Machine Learning and AI

- Methods and Algorithms -

Personnal Notes

François Bouvier d'Yvoire

CentraleSupélec & Imperial College

Current Branch : `ProbaInference`
Commit : `0da884432d33167cec3bc27d1e880e939dc06ef8`

Contents

Todo list

Chapter 1

Common Machine Learning algorithms

This chapter is dedicated to the most common ML algorithms, a major part of the notes come from the mml-books.com

Find better paragraph layout

Add bibtex reference

1.1 Graphical Model for Probabilistic Inference

Based on the Probabilistic Inference Course of Marc Deisenroth (Imperial College)

1.1.1 Probabilistic Pipeline

Here is a simple pipeline of the inference process with a model

1.1.2 Probabilistic graphical Model

In order to deal with complex and big probalistic model, we can use different kind of graphs which represent relationships between random variables. We define the random variables as nodes and the probabiistic or functional relationship between variables as edges in the graphs. With them you can :

- Visualize the structure
- Have insights into properties (such as conditional independance)
- Design or Motivate new models
- Compute some inference and learning as graphical manipulations

There exists 3 kinds of model : Bayesian networks (directed graphical models), Markov random fields (undirected graphical models) and factor graphs (with nodes which are not random variables)/

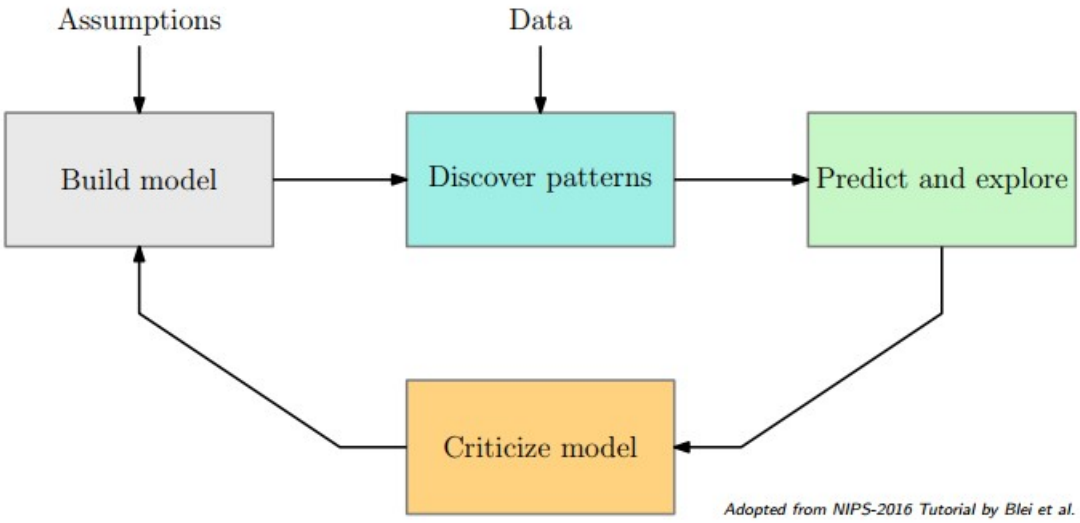

Figure 1.1 – Simple Pipeline of how to build model for inference

Bayesian Networks

add basic graphs exemple

They are defined by

- Nodes : Random variables
- Shaded nodes: Observed random variables
- Other : Latent Variables
- Edges : (a, b) $\iff$ conditionnal distribution $p(b|a)$

add graphicals model of linear regression

D-Separation: A set A of nodes is d-Separated (conditionnaly independant) from B by C iff all path between A and B are blocked. C is tht set of observed variables in the graphical model.

