1. 20 Newsgroups predictions

1.1 The results of three different algorithms and baseline

a) SVM (support vector machine)
The accuracy of training set is 95.4%
The accuracy of test set is 66.3%
The train loss is 0.046
The test loss is 0.337

b) Random Forest Classifier
The accuracy of training set is 78.7%
The accuracy of test set is 59.7%
The train loss is 0.213
The test loss is 0.403

c) Multinomial Naïve Bayes
The accuracy of training set is 95.9%
The accuracy of test set is 70.1%
The train loss is 0.041
The test loss is 0.299

d) Bernoulli Naïve Bayes
The accuracy of training set is 59.9%
The accuracy of test set is 45.8%
The train loss is 0.401
The test loss is 0.542

1.2 Explain in your report how you picked the best hyperparameters.

a) SVM (support vector machine)
Penalty parameter C = 1.0
Kernel = 'linear'

b) Random Forest Classifier
'min_samples_leaf': 2
'n_estimators': 85
'max_depth': 45

c) Multinomial Naïve Bayes
alpha (smoothing parameter) = 0.012122845691382765
fit_prior = 'False', which means that a uniform prior is used.

I use GridSearchCV method to find the best hyperparameters. Given an estimator, I do exhaustive search over specified parameter values, which are optimized by cross-validated grid-search over the parameter grid. This method would choose the best one with high score and here I choose the "accuracy" for evaluation. After exhaustive searching, it returns best-estimator and best parameters.

For the algorithm SVM, the candidates for penalty parameters C are from 1 to 10.

For the algorithm Random Forest Classifier, the candidates for min_samples_leaf are [1,2,4], for n_estimators are [80,85,90,100], for max_depth are [30,35,40,45,50].
For the algorithm Multinomial Naïve Bayes, the candidates for alpha are arithmetic array from 0.0001 to 1.

1.3 Explain why you picked these 3 methods.

I choose these 3 methods based on the properties of text. When learning text classifiers, one has to deal with many features and few of them are irrelevant. In addition, documents vectors are sparse and most text categorization problems are linearly separable.

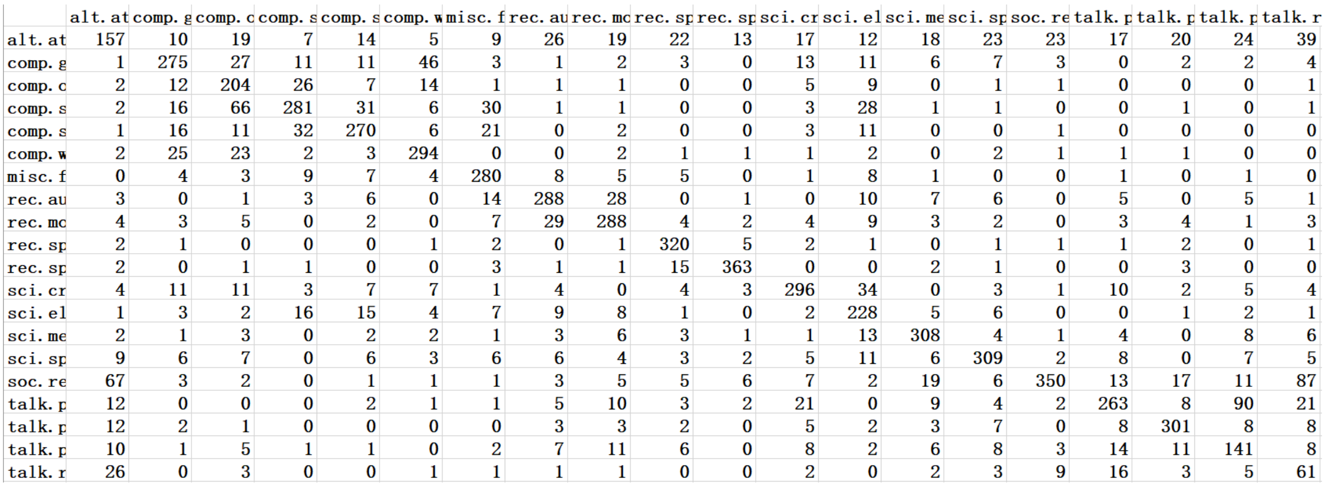
SVM uses overfitting protection and is well suited for problems with dense concepts and sparse instances. The idea of SVM is to find linear separators. So I choose SVM, which should perform well for text categorization.

