Deep Learning[1] Notes

Yan JIN

22 novembre 2016

1 3.8 Expectation, Variance and Covariance

$$Expectation : \mathbb{E}_{X\sim P}[f(x)] = \int p(x)f(x)dx$$
$$Variance : Var(f(x)) = \mathbb{E}[(f(x) - \mathbb{E}[f(x)])^2]$$
$$Covariance : Cov(f(x), g(x)) = \mathbb{E}[(f(x) - \mathbb{E}[f(x)])(g(x) - \mathbb{E}[g(x)])]$$

2 5.4 Estimators, Bias and Variance

2.1 5.4.1 Point Estimation

2.2 5.4.2 Bias

$$bias(\hat{\boldsymbol{\theta}}_m) = \mathbb{E}(\hat{\boldsymbol{\theta}}_m) - \boldsymbol{\theta}$$

Proof of formula (5.39) :

$$\hat{\mu}_m = \mathbb{E}[x^{(i)}] = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$$
$$\mu = \mathbb{E}[\hat{\mu}_m]$$
$$Var(x^{(i)}) = \mathbb{E}[(x^{(i)} - \mu)^2] = \sigma^2$$
$$Var(\hat{\mu}_m) = \mathbb{E}[(\hat{\mu}_m - \mu)^2] = \frac{\sigma^2}{m}$$

so :

$$\begin{aligned}
\mathbb{E}[\hat{\sigma}_m^2] &= \mathbb{E}[\frac{1}{m}\sum_{i=1}^{m}(x^{(i)}-\hat{\mu}_m)^2] \\
&= \frac{1}{m}\mathbb{E}[\sum_{i=1}^{m}(x^{(i)}-\mu+\mu-\hat{\mu}_m)^2] \\
&= \frac{1}{m}\mathbb{E}[\sum_{i=1}^{m}(x^{(i)}-\mu)^2+2\sum_{i=1}^{m}(x^{(i)}-\mu)(\mu-\hat{\mu}_m)+\sum_{i=1}^{m}(\mu-\hat{\mu}_m)^2] \\
&= \frac{1}{m}\mathbb{E}[\sum_{i=1}^{m}(x^{(i)}-\mu)^2+2m(\hat{\mu}_m-\mu)(\mu-\hat{\mu}_m)+m(\mu-\hat{\mu}_m)^2] \\
&= \frac{1}{m}\mathbb{E}[\sum_{i=1}^{m}(x^{(i)}-\mu)^2-m(\hat{\mu}_m-\mu)^2] \\
&= \frac{1}{m}(\sum_{i=1}^{m}\mathbb{E}[(x^{(i)}-\mu)^2]-m\mathbb{E}[(\hat{\mu}_m-\mu)^2]) \\
&= \frac{1}{m}(mVar(x^{(i)})-mVar(\hat{\mu}_m)) \\
&= Var(x^{(i)})-Var(\hat{\mu}_m) \\
&= \sigma^2-\frac{\sigma^2}{m}=\frac{m-1}{m}\sigma^2
\end{aligned}$$

2.3 5.4.2 Variance and Standard Error

Variance :$Var(\hat{\theta})$

Standard Error : $SE(\hat{\theta})=\sqrt{Var(\hat{\theta})}$

2.4 5.4.4 Trading off Bias and Variance to Minimize Mean Square Error

Proof (5.54) :

$$\begin{aligned}
MSE &= \mathbb{E}[(\hat{\theta}_m-\theta)^2] \\
&= \mathbb{E}[\hat{\theta}_m^2]-2\theta\mathbb{E}(\hat{\theta}_m)+\theta^2 \\
Bias(\hat{\theta}_m)^2 &= (\mathbb{E}[\hat{\theta}_m]-\theta)^2 \\
&= \mathbb{E}[\hat{\theta}_m]^2-2\mathbb{E}[\hat{\theta}_m]\theta+\theta^2 \\
Var(\hat{\theta}_m) &= \mathbb{E}[(\hat{\theta}_m-\mathbb{E}[\hat{\theta}_m])^2] \\
&= \mathbb{E}[\hat{\theta}_m^2-2\hat{\theta}_m\mathbb{E}[\hat{\theta}_m]+\mathbb{E}[\hat{\theta}_m]^2] \\
&= \mathbb{E}[\hat{\theta}_m^2]-\mathbb{E}[\hat{\theta}_m]^2 \\
\Rightarrow MSE &= Bias(\hat{\theta}_m)^2+Var(\hat{\theta}_m)
\end{aligned}$$

