Final Exam, Machine Learning (MIRI) June 16th, 2021

Instructions

- You have **2h** to solve the exam.
- Please **note your seat for covid-tracking purposes in all your pages** together with your full name.
- You **can** use:
  - Any paper notes and books you bring to the exam
  - A calculator
  - A PDF viewer on your laptop/tablet for lecture slides or other previously downloaded material
- You **cannot** use:
  - Any connectivity to wifi or network at all. Make sure you download all material before the exam.
  - Any software on your laptop, including python/R/matlab or numeric solvers.

  If in doubt, ask for permission before. Any violation of this rule will be considered cheating.
- Be concise but clear. We may penalize answers that are unnecessarily lengthy.
- Good luck!

All questions have equal weight. Justify all your answers except Question 5.

Question 1

Ordinal regression refers to a type of supervised learning problem where (discrete) taget labels show a natural ordering. For example, classifying wines in a 1-10 scale, or predicting customer satisfaction into one of `excellent`, `good`, `average`, or `bad`.

Given the type of supervised learning problem that you have come accross during our course, what modifications or extensions can you think of that could solve the ordinal regression problem? You can mention how to alter learning algorithms, or cost functions, or anything you can think of.

Solution There is no single solution to this question, as it is the subject of ongoing research. Accepted solutions are along the lines of:

- Treat the problem as a multi-class classification problem. In this case, something along the lines of specifying different costs for different types of mistakes should be mentioned. For example, predicting 1 when true label is 5 should be more penalized than predicting 4 when true label is 5.
- Treat the problem as a regression problem by mapping discrete ordered labels into, for example, integers that respect the ordinal order. EG map excellent to +2, good to +1, average to 0, and bad to -1. Other values could be chosen if one wants to separate more some labels. Then, to predict one just needs to round. Some observation should be made along the lines of what to do with "out-of-range" predictions, e.g. clipping to max or min value possible, or issuing an alert or something like that. For this to work reasonably well, distance between mapped ordinal values should reflect distance over the ordinal categories.
- Convert ordinal regression with K levels into $K-1$ binary classification problems: eg train one classifier to distinguish between excellent vs. rest, train another one to distinguish excellent+good vs average+bad, train another one to distinguish bad from rest. And then, to assign final level combine information given by $K-1$ classifiers in some meaningful way.

clipping
range
bhanda
tala mathi
aauda error

If want more information on this topic, you can start reading this paper.

Question 2

A data analyst has received a (small) set of expensive, labelled data and wants to build a good predictive classification model. She comes up with the following protocol:

1. Split all available data into training, validation and test sets using 50/25/25 proportions at random
2. Train SVM model with default parameters and polynomial kernel of degree 3 on the training set
3. Train and optimize Random Forest, optimizing its hyper-parameters using OOB on the training set
4. Choose best of SVM, and RF models using error on the validation set
5. Estimate true error of selected model using test set

Please criticize (in a constructive manner) this solution.

Solution While the solution is reasonable, there are several weak points that we can improve on. In what follows, I will list strong and weak points of this solution:

Strong points:

- She uses validation error (so, estimate of *predictive* error) as a guide to optimize and select models
- She does not look into the test data until the end, in order to have a good estimate of true error (generalization error)
- She uses a validation metric to optimize the RF model (OOB error)

Weak points:

- Using a training/validation/test split is not the best option when one has scarce data, the results will be very sensitive to the particular partition made, so probably it would be a better idea to do cross-validation with a larger k in order to obtain more stable estimates
- Using default hyperparameters and a polynomial kernel of degree 3 for the SVM model seems a bit arbitrary; probably it would be a good idea to optimize hyperparameters using some resampling technique
- Probably it would be a good idea to check first whether a linear model does well (regularized logistic or linear regression, or some other GLM model)
- Before checking true error on test set, one could re-train the final model on all training + validation data to make full use of "non-test" data

In summary, the main problem with this solution is that she does not make the best use of the available data, which we are told is scarce. Therefore, she may not find the best model that she could were she to use a more stable resampling technique (eg loocv) instead of a single train/val split. But other than that, she commits no major "ml sins". Also it is strange that she optimizes one model (RF) but not the other one that she uses (SVM) and as a result she may not find a possible configuration for SVM that may work well in her data.

