Machine Learning @ Data science 2019: Set 4

Jordi Morera Serra

March 17, 2019

Exercise 12

We want to compute f^* that minimizes the cost functional function $A(f) = \mathbb{E}\phi(-f(X)Y)$.

Applying the law of total expectation, we get

$$A(f) = \mathbb{E}_X \mathbb{E}_{X|Y}[\phi(-f(X)Y)|X]$$

For fixed X, as $Y \in \{-1, 1\}$, we can rewrite the cost functional function as

$$A(f) = \mathbb{E}_X[\eta(X)\phi(-f(X)) + \eta(X)\phi(f(X))]$$

Where $\eta(X)$ is $\mathbb{P}(Y = 1|X = x)$.

For a fixed X, we compute the derivative of the expression inside the expectation

$$\frac{d}{df}\big(\eta \cdot \phi(-f) + \eta \cdot \phi(f)\big) = -\eta \cdot \phi'(-f) + \eta\phi'(f)$$

F.O.C

$$-\eta \cdot \phi'(-f) + \eta\phi'(f) = 0 \qquad \Rightarrow \qquad \frac{\phi'(f)}{\phi'(-f)} = \frac{\eta}{1-\eta}$$

Assuming $\frac{\phi'(f)}{\phi'(-f)} = \theta(f)$, we obtain $f^* = \theta^{-1}(\frac{\eta}{1-\eta})$.

Note that if $\phi : \mathbb{R} \to \mathbb{R}_+$, positive, increasing, strictly convex function, then

$$\phi'(f) > \phi'(-f) \qquad iif \qquad \eta(x) > 1 - \eta(x)$$

This will only happen when $f > -f$ which implies that $f > 0$.

Therefore, this is the bayes classifier.

Exercise 13

Let's first compute the Kernel function for $d = 1$

$$\begin{aligned} K(x, y) &= \langle \Phi(x), \Phi(y) \rangle \\ &= \sum_{n=0}^{\infty} \frac{1}{\sqrt{n!}} x^n e^{-x^2/2} \frac{1}{\sqrt{n!}} y^n e^{-y^2/2} \\ &= e^{-x^2/2} e^{-y^2/2} \sum_{n=0}^{\infty} \frac{1}{n!} (xy)^n \end{aligned}$$

Using the hint, we finally get

$$\begin{aligned} K(x, y) &= e^{xy} e^{-x^2/2 - y^2/2} \\ &= e^{-\frac{1}{2}(x-y)^2} \end{aligned}$$

Generalizing for any d, if $X, Y \in \mathbb{R}^d$

$$
\begin{aligned}
K(X,Y) &= \langle \Phi(X), \Phi(Y) \rangle \\
&= \sum_{n=0}^{\infty} \frac{1}{\sqrt{n!}} \frac{1}{\sqrt{n!}} (X^t Y)^n e^{-\frac{1}{2}(X^t X + Y^t Y)} \\
&= e^{-\frac{1}{2}(X-Y)^t (X-Y)}
\end{aligned}
$$

Which is a Gaussian kernel.
Finally, the corresponding feature map is

$$
\phi(X)_n = \frac{1}{\sqrt{n!}} ||X||^n e^{-\frac{||X||^2}{2}}
$$

Exercise 14

$$\mathcal{R}(f_w) = \mathbb{E} \sup_{w \in \mathcal{X}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i f_w(x_i)$$

$$\mathbb{E} \sup_{w \in \mathcal{X}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i K(x_i, w)$$

$$= \mathbb{E} \sup_{w \in \mathcal{X}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i \langle \phi(x_i), w \rangle$$

By the linearity properties of the inner product and as $\frac{\sum \sigma_i \phi(x_i)}{||\sum \sigma_i \phi(x_i)||} = 1$, we can rewrite last step as

$$= \mathbb{E} \sup_{w \in \mathcal{X}} \frac{1}{n} \langle \sum_{i=1}^{n} \sigma_i \phi(x_i), w \rangle$$

$$= \mathbb{E} \sup_{w \in \mathcal{X}} \frac{1}{n} \langle \sum_{i=1}^{n} \sigma_i \phi(x_i), \frac{\sum_{i=1}^{n} \sigma_i \phi(x_i)}{||\sum_{i=1}^{n} \sigma_i \phi(x_i)||} w \rangle$$

We know $||w|| \leq 1$, therefore

$$= \mathbb{E} \frac{1}{n} \langle \sum_{i=1}^{n} \sigma_i \phi(x_i), \frac{\sum_{i=1}^{n} \sigma_i \phi(x_i)}{||\sum_{i=1}^{n} \sigma_i \phi(x_i)||} \rangle$$

$$= \mathbb{E} \frac{1}{n} \frac{1}{||\sum_{i=1}^{n} \sigma_i \phi(x_i)||} \langle \sum_{i=1}^{n} \sigma_i \phi(x_i), \sum_{i=1}^{n} \sigma_i \phi(x_i) \rangle$$