3 Frequentist Statistics and Baysian Statistics

Frequentist : Estimate a single value of $\boldsymbol{\theta}$, then making all predictions thereafter based on that **one** estimate ;

Baysian : Consider **all** possible values of $\boldsymbol{\theta}$ when making a prediction ;

Frequentist : The true parameter value $\boldsymbol{\theta}$ is **fixed but unknown**, while $\hat{\boldsymbol{\theta}}$ is a random variable and a function of **the dataset**(which is seen as **random**) ;

Baysian : **Dataset** is directly observed and is **not random** ; the true parameter value $\boldsymbol{\theta}$ is **unknown or uncertain** and thus is represented as a random variable ;

Differences between MLE(Maximum Likelihood Estimation) and Bayesian estimation :

1. MLE : Make predictions using a **point estimate** of $\boldsymbol{\theta}$;
   Bayesian : Using a **full distribution** over $\boldsymbol{\theta}$;
2. MLE : Address the uncertainty on a given point estimate of $\boldsymbol{\theta}$ by evaluating its **variance** ;
   Bayesian : Simply **integrate over it** ;
3. Baysian : Use a priori, which expresses a preference for **simpler and smooth models**, and seems as a source of **subjective human judgment** impacting the predictions ;
4. Baysian : **Generalize much better** when training data is **small**, but **high computation cost** when training data is **large** ;

3.1 Frequentist Statistics - Maximum Likelihood Estimation (MLE)

For data samples $x^{(1)}, ..., x^{(m)}$ drawn independently from **the true but unknown** data generating distribution $p_{data}(\boldsymbol{x})$

$p_{model}(\boldsymbol{x;\theta})$ is a parametric family of probability distribution over the space indexed by $\boldsymbol{\theta}$ for estimating the $p_{data}(\boldsymbol{x})$.

$$\boldsymbol{\theta_{ML}} = \underset{\boldsymbol{\theta}}{argmax}\ p_{model}(\mathbb{X};\boldsymbol{\theta}) =$$

3.2 Baysian Statistics

Prior probability distribution(the prior) : $p(\boldsymbol{\theta})$

For data samples $x^{(1)}, ..., x^{(m)}$, we reform the belief about $\boldsymbol{\theta}$(the posterior$p(\boldsymbol{\theta}|x^{(1)}, ..., x^{(m)})$) by the data likelihood $p(x^{(1)}, ..., x^{(m)}|\boldsymbol{\theta})$ and the prior $p(\boldsymbol{\theta})$ via **Bayes' rule** :

$$p(\boldsymbol{\theta}|x^{(1)}, ..., x^{(m)}) = \frac{p(x^{(1)}, ..., x^{(m)}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(x^{(1)}, ..., x^{(m)})}$$

3.3 Maximum A Posteriori (MAP) Estimation

$$\boldsymbol{\theta}_{MAP} = \underset{\boldsymbol{\theta}}{argmax}\ p(\boldsymbol{\theta}|\boldsymbol{x}) = \underset{\boldsymbol{\theta}}{argmax}\ log\ p(\boldsymbol{x}|\boldsymbol{\theta}) + log\ p(\boldsymbol{\theta})$$

MAP has the advantage of leveraging information that is brought by **the prior** and cannot be found in **the training data**. This information helps to **reduce the variance** in the MAP point estimate (compare to ML estimate), but **increase bias**.

$$MLE(log\ p(\boldsymbol{\theta}|\boldsymbol{x}))\ +\ Regularization\ with\ weight\ decay(log\ p(\boldsymbol{\theta})) = MAP\ to\ Bayesian\ inference.$$

4 Chapter 11 Practical Methodology

Practical design process :

1. Determine error metric and target value ;
2. Establish a Baseline Model ;
3. Determine bottlenecks in performance ;
4. Repeatedly make incremental changes : gathering new data, adjusting hyperparameters, or changing algorithms ;

4.1 11.3 Determining Whether to Gather More Data

1. Determine whether the performance on the training set is acceptable ;
   If performance on the training set is poor :
2. Increase the size of the model : add more layers ; add more hidden unites to each layer ; turning the learning rate etc.
   If still not work well : data needed to be cleaned or gathered ;
   Else :
3. Measure performance on test set ;
   If performance good, done !
   Else if test set performance is much worse than training set performance :
4. Gather data ;
   If not easy to gather data :
5. Reduce the size of the model ; Improve regularization(adjust weight decay coefficients or add dropout) ;
   If test set performance is still unacceptable :
6. Gather data ;

Reference

[1] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. Book in preparation for MIT Press, 2016.