# Assignment 3

Tanuj Kumar ta.kumar@mail.utoronto.ca 1002197133

December 2017

## 1 20 Newsgroups Classification

### 1.1 A note on feature reduction and memory limitations

For the below data, there is no feature reduction. This very clearly will result in overfitting, as explained below. Feature reduction was attempted insofar as using the Truncated SVD method (an extension of the PCA dimensionality reduction method, albeit intended for sparse matrices), but even at a low number of principal components, there were constant memory errors. Thus, mathematical dimensionality reduction was limited by the architecture's limited memory.

As an extension, some non-PCA feature reduction methods could be possible. One such is simply the omitting of sparse features. A quick experiment showed that about 80,000 features had scores of less than 5 times of appearance. Thus a feature reduction of this sort would bring the feature count to about 20,000. This is still an order of magnitude larger than training and test sets, but is likely to reduce overfitting while keeping accuracy relatively intact.

### 1.2 Chosen Classifiers and Hyperparameters

Four distinct classifiers were used due to feature reduction limitations. Although we focus on the first three, the results of the Random Forest classifier will be included as well. For all of the below, the *tf-idf* feature representation was used.

- **Multinomial Naive Bayes**. $\alpha = 0.01$
- **L2 Hinge-Loss Support Vector Machine**. $\alpha = 0.0001$, hinge loss, $L2$-regularization, $C = 1.0$ penalty.
- **Multiclass L2 Logistic Regression**. $L2$-penalty regularization, $C = 1.0$ penalty cost, newton-cg solver, multinomial one-hot classifier.
- **100-Estimator Random Forest**. $estimators = 100$, Gini Impurity estimation.

### 1.2.1 Hyperparameter Tuning Method

For the above parameters save for Random Forest, the use of scikit-learn's **GridSearchCV** was a grid-search cross-validator over a dictionary of provided possible parameter values. This mechanism works by automating the process of "setting a hyperparameter or hyperparameters to certain values, then cross-validating" by using a grid search algorithm to find the best possible hyperparameter values in each search instance and build upon them until you get a classifier that has all possible optimized outputs. For Random Forest, memory errors occurred when using estimators of anything greater than 500, and the classifier itself took the longest time of all, so the estimator hyperparameter was found using hand-tuning the estimator number until 100 was reached as a reasonable value.

## 1.3 Loss Results

Our three selected algorithms are bolded. The baseline is italicized. Random Forest data provided for reference.

| classifier | *BernoulliNB* | **MultinomalNB** | **SVM** | **MultiLogReg** | RandForest |
|---|---|---|---|---|---|
| Train Loss | 0.5988 | 0.9589 | 0.9540 | 0.9073 | 0.9747 |
| Test Loss | 0.4579 | 0.7002 | 0.6966 | 0.6737 | 0.5916 |

Overall, Multinomial Naive Bayes did the best, with the Support Vector Machine a very close second.

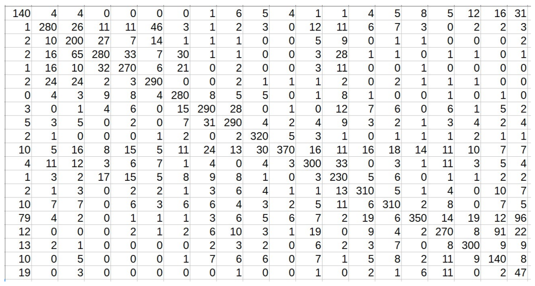
### 1.3.1 Train-Test Loss Difference and Other Experiments

As can be seen above, there is a massive difference in the training accuracy versus the testing accuracy for every classifier above. This is to be expected without feature reduction. The number of features is an order of $10^4$ while the test data has a much smaller size, in an order of $10^2$. The result is massive overfitting, as seen above, because there are too many feature values creating excessive noise when working on a smaller data set (especially one that is two orders of magnitude smaller). See the feature reduction section above. Despite this, all four experimental classifiers did better than the Bernoulli Naive Bayes baseline. Two other classifiers were attemptined: Gaussian Naive Bayes and k-Nearest Neighbours. Gaussian Naive Bayes did not work due to the sparsity of the matrix, while k-Nearest Neighbours had a terrible accuracy of around $0.12$ even after hyperparameter tuning with cross validation, due to the excessive amount of features making accurate fit assessments much too hard.

