Machine Learning: From Theory to Practice

Exam answers (2017)

January 9, 2018

1. **Explain why one needs to define a performance measure to talk of learning.**

   A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E. Therefore, P must be carefully defined for the learning task to "learn" something pertinent for the given task.

2. **How is the k-Nearest-Neighbors classification method related to conditional density estimation?**

   It allows us to estimate $Y|\mathbf{X}$ (or $\mathbb{E}[Y|\mathbf{X}]$) directly without resorting to an explicit parametric model; the idea behind it is that the behavior of $Y|\mathbf{X}$ is approximately constant in a given neighborhood. Thus k-NN gives us a local conditional density estimate: $\widehat{\mathbb{P}\{Y=1|\mathbf{X}\}} = \frac{\sum_{\mathbf{X_i}\in\nu_\mathbf{X}} \mathbf{1}\{Y_i=+1\}}{|\nu_\mathbf{X}|}$

3. **Why is there a need to make a competition between different learning algorithm?**

   All machine learning algorithms have different characteristics, that make them each more adapted to certain situations. However it is always interesting to compare several algorithms for the same task, either to find the one which performs the best, or to merge different models with ensemble methods.

4. **What is a V-fold cross validation scheme and how to use it?**

   Split dataset $\mathcal{D}$ in V sets $\mathcal{D}_v$ of almost equal size. Then for $v \in \{1, ..., V\}$, learn $\widehat{f}^{-V}$ from the dataset $\mathcal{D}$ minus the set $\mathcal{D}_v$ and compute the empirical error $\mathcal{R}_n^{-v}(\widehat{f}^{-v}) = \frac{1}{n_v}\sum_{(\mathbf{X}_i,Y_i)\in\mathcal{D}_V} |Y_i - \widehat{f}^{-v}(\mathbf{X}_i)|^2$. Finally, compute the average empirical error: $\mathcal{R}_n^{CV}(\widehat{f}) = \frac{1}{V}\sum_{v=1}^{V} \mathcal{R}_n^{-v}(\widehat{f}^{-v})$.

5. **How is justified the choice of V = 5 or V = 10?**

   Fine analysis shows that the larger V, the better. The choice the depends on an accuracy/speed trade-off, and is usually chosen between 5 and 10.

6. **Explain the underlying principle of the penalization schemes.**

   The penalization approach consists of using the empirical loss criterion and penalizing it by a term increasing with the complexity of $\mathcal{S}$: $R_n(\widehat{f_\mathcal{S}}) \to R_n(\widehat{f_\mathcal{S}}) + \text{pen}(\mathcal{S})$. It stems from the realization that the empirical loss computed on an estimator selected in a family according to the data is biased (optimistic estimation of the risk): the goal is then to estimate an upper bound of this optimism for a given family (penalty) and to add it to the empirical loss.

7. **What is the principle of the Principal Component Analysis?**

   The goal is to construct a map ϕ from the space $\mathcal{X}$ of our problem into a space $\mathcal{X}'$ of smaller dimension. The PCA algorithm is as follows: first, compute the empirical mean $m = \frac{1}{n}\sum_{i=1}^{n} \mathbf{X}_i$ and the empirical covariance matrix $\Sigma = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{X}_i - m)(\mathbf{X}_i - m)^\top$. Finally, compute the d' first eigenvectors of this matrix $V^{(1)}, ..., V^{(d')}$ and set $\phi(\mathbf{X}) = V^\top(\mathbf{X} - m)$

8. **What is its relation with the SVD?**

   In the PCA, covaiance matrix $C = X^\top X$ is symmetric and can then be diagonalized as $C = QLQ^\top$, where Q is a matrix of eigenvectors and L is a diagonal matrix with eigenvalues λ_i in decreasing order on the diagonal. If we perform SVD on the data, we get $X = USV^\top$, where U and V are orthonormal and S is nonzero only on the diagonal. If we come back to our covariance matrix, it can then also be written as $C = USV^\top VS^\top U = U(SS^\top)U$, where $SS\top$ is diagonal. We notice that $U(SS^\top)U$ has the same structure as $QLQ^\top$; we can say that the U matrix in the SVD is the Q matrix in the PCA. PCA is also known as truncated SVD.

9. **Describe the two kind of ensemble methods proposed to stabilize decision trees.**

   Parallel construction: constructing several trees from bootstrapped samples and average the responses (bagging) and/or adding more randomness in the tree construction (random forest). Sequential construction: constructing a sequence of trees by reweighting sequentially the samples according to their difficulties (adaboost) or reinterpreting as a stagewise additive model (boosting).

