UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

Advanced Machine Learning

Lecture Notes

January 1, 1980

Aleix Boné

Contents

List of Figures

List of Tables

1 Setting of the learning problem

1.1 Learning problem definition

Can be defined as 3 components:

- A generator G of random data x from a probability distribution $P(x)$which is fixed and unknown to us.
- A supervisor S which returns an output value $y \in Y$ for each everyinput $x \in X$ according to a conditional probability distribution $P(y|x)$, also fixed and unknown to us.
- A learning machine (algorithm) L which is capable of implementing(computing) any of a set of functions (models). f_θ where θ is a set of parameters belonging to the parameter space θ.

The selection of the best possible model according to L is done based on a learning data sample $D = (x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ (where x_1 are vectors)

1.2 Why is the relation between X and Y stochastic?

In most of the cases, randomness is ignorance of the true effects.

1. Ignorance of the functional dependence on a not measured variable z.
2. Technical invisibility

1.3 Loss function

loss fn is a function $L : Y \times Y \longrightarrow \mathbb{R}+$ such that:

1. $L(y, y') = L(y', y)$ (symmetry)
2. $L(y, y') = 0 \Longleftarrow y = y'$

Examples of loss functions:

- $L(y, y') := 1$ if $y \neq y'$ (0-1 loss) (zero-one) L_{01}
- $L(y, y') := (y - y')^2$ (squared error)

1.4 Hypothesis space

$$\mathcal{F} = \{f_\theta : \mathcal{X} \to \mathcal{Y} \mid \theta \in \Theta \subseteq \mathbb{R}\}$$

1.5 classifier

Is a function $f : X \longrightarrow Y$

Y is a set of different symbols.

1.6 Risk

The risk of a function is its **expected loss**. Also known as true error or generalization error.

$$\mathbb{R}[f] := \mathbb{E}_{(x,y)\ p}[L(f(x), y)]$$

A function that takes a function as an argument is called a functional. Risk is a functional.

1.7 Empirical error

The average error or loss function evaluated on a specific data sample D.

$$\hat{R}_{D^n}[f] = \frac{1}{n}\sum_{i=1}^{n} L(f(x_i), y_i)$$

This is also a functional.

1.8 Training error

The training error is the **empirical error** obtained in the data sample used fortraining.

1.9 Model error proposition

Proposition: for a fixed model f the expected value of the empirical error based on a data sampleis equal to the true error.

$$\mathbb{E}_{D^{iid}p^n}[\hat{R}_{D^n}[f]] = R[f]$$

Proof. • TODO

□

1.10 discriminative vs generative classifiers

1.10.1 Generative

Example: Gaussian Naive Bayes

1.10.2 Discriminative

Example: Logistic regression

1.11 Risk minimizers

1.11.1 Proposition

The function minimizing the risk on the zero-one loss is the Bayes classifier.Which is the one that given x, assigns the class with the highest posterior probability.

$f^* = argmax_{P(w|x)}$

And the risk of f^*_{01} is the Bayes risk.

The upper bound of the Bayes risk is 0.5

1.12 Regression under the square error / loss function

1.12.1 Standard setting of the regression problem

Proposition We cannot compute this in practice since we do not know the distribution of the data.

Proposition The bias / variance decomposition of the mean squared error.

$$MSE[f] = Bias(f)^2 + Var(f)$$

There is a trade-off between bias and variance, known as the bias-variance dilemma.

Remainder Given a dataset $D = x^i, y^i$ of size $n(x^i, y^i)$ p We choose a loss function L and a hypothesis space F.

$$F := x \to f_\theta(x), \theta \in \Theta$$

Observations The set of functions we chose matters a lot. In some sense F should be large to minimize the chancesthat the choosen function is not in F.

- Chosing a Machine Learning Algorithm of F given a particular $D_n$obviousley depends on this D_n.D_n! $\rightsquigarrow f_\theta = f_n$
- The best possible solution $f^*_{sq.}$ may not belong to F.

In case $f^*_{sq.} \notin F$ then we should find:$\hat{f}_{sq.} = argmin_{||f - f^*_{sq.}||}$We could use different norm functions.

The expression being integrated is:

$$\left(f_m(x) - f^*_{sq.}(x)\right)^2$$

Expected value of the model for all possible datasets in D_n:

$Bias^2(f_n(x))$: how the average prediction over all possible D_n at point $x$differs from the best possible predicition.

$$Bias^2(f) = \int_{\mathbb{R}^d} \left(\overline{f_n}(x) - f^*_{sq.}(x)\right)^2 p(x)dx Var(f) = \int_{\mathbb{R}^d} \mathbb{E}_{D_n}\left[\left(f_n(x) - \overline{f_n}(x)\right)^2\right] p(x)dx$$

$$MSE[f] = Bias(f)^2 + Var(f) + \sigma^2$$

where σ^2 is the variance of the noise.