## 1.4 Confusion Matrix

Below is the confusion matrix for the best classifier, **Multinomial Naive Bayes**. (Note, if for whatever reason MNB is not necessarily a valid classifier, the SVM matrix, which was the next best algorithm, looks essentially identical).

Correct classification chart:



| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 140 | 4 | 4 | 0 | 0 | 0 | 0 | 1 | 6 | 5 | 4 | 1 | 1 | 4 | 5 | 8 | 5 | 12 | 16 | 31 |
| 1 | 280 | 26 | 11 | 11 | 46 | 3 | 1 | 2 | 3 | 0 | 12 | 11 | 6 | 7 | 3 | 0 | 2 | 2 | 3 |
| 2 | 10 | 200 | 27 | 7 | 14 | 1 | 1 | 1 | 0 | 0 | 5 | 9 | 0 | 1 | 1 | 0 | 0 | 0 | 2 |
| 2 | 16 | 65 | 280 | 33 | 7 | 30 | 1 | 1 | 0 | 0 | 3 | 28 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 16 | 10 | 32 | 270 | 6 | 21 | 0 | 2 | 0 | 0 | 3 | 11 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 2 | 24 | 24 | 2 | 3 | 290 | 0 | 0 | 2 | 1 | 1 | 1 | 2 | 0 | 2 | 1 | 1 | 1 | 0 | 0 |
| 0 | 4 | 3 | 9 | 8 | 4 | 280 | 8 | 5 | 5 | 0 | 1 | 8 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 3 | 0 | 1 | 4 | 6 | 0 | 15 | 290 | 28 | 0 | 1 | 0 | 12 | 7 | 6 | 0 | 6 | 1 | 5 | 2 |
| 5 | 3 | 5 | 0 | 2 | 0 | 7 | 31 | 290 | 4 | 2 | 4 | 9 | 3 | 2 | 1 | 3 | 4 | 2 | 4 |
| 2 | 1 | 0 | 0 | 0 | 1 | 2 | 0 | 2 | 320 | 5 | 3 | 1 | 0 | 1 | 1 | 1 | 2 | 1 | 1 |
| 10 | 5 | 16 | 8 | 15 | 5 | 11 | 24 | 13 | 30 | 370 | 16 | 11 | 16 | 18 | 14 | 11 | 10 | 7 | 7 |
| 4 | 11 | 12 | 3 | 6 | 7 | 1 | 4 | 0 | 4 | 3 | 300 | 33 | 0 | 3 | 1 | 11 | 3 | 5 | 4 |
| 1 | 3 | 2 | 17 | 15 | 5 | 8 | 9 | 8 | 1 | 0 | 3 | 230 | 5 | 6 | 0 | 1 | 1 | 2 | 2 |
| 2 | 1 | 3 | 0 | 2 | 2 | 1 | 3 | 6 | 4 | 1 | 1 | 13 | 310 | 5 | 1 | 4 | 0 | 10 | 7 |
| 10 | 7 | 7 | 0 | 6 | 3 | 6 | 6 | 4 | 3 | 2 | 5 | 11 | 6 | 310 | 2 | 8 | 0 | 7 | 5 |
| 79 | 4 | 2 | 0 | 1 | 1 | 1 | 3 | 6 | 5 | 6 | 7 | 2 | 19 | 6 | 350 | 14 | 19 | 12 | 96 |
| 12 | 0 | 0 | 0 | 2 | 1 | 2 | 6 | 10 | 3 | 1 | 19 | 0 | 9 | 4 | 2 | 270 | 8 | 91 | 22 |
| 13 | 2 | 1 | 0 | 0 | 0 | 0 | 2 | 3 | 2 | 0 | 6 | 2 | 3 | 7 | 0 | 8 | 300 | 9 | 9 |
| 10 | 0 | 5 | 0 | 0 | 0 | 1 | 7 | 6 | 6 | 0 | 7 | 1 | 5 | 8 | 2 | 11 | 9 | 140 | 8 |
| 19 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 2 | 1 | 6 | 11 | 0 | 2 | 47 |

Figure 1: Confusion matrix for MNB. $C_{ij}$ means MNB classified word as class i when it is actually j.