10. **What is the gradient boosting algorithm?**

    Informally, what a boosting algorithm does is combining weak learners (ie. only slightly better than random guess) into a strong learner. It does so iteratively, by reweighting the training data after each iteration through a weak learner (stronger weights to misclassified/predicted data), and using those newly reweighted data into the next learner. Misclassification is evaluated through a loss function that the learners try to minimize; in the case of gradient boosting, it is done by taking a gradient step at each iteration (after replacing the function by a first order approximation), that can be combined with line search to find the optimal step.

11. **What is the principle of the graph spectral clustering?**

    Given an enumerated set of data points, the similarity matrix may be defined as a symmetric matrix A, where $A_{ij} \geq 0$ represents a measure of the similarity between data points with indices i and j. The general approach to spectral clustering is to use a standard clustering method (such as a k-NN graph) on relevant eigenvectors of a Laplacian matrix of A. The eigenvectors that are relevant are the ones that correspond to smallest several eigenvalues of the Laplacian except for the smallest eigenvalue which will have a value of 0.

12. **Provide a loss function with appropriate penalties that allows to address semi- supervised regression.**

    Not sure about this one.

13. **Explain how matrix factorization is a way to solve collaborative filtering in recommendation systems.**

    Let's consider items for sale on an e-commerce site. We represent user $i \in \{1, ..., I\}$ as a vector of preferences for a few k factors p_i (eg. ease-of-use, value-for-money, aesthetics $\rightarrow p_i = [0.7, 0.5, 0.1]$). Then let's represent item $j \in \{1, ..., J\}$ as a vector q_j where each element expresses how much the item exhibits that factor (eg. toothbrush $\rightarrow q_j = [0.9, 0.8, 0.1]$). We can then estimate an item j potential rating by user i with $p_i^\top q_j$. We can therefore represent all users in a matrix P $(I \times k)$, all sold items in a matrix Q $(I \times k)$ and approximate all ratings by $R = PQ$.

14. **Explain what is a representer theorem and how it can be useful in the framework of Reproducing Kernel Hilbert Spaces.** Let $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ be a positive definite symmetric kernel and $\mathcal{H}_k$ its corresponding RKHS. Then for any strictly increasing function $\Omega : \mathbb{R} \rightarrow \mathbb{R}$ and any loss function $L : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$, any minimizer of $J(f) = L(f(x_1), ..., f(x_n)) + \lambda\Omega(\|f\|_{\mathcal{H}})$ admits an expansion of the form $f^*(\cdot) = \sum_{i=1}^n \alpha_i k(x_i, \cdot)$. It is useful because it converts a minimization problem in infinite dimensional space into a finite dimensional space.

15. **What is the kernel trick?**

    In the dual formulation of the SVM/SVR problem, we notice that each time the training data appear in the objective dual function, they appear as dot products. Therefore we just need to compute the scalar products during the learning phase as well as the prediction phase. So whatever space/set $\mathcal{X}$ we just need a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that k computes inner products. k needs to be a positive definite symmetric kernel; then there exists a Hilbert space $\mathcal{H}$, called feature space, and a feature map $\phi : \mathcal{X} \to \mathcal{H}$ such that $k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$ where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is the dot product associated with $\mathcal{H}$.

16. **What are the parameters of a Neural Network?**

    The parameters of a neural network are its number of connections between neurons and their associated weights.

17. **How the Convolution Neural Networks used for instance in image processing reduce the number of those parameters?**

    In a traditional neural network, each input (pixel value) is connected to every neuron in the first layer. So each neuron in the first layer is getting input from every part of the image. With a convolutional network, each neuron only receives input from a small local group of the pixels in the input image. This is called local connectivity: all of the inputs that go into a given neuron are actually close to each other, and thus share the same weight. Therefore the number of weights to be computed each time is reduced.

18. **What is the principle of the back-prop algorithm?**

    The error contribution of each neuron is calculated after a batch of data is processed and distributed back through the network layers by adjusting the weight of neurons accordingly to minimize the loss function (usually gradient descent).

19. **How to construct a bag of words representation of a text?** A dictionary is first constructed with every different words of the text. Vectors are then formed for each given sample (ie. sentences) to contain the count of each of words of the dictionary; the length of each vector is then equal to the total number of different words in teh text. Each sample is therefore represented in what is called a vector space, and all vectors have the same length. The whole text can then be represented as a matrix, with each sentence as a row.

20. **How to use this representation to classify text in an unsupervised way?**

    Topic modeling is an example of unsupervised text classification that can be done via the bag of words approach, using models such as Latent Dirichlet Allocation or Non-Negative Matrix factorization; both different approaches make use of the text representation in a vector space and take as input a bag of words matrix.