Theoretical task 3 - solution

due 9:00 February 5 (Friday).

Remark: No late submissions allowed this time. All solutions should be short, precise and clearly written.

1. **Prove that if particular feature x^i has arbitrary continuous distribution with cumulative distribution function $F(u)$, then monotonous transformation with F will yield uniformly distributed feature:**
$$F(x^i) \sim Uniform[0,1]$$

   Define $\xi = F(x^i)$. Cumulative distribution function of ξ is $G(u) = P(\xi \leq u) = P(F(x^i) \leq u) = P(x^i \leq F^{-1}(u)) = F(F^{-1}(u)) = u \quad \forall u \in [0,1]$. As $F : (-\infty, +\infty) \to [0,1]$, so $P(\xi \leq u) = 0 \, \forall u < 0$ and $P(\xi \leq u) = 1 \, \forall u > 1$. So density function is

$$g(u) = G'(u) = \begin{cases} 0, & u < 0 \\ 1, & u \in [0,1] \\ 0, & u > 1 \end{cases}$$

   The random variable, having such density function is Uniform[0,1].

2. **Suppose that you have a random classifier, assigning probabilities**

$$\begin{aligned} p(y = +1|x) &= \xi \\ p(y = -1|x) &= 1 - \xi \end{aligned}$$

   where ξ is a random variable uniformly distributed on $[0,1]$ independent of x. Plot the ROC curve for this classifier and justify your result.

   The ROC curve is a set of points $FRP(\mu)$, $TPR(\mu)$ for classifier

$$\widehat{y}(x) = \begin{cases} +1, & p(y = +1|x) = \xi > \mu \\ -1, & p(y = +1|x) = \xi \leq \mu \end{cases}$$

   So our classifier predicts class randomly with probability

$$\widehat{y}(x) = \begin{cases} +1, & \text{with probability } P(\xi > \mu) = 1 - \mu \\ -1, & \text{with probability } P(\xi \leq \mu) = \mu \end{cases}$$

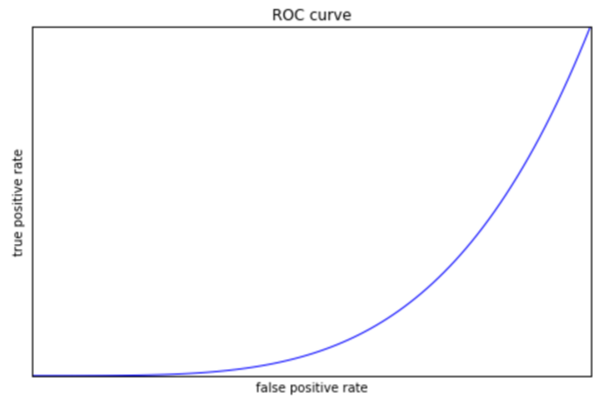
   The key point is that predicted class is independent on x and the true class, so

$$TPR(\mu) = P(\widehat{y} = +1|y = +1, x, \mu) = P(\widehat{y} = +1|\mu) = 1 - \mu$$

$$FPR(\mu) = P(\widehat{y} = +1|y = -1, x, \mu) = P(\widehat{y} = +1|\mu) = 1 - \mu$$

   Since $\mu \in [0,1]$ points ($TPR(\mu)$, $FPR(\mu)$) draw a straight line between (0,0) and (1,1).

3. **Suppose that you have a classifier with the convex ROC curve lying below the line $y = x$ and shown here:**



How can you make this classifier yield you a higher AUC than the random classifier from task (2)? Justify your solution.

$$TPR(\mu) = P(\widehat{y}_\mu = +1 | y = +1, \mu)$$

$$FPR(\mu) = P(\widehat{y}_\mu = +1 | y = -1, \mu)$$

Consider inverted classifier which predicts $+1$ if original prediction was -1 and vice versa. Denote its ROC characteristics as $TPR'(\mu)$, $FPR'(\mu)$. It predicts correctly positive class if original prediction was negative, and vice versa, so

$$TPR'(\mu) = P(\widehat{y}_\mu = -1 | y = +1, \mu) = 1 - P(\widehat{y}_\mu = +1 | y = +1, \mu) = 1 - TPR(\mu)$$

$$FPR'(\mu) = P(\widehat{y}_\mu = -1 | y = -1, \mu) = 1 - P(\widehat{y}_\mu = +1 | y = -1, \mu) = 1 - FPR(\mu)$$

Since $TPR(\mu) < FPR(\mu) \, \forall \mu$ so

$$TPR'(\mu) = 1 - TPR(\mu) > 1 - FPR(\mu) = FPR'(\mu) \, \forall \mu$$

So all points of the ROC curve of the inverted classifier will lie above $TPR(\mu) = FPR(\mu)$, $\forall \mu$ line, which is a ROC curve for random classifier from previous task.