| class | correct % |
|---|---|
| alt.atheism | 0.566 |
| comp.graphics | 0.651 |
| comp.os.ms-windows.misc | 0.712 |
| comp.sys.ibm.pc.hardware | 0.594 |
| comp.sys.mac.hardware | 0.724 |
| comp.windows.x | 0.812 |
| misc.forsale | 0.828 |
| rec.autos | 0.749 |
| rec.motorcycles | 0.761 |
| rec.sport.baseball | 0.930 |
| rec.sport.hockey | 0.600 |
| sci.crypt | 0.723 |
| sci.electronics | 0.721 |
| sci.med | 0.824 |
| sci.space | 0.760 |
| soc.religion.christian | 0.552 |
| talk.politics.guns | 0.584 |
| talk.politics.mideast | 0.817 |
| talk.politics.misc | 0.619 |
| talk.religion.misc | 0.505 |

## 1.5 Two Obfuscated Classes and Justification

A higher probability in the above table typically implies that the corpus of words associated with that news group has specific characteristics that make the vocabulary of that topic unique to it, when compared to the topics of other newsgroups. rec.sport.baseball and misc.forsale have the highest correctness score, which makes sense insofar as baseball as a sport has a very specific vocabulary and putting items up for sale typically repeats the same common purchase-and-selling related words.

**By percentage of correctness,** the two most obfuscated classes were **alt.atheism**

and **talk.religion.misc**. The correctness score for talk.religion.misc barely surpassed 50 percent, while the rest of the words were classified as alt.atheism. This is likely due to the fact that a miscellaneous religion discussion board would likely have much in common with an atheism board when it comes to debates around religion and atheism, much of which will share similar vocabulary.

**By non-maximum diagonal numbers,** the two most obfuscated classes were **soc.religion.christian** and **talk.religion.misc**, at 96 misclassifications. This is likely due to the similarity of topic (religious), and the fact that the Usenet newsgroups were predominantly oriented around usergroups that were majority Christian in the west.

## 2 SVM 4-to-9 Binary Classification

### 2.1 SGD with Momentum

Below is a chart of the graph of SGD with momentum (red) and without momentum (blue) on the optimizing function $f(w) = 0.01w^2$.

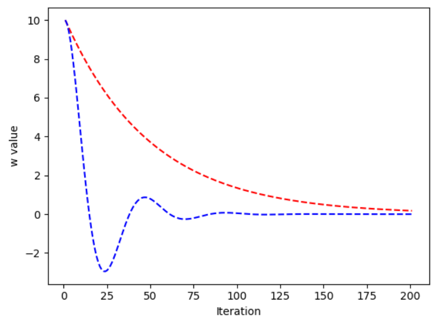


Figure 2: Graph of $w$

The momentum adds a giant drop from initialization. In fact, above there is a slight drop *below* 0 that is later corrected.

### 2.2 SVM Implementation: Pegasos and Piecewise Gradient

Check **q2.py** for the code implementation.

Two gradients provided the basis of experimentation. The first was the **Pegasos** method as covered in the paper **Pegasos: Primal Estimate sub-GrAdient SOlver for SVMs** by Shalev-Schwartz et al. The second gradient was a simple analytical differentiation with respect to **w**. For the gradient of the hinge loss:

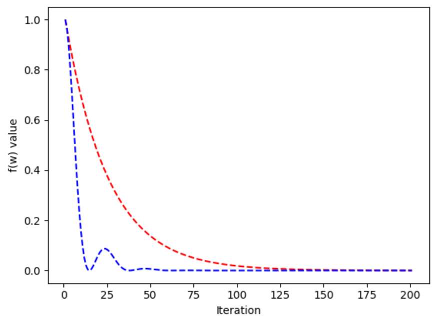


Figure 3: Graph of $f(w)$

$$\nabla_{\mathbf{w}} l_{hinge} = \begin{cases} 0 & y\mathbf{w}^T\mathbf{x} >= 1 \\ -y\mathbf{x} & y\mathbf{w}^T\mathbf{x} < 0 \end{cases}$$

And then putting together the entire gradient is trivial as above (see **grad** function). We use this second above gradient due to its speed and simplicity in the actual implementation. Note that in this context **the bias was included in the w and X terms**.