Applying Cauchy–Schwarz and Jensen's inequality

$$\leq \mathbb{E} \frac{1}{n} \frac{||\sum_{i=1}^{n} \sigma_i \phi(x_i)||^2}{||\sum_{i=1}^{n} \sigma_i \phi(x_i)||}$$

$$\leq \sqrt{\mathbb{E} || \frac{1}{n} \sum_{i=1}^{n} \sigma_i \phi(x_i)||^2}$$

$$= \frac{1}{n} \sqrt{\mathbb{E}[|| \sum_{i=1}^{n} \sigma_i \phi(x_i)||^2 + 2 \sum_{i<j} \sigma_i \phi(x_i) \sigma_j \phi(x_j)]}$$

As σ_i,σ_j are i.i.d, thus $\mathbb{E}\sigma_i\sigma_j = 0$. Also $\sigma_i \phi(x_i)$ are i.i.d

$$= \frac{1}{n} \sqrt{\mathbb{E}[|| \sum_{i=1}^{n} \sigma_i \phi(x_i)||^2]}$$

$$= \frac{1}{n} \sqrt{\mathbb{E}[\sum \sigma_i \sigma_i \phi(x_i) \phi(x_i)]}$$

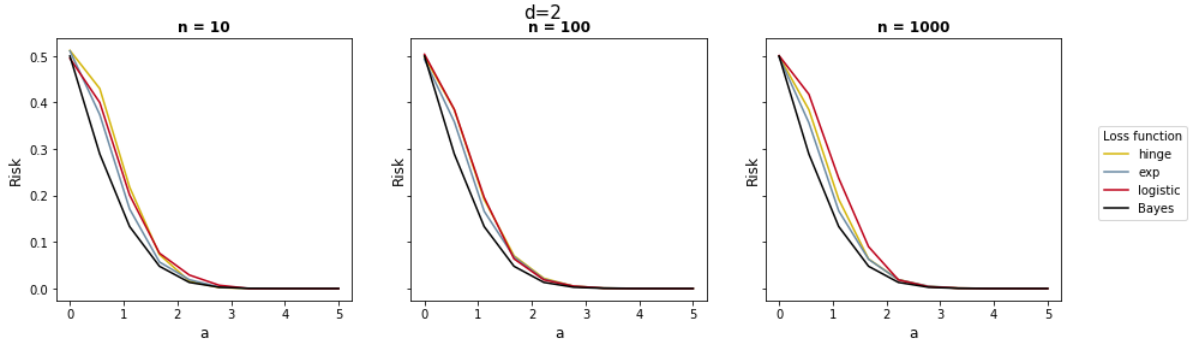
$$= \frac{1}{n} \sqrt{\mathbb{E}[\sum_{i=1}^{n} K(x_i, x_i)]}$$

Considering $K(x_i,x_i) \in [-1, 1]$, and therefore $K(x_i,x_i) \leq 1$,we get finally

