CSC411 Fall 2017
Assignment 3 Report

Tianbao Li

2017/12/04

1 20 Newsgroups prediction

1.1 Data summary

Here is the detailed information of the dataset:

- data set: The 20 newsgroups text
- data amount: 11314
- feature amount: 101631
- data representation: tf-idf
- baseline: Bernoulli Naive-Bayes classifier

1.2 Model training

To train model to outperform the baseline, here we useopen-souece code of scikit-learn. Here, I picked several different algorithms and trained their hyper-parameters with k-fold validation. The detailed better hyper-parameters, train losses (0-1 loss), test losses (0-1 loss) are shown in Table 1. From the result of the validation losses, the best three models are Neural Networks, Multinomial Naive Bayes, Logist Regression, and has the same best results in test dataset.

Model	Hyper-pramater	Train loss	test loss
BernoulliNB		0.598727240587	0.457912904939
MultinomialNB	$\alpha = 0.01$	0.958900477285	0.700212426978
Logist Regression	$C = 500$	0.974721583878	0.683483802443
SGD	$\alpha = 0.0001$	0.962877850451	0.671136484334
SVM	$C = 1$	0.895704436981	0.67750929368
KNN	$K = 1$	0.973749337104	0.113382899628
Decision Tree	$K = 601$	0.974721583878	0.4026818906
Neural Networks	$\alpha = 0.0001$	0.947587060279	0.712294211365

Table 1: Model comparison

1.3 Hyper-parameters training

To train models by choosing better-fitting parameters, I splited training data by KFold and run cross valiadation by cross_val_score.

Here, taking multinomial Naive Bayes as an example. Hyper-parameter α is used for smoothing. I picked several possible α value and compared by validation loss, then chose a better α.

```
splits = 5
kf = KFold(splits, shuffle = True, random_state = 0)
As = [1e-10, 1e-8, 1e-6, 1e-4, 1e-2, 0.1, 0.5, 1.0, 2, 5, 10]
scores = []
for a in As:
    model = MultinomialNB(alpha = a)
    score = cross_val_score(model, tfidf_train, train_labels, cv = kf)
    scores.append(np.mean(score))
opt_A_index = int(np.argmax(scores))
opt_A = As[opt_A_index]
```

1.4 Model selection

For the three well-working models, they have their advantages for solving this problem.

- **Neural Networks** can work well for complex models, especially when it comes deeper.
- **Multinomial Naive Bayes** implements the naive Bayes algorithm for multinomially distributed data, text classification usually holds such structure.
- **Logistic Regression** is famous for linear classification, which the newsgroup data has similar distribution.

1.5 Class confusion

For the best classifier, Neural Networks, the confusion matrix among 20 groups is shown in Table 2. The two classed that Neural Networks classifier confuses about are class 16 and 18.

163	6	5	1	1	0	0	4	4	7	5	3	2	11	7	20	7	28	20	38
1	287	26	15	6	49	2	1	1	2	2	8	11	8	11	3	1	1	1	4
3	19	241	37	10	31	2	2	1	0	0	5	15	1	2	1	2	0	0	2
1	9	39	256	22	7	18	1	1	0	0	2	26	1	0	0	1	2	0	2
1	9	12	32	284	5	14	1	1	0	0	2	11	2	1	0	2	0	0	0
0	16	12	4	3	278	0	0	0	0	0	1	2	0	0	0	0	0	1	0
0	4	3	10	8	4	305	10	5	3	0	1	10	3	3	0	1	0	0	1
15	7	17	9	21	5	20	322	32	15	12	19	21	22	23	14	19	8	13	10
3	5	4	0	4	1	6	16	312	3	1	2	10	7	5	1	4	7	1	2
4	2	1	0	1	2	3	3	3	334	11	8	2	0	1	3	2	3	0	1
2	0	0	1	0	0	0	1	0	14	354	0	0	4	1	0	0	1	2	0
2	6	6	2	2	4	1	2	1	1	0	289	13	0	0	0	7	3	1	1
4	9	2	24	21	5	8	15	18	3	1	14	245	10	9	1	2	1	3	1
2	0	2	0	0	0	0	1	3	3	1	2	8	300	5	2	2	2	5	6
9	7	11	1	1	3	3	3	6	1	1	3	9	3	302	4	7	0	8	6
53	1	3	0	0	1	1	1	2	4	4	3	2	7	6	321	9	13	5	68
9	1	2	0	1	0	3	2	3	2	2	19	1	5	4	0	258	6	89	20
9	1	2	0	0	0	1	2	0	1	2	3	3	3	1	0	8	287	6	6
7	0	4	0	0	0	2	5	4	4	2	10	2	8	13	5	18	13	149	11
31	0	2	0	0	0	1	4	1	0	1	2	0	1	0	23	14	1	6	72