### 2.3 MNIST Binary 4-vs-9 Classification

Denote our two SVMs as $SVM$ for momentumless minibatch SGD and $SVM_M$ momentum minibatch SGD. All hyerparameters are as provided in the question. Results:

| classifier | $SVM$ | $SVM_M$ |
|---|---|---|
| train loss | 0.358 | 0.346 |
| test loss | 0.370 | 0.352 |
| train acc. | 0.896 | 0.919 |
| test acc. | 0.892 | 0.913 |

The plotted image output. $SVM$ is on the left, $SVM_M$ is on the right:

## 3 Kernels

### 3.1 Positive Semidefinite and Quadratic Form

Assume $K \in \mathbb{R}^{d\times d}$ is symmetric.

- **Proof**: If $K$ is symmetric and positive semidefinite, then $\mathbf{x}^T K\mathbf{x} \geq 0, \forall \mathbf{x} \in \mathbb{R}^d$.
    - Assume $K$ is symmetric and positive semidefinite.

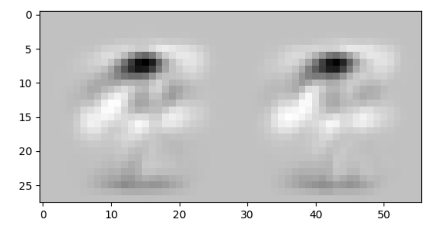


Figure 4: Plot of **w** for the two SVMs.

- Assume for purposes of contradiction that $\exists \mathbf{x} \in \mathbb{R}^d : \mathbf{x}^T K \mathbf{x} < 0$. Let this vector be $\mathbf{x}'$.
- Because $K$ is symmetric and real, $\exists Q : Q^T Q = QQ^T = I$ and $K$ has the eigendecomposition $K = Q^T \Lambda Q$ where eigenvalue $\lambda_i = \Lambda_{ii}$.
- Define $\mathbf{y}$ as $\mathbf{y} = Q\mathbf{x}'$ and thus $\mathbf{x}' = Q^T \mathbf{y}$ where $\mathbf{y} \in \mathbb{R}^d$.
- Then, $\mathbf{x'}^T Q^T \Lambda Q \mathbf{x'} < 0 \rightarrow (Q^T \mathbf{y})^T \Lambda (Q^T \mathbf{y}) < 0$
- Then, $\mathbf{y}^T Q Q^T \Lambda Q^T Q \mathbf{y} < 0 Q Q^T \rightarrow \mathbf{y}^T \Lambda \mathbf{y} < 0$
- That is just $\lambda_1 y_1^2 + ... + \lambda_d y_d^2 < 0$
- But we know that each $\lambda_i >= 0$. Thus each $y_i^2 < 0$ but this is impossible. **This is a contradiction**.
- Therefore, it follows that $\mathbf{x}^T K \mathbf{x} \geq 0, \forall \mathbf{x} \in \mathbb{R}^d$ ■.

- **Proof**: If $K$ is symmetric and $\mathbf{x}^T K \mathbf{x} \geq 0 \forall \mathbf{x} \in \mathbb{R}^d$, then $K$ is positive semidefinite.
  - Assume $K$ is symmetric and $\mathbf{x}^T K \mathbf{x} \geq 0 \forall \mathbf{x} \in \mathbb{R}^d$.
  - By the definition, $K$ is positive semidefinite if $K$ is symmetric and has no negative eigenvalues.
  - Because $K$ is symmetric, it has exactly $n$ eigenvalues and thus $n$ eigenvectors. Let $\mathbf{v}$ be an eigenvector of $K$ and let $\lambda_v$ be its corresponding eigenvalue.
  - By assumption, $\mathbf{v}^T K \mathbf{v} \geq 0 \rightarrow \mathbf{v}^T \lambda_v \mathbf{v} \geq 0 \rightarrow \lambda_v \mathbf{v}^T \mathbf{v} \geq 0$.
  - Note that $\mathbf{v}^T \mathbf{v} \geq 0$. Which immediately follows that $\lambda_v \geq 0$.
  - Thus, for any arbitrary eigenvalue $\lambda$, it follows that $\lambda \geq 0$
  - Thus, $K$ is symmetric and has non-negative eigenvalues. Therefore, $K$ is positive semidefinite. ■