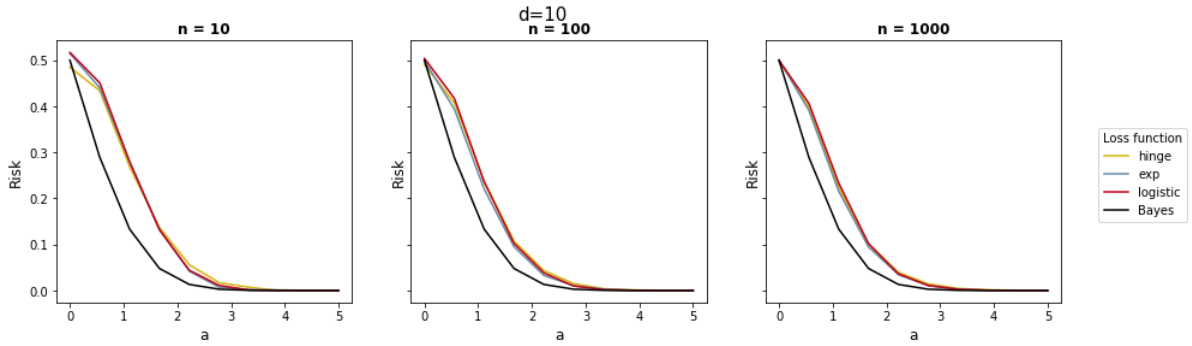
$$\leq 1/\sqrt{n}$$

Exercise 15

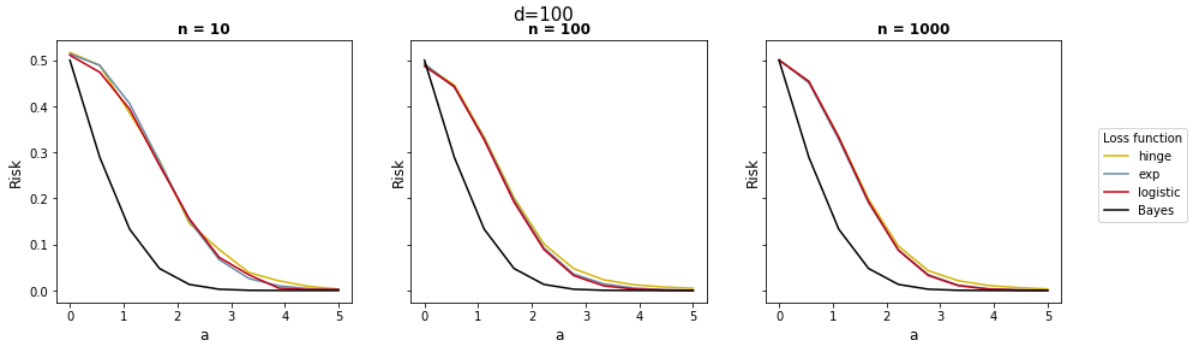
In figure 1 we plotted the estimation of the risk obtained for different scenarios stopping the gradient descent algorithm after 20 weight updates.



(a) Risk comparison for different training set scenarios in $\mathbb{R}^2$



(b) Risk comparison for different training set scenarios in $\mathbb{R}^{10}$



(c) Risk comparison for different training set scenarios in $\mathbb{R}^{100}$

Figure 1

Remarks: Risk

- We clearly see the risk decreasing with the margin. This is what we expected since as the margin increases, the two conditional distributions are further apart and the classifier fitted with the training data is more likely to predict correctly out-training-sample data. Also note that when a is high enough it converges to the bayes risk, which is 0 for linearly separable data.
- When n increases, since a bigger training sample is available, the risk decreases no matter d. On the other hand, for a given n, having higher d adds noise to our classification problem since all dimensions but the first one are not relevant. Therefore, the classification problem is harder and the risk increases.

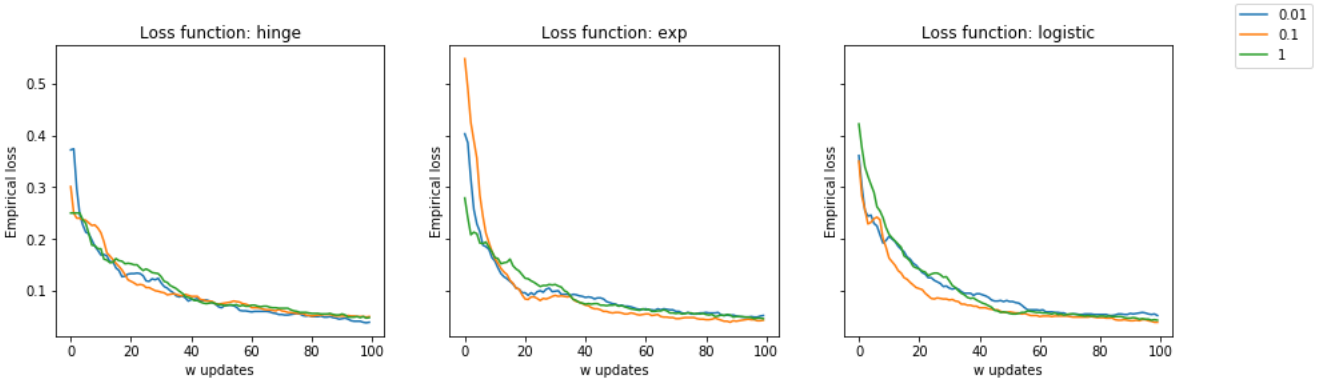


Figure 2: Risk convergence different learning rates for for d = 100, n = 1000 and a = 2

In figure 2 we plotted the evolution of the empirical in-sample risk over the stochastic gradient descent updates. Note that this plot corresponds to a single trial of a specific n, d and a scenario since the main objective is to illustrate the ideas explained below.

Remarks: Surrogate loss functions

- In figure 1 we can see that, out of sample risk wise, the three surrogate loss functions seem to perform quite uniformly in all our scenarios . However, in figure 2 we notice that hinge surrogate loss function seems to converge with higher number of iterations than the other two no matter the learning rate parameter. Although the figure shows only one single trial, is was found the same situation for other multiple trials.
- Regarding the learning rate parameter, it was not found any general trend for our set up. However, one can show that the learning parameter has capital influence in the training process and a bad choice can prevent the algorithm from converging. Generally, an adaptative learning rate reaches better performance.