Random Forest Classifier can handle discrete variables, and it is suitable for multi-classification problems. It can, to some extent, prevent over-fitting and increase stability.

Multinomial Naïve Bayes is also well suited for problems with sparse instances. Feature vectors represent the frequencies with which certain events have been generated by a multinomial distribution.

1.4 For your best classifier compute and show the confusion matrix.

The best classifier is Multinomial Naïve Bayes with alpha = 0.012122845691382765



	alt. at	comp. g	comp. c	comp. s	comp. s	comp. w	misc. f	rec. au	rec. mc	rec. sp	rec. sp	sci. cr	sci. el	sci. me	sci. sp	soc. re	talk. p	talk. p	talk. p	talk. r
alt. at	157	10	19	7	14	5	9	26	19	22	13	17	12	18	23	23	17	20	24	39
comp. g	1	275	27	11	11	46	3	1	2	3	0	13	11	6	7	3	0	2	2	4
comp. c	2	12	204	26	7	14	1	1	1	0	0	5	9	0	1	1	0	0	0	1
comp. s	2	16	66	281	31	6	30	1	1	0	0	3	28	1	1	0	0	1	0	1
comp. s	1	16	11	32	270	6	21	0	2	0	0	3	11	0	0	1	0	0	0	0
comp. w	2	25	23	2	3	294	0	0	2	1	1	1	2	0	2	1	1	1	0	0
misc. f	0	4	3	9	7	4	280	8	5	5	0	1	8	1	0	0	1	0	1	0
rec. au	3	0	1	3	6	0	14	288	28	0	1	0	10	7	6	0	5	0	5	1
rec. mc	4	3	5	0	2	0	7	29	288	4	2	4	9	3	2	0	3	4	1	3
rec. sp	2	1	0	0	0	1	2	0	1	320	5	2	1	0	1	1	1	2	0	1
rec. sp	2	0	1	1	0	0	3	1	1	15	363	0	0	2	1	0	0	3	0	0
sci. cr	4	11	11	3	7	7	1	4	0	4	3	296	34	0	3	1	10	2	5	4
sci. el	1	3	2	16	15	4	7	9	8	1	0	2	228	5	6	0	0	1	2	1
sci. me	2	1	3	0	2	2	1	3	6	3	1	1	13	308	4	1	4	0	8	6
sci. sp	9	6	7	0	6	3	6	6	4	3	2	5	11	6	309	2	8	0	7	5
soc. re	67	3	2	0	1	1	1	3	5	5	6	7	2	19	6	350	13	17	11	87
talk. p	12	0	0	0	2	1	1	5	10	3	2	21	0	9	4	2	263	8	90	21
talk. p	12	2	1	0	0	0	0	3	3	2	0	5	2	3	7	0	8	301	8	8
talk. p	10	1	5	1	1	0	2	7	11	6	0	8	2	6	8	3	14	11	141	8
talk. r	26	0	3	0	0	1	1	1	1	0	0	2	0	2	3	9	16	3	5	61