Question 3 do concatenation

During our last class, we saw that if k_1 and k_2 are *kernel* functions on $\mathbb{R}^d$, then

$$k_3(u, v) := k_1(u, v) + k_2(u, v)$$

is also a kernel (with $u, v \in \mathbb{R}^d$). Please describe the feature map associated with this new kernel k_3.

Solution The feature map associated with k_3 is the concatenation of feature maps of k_1 and k_2. Namely, if (in the following $< u, v >$ denotes the inner product of vectors u and v):

- ϕ_1 is the map such that $k_1(u, v) =< \phi_1(u), \phi_1(v) >$ and
- ϕ_2 is the map such that $k_2(u, v) =< \phi_2(u), \phi_2(v) >$, then
- $\phi_3(u) := (\phi_1(u), \phi_2(u))$ (meaning the vector that contains all features from ϕ_1 and then the ones from ϕ_2, in any order).

Clearly,

$$\begin{aligned} k_3(u,v) &= < \phi_3(u), \phi_3(v) > \\ &= < (\phi_1(u), \phi_2(u)), (\phi_1(v), \phi_2(v)) > \\ &= < \phi_1(u), \phi_1(v) > + < \phi_2(u), \phi_2(v) > \\ &= k_1(u,v) + k_2(u,v) \end{aligned}$$

as required.

Question 4

Explain in your own words the difference between the **posterior** and the **predictive** distribution in Bayesian learning.

Solution The posterior distribution is a probability distribution that describes how probable model parameter values are after observing some data. It describes the parameter space, and combines our prior knowledge with the observed data.

The predictive distribution is instead a distribution over response values (predicted values) that indicates how probable each response is. It is therefore a distribution not over parameters but over the space of target values instead. It is computed by marginalization (integrating out) of all possible parameter values weighted by their posterior probabilities. It is a direct application of the product and sum rules.

Sure! In Bayesian learning, the posterior distribution represents our updated belief about the parameters of a model after taking into account the observed data. It's the distribution of the model parameters given the data.

On the other hand, the predictive distribution represents our uncertainty about future observations given the observed data. It's the distribution of the yet-to-be-observed data points, given the data we have already observed.

So, in short, the posterior distribution tells us about the parameters of the model after observing the data, while the predictive distribution tells us about the distribution of future observations given the data we have observed.

Question 5

Please mark whether the following statements are **true** or **false**; the score for this question is given by the formula $2 \times \frac{\text{nr. of correct} - \text{nr. of incorrect questions}}{15}$ when positive, otherwise it is 0.

- ☐ Training error is always lower than test error FALSE: generally it is, but it cannot be guaranteed to be so
- ☐ Lasso and ridge regression both help in reducing overfitting TRUE: it is the main point of regularizing
- ☐ Lasso regression is preferable to ridge regression because it produces sparse models FALSE: it depends on the data, sparse models do not always work best
- ☐ The activation functions of output neurons of a neural network are determined mainly by the nature of the target variable one wants to predict TRUE
- ☐ Linear regression assumes Gaussian input variables FALSE: assumes Gaussian response assumes gaussian noise
- ☐ Naive Bayes assumes Gaussian input variables FALSE: a particular case, Gaussian NB does, but in general NB can be applied to any combination of input variable distributions
- ☐ Bayes formula is used in Bayesian learning to obtain posterior distributions TRUE

- ☐ It is not possible to train a neural network for both regression and classification at the same time FALSE: all we need to do is have a differentiable error function combining both regression error and classification error; a simple sum may do
- ☐ Bigger training sets help to reduce overfitting TRUE: with more and more data overfitting becomes harder
- ☐ It is impossible to evaluate the quality of a clustering result because we never have the ground truth FALSE: we have heuristics like the CH-index that do this
- ☐ Cross-validation is a resampling method used to select a good model TRUE
- ☐ Gaussian Naive Bayes assumes Gaussian input variables TRUE
- ☐ Backprop is an algorithm used in neural network learning to obtain partial derivatives of an error function with respect to its weights TRUE
- ☐ The negative log-likelihood can always be used as an error function in supervised learning TRUE
  if probabilistic model, true
  else, false
- ☐ The EM algorithm is particularly suited to learn probabilistic models with partially observed data TRUE