CUB Spring 2024. Machine Learning.
Mid-term exam. Test variant.
REFERENCE SOLUTION

Task 1. Credit scoring classification problem. Input object – a bank client. Target variable – binary label, whether to give money to the client or not. Possible features are client gender, income, job type, credit goal, ownership of real estate, ownership of auto, etc.

Task 2. Example of nominal feature: Country name with values Germany, France, Italy, etc. Example of set-valued feature: a list of tags for video description from the list of all possible tags.

Task 3. Let's first consider the function $G(Z) = \frac{1}{2}\|Z\|_F^2$ and compute its differential:

$$dG(Z) = d\frac{1}{2}\|Z\|_F^2 = \frac{1}{2}d\mathrm{tr}(Z^TZ) = \frac{1}{2}\mathrm{tr}(dZ^TZ + Z^TdZ) = \frac{1}{2}\underbrace{\mathrm{tr}(dZ^TZ)}_{=\mathrm{tr}((dZ^TZ)^T)} + \frac{1}{2}\mathrm{tr}(Z^TdZ) = \mathrm{tr}(Z^TdZ)$$

Then

$$dF(X) = d[G(X - X_0) + \boldsymbol{\lambda}^T(X\boldsymbol{a} - \boldsymbol{b})] = \mathrm{tr}(X - X_0)^Td(X - X_0) + \underbrace{\boldsymbol{\lambda}^TdX\boldsymbol{a}}_{\in\mathbb{R}} = \mathrm{tr}(X - X_0)^TdX + \mathrm{tr}(\boldsymbol{\lambda}^TdX\boldsymbol{a}) =$$
$$= \mathrm{tr}(X - X_0)^TdX + \mathrm{tr}(\boldsymbol{a}\boldsymbol{\lambda}^TdX) = \mathrm{tr}(\nabla_X F(X)^TdX).$$

$$\nabla_X F(X) = X - X_0 + \boldsymbol{\lambda}\boldsymbol{a}^T = 0 \;\Rightarrow\; X_{min} = X_0 - \boldsymbol{\lambda}\boldsymbol{a}^T.$$

Let's verify answer's consistency by checking dimension. X should be a matrix of size $n \times m$. $X_0 \in \mathbb{R}^{n\times m}$, $\boldsymbol{\lambda} \in \mathbb{R}^n$, $\boldsymbol{a} \in \mathbb{R}^m$. Hence $X_0 - \boldsymbol{\lambda}\boldsymbol{a}^T \in \mathbb{R}^{n\times m}$.

Task 4. The values m, s, min and max depends on all feature values and thus are sensitive to presence of outliers. The values M and S are determined only by the middle of sorted feature values and thus are not sensitive to adding big values to features of some objects. So the option (f) is the only possible answer.

Task 5. Gradient descent optimizer is sensitive to poor scaling of optimization function. If function changing rate is different w.r.t. different directions in multidimensional space then this may significantly slow down convergence speed of gradient descent. Data normalization helps in proper scaling of loss function and thus helps in getting faster solution from optimization. So the right answer is (a).

Task 6. The value of regression function for two identical features $w_1x + w_2x$ would be the same if we fix value of $w_1 + w_2$. L_1-regularization doesn't distinguish between situations $w_1 = 1, w_2 = 0$ and $w_1 = w_2 = 1/2$, because in both cases $|w_1| + |w_2| = 1$. For L_2-regularization:

$$w_1 = 1, w_2 = 0 \;\Rightarrow w_1^2 + w_2^2 = 1,$$
$$w_1 = w_2 = 1/2 \;\Rightarrow w_1^2 + w_2^2 = 1/2.$$

Actually if we consider the optimization problem

$$w_1^2 + w_2^2 \to \min_{w_1,w_2},$$
$$w_1 + w_2 = \mathrm{const},$$

then the solution will be $w_1 = w_2$.

Following similar reasoning for Elastic-Net regularization:

$$w_1 = 1, w_2 = 0, \;\Rightarrow\; \lambda_1(w_1^2 + w_2^2) + \lambda_2(|w_1| + |w_2|) = \lambda_1 + \lambda_2/2,$$
$$w_1 = w_2 = 1/2, \;\Rightarrow\; \lambda_1(w_1^2 + w_2^2) + \lambda_2(|w_1| + |w_2|) = \lambda_1/2 + \lambda_2/2.$$

So the answer is (b) and (c).