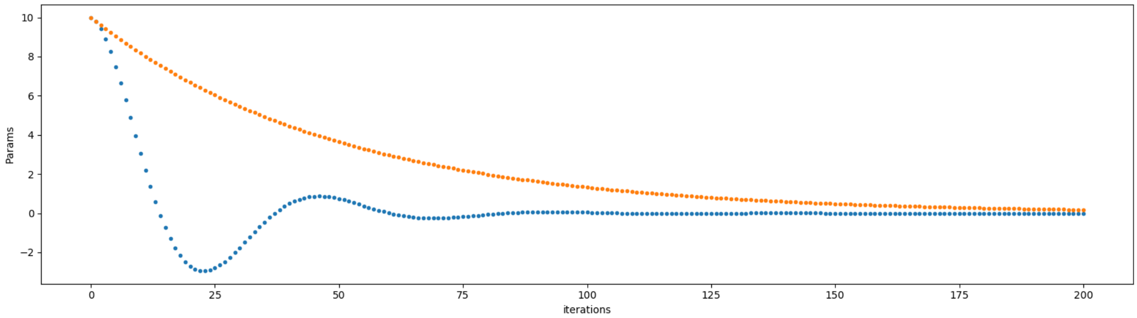
1.5 What were the two classes your classifier was most confused about?

The classes that the classifier was most confused about are class1—'alt.atheism' and class 16—'soc.religion.christian'.

I computed the ratio of mis-classified samples of class i to class j. $\alpha_{ij} = \frac{C_{ij}}{\sum_k C_{kj}}$. Then I added the ratio of two relevant classes. $I_{ij} = \alpha_{ij} + \alpha_{ji}$. Then I compared all the I's and select the largest one to get the most confused classes i and j.

2. Training SVM with SGD

2.1 Find the minimum of $f(w) = 0.01w^2$ with learning rate $\alpha = 1.0$ and $\beta = 0.0$ and 0.9 respectively.



2.3 Apply on 4-vs-9 digits on MNIST

Train two SVM models using gradient descent with a learning rate of α = 0:05, a penalty of C = 1:0, minibatch sizes of m = 100, and T = 500 total iterations.

First model: β = 0

(1) The training loss: 0.564242212128

(2) The test loss: 0.524289591156

(3) The classification accuracy on the training set is: 92.5%

(4) The classification accuracy on the test set is: 92.5%

Second model: β = 0.1

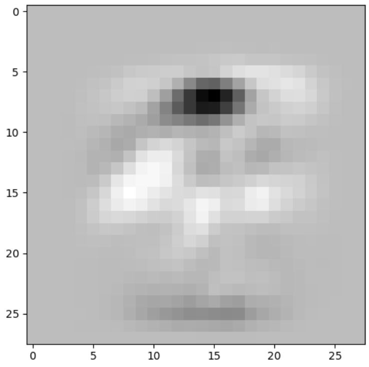
(5) The training loss: 0.546157589616

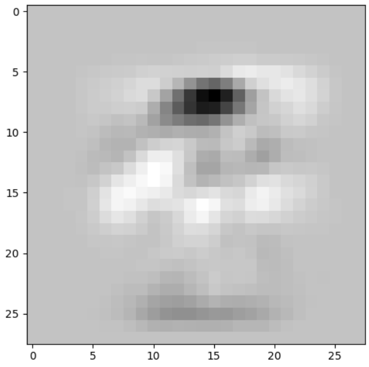
(6) The test loss: 0.480811285096

(7) The classification accuracy on the training set is: 90.5%

(8) The classification accuracy on the test set is: 90.4%

(9) Plot w's of two models as 28 × 28 images.





3 Kernels

3.1 Proof:

By definition, the necessity is obvious. We can prove the sufficiency by contradiction.
Since K is positive semidefinite matrix, according to spectral decomposition, we have $Q^T \Lambda \mathrm{Q}$ = K, where Λ is $\begin{bmatrix} \lambda_1 & \cdots & \\ \vdots & \ddots & \vdots \\ & \cdots & \lambda_n \end{bmatrix}$. All the eigen values are diagonal and Q is the eigen vector matrix.
Let's assume there exists x, s.t. $x^T K x < 0$. We have: $x^T K x = x^T Q^T \Lambda \mathrm{Q}\, x = y^T \Lambda \mathrm{y} = \sum_{i=1}^{n} \lambda_i y_i^2 \geq 0$, which contradicts with our assumption.

