predict the result on novel input x_*, meaning ones that weren't seen before.
- **Probability notation:** Probability of output y given the input x, the training dataset D, and the model M.

- If the model is known and we do not wish to compare models, we drop the M so that: $p(y|x, D, M) \equiv p(y|x, D)$.
- Our best guess (most probable class label, mode of the distribution, MAP: maximum a posteriori estimate) will maximize this probability.

$$\hat{y} = argmax_{c=1}^{C} \ (p(y|x, D, M)$$

1.3.2 Regression

- Just like classification but the response variable y is **continuous**.
- Will be explored further.

1.4 Models for supervised learning

1. **Parametric models:** $p(y|\mathbf{x})$ Fixed number of parameters. Usually faster but require *stronger assumptions* about the nature of the data distribution.
2. **Non-parametric models:** $p(\mathbf{x})$ More flexible but computationally hungry for large datasets.

1.4.1 Linear regression

Can be written as follow:

$$y(\mathbf{x}) = \mathbf{w^T}.\mathbf{x} + \epsilon$$

where $\mathbf{w^T}$ is the vector containing **weights**, and ϵ the **residual error** (or noise) between our linear predictions and the input data.
We often assume that the error vector follows a **Gaussian** or **normal** distribution:

$$\epsilon \sim \mathcal{N}(\mu, \sigma^2)$$

where μ is the **mean** and σ^2 the variance, and $\mathcal{N}$ represents the normal distribution. The parameters of the model can then be defined such that:

$$\theta = (\mathbf{w}, \sigma^2)$$

Linear regression can be used to model **non-linear** relationships by introducing a **basis function** $\mathbf{\Phi}(x)$:

$$p(y|x, \theta) = \mathcal{N}(y|w^T \mathbf{\Phi}(x), \sigma^2)$$

2 A brief review of probability theory

- Probability of a union of two events:

$$p(A \wedge B) = p(A) + p(B) - p(A \vee B)$$

- Joint probability:

$$p(A, B) = p(A \vee B) = p(A|B)p(B)$$

- **Marginal distribution**: Rotating over all the possible values of another variable B. We write the joint probability of A and B as the product of a conditional probability and a marginal distribution.

$$\underbrace{p(A, B)}_{joint\,probability} = \sum_b \underbrace{p(A|B = b)}_{conditional} \underbrace{p(B = b)}_{marginal}$$

$$p(A) = \int_b p(A)p(A|B = b)db$$

- **Conditional probability**

$$p(A|B) = \frac{p(A, B)}{p(B)}$$

- **Bayes rule**

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}$$

For practical cases:

$$\underbrace{p(\theta|data)}_{posterior} = \frac{\overbrace{p(data|\theta)}^{\propto likelihood} \overbrace{p(\theta)}^{prior}}{p(data)}$$

also:

$$p(w|x, y, M) = \frac{p(w|M)p(y|x, w, M)}{p(y|x, M)}$$

(x is removed in $p(w|M)$ because the weights **w** are independent of x)

- **Mean** (expected value) and **variance**

$$\mu_X = E(X) = \sum_\chi xp(x) = \int_\chi xp(x)dx$$

$$\sigma^2 = var[X] = E[(X - \mu)^2] = \int (x - \mu)^2 p(x)dx$$

- **Gaussian (normal) distribution**

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$$

- **Covariance:** Measures the degree of correlation between two random variables X and Y:

$$cov[X, Y] = E((X - E(X)).(Y - E(Y))) = E(XY) - E(X)E(Y)$$

Covariance can be between 0 and infinity. Correlation is normalized between -1 and 1.

- **Monte Carlo approximation**

$$z = \int f(\chi)p(\chi)d\chi = \frac{1}{T}\sum_{s=1}^{S} f(x_s)$$

3 Introduction

3.1 Modelling data

3.1.1 Purpose of models

The purpose of models is:

- Making predictions
- Generalizing: interpolation, extrapolation
- Generating more data from a similar distribution as the training set
- Compressing and summarizing data
- Interpreting statistical relationships in data
- Evaluating the relative probability of a hypothesis on data

3.1.2 Origin of models

The origin of models can be:

- **First principles:** (i.e: Newtonian mechanics model, high level of accuracy)
- **Observations and data:** (i.e: annual production of timber depending on climate and geographical factors)

Definition – Machine learning is a broad term that covers theory and practice of mathematical models which to a significant degree rely on data.