Task 7. Here we need to solve the following optimization problem:

$$F(c) = \sum_{i=1}^{N} (y_i - c)^2 \to \min_{c \in \mathbb{R}}$$

Let's take a derivative and equate it to zero:

$$F'(c) = \sum_{i=1}^{N} 2(c - y_i) = 2cN - 2\sum_{i=1}^{N} y_i = 0 \;\Rightarrow\; c_{opt} = \frac{1}{N}\sum_{i=1}^{N} y_i.$$

Task 8. For stochastic gradient descent it is possible to prove the theorem that function residual after k iterations is upper bounded by

$$\frac{\|\boldsymbol{x}_0 - \boldsymbol{x}_{opt}\|_2^2 + G^2 \sum_{i=0}^{k} \alpha_k^2}{2\sum_{i=0}^{k} \alpha_i}.$$

In case of using constant learning rate $\alpha_k = h$ this bound transforms to

$$\frac{\|\boldsymbol{x}_0 - \boldsymbol{x}_{opt}\|_2^2 + G^2 h^2 (k+1)}{2(k+1)h} = \frac{\|\boldsymbol{x}_0 - \boldsymbol{x}_{opt}\|_2^2}{2(k+1)h} + \frac{G^2 h}{2}.$$

It doesn't converge to zero with $k \to +\infty$. So the right answer is (d).

Task 9. Here we have $TP = 4, TN = 2, FP = 1, FN = 3$. Hence,

$$\begin{aligned}
\text{accuracy} &= \frac{TP + TN}{TP + TN + FP + FN} = \frac{4+2}{4+2+1+3} = \frac{6}{10} = \frac{3}{5}, \\
\text{precision} &= \frac{TP}{TP + FP} = \frac{4}{4+1} = \frac{4}{5}, \\
\text{recall} &= \frac{TP}{TP + FN} = \frac{4}{4+3} = \frac{4}{7}, \\
F_1 &= \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2(4/5)(4/7)}{4/5 + 4/7} = \frac{2 \cdot 4 \cdot 4}{4 \cdot 7 + 4 \cdot 5} = \frac{32}{28 + 20} = \frac{32}{48} = \frac{2}{3}.
\end{aligned}$$

Task 10. We don't want to miss any cases of presence of dangerous disease. With high precision the algorithm will be too conservative and output positive class only for absolutely certain cases. With high recall the test would try to cover the most number of disease cases without missing anything. Even in case of false alarm an additional more precise test would allow to detect the disease. F_1-measure combines precision and recall and thus may reduce recall (increase missing rate) for the price of higher precision. So the best option here is (c).

Task 11. Let's sort all objects from the dataset w.r.t. their classifier score:

$$[0.9, 0.9, 0.7, 0.7, 0.7, 0.6, 0.5, 0.1, 0].$$

The corresponding class labels will be:

$$[-1, 1, -1, 1, 1, -1, -1, 1, -1].$$

Hence in total there are 7 possible thresholds for score values (all objects belong to positive class, all objects with score ≥ 0.1 belong to positive class, etc.). Then the points of ROC-curve can be computed as follows:

Threshold	TRP	FPR
0	1	1
0.1	1	4/5
0.5	3/4	4/5
0.6	3/4	3/5
0.7	3/4	2/5
0.9	1/4	1/5
1	0	0

Task 12. In unconstrained SVM view the coefficient $1/C$ serves as L_2-regularization coefficient. So increasing C reduces the regularization and thus allows the model to better adapt to the training set.

Polynomial kernel function with degree d corresponds to considering feature space with polynomial features $x_{i_1}x_{i_2}\dots x_{i_k}$ for all $k \le d$. So increasing d corresponds to considering richer feature space and thus better fit to the training set.