3.2 Solution:

(1) If $\mathrm{k(x,y)} = \alpha, K_{ij} = K\left(x^{(i)}, y^{(j)}\right) = \alpha.$ Let vector with dimension (1*d) be b=$[1\ 1 \ldots 1]$, we have: $\mathrm{K} = \alpha b^T b$. So $x^T K x = x^T \alpha b^T b x = \alpha (bx)^T bx = \alpha \parallel bx \parallel^2 \geq 0$. Thus K is a positive semidefinite. Thus $\mathrm{k(x,y)} = \alpha$ is a kernel.

(2) If $\mathrm{k(x,y) = f(x) * f(y)}$. Let $\phi(\mathrm{x}) = \mathrm{f(x)}, \phi(\mathrm{y}) = \mathrm{f(y)}$, we have $\mathrm{k(x,y) = f(x) * f(y)} = \phi(\mathrm{x}) * \phi(\mathrm{y})$.

(3) If $k_1(x,y)$ and $k_2(x,y)$ are kernels, we have $k_1(x,y) = \phi_1(x) * \phi_1(y)$, $k_2(x,y) = \phi_2(x) * \phi_2(y)$.

Let $\phi_1(x) = (\phi_1^1(x), \phi_1^2(x), \ldots.\phi_1^{N_1}(x))$, $\phi_2(x) = (\phi_2^1(x), \phi_2^2(x), \ldots.\phi_2^{N_2}(x))$

$$k(x,y) = ak_1(x,y) + bk_2(x,y) = a \sum_{i=1}^{N_1} \phi_1^i(x)\phi_1^i(y) + b \sum_{j=1}^{N_2} \phi_2^j(x)\phi_2^j(y)$$
$$= \left[\sqrt{a}\phi_1^1(x), \sqrt{a}\phi_1^2(x)..\sqrt{a}\phi_1^{N_1}(x), \sqrt{b}\phi_2^1(x), \sqrt{b}\phi_2^2(x)..\sqrt{b}\phi_2^{N_2}(x)\right] \cdot$$
$$\left[\sqrt{a}\phi_1^1(y), \sqrt{a}\phi_1^2(y)..\sqrt{a}\phi_1^{N_1}(y), \sqrt{b}\phi_2^1(y), \sqrt{b}\phi_2^2(y)..\sqrt{b}\phi_2^{N_2}(y)\right]^T$$

So we define $\phi(\mathrm{x}) = \left[\sqrt{a}\phi_1^1(x), \sqrt{a}\phi_1^2(x)..\sqrt{a}\phi_1^{N_1}(x), \sqrt{b}\phi_2^1(x), \sqrt{b}\phi_2^2(x)..\sqrt{b}\phi_2^{N_2}(x)\right]$, we get:

$$\mathrm{k(x,y)} = \phi(\mathrm{x}) * \phi(\mathrm{y}).$$

Thus $(x,y) = ak_1(x,y) + bk_2(x,y)\ is\ a\ kernel.$

(4) If $k_1(x,y)$ is a kernel, we have $k_1(x,y) = \phi_1(x) * \phi_1(y)$.

We define $\phi(\mathrm{x}) = \frac{\phi_1(x)}{\|\phi_1(x)\|}$, so $\mathrm{k(x,y)} = \frac{k_1(x,y)}{\sqrt{k_1(x,x)}\sqrt{k_1(y,y)}} = \frac{\phi_1(x)}{\|\phi_1(x)\|} * \frac{\phi_1(y)}{\|\phi_1(y)\|} = \phi(\mathrm{x}) * \phi(\mathrm{y})$.

Thus $\mathrm{k(x,y)}$ is a kernel.