Stats 315B Homework

Rachael Caelie (Rocky) Aikens, Daniel Sosa, Christine Tataru

Problem 1

Problem 2

$$\frac{\partial \hat{F}}{\partial a_m} = B(\mathbf{x}|\mu_m, \sigma_m) \tag{1}$$

$$\frac{\partial \hat{F}}{\partial \sigma_m} = \Sigma_{m=1}^{M} a_m (\frac{1}{\sigma_m^3} \Sigma_{j=1}^{n} (x_j - \mu_{jm})^2) B(\mathbf{x}|\mu_m, \sigma_m) \tag{2}$$

$$\frac{\partial \hat{F}}{\partial \mu_{mj}} = \Sigma_{m=1}^{M} (\frac{a_m}{\sigma_m^2} (x_j - \mu_{jm})) B(\mathbf{x}|\mu_m, \sigma_m) \tag{3}$$

Problem 3

Problem 4

Problem 5

K-fold validation is a method used for cross-validation, or to attempt to find regularization parameters that sufficiently generalize a model in question to other data (favoring bias over high variance). In this procedure, the data is randomly divided into K sets, whereby the model is built and trained on $K-1$ sets and the left out set is used for validation/regularization parameter tuning. This procedure is repeated K times and the parameters are the average parameters found from each iteration of training and validating.

Increasing K...

Advantages:

1. Averaging over more folds to find optimal parameters. Averaging over more folds decreases variance, so we're more confident in the parameters chosen.
2. Training set is larger, each fold creates a model that better estimates the training data likely.

Disadvantages:

1. Increasing number of folds decreases the size of the data used for training in each fold (each fold produces a worse model).
2. Validation set is smaller, estimate of parameters may be poor
3. As a result of 2, the KFCV may fail to produce parameter settings that don't overfit the data
4. More computationally expensive.

Cross-validation will estimate the performance of the actual predicting function when the data held out for validation is drawn from the same distribution as the rest of the training data, otherwise the CV may be skewed by influential outliers for example.

Problem 6

Training separate neural nets for each $y_m \ldots$

Advantages:

1. A more accurate model for each y_m.
2. Perhaps different input data is important for different y_m's and by separating the neural nets, the learned parameters are more robust to uninformative variables.

Disadvantages:

1. Computationally very expensive potentially
2. Inefficient way to make a model if there do exist dependencies between the y_m's
3. Not attempting to capture inherent hierarchical "structure" from the input data (as in the case with filters in image processing for example).

Training separate neural nets might make sense when the y_m's are dependent on different variables entirely and have little correlation. Training with the same neural net would make sense when there's more of a dependency structure between the y_m's.

Problem 7