MATH 342W / 642 / RM 742 Spring 2024 HW #5

Benjamin Minkin

Thursday 16th May, 2024

Problem 1

These are some questions related to probability estimation modeling and asymmetric cost modeling.

(a) [easy] Why is logistic regression an example of a "generalized linear model" (glm)?

It is a linear model as the independent variables have a linear relationship with the log-odds of a y=1 being predicted. It is GLM because it uses the logistic CDF as a link function to shrink to sample space between 0 and 1. This is a generalization hence the GLM.

(b) [easy] What is $\mathcal{H}_{pr}$ for the probability estimation algorithm that employs the linear model in the covariates with logistic link function?

These are all possible logistic CDF sigmoids. Because the CDF always ranges from 0-1, they are all valid link functions for binary classification models.

(c) [easy] If logistic regression predicts 3.1415 for a new $\boldsymbol{x}_*$, what is the probability estimate that $y = 1$ for this $\boldsymbol{x}_*$?

This is equivalent to:

$$\frac{1}{1 + e^{-3.1415}} = 0.96$$

(d) [harder] What is $\mathcal{H}_{pr}$ for the probability estimation algorithm that employs the linear model in the covariates with cloglog link function?

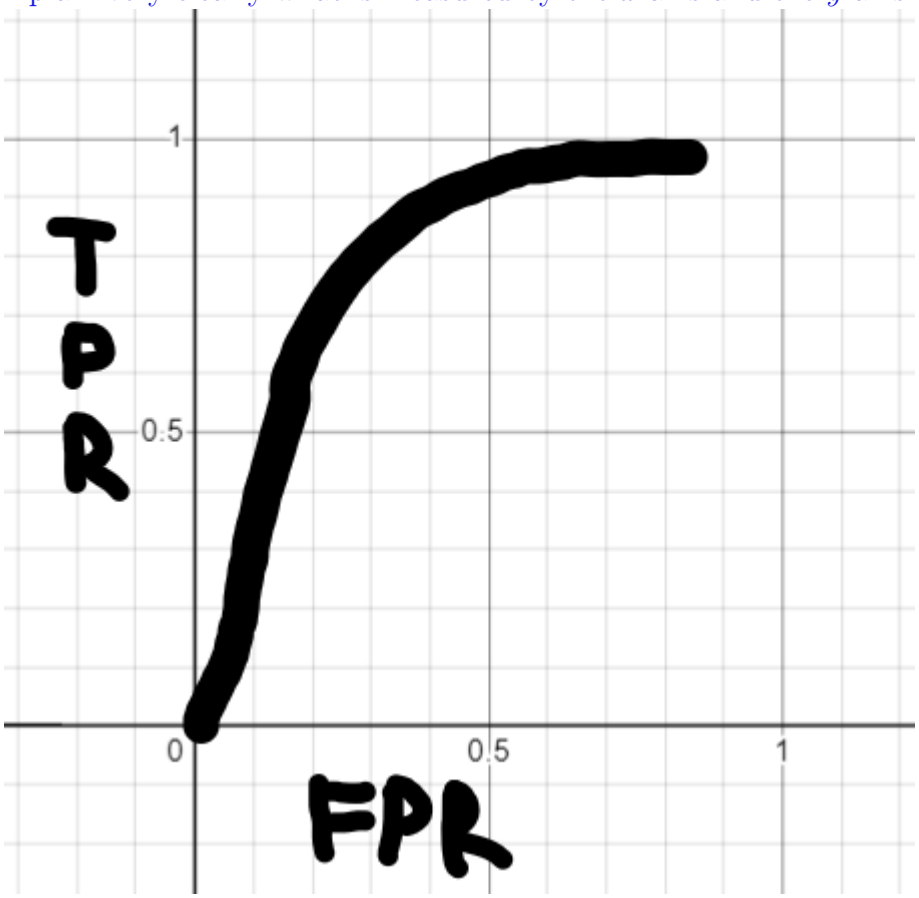
These are all possible Gumbel CDF sigmoids.