Table 2: Confusion matrix

2 Training SVM with SGD

2.1 SGD with momentum

Parameters:

- function: $f(w) = 0.01w^2$
- initianlization: $w_0 = 10.0$
- learning rate: $\alpha = 1.0$
- momentum: $\beta_1 = 0.0, \beta_2 = 0.9$

Changes for w_t is shown as Figure 1.

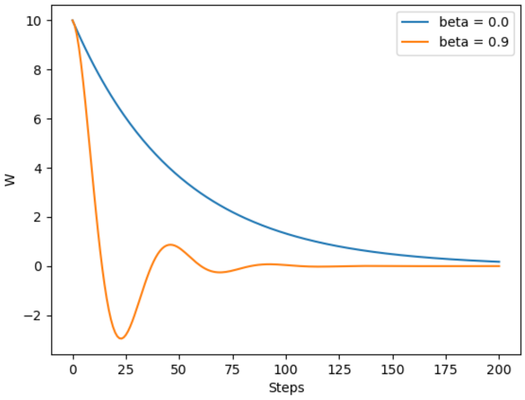

Figure 1: w_t for $\beta = 0.0$ and 0.9

2.2 Training SVM

Given the formula of SVM, the gradient during training can be shown as

$$\frac{\partial \ell}{\partial w} = \begin{cases} -C * y^{(i)}\mathbf{x}^{(i)} + w = & y^{(i)}(w^{\mathrm{T}}\mathbf{x}^{(i)} + b) < 1 \\ 0 = & y^{(i)}(w^{\mathrm{T}}\mathbf{x}^{(i)} + b) \geq 1 \end{cases} \tag{1}$$

2.3 Apply on 4-vs-9 digits on MNIST

Here, I trained SVM on MNIST. The trained models for $\beta = 0$ is reported as follows:

- β: 0.0
- optimal hyper-parameter: $C = 1$
- training loss: 0.748119588312
- test loss: 0.720131532536
- training accuracy: 0.927437641723
- test accuracy: 0.92636924193

The trained models for $\beta = 0.0$ is reported as follows:

- β: 0.1

- optimal hyper-parameter: $C = 1$
- training loss: 0.483890097883
- test loss: 0.500220862748
- training accuracy: 0.940770975057
- test accuracy: 0.937250634748

The w figures for $\beta = 0.0$ and 0.1 are shown in Figure 2.

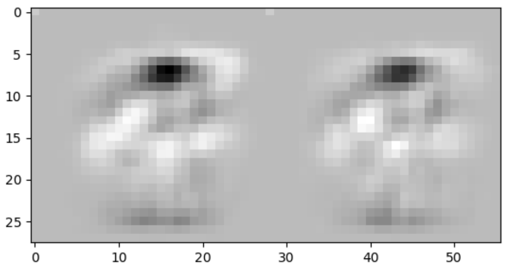

Figure 2: w figure for $\beta = 0.0$ (left) and 0.1 (right)

3 Kernels

3.1 Positive semidefinite and quadratic form

Proof that a symmetric matrix $K \in \mathbb{R}^{d\times d}$ is positive semidefinite iff for all vectