# Reports on PRML Reading Talk

**Siyu Wang**
Department of Computer Science
Tsinghua University
thuwangsy@gmail.com

## 1 Introduction to ML, probability basics–prml chap.1,2, mlapp chap.1,2

The key purpose of machine learning is to extract good **features** and **patterns** from **data**. To realize this purpose, we mainly use function fitting as the tool to solve this problem. In this section we will focus on a polynomial fitting problem, solving it from different angles and considering differences and relations among them.

But we must keep in mind that, function fitting is only a mathematical tool, not our purpose.

Consider the following problem: given some observed points:

$$D = \{(x_1, t_1), (x_2, t_2), \cdots, (x_N, x_N)\}$$

in which, $t = \sin(2\pi x) + \mathcal{N}(0, \sigma^2)$. Then we want to find a polynomial function

$$t = y(x, \mathbf{w}) = \mathbf{w_0} + \mathbf{w_1 x} + \mathbf{w_2 x^2} + ... + \mathbf{x_M x^M} \tag{1.1}$$

to fit these points, so when given new values of $x$, we can predict the corresponding $t$.

### From function fitting angle

We want to optimize the unknown parameter $\mathrm{w}$ in equation.1.1 by minimize the *error function*

$$E(\mathrm{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathrm{w}) - t_n\}^2. \tag{1.2}$$

So we can set the error function's derivative with respect to **w** to zero and easily get the optimal parameter $\mathbf{w}^*$.

To reduce the over-fitting problem, we may make a little change about the error function by adding a regularization item:

$$E(\mathrm{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathrm{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathrm{w}\|^2. \tag{1.3}$$

### From probabilistic angle

Here, we shall assume that, given the value of $x$, the corresponding value of $t$ has a Gaussian distribution with a mean equal to the value $y(x, \mathrm{w})$ and a fixed variance $\beta$. So we have

$$p(t|x, \mathrm{w}, \beta) = \mathcal{N}(y(x, \mathrm{w}), \beta^{-1}) \tag{1.4}$$

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

**MLE: maximum likelihood estimation**

Given $N$ input value $\mathbf{x} = (x_1, \cdots, x_N)^T$ and their corresponding target value $\mathbf{t} = (t_1, \cdots, t_N)^T$, we have the likelihood function with the form as

$$p(\mathbf{t}|\mathbf{x}, \mathrm{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n | y(x_n, \mathrm{w}), \beta^{-1}). \tag{1.5}$$

It's convenient to maximize the logarithm of the likelihood function

$$\ln p(\mathbf{t}|\mathbf{x}, \mathrm{w}, \beta) = -\frac{\beta}{2} \sum_{n=1}^{N} \{y(x_n, \mathrm{w}) - t_n\}^2 + \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi). \tag{1.6}$$

So when determining $\mathbf{w}$, maximizing likelihood function is equal to minimizing the error function in equation.1.2. And we obtain

$$\frac{1}{\beta_{ML}} = \frac{1}{N} \sum_{n=1}^{N} \{y(x_n, \mathrm{w}_{ML}) - t_n\}^2. \tag{1.7}$$

So we get a probability distribution of $t$ when given a new value of $x$

$$p(t|x, \mathrm{w}_{ML}, \beta_{ML}) = \mathcal{N}(t|y(x, \mathrm{w}_{ML}, \beta_{ML}) \tag{1.8}$$

**MAP: maximum a posteriori estimation**

Now we take a step towards a more Bayesian approach and introduce a prior distribution over the polynomial coefficient w and for simplicity, consider a Gaussian distribution of the form

$$p(\mathrm{w}|\alpha) = \mathcal{N}(\mathrm{w}|0, \alpha^{-1}\mathrm{I}) = (\frac{\alpha}{2\pi})^{(M+1)/2} \exp\{-\frac{\alpha}{2}\|w\|^2\} \tag{1.9}$$

where $\alpha$ is the precision of the distribution and $M + 1$ is the total number of elements in the vector $\mathbf{w}$. Using Bayes' theorem, we can get the posterior distribution for $\mathbf{w}$, which is proportional to the product of the prior distribution and the likelihood function

$$p(\mathrm{w}|\mathbf{x}, \mathbf{t}, \alpha, \beta) \propto p(\mathbf{t}|\mathrm{w}, \mathbf{x}, \beta)p(\mathrm{w}|\alpha). \tag{1.10}$$

And finally we find that the maximum of the posterior is given by the minimum of

$$\frac{\beta}{2} \sum_{n=1}^{N} \{y(x_n, \mathrm{w}) - t_n\}^2 + \frac{\alpha}{2} \mathrm{w}^T \mathrm{w}. \tag{1.11}$$

Thus we see that maximizing the posterior distribution is equivalent to minimizing the regularized sum-of-square error function in the form (1.3)

**Inference and decision**

Here, we come to a classification problem and we have three different approaches to doing inference and decision.

1. First solve $p(\mathrm{x}|\mathcal{C}_k)$ as well as $p(\mathcal{C}_k$ for each class individually, then figure out $p(\mathcal{C}_k|\mathrm{x}) = \frac{p(\mathrm{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathrm{x})}$ and $p(\mathrm{x})$ can be gotten from $p(\mathrm{x}) = \sum_k p(\mathrm{x}|\mathcal{C}_k)p(\mathcal{C}_k)$. This is equivalent to model the joint distribution $p(\mathrm{x}, \mathcal{C}_k)$
2. Determine posterior class probabilities $p(\mathcal{C}_k|\mathrm{x})$ and then use decision theory. *discriminative model*.
3. Find $f(\mathrm{x})$ which maps x directly to a class label.

The first approach is too time-consuming and demanding, while the last one is simplest but not so robust as approach.2 because we can see a lot of information from the posterior probabilities.

**For others**

Other than the main ideas and problems discussed above, there are still some other small items here:*Gaussian probability distribution, exponential family, students' t distribution, conjugate prior,* and so on. For more details about these, we can refer to the original book.

## 2 Statistics, information theory basics–mlapp chap.5,6

## 3 Linear models(1)–mlapp chap.7, prml chap.3

## 4 Linear models(2)–mlapp chap.8, prml chap.4

## 5 Kernels–mlapp chap.14. prml chap.6-7

## 6 Graphical models–mlapp chap 10, 19. prml chap 8.1-8.3

## 7 Latent variable model, clustering and EM. prml chap9. mlapp chap11, 25

## 8 Continuous latent variables. prml chap 12. mlapp 12-13

## 9 Exact inference–mlapp chap20. prml chap 8.4

## 10 Variational inference–prml chap10. mlapp chap21-22

## 11 Monte Carlo– prml11. mlapp 23,24

## 12 Sequential data–prml chap13. mlapp chap 17-18.

## 13 Gaussian Process–mlapp chap15

## 14 Neural network–mlapp chap 16. prml chap5.

## 15 Deep learning–mlapp chap 28.

## 16 Combining models. prml chap 14