(e) [difficult] Generalize linear probability estimation to the case where $\mathcal{Y} = \{C_1, C_2, C_3\}$. Use the logistic link function like in logistic regression. Write down the objective function that you would numerically maximize. This objective function is one that is argmax'd over the parameters (you define what these parameters are — that is part of the question).

Once you get the answer you can see how this easily goes to $K > 3$ response categories. The algorithm for general K is known as "multinomial logistic regression", "polytomous

LR", "multiclass LR", "softmax regression", "multinomial logit" (mlogit), the "maximum entropy" (MaxEnt) classifier, and the "conditional maximum entropy model". You can inflate your resume with lots of jazz by doing this one question!

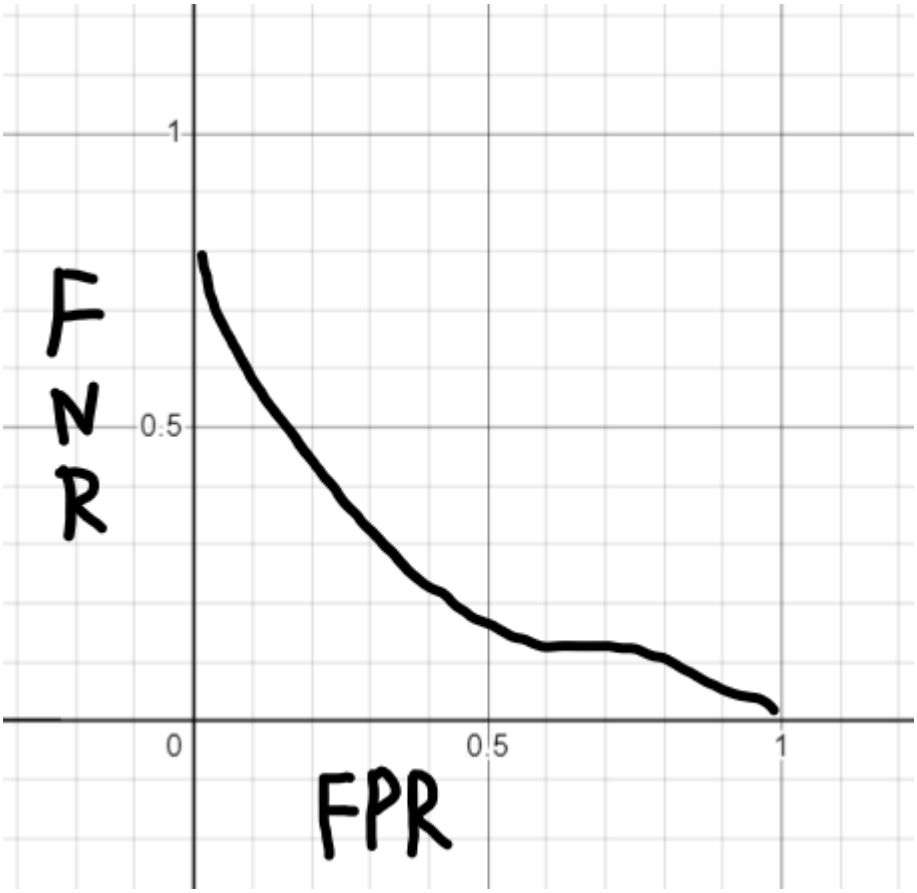
(f) [easy] Graph a canonical ROC and label the axes. In your drawing estimate AUC. Explain very clearly what is measured by the x axis and the y axis.



(g) [easy] Pick one point on your ROC curve from the previous question. Explain a situation why you would employ this model.

At point (0.1,0.5) the mode is having pretty good accuracy. I would use this in a case where there is symmetric cost to making errors.

(h) [harder] Graph a canonical DET curve and label the axes. Explain very clearly what is measured by the x axis and the y axis. Make sure the DET curve's intersections with the axes is correct.