Definition 1
A Path is **Blocked** *id it includes a node such that either :*

- *The arrow on the path meet either head-to-tail or tail-to-tail at the node, and the node is in C*

- *The arrowe meet head-to-head at the node, and neither the node nor any of its descendants is in the set C*

1.2 Linear Regression

The Linear regression problem correspond to find a linear mapping $f(x)$ based on noisy observation $y = f(x) + \epsilon$, where ϵ is a noise. Finding the regression function require :

- Choice of parameters (function classes, dimension)
- Choice of probabilistic model (Loss function, ...)
- Avoiding under and overfitting
- Modeling uncertainty on data

Definition 2

- $p(x,y)$ *is te joint distribution*
- $p(x)$ *and* $p(y)$ *are the marginal distributions*
- $p(y|x)$ *is the conditional distribution of y given x*
- *in the context of regression,* $p(y|x)$ *is called likelihood,* $p(x|y)$ *the posterior,* $p(x)$ *the prior and* $p(y)$ *the marginal likelihood or evidence.*
- *they are related by the Bayes' Theorem :* $p(x|y) = \frac{p(y|x)p(x)}{p(y)}$

1.2.1 Conjugacy

In order to compute the posterior, we require some calculations which imply the prior, and can be intractable as a closed form. But given a likelihood, it can exist prior which give closed-form solution for the posterior. This is the principle of conjugacy (between the likelihood and the prior)

1.2.2 Maximum Likelihood Estimation (MLE)

Closed-Form Solution

In some cases, a closed-form solution exist, which make computation easy (but not necesseraly cheap)

Maximum A Posteriori Estimation (MAP)

1.3 Gradient Descent

1.3.1 Simple Gradient Descent

1.3.2 Gradient Descent with Momentum

1.3.3 Stochastic Gradient Descent

1.4 Model Selection and Validation

1.4.1 Cross-Validation

1.4.2 Marginal Likelihood

1.5 Bayesian Linear Regression

1.5.1 Mean and Variance

1.5.2 Sample function

Chapter 2

Bayesian Algorithms

2.1 Gaussian Process

This section is made with a great inspiration from the Probabilistic Inference from Marc Deisenroth (Imperial College London)

2.1.1 Problem setting

For a set of observation $y_i = f(x_i) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2$, we want to fine a distribution over **functions** $p(f)$ that explains the data. It's not exactly the same as a linear regression problem because we do not look for only one function.

This is a really power full process used in a lot of different problematic thanks to its robustness and known properties (in comparison to deep learning methods).

- Reinforcement Learning and Robotics
- Bayesian optimization
- Geo-statistics
- Sensor networks
- Time-series modelling and forecasting
- High-energy physics
- Medical application

Formally a Gaussian Process is a multivariate Gaussian distribution with infinite variables (countable or even uncountable).

2.1.2 Definition

A Gaussian process is defined by a mean function $m(\dot{)}$ and a covariance function (=kernel) $k(\dot{,}\dot{)}$.

The mean function represents the average of all the function.
the Covariance function allows us to compute covariance between any two functions. Notes that the function are unknown, and only this correlations are fully known.

2.1.3 Gaussian Process Inference

Considering X training inputs and y training target, the Bayes' theorem in the case of Gaussian Process become

$$p(f|X,y) = \frac{p(y|f,X)p(f)}{p(y|X)}$$

Which gives us a Likelihood $p(y|f,X) = \mathcal{N}(f(X),\sigma^2\boldsymbol{I})$, a Marginal Likelihood $p(y|X) = \int p(y|f,X)p(f|X)df$ and a Posterior $p(f|y,X) = \mathbf{GP}(m_{post},k_{post})$

How can we manage to work with the distribution over function, and then infinite dimension for calculus, etc. ? This is possible because each time you consider only finite sample, computing the joint distribution boils down to work on finite-dimensional multivariate Gaussian distributions

Then we can use the gaussian properties (including gaussian prior as conjugate, ...) to make prediction :
(We define X_*, f_*, etc as the test data and predicted function)

$$p(f(x_*)|X,y,x_*) = \mathcal{N}(m_{post}(x_*),k_{post}(x_*,x_*))$$

This can also be seen as

$$p(f(x_*)|X,y,x_*) = \mathcal{N}(\mathrm{E}[f_*|X,y,X_*],V[f_*|X,y,X_*])$$

with $\mathrm{E}[f_*|X,y,X_*]$ = prior mean + "Kalman gain" * error $= m(X_*) + k(X_*,X)(\boldsymbol{K} + \sigma_n^2\boldsymbol{I})^{-1}(y - m(X))$ and $V[f_*|X,y,X_*] = k(X_*,X_*) - k(X_*,X)(\boldsymbol{K} + \sigma_n^2\boldsymbol{I})^{-1})k(X,X_*)$