Since σ cannot be reduced, it is called irreducible error.The rest is called reducible error.

Getting better data we reduce the irreducible error. (ignorance)

2 Consistency

We depart from a specific data sample D_nThe training error of a model f is $\hat{R}_n^{[f]} := \frac{1}{n}\sum_{i=1}^{n} L(f(x^i), y^i)$

So when we learn a specific model f_n from D_n its **training error** is $\hat{R}_n(f_n)$The true error of f_n is $R(f_n)$

The Empirical Risk Minimization (ERM) prescribes to minimize the training error.

Consistency The ERM is said to be consistent if:

- $\hat{R}_n(f_n) \rightarrow infimum_{f \in F} R(f)$ as $n \rightarrow \infty$
- $R(f_n) =\rightarrow infimum_{f \in F} R(f)$ as $n \rightarrow \infty$

These convergences are in probability.

Graphically:

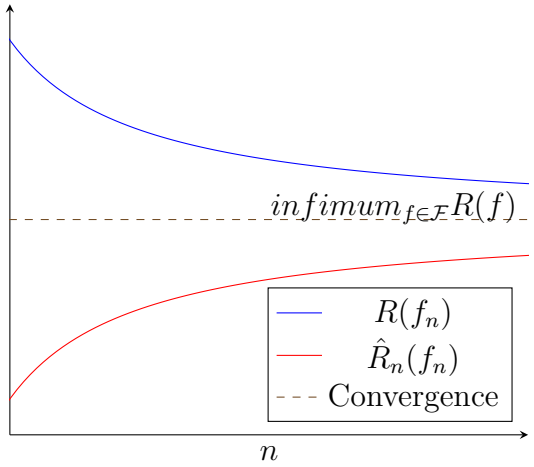

Figure 1: Convergence of the training error and the true error

Theorem (Vapnik-Chervonenkis) P1 A necessary and sufficient condition for consistency is that the hypothesis space F is compact.

$$\forall \varepsilon >$$

3 A glimpse into Bayesian methods (for supervised ML)

We have a data sample $D = \{x^i, y^i\} i = 1, \ldots, n$ where $x \in \mathcal{X} = \mathbb{R}^d, y^i \in \mathcal{Y}$ We consider a hypothesis space $\mathcal{F} := \{f_\theta(x), \theta \in \Theta \subset \mathbb{R}^d\}$.

The Bayesian idea is not to consider the optimal solution $\hat{\theta}$ as a point solution, but as a set of possible solutions, some of which are more likely (to be correct) than others.

The available data D serves to reduce the uncertainty of the distribution.

The Bayesian machinery makes use of the Bayes formula.

$$P(D|\theta) \qquad \text{(likelihood)}$$
$$P(\theta|D) \qquad \text{(posterior)}$$
$$P(\theta) \qquad \text{(prior)}$$
$$\int_\Theta P(D|\theta')P(\theta')d\theta' = P(D) \qquad \text{(EXPECTED } \equiv \text{EVIDENCE)}$$

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{\int_\Theta P(D|\theta')P(\theta')d\theta'}$$

3.1 Procedure

Modelling is choosing

1. Collect the data D
   (a) Decide the functional form of the models $f_\theta(x)$
   (b) Choose the prior $P(\theta)$ about the parameters
2. Calculate the likelihood function $P(y|x, \theta)$ (The probability of getting the data D for a specific value of θ)
3. Calculate the posterior distribution of θ

$$P(\theta|D) = \frac{P(\theta)\prod_{i=1}^n}{P(y^i|x^i, \theta)\int_\Theta d\theta' P(\theta')\prod_{i=1}^n P(y^i|x^i, \theta')}$$

1. Decide a method for making predictions (when new data x^0 comes):

(a) Eliminate the prior $P(\theta)$ -> $P(\theta|D) \propto \mathcal{L}(\sigma; D) \cdot P(\theta)$

i. Then, the only thing that matters is the likelihood

ii. You then find a set of parameters θ that maximizes the likelihood function $\hat{\theta_{ML}} argmax_{\theta \in \Theta} \mathcal{L}(\sigma; D)$

(b) Half Bayesian $P(\theta|D) \propto \mathcal{L}(\sigma; D) \cdot P(\theta)$

i. $\hat{\theta_{MAP}} argmax_{\theta \in \Theta} \mathcal{L}(\sigma; D) \cdot P(\theta)$ (Maximum a posteriori)

ii. Since we have a multiplication to minimize, we use logarithms.

iii. It is then: $\hat{\theta_{MAP}} argmax_{\theta \in \Theta} \log \mathcal{L}(\sigma; D) + \log P(\theta)$

(c) Calculate the average of the posteriors $P(\theta|D)$ instead of the maximum. (average -> expected value)

i. $\hat{\theta_{avg}} = \int_\Theta \theta P(\theta|D) d\theta$

ii. In practice, this integral is intractable, we have to use numerical integration. Which is quite delicate.