4. **Suppose your training set consists of N samples and you generate bootstrap pseudosample of the same size.**

   (a) **What is the probability, that a particular observation will not appear in the bootstrap pseudosample at all?**
   Consider first sample (x_1, y_1). Since at each position of the bootstrap sample we pick random sample, the probability to take first sample for the first position is $\frac{1}{N}$. The probability not to take first sample at first position is $1 - \frac{1}{N}$. The probability not to select first sample for all N positions of the bootstrap sample is $(1 - \frac{1}{N})^N$

   (b) **What is the limit of this probability as $N \to \infty$?**
   Denote $F(N) = (1 - \frac{1}{N})^N$. $\ln F(N) = N \ln(1 - \frac{1}{N})$. Since from Taylor expansion $\ln(1+x) = 1 + x + \overline{o}(x)$, so $\ln F(N) = N(-\frac{1}{N} + \overline{o}(\frac{1}{N})) = -1 + \overline{o}(1)$. So

$$\lim_{N \to \infty} \ln F(N) = -1$$

Thus, by taking exponent of both parts, we obtain

$$\lim_{N\to\infty} F(N) = e^{-1}$$

5. **Under what selection of $h(x)$ and $K(u)$ will Nadaraya-Watson regression transform to K-nearest neighbours regression?**

   $h(x) = \rho(x, z_K)$, $K(u) = \mathbb{I}[|u| \leq 1]$.

6. **Explain, why the number of SVM misclassifications, obtained from leave-one-out validation is no greater than the number of support vectors?**

   See page 15 of the lecture on SVM. For support inequality constraints are satisfied as equalities: $y_i(w^T x_i + w_0) = 1 - \xi_i$ while for other (uninformative) objects inequality constraints are satisfied as strict inequalities:

$$y_i(w^T x_i + w_0) > 1$$

   So uninformative objects are correctly classified and the solution depends only on support vectors (property of SVM). That's why LOO predictions will be correct for all non-informative objects and the errors can appear only for support vectors.

7. **Prove that polynomial kernel $K(x, x') = (\alpha\langle x, x'\rangle + \beta)^M$ and Gaussian kernel $K(x, x') = e^{-\gamma\langle x-x', x-x'\rangle}$ ($\alpha > 0, \beta > 0, \gamma > \mathbf{0}$, $M = 1, 2, 3, ...$) are valid Mercer kernels.**

   Using that $\langle x, z\rangle$ is a valid Mercer kernel by definition and looking at transformations from lecture "Kernel trick", page 7, which generate new valid Mercer kernels out of existing Mercer kernels, we obtain that:

   (a) $\langle x, z\rangle$-kernel => $\alpha\langle x, z\rangle$-kernel => {since β is also a kernel} $\alpha\langle x, x'\rangle + \beta$ => kernel => $(\alpha\langle x, x'\rangle + \beta)^M$ - kernel, given constraints on α, β, M.

   (b) $\langle x, z\rangle$-kernel=>$2\gamma\langle x, z\rangle$-kernel=>$e^{2\gamma\langle x,z\rangle}$-kernel=>$e^{2\gamma\langle x,z\rangle}$-kernel=>$\varphi(x)e^{2\gamma\langle x,z\rangle}\varphi(z)$-kernel for any $\varphi(u)$, in particular for $\varphi(u) = e^{-\gamma\langle u,u\rangle}$, so $e^{-\gamma\langle x,x\rangle}e^{2\gamma\langle x,z\rangle}e^{-\gamma\langle z,z\rangle} = e^{-\gamma\langle x-z,x-z\rangle} = e^{-\gamma\|x-z\|^2}$ is a kernel.

8. **Draw a neural network (structure, weights, thresolds), implementing a XOR function for binary inputs, shown below:**

x^1	x^2	x^1 XOR x^2
0	0	0
0	1	1
1	0	1
1	1	0

   The network is supposed to use only $\mathbb{I}[u \geq threshold]$ activation functions.

   Many solutions are possible. One of them is a two layered network. On the first layer $z^1 = x^1 AND\, x^2 = \mathbb{I}[x^1 + x^2 \geq 2]$ and $z^2 = x^1 OR\, x^2 = \mathbb{I}[x^1 + x^2 \geq 1]$ are calculated. On the second layer $NOT(z^1)\, AND\, z^2 = \mathbb{I}[z^2 - z^1 \geq 1]$ gives exactly $x^1\, XOR\, x^2$.