RBF kernel function $\exp(-\|\boldsymbol{x}-\boldsymbol{x}_i\|^2/(2\sigma^2))$ with fixed $\boldsymbol{x}_i$ and with smaller σ value corresponds to quicker varying function as a function of $\boldsymbol{x}$. So for smaller σ the prediction of the form $\sum_i \lambda_i \exp(-\|\boldsymbol{x}-\boldsymbol{x}_i\|^2/(2\sigma^2))$ may better fit the training set.

The right answers are (a), (c) and (f).

Task 13. The decision rule in SVM is determined only by support vectors. If the object is not support vector than excluding of this object from the training set doesn't change SVM decision rule, and also non-support object is always correctly classified. So in leave-one-out procedure an error may occur only in case of excluding support vector, and thus the total number of errors doesn't exceed M.

Task 14. First let's write down Lagrange function:

$$L(\boldsymbol{x},\lambda) = \boldsymbol{c}^T\boldsymbol{x} + \lambda(\|\boldsymbol{x}\|_2^2 - b)$$

Dual function by definition is:

$$D(\lambda) = \min_{\boldsymbol{x}} L(\boldsymbol{x},\lambda).$$

So let's differentiate Lagrange function w.r.t. $\boldsymbol{x}$:

$$dL(\boldsymbol{x},\lambda) = \boldsymbol{c}^T d\boldsymbol{x} + \lambda d(\boldsymbol{x}^T\boldsymbol{x} - b) = \boldsymbol{c}^T d\boldsymbol{x} + 2\lambda\boldsymbol{x}^T d\boldsymbol{x} = \nabla_{\boldsymbol{x}} L(\boldsymbol{x},\lambda)^T d\boldsymbol{x}.$$

Hence,

$$\nabla_{\boldsymbol{x}} L(\boldsymbol{x},\lambda) = \boldsymbol{c} + 2\lambda\boldsymbol{x} = 0 \;\Rightarrow\; \boldsymbol{x}_{min} = -\frac{1}{2\lambda}\boldsymbol{c}.$$

Substituting this result back to the Lagrange function we find:

$$D(\lambda) = L(\boldsymbol{x}_{min},\lambda) = -\frac{\|\boldsymbol{c}\|^2}{2\lambda} + \lambda\frac{\|\boldsymbol{c}\|^2}{4\lambda^2} - \lambda b = -\frac{\|\boldsymbol{c}\|^2}{4\lambda} - \lambda b.$$

Hence, dual optimization problem would be the following:

$$\begin{aligned} &-\frac{\|\boldsymbol{c}\|^2}{4\lambda} - \lambda b \to \max_{\lambda},\\ &\lambda \ge 0. \end{aligned}$$

Task 15. SVM decision rule is the following:

$$\operatorname{sign}\left(\sum_{j=1}^{M} \lambda_j y_j K(\boldsymbol{x},\boldsymbol{x}_j)\right),$$

where $\boldsymbol{x}_j$ are support vectors, λ_j are solution of dual SVM problem. In case of RBF kernel we need to compute Euclidean distances between test object $\boldsymbol{x}$ and all support vectors $\boldsymbol{x}_1,\dots,\boldsymbol{x}_M$. This costs $O(DM)$. All other operations in the decision rule are cheaper. So the final computational complexity is $O(DM)$.

Task 16. Here we need to find derivative w.r.t. $\boldsymbol{w}$ for the function

$$\max(0, 1 - y\boldsymbol{w}^T\boldsymbol{x}).$$

If $1 - y\boldsymbol{w}^T\boldsymbol{x} \le 0$ then the function is constant 0 and its derivative is also 0. Otherwise the derivative is $-y\boldsymbol{x}$. Hence one iteration of SGD will be

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \alpha[y\boldsymbol{w}_k^T\boldsymbol{x} < 1]y\boldsymbol{x},$$

where $(\boldsymbol{x}, y)$ is randomly sampled object on the current iteration k.