(d) Fully Bayesian: you have to work with the full posterior

i. You don't return any model. You return a distribution over models.

ii. Since we have a distribution, we can compute confidence integrals (bayesian confidence intervals) (not in the scope of the course)

iii. Suppose that a new data point comes, x^0:

$$P(y^0|x^0, D) = \int_\Theta P(y^0|x^0, \theta) P(\theta|D) d\theta$$

In the first 2 cases, we obtain a single unique model.In all cases, the data is constant.

3.1.1 Fully worked example

We depart from a data sample and face a regression task.

So $x^i \in \mathbb{R}^d, y^i \in \mathbb{R}$ and we want to perform linear regression.

By convention $\underline{\omega} := (\omega, \omega_0)$

- What loss function do we use?
- What king of regularizer (penalty on model complexity) should I use?

Procedure

1. For an arbitrary input $x \in \mathbb{R}^d$, we want to predict $y \in \mathbb{R}$

$$
\begin{aligned}
\epsilon &\sim \mathcal{N}(0, \sigma^2) \\
P(y|x, \underline{\omega}) &= \mathcal{N}(y|f_{\underline{\omega}}(x), \sigma^2) \\
\beta &= \frac{1}{\sigma^2} \text{ (precision)} \\
P(\underline{\omega}) &= \mathcal{N}(\underline{\omega}; 0, \Sigma) \\
\Sigma &:= \text{ covariance matrix (Not viable in practice, we use other)} \\
P(\underline{\omega}|\alpha) &= \mathcal{N}(\underline{\omega}; 0, \sigma_\omega^2 I)) \\
\alpha &:= \frac{1}{\sigma_\omega^2}
\end{aligned}
$$

A parameter that controls other parameters is called a hyperparameter. In this case, σ_ω is an hyperparameter.

1. The likelihood function

$$\mathcal{L}(\underline{\omega}; \beta) = \prod_{i=1}^{n} P(y^i|x^i, \underline{\omega}) = \prod_{i=1}^{n} \mathcal{N}(y^i|f_{\underline{\omega}}(x^i), \sigma^2)$$

1. The posterior distribution

$$P(\underline{\omega}|y, X, \alpha, \beta) = \frac{\mathcal{L}(\underline{\omega}; \beta) P(\underline{\omega}|\alpha)}{\int_{\mathbb{R}} \mathcal{L}(\underline{\omega}; \beta) P(\underline{\omega}|\alpha) d\underline{\omega}}$$

Recap

Model

$$P(y|X, \underline{\omega}, \beta) = \mathcal{N}(y|f_{\underline{\omega}}(x), \beta) = \sqrt{\frac{\beta}{2\pi}} \exp\left(-\frac{\beta}{2}(y - f_{\underline{\omega}}(x))^2\right)$$

Prior

$$P(\underline{\omega}|\alpha) = \mathcal{N}(\underline{\omega}; 0, \sigma_\omega^2 I)) = \sqrt{\frac{\alpha}{2\pi}} \exp\left(-\frac{\alpha}{2}\underline{\omega}^T \underline{\omega}\right)$$

Result The posterior distribution turns out to be:

$$P(\underline{\omega}|D) = \mathcal{N}(\underline{\omega}; \mu, \Sigma)\mu = \beta\Sigma X^T y\Sigma = \left(\alpha I + \beta X^T X\right)^{-1}$$

1. Maximum likelihood: $\hat{\underline{\omega}} = (X^T X)^{-1} X^T y$
2. Maximum posteriori: $\hat{\underline{\omega}} =$??? (Standard ridge regression)
3. Average posteriori: $\hat{\underline{\omega}} =$???
4. Full Bayesian: $\hat{\underline{\omega}} =$???

References

[1] Aggarwal, Charu C. Neural networks and deep learning: a textbook. Cham, Switzerland: Springer, 2018. xxiii+497. ISBN: 978-3-319-94462-3.

[2] Bishop, Christopher M. Pattern recognition and machine learning. Information science and statistics. New York: Springer, 2006. xx+738. ISBN: 978-0-387-31073-2.

[3] Chollet, François. Deep learning with Python. Shelter Island, New York: Manning Publications Co., 2018. xxi+361. ISBN: 978-1-61729-443-3.

[4] Goodfellow, Ian. Deep learning. In collab. with Bengio, Yoshua and Courville, Aaron. Adaptive computation and machine learning. Cambridge, Massachusetts: The MIT Press, 2016. xxii+775. ISBN: 978-0-262-03561-3.

[5] Shawe-Taylor, John. Kernel methods for pattern analysis. In collab. with Cristianini, Nello. Cambridge: University Press, 2004. xiv+462. ISBN: 978-0-521-81397-6.