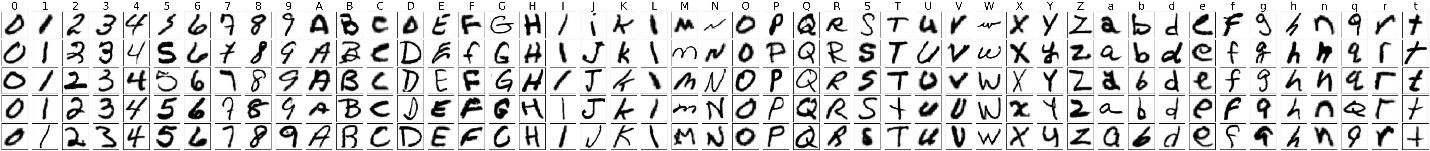
Mini-Project: Categorical Naive Bayes

I have executed the project in mini\_project.py, with the generated plots as pdfs. I will outline my project here along with the plots.

# 1. Visualizing the EMNIST Dataset

I create a dictionary mapping each class label to a list of ids having that label. I then loop through all classes, sample 5 images from each class and lay them out in a column.

**Task 1: Display samples from the dataset in a 5 x C table.**



# 2. Building a Model

I create a class CategoricalNaiveBayes, inheriting from the BaseEstimator, creating the following classes – fit(), predict(), predict\_log\_proba(), predict\_proba(), score().

**Task 2.1: I use the following priors –**

**Task 2.2: For Prediction, I use -**

**Task 2.3: Scoring** is done by averaging up log likelihoods of the true class for each sample in the dataset.

# 3. Learning Curves for (almost) Balanced Training Data

**A graph of different colored lines

AI-generated content may be incorrect.**

**A graph of different colored lines

AI-generated content may be incorrect.**

# 4. Learning Curves for Imbalanced Training Data

# A group of graphs with numbers and lines AI-generated content may be incorrect.

# A group of graphs with numbers and lines AI-generated content may be incorrect.

# 5. Discussion

Note about varying alpha – the class prior only contributes 1 term to the scores, as opposed to 784 terms from feature priors, one for each pixel. This leads to the overlapped plots for MAP as we vary alpha. Here is the debug statement to confirm the values –

DEBUG: Training models on full training set and checking class priors:

MLE | Class prob std: 0.001216 | Train score: -226.7364 | Val score: -232.5047

MAP (α=1) | Class prob std: 0.001211 | Train score: -227.9099 | Val score: -230.9229

MAP (α=10) | Class prob std: 0.001167 | Train score: -227.9099 | Val score: -230.9228

MAP (α=50) | Class prob std: 0.001006 | Train score: -227.9099 | Val score: -230.9224

MAP (α=100) | Class prob std: 0.000858 | Train score: -227.9100 | Val score: -230.9220

MAP (α=200) | Class prob std: 0.000663 | Train score: -227.9102 | Val score: -230.9217

**5.1:** We observe an interesting difference between MLE and MAP models – for MLE, the training scores are overall better than MAP, and the training scores increase as the training set size is increased (unlike MAP). Looking at validation scores we can see that MAP outperforms MLE and the scores improve with increasing training set size for both. This is clear evidence that MLE overfits the training set and MAP performs the necessary regularization to avoid overfitting. We also have evidence that increasing the training set size also helps improve generalization.

From plots for imbalanced data, we clearly see that increasing alpha\_class value has the most dramatic effect on validation scores which again is not observed in the training scores. We don’t see much difference by increasing alpha\_class beyond 10.

beta = 1 provides the best generalization across all plots. We also observe a convergence of training scores for all plots as training set size increases, implying that regularization matters even more when the training set is smaller.

**5.2:** Looking at the results, I would select the MAP model with beta = 1 and alpha = 1 (though varying it does not seem to impact validation scores for high dimensional models). For this EMNIST dataset, I will select an alpha\_class value of more than 10 (try to create a balanced training set). The results show that from the compared models, this one is the least overfitting and performs well.