## 3.2 Kernel Properties

### 3.2.1 Constant Functions are Kernels

- **Method one**.
- Define $\phi : \mathbf{x} \to \sqrt{\alpha}$ where $\alpha > 0$.
- Then, $k(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$
- Then, $\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle = \sqrt{\alpha}\sqrt{\alpha} = \alpha$. $\blacksquare$
- **Method two**. We will prove that $K$ is positive semidefinite.
- For $K(\mathbf{x}, \mathbf{y}) = \alpha$, the corresponding Gram matrix has $K_{ij} = \alpha$ for all $i, j$. Thus $K$ is obviously symmetric, as it is a matrix with all elements $\alpha$. We can scale out $\alpha$ from $K$ to get $\alpha I_K$ where $I_K$ is the matrix of all ones.
- Since all elements are identical, $K$ has a rank of 1, and by the rank-nullity theorem, this means that $K$ has a nullity of $d - 1$.
- Therefore, there exists an eigenvalue $\lambda$ with geometric multiplicity $d - 1$.
- This divides the characteristic polynomial $\chi_K = \lambda^d - c\lambda^{d-1}$ where $c = tr(K) = d$ and $\chi_K = \lambda^{d-1}(\lambda - c)$. Therefore $\lambda = d$ and $\alpha \geq 0$, so every eigenvalue is $\geq 0$. $\blacksquare$.

### 3.2.2 Function-Mapping Product Kernels

- Define $\phi : \mathbf{x} \to f(\mathbf{x})$ for $f : \mathbb{R}^d \to \mathbb{R}$.
- $K(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$
- Then, $\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle = \langle f(\mathbf{x}), f(\mathbf{y}) \rangle = f(\mathbf{x}) \cdot f(\mathbf{y})$
- Thus, $K$ is a kernel. $\blacksquare$

### 3.2.3 Linear Combinations of Kernels

- Define the Gram matrix of kernel $K$ as $K = aK_1 + bK_1$, where $K_1, K_2$ are the Gram matrices of kernels $K_1$, $K_2$. We will show $K$ is positive semidefinite. Let $\mathbf{x} \in \mathbb{R}^d$
- $\mathbf{x}^T K \mathbf{x} = \mathbf{x}^T (aK_1 + bK_1)\mathbf{x}$.
- Then, $\mathbf{x}^T (aK_1 + bK_1)\mathbf{x} = (a\mathbf{x}^T K_1 \mathbf{x} + b\mathbf{x}^T K_2 \mathbf{x})$.
- By assumption, $\mathbf{x}^T K_1 \mathbf{x} \geq 0$ and $\mathbf{x}^T K_2 \mathbf{x} \geq 0$. And since $a, b > 0$, it follows that the sum $\mathbf{x}^T K \mathbf{x} \geq 0$.
- Therefore $K$ is positive semidefinite. Thus, $K$ is a valid kernel. $\blacksquare$

### 3.2.4 Nonzero Kernel Quotient

- Because $K_1(\mathbf{x}, \mathbf{x}) > 0$, let $\phi_1$ be the corresponding mapping of $K_1$. Then we have that $\sqrt{K_1(\mathbf{x}, \mathbf{x})} > 0$. Therefore we can define the following mapping:
- $\phi : \mathbf{x} \to \frac{\phi_1(\mathbf{x})}{\sqrt{\langle \phi_1(\mathbf{x}), \phi_1(\mathbf{x}) \rangle}}$
- Then, $K(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$
- And, $\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle = \langle \frac{\phi_1(\mathbf{x})}{\sqrt{\langle \phi_1(\mathbf{x}), \phi_1(\mathbf{x}) \rangle}}, \frac{\phi_1(\mathbf{y})}{\sqrt{\langle \phi_1(\mathbf{y}), \phi_1(\mathbf{y}) \rangle}} \rangle$
- Because $\mathbb{R}^d$ is a *real inner product space*, sesquilinearity implies that $\langle ax, by \rangle = ab\langle x, y \rangle$. Thus:
- $\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle = \frac{1}{\sqrt{\langle \phi_1(\mathbf{x}), \phi_1(\mathbf{x}) \rangle}} \frac{1}{\sqrt{\langle \phi_1(\mathbf{y}), \phi_1(\mathbf{y}) \rangle}} \langle \phi_1(\mathbf{x}), \phi_1(\mathbf{y}) \rangle$
- Which is just: $\frac{K_1(\mathbf{x}, \mathbf{y})}{\sqrt{K_1(\mathbf{x}, \mathbf{x})}\sqrt{K_1(\mathbf{y}, \mathbf{y})}}$ as required $\blacksquare$.