3.1.3 Priors

Every model relies on priors:

- **Knowledge**
- **Assumptions** (could be true or false)
- **Simplifying assumptions** (not necessarily true, but good enough – i.e: the mistake associated with the assumption is fairly small even though it might not be necessarily true)

3.1.4 Components of a model

Time series have:

- Unobserved/hidden/latent variables ($x(t)$, $x(t-1)$)
- Observations (shaded $y(t)$, $y(t-1)$)
- Parameters to link everything
  - Transitions (between latent variables)
  - Emissions (from a lantent variable to an observation)

Note: The number of latent variables increases with the number of observations, but the number of parameters doesn't!
– **Learning/training models:** "What to do with all this data?"
Depending on the data, some models include: inference, estimation, sampling, and marginalization.

3.1.5 Practical modelling

1. Treat (training) the unobserved quantities (latent variables, observations, parameters)
2. Make predictions based on test cases, interpret the trained model (can we figure out what the model is trying to tell us about the data?)
3. Evaluate the accuracy of the data
4. Model selection and criticism (choose the right model or variant of the model, identify limitations)

There is not "true" or "correct" model – *"All models are wrong, but some are useful"* - George E.T. Box

3.2 Linear in the parameters regression

Let's start off with a dataset $D = x_i, y_{i}{}_{i=1}^{N}$. From a dataset of N points, we would like to infer the coordinate y_* at x_*. A simple model to do that would be polynomial **linear in the parameters** regression:

$$f_w(x) = w_0 + w_1 x + w_2 x^2 + ... + w_M x^M$$

where w_i are the corresponding weights of the polynomial, and the **parameters** of the model.

– **Relevant questions:**

Model structure: Should we choose a polynomial? What degree M should we choose?
Parameters: What values of w_i do we choose?

3.2.1 Least squares approach

Let's find the "best" polynomial (degree M and weights w_i) according to the least squares approach (minimizing the variance or sum of squared error e_i^2):

$$e_i(x)^2 = (y_i(x_i) - f_w(x_i))^2$$

$$E(x) = \sum_{i=1}^{N} e_i^2$$

– **Some notations:**

- **Training target** (input data that we would like to fit):

$$y = [y_1, ..., y_N]$$

- **Prediction** (the functions that will attempt to fit the data y):

$$f = [f_w(x_1), ..., f_w(x_N)]$$

- **Errors**:

$$e = f - y = [e_1, ..., e_N]$$

- **Parameters (what we're solving for!):** The **weights** of f:

$$w = [w_0, ..., w_M]$$

The sum of errors we would like to minimize is (defined as a variance to minimize):

$$E(w) = ||e||^2 = e^T e = (f - y)^T (f - y)$$

We define the **basis function** of the model (linear in the parameters model) as a matrix that contains all the x^j (the matrix contains a list of all polynomials of unknown x with maximum order from 0 to M) :

$$\Phi = \Phi_{i,j} = \begin{bmatrix} \phi_0(x_1) & ... & \phi_M(x_1) \\ ... & ... & ... \\ \phi_0(x_M) & ... & \phi_M(x_M) \end{bmatrix}$$

$$\text{where } \phi_j(x) = x^j$$

Ultimately, we can define the prediction function as a **linear in parameters** model that is the product of the *basis function* and the *weights*:

$$\boxed{f = \Phi.w}$$

To minimize the sum of squared errors $E(w)$, the gradient of the function should be equal to the zero vector:

$$\frac{\partial E(w)}{\partial w} = 2\Phi^T \Phi w - 2\Phi^T y = \text{"0"}$$

The weight vector $\hat{w}$ that minimizes $E(w)$:

$$\boxed{\hat{w} = (\Phi^T Phi)^{-1} \Phi^T y}$$

The error vector is <u>minimal</u> when it orthogonal with all columns of Φ:

$$\Phi^T.e = 0$$

When we go to higher values of M, we are able to fit a function that minimizes the error term but we may run into the problem of **overfitting**. In that case, **additional assumptions** are needed.

3.2.2 Probabilistic approach: Likelihood and concept of noise

(Probabilistic view of what was done in the previous section. We're doing the exact same thing but using probabilities, only to end up getting equivalent weights)

<u>Definition:</u> The **likelihood** is the probability of the data given the parameters.

Example:

- $p(y|w, \sigma^2)$ is the probability of the observed data given the weights and noise.
- $L(w) \propto p(y|w, \sigma^2)$ is the **likelihood** of the weights.

The **maximum likelihood** of the weights:

$$\hat{w} = argmax(L(w)) = argmax(e^{-\frac{E(w)}{2\sigma^2_{noise}}}) = argmin(E(w))$$

which is equivalent to the result that we got using least squares. But we still haven't solved the problem: we still overfit!

