Benjamin Limoges
Homework 9

1. “Maximizing the sharpness” of a learner is equivalent to saying that the learner picks instances to label that increases how certain it thinks it is across the rest of the unlabeled data. To see that the log loss function maximizes the sharpness, you can plot the loss versus how certain the learner is. On this graph, the end points (where the learner is 100% certain that class 2, or 100% that it is class 1) the loss from predicting a point is 0. Since loss cannot be negative, then the function is either flat between the point (there is never a loss) or that it is positive at a point (meaning that the learner still has room to learn). Functions that are flat have uniform loss (potentially at 0) and are not sharp learners, ones that have positive loss are. Using Rolle's Theorem, we can see that there must be at least one point in between the two endpoints where the first derivative is 0, leading us to a maximum. Since logloss takes this shape, and the learner is trying to maximize sharpness, then the authors' statement is a valid claim.

2. Assuming that the learner is maximizing the conditional probability, the major impact of using logloss over 0/1 (or hinge) loss for active learning is the sacrifice in classification accuracy. This is because the logloss learner may penalize correct classifications – if the learner penalizes correct classifications it naturally cannot be more accurate than one that does not. This isn't to say logloss is a bad loss function to use (or even sub-optimal); one major advantage of using logloss is that it gives posterior probabilities.

3. Skewed probabilities are a major issue anytime you have an underlying distribution that is a multinomial one. When we use Bayes Rule to find the probability of a class conditional on a document, we are going to have to multiply all of the conditional probabilities of a word occuring given a class label. If any one of these conditional probability is 0 for any class label, then the resulting left hand side will be 0 since it is a product of these conditional probabilities. When the product is 0, it is telling the learner that it is certain that there is no gain to labeling these types of documents. By using a smoothing technique (in this case Laplacian), we can avoid this and still learn, even if that word has never been documented with a specific class label.

4. My heuristic would be to look at a graph of the accuracy over the validation site. Similar to determining optimal clustering, I would see if the difference in accuracy plateaued (this is in effect the same as looking at the rate at which accuracy is increasing.) If the accuracy is relatively constant for a few requested labels, then it is probable that no more labels are needed.

   There are a few issues with this approach. One issue is that this will require there to be “too many” labeled instances; since it is backwards looking, it will need to label a few too many instances to make sure that it has plateaued. Another con is that number of consecutive plateaued instances is a parameter in this approach – therefore this method is sensitive to the choice of this parameter. If it is set too large then the learner will request too many points be labeled incurring a large cost; if it is too low then it may miss out on accuracy gains.

   The final issue is that the accuracy calculation is stochastic rather than deterministic. This means that there are random fluctuations – I'm not guaranteed that the accuracy function is monotonically increasing and concave. This means my method is only a good method on average – there are random fluctuations that could cause my learner to request too many labels.

5. I am quite partial to the pool based uncertainty sampling discussed in class. The pros to uncertainty sampling is that relatively cheap to implement, as well as rather easy to implement since it is a simple greedy algorithm. Since I'm a poor programmer, I have a tendency to pick easy and elegant solutions. Uncertainty sampling is also a good method for large set of possible hypothesis spaces.

   There are quite a few cons with uncertainty sampling. The foremost is that uncertainty sampling works best with large, compact, and connected classes. It is entirely possible for uncertainty sampling to miss extreme minority classes, which could have extremely negative consequences depending on the problem. It also can miss pieces in identifying classes if the class is disconnected, such as the two triangles example in class.