(i) [easy] Pick one point on your DET curve from the previous question. Explain a situation why you would employ this model.

At point (0.4,0.2) I would use this model when there is symmetry in cost between errors.

(j) [difficult] [MA] The line of random guessing on the ROC curve is the diagonal line with slope one extending from the origin. What is the corresponding line of random guessing in the DET curve? This is not easy...

Problem 2

These are some questions related to bias-variance decomposition. Assume the two assumptions from the notes about the random variable model that produces the δ values, the error due to ignorance.

(a) [easy] Write down (do not derive) the decomposition of MSE for a given $\boldsymbol{x}_*$ where $\mathbb{D}$ is assumed fixed but the response associated with $\boldsymbol{x}_*$ is assumed random.

$$bias(f(x), g(x))^2$$

(b) [easy] Write down (do not derive) the decomposition of MSE for a given $\boldsymbol{x}_*$ where the responses in $\mathbb{D}$ is random but the $\boldsymbol{X}$ matrix is assumed fixed and the response associated with $\boldsymbol{x}_*$ is assumed random like previously.

$$var(g(x)|x) + E[f(x*) - g(x)^2]^2$$

(c) [easy] Write down (do not derive) the decomposition of MSE for general predictions of a phenomenon where all quantities are considered random.

$$var(g(x)) + var(f(x) + cov(g(x), f(x))$$

(d) [difficult] Why is it in (a) there is only a "bias" but no "variance" term? Why did the additional source of randomness in (b) spawn the variance term, a new source of error?

The randomness in the response add the new variance term. In (a) the variance is 0 because the term is fixed.

(e) [harder] A high bias / low variance algorithm is underfit or overfit?

Likely underfit.

(f) [harder] A low bias / high variance algorithm is underfit or overfit?

Likely overfit.

(g) [harder] Explain why bagging reduces MSE for "free" regardless of the algorithm employed.

Bagging uses averages and randomness to get a better prediction. This is free because it does not require more data for training.

(h) [harder] Explain why RF reduces MSE atop bagging M trees and specifically mention the target that it attacks in the MSE decomposition formula and why it's able to reduce that target.

Random forest method atttacks the variance term in the MSE decomposition. IT uses random subsets of features to lessen the covariance between iterations. This decrease in covariance leads to a decrease in overall variance.

(i) [difficult] When can RF lose to bagging M trees? Hint: think hyperparameter choice.

If the hyperparameters are either too high or too low for the specific data, bagging can do better.

Problem 3

These are some questions related to missingness.

(a) [easy] [MA] What are the three missing data mechanisms? Provide an example when each occurs (i.e., a real world situation). We didn't really cover this in class so I'm making it a MA question only. This concept will NOT be on the exam.

(b) [easy] Why is listwise-deletion a *terrible* idea to employ in your $\mathbb{D}$ when doing supervised learning?

This is because it deletes large amounts of data. If there is an underlying reason for missingness, this can introduce serious bias to the model.

(c) [easy] Why is it good practice to augment $\mathbb{D}$ to include missingness dummies? In other words, why would this increase oos predictive accuracy?

This makes it possible to take missingness into account as a variable. If there is a reason for the missing entries this will try to adjust for that.

(d) [easy] To impute missing values in $\mathbb{D}$, what is a good default strategy and why?

A good strategy is to use the average of that feature. This is the best guess for what it could have been.

Problem 4

These are some questions related to gradient boosting. The final gradient boosted model after M iterations is denoted G_M which can be written in a number of equivalent ways (see below). The g_t's denote constituent models and the G_t's denote partial sums of the constituent models up to iteration number t. The constituent models are "steps in functional steps" which have a step size η and a direction component denoted $\tilde{g}_t$. The directional component is the base learner $\mathcal{A}$ fit to the negative gradient of the objective function L which measures how close the current predictions are to the real values of the responses:

$$\begin{aligned}
G_M &= G_{M-1} + g_M \\
&= g_0 + g_1 + \ldots + g_M \\
&= g_0 + \eta\tilde{g}_1 + \ldots + \eta\tilde{g}_M \\
&= g_0 + \eta\mathcal{A}\left(\langle \boldsymbol{X}, -\nabla L(\boldsymbol{y}, \hat{\boldsymbol{y}}_1)\rangle, \mathcal{H}\right) + \ldots + \eta\mathcal{A}\left(\langle \boldsymbol{X}, -\nabla L(\boldsymbol{y}, \hat{\boldsymbol{y}}_M)\rangle, \mathcal{H}\right) \\
&= g_0 + \eta\mathcal{A}\left(\langle \boldsymbol{X}, -\nabla L(\boldsymbol{y}, g_1(\boldsymbol{X}))\rangle, \mathcal{H}\right) + \ldots + \eta\mathcal{A}\left(\langle \boldsymbol{X}, -\nabla L(\boldsymbol{y}, g_M(\boldsymbol{X}))\rangle, \mathcal{H}\right)
\end{aligned}$$

(a) [easy] From a perspective of only multivariable calculus, explain gradient descent and why it's a good idea to find the minimum inputs for an objective function L (in English).

The gradient is used to find the most steep angle either up or down. Gradient descent is an algorithm that uses that information to iteratively move towards a local maximum or minimum. This will be used to be to find the best inputs for the model a.k.a. the local maximum of the objective function.

(b) [easy] Write the mathematical steps of gradient boosting for supervised learning below. Use L for the objective function to keep the procedure general. Use notation found in the problem header.

1. Choose a random starting point.
2. Calculate the gradient at that point
3. Use a $\mathcal{A}$ as a base learner
4. Take a step of size η towards the local max
5. Update the model
6. Repeat steps 2-5 M times

(c) [easy] For regression, what is $g_0(\boldsymbol{x})$?

That is the initial model. In this case it is a random starting point. An educated $g_0(\boldsymbol{x})$ is a degenerate function that yields the mean of the responses.

(d) [easy] For probability estimation for binary response, what is $g_0(\boldsymbol{x})$?

An educated $g_0(\boldsymbol{x})$ for binary response is the mode of the responses.

(e) [harder] What are all the hyperparameters of gradient boosting? There are more than just two.

1. Base learner selection
2. Size of step taken
3. Size of M times to iterate
4. How many times to run the gradient boosting algorithm to infer towards the absolute max from local maxes.

(f) [easy] For regression, rederive the negative gradient of the objective function L.

(g) [easy] For probability estimation for binary response, rederive the negative gradient of the objective function L.

(h) [difficult] For probability estimation for binary response scenarios, what is the unit of the output $G_M(\boldsymbol{x}_\star)$?

The output will be the probability that the output is 1 given the selected inputs.

(i) [easy] For the base learner algorithm $\mathcal{A}$, why is it a good idea to use shallow CART (which is the recommended default)?

This is because shallow CART is a weak learner. This counter balances the overfitting that can occur using a deeper CART.

(j) [difficult] For the base learner algorithm $\mathcal{A}$, why is it a bad idea to use deep CART?

This may cause overfitting. Deep trees paired with random forests and used as an ensamble model will be focusing on the training data very closely. Using shallow tree can somewhat counteract this.

(k) [difficult] For the base learner algorithm $\mathcal{A}$, why is it a bad idea to use OLS for regression (or logistic regression for probability estimation for binary response)?

OLS may not be the best choice as it is not a weak learner per-se. This can cause overfitting as mentioned above.

(l) [difficult] If M is very, very large, what is the risk in using gradient boosting even using shallow CART as the base learner (the recommended default)?

There are three main ways to induce overfitting.
1. The random forest has too many trees
2. The trees are too deep
3. The process is repeated too many times

(m) [difficult] If η is very, very large but M reasonably correctly chosen, what is the risk in using gradient boosting even using shallow CART as the base learner (the recommended default)?

If the steps taken are too large, it can over shoot the local maximum. This could cause a jump from valley to valley skipping over the hill entirely.