Task 17. Suppose that the algorithm outputs some constant a for all input objects. Then the mean error of such algorithm would be

$$p(y=+1)L(+1,a) + p(y=0)L(0,a) = p(1-a)^2 0.9 + (1-p)a^2 0.1 \to \min_{a\in[0,1]}$$

Let's differentiate this function w.r.t. a and equate the derivative to zero:

$$\frac{d}{da} = -2p(1-a)0.9 + 2(1-p)a0.1 = -2p0.9 + 2pa0.9 + 2(1-p)a0.1 = 0 \;\Rightarrow\; a_{opt} = \frac{2p0.9}{2p0.9 + 2(1-p)0.1} = \\ = \frac{0.9p}{0.8p + 0.1} = \frac{9p}{8p+1} = \frac{9}{8 + 1/p}.$$

Let's check that the found $a_{opt} \in [0,1]$. If $p = 1$, then $a_{opt} = 1$. If $p \to 0$, then $a_{opt} \to 0$.

Task 18. Let's compute likelihood function for the observed dataset given q:

$$p(X|q) = p(x_1, x_2, \ldots, x_N|q) = \prod_{i=1}^{N} p(x_i|q) = \prod_{i=1}^{N} (1-q)^{x_i - 1} q = (1-q)^{\sum_i x_i - N} q^N$$

Then

$$q_{ML} = \arg\max_{q\in[0,1]} p(X|q) = \arg\max_{q\in[0,1]} \log p(X|q) = \arg\max_{q\in[0,1]} \left(\sum_{i=1}^{N} x_i - N\right)\log(1-q) + N\log q$$

Let's differentiate the obtained expression w.r.t. q and equate the result to zero:

$$\frac{d}{dq}\log p(X|q) = -\frac{\sum_i x_i - N}{1-q} + \frac{N}{q} = 0 \;\Rightarrow\; -q\left(\sum_i x_i - N\right) + N(1-q) = 0 \;\Rightarrow\; q_{ML} = \frac{N}{\sum_i x_i}.$$

Since all $x_i \geq 1$ then $N/\sum_i x_i$ is always between 0 and 1 and we don't need to deal with condition $q \in [0,1]$ by introducing Lagrange function.

Task 19. Here we need to train a separate linear classifier for all distinct pair of classes. Each classifier has $D+1$ parameters (all feature weights plus bias term). Number of distinct pairs is binomial coefficient $\binom{K}{2} = \frac{K(K-1)}{2}$. So the total number of parameters for training would be

$$\frac{K(K-1)}{2}(D+1).$$

Task 20. Gini index is defined as

$$H = \sum_{k=1}^{K} p_k(1-p_k).$$

Since there are only two classes then $p_1 = 1 - p_2$ and Gini index can be simplified to

$$2p(1-p),$$

where p is a ratio of positive objects. Let's compute Gini index for all 8 objects:

$$H = 2 \cdot \frac{7}{8} \cdot \frac{1}{8} = \frac{14}{64} = \frac{7}{32}.$$

It is obvious that there are only two potential interesting splits: $R_l = \{1,2,3\}$, $R_r = \{4,5,6,7,8\}$ and $R_l = \{1,2,3,4\}$, $R_r = \{5,6,7,8\}$. For the first split we have

$$H(R_l) = 0,\; H(R_r) = 2 \cdot \frac{1}{5} \cdot \frac{4}{5}.$$

The impurity reduction criterion has the following value:

$$Q = H(R) - \frac{|R_l|}{|R|}H(R_l) - \frac{|R_r|}{|R|}H(R_r) = \frac{7}{32} - \frac{3}{8}\cdot 0 - \frac{5}{8}\cdot\frac{8}{25} = \frac{7}{32} - \frac{1}{5} = \frac{3}{5\cdot 32}$$

For the second split:

$$H(R_l) = 2\cdot\frac{1}{4}\cdot\frac{3}{4},\ H(R_r) = 0,$$

and the impurity reduction is:

$$Q = H(R) - \frac{|R_l|}{|R|}H(R_l) - \frac{|R_r|}{|R|}H(R_r) = \frac{7}{32} - \frac{1}{2}\cdot 0 - \frac{1}{2}\cdot\frac{3}{8} = \frac{7}{32} - \frac{3}{16} = \frac{1}{32}.$$

Looking at these results we see that 1) the impurity reduction is positive, so it is beneficial to do a split and 2) the second split has higher impurity reduction and thus is preferred.