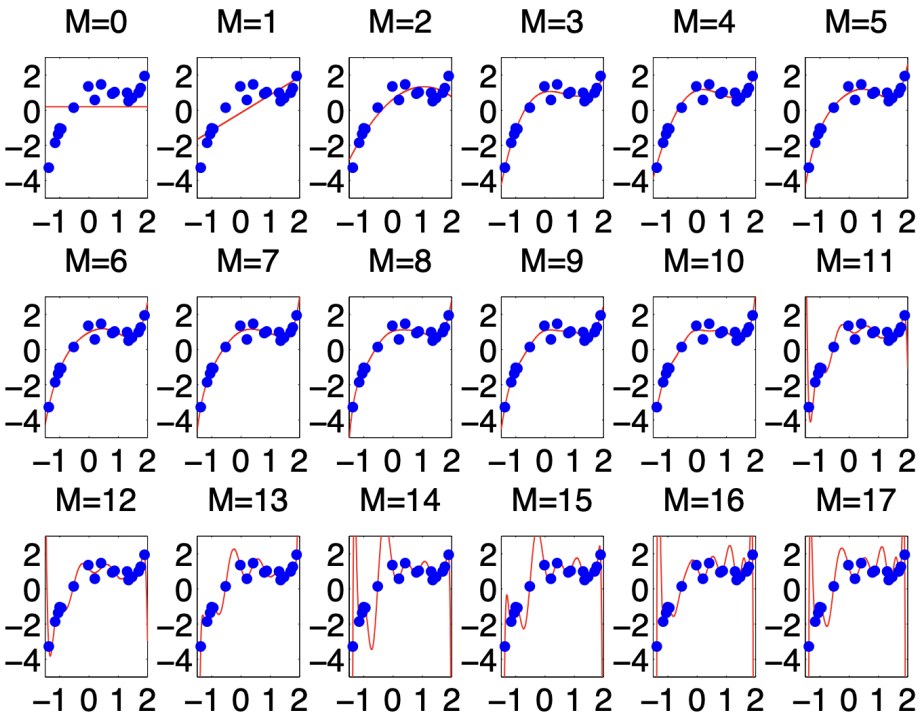


Figure 1: Overfitting illustrated. In this example, we are solving for the weights for each case of M

3.3 Comments from QnA

– Why is the error vector e minimal if it's orthogonal to all columns of ϕ?

ϕ is a fixed function, it doesn't have any parameters in it. Still linear in parameters because the product of w and ϕ matrices is linear (even at high polynomial orders).

– What is Euclidian geometry?

Simple geometries, straight lines, basic shapes (including circle).

– Bayesian methods, why are they not as popular?

Not so sure about that. Maximum likelihood (used often, single value that best explains the data, quite popular/successful).

4 Bayesian inference

Using Bayesian probabilities for statistical inference.

4.1 Maximum likelihood vs Marginal likelihood

- **Maximum likelihood**

  1. Calculate the Gaussian likelihood:

$$p(y|x,w,M) = (\frac{1}{\sqrt{2\pi\sigma^2_{noise}}})^N \Pi_{n=1}^N e^{-\frac{(y_n - f_w(x_n))^2}{\sigma^2_{noise}}}$$

2. Determine the weights **w** which maximize the likelihood **(maximum likelihood**:

$$w_{ML} = argmax\ p(y|x, w, M)$$

3. Make predictions!

$$p(y_*|x_*, w_{ML}, M)$$

- **Marginal likelihood** (or "evidence")

1. Calculate the marginal likelihood:

$$p(y|x, M) = \sum p(w|x, M)\ \overbrace{p(y|x, w, M)}^{likelihood} dw = \mathcal{N}(w; \mu, \Sigma)$$

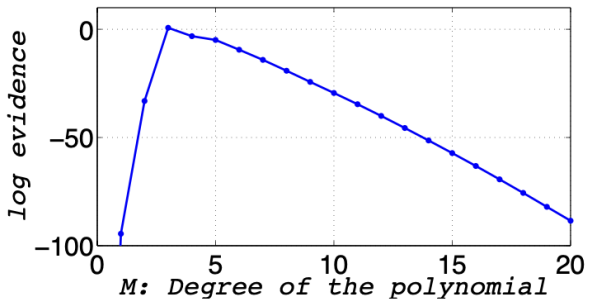
Depends ONLY on your training sets and model

2. Find the point where

$$log(marginal\, likelihood) = 0$$

3. Make predictions:

$$p(y_*|x_*, x, y, M) = \int p(y_*, w|x_*, x, y, M)dw = \int p(y_*|w, x_*, M)\, p(w|x, y, M)dW$$



How can probabilities be higher than 1? Not the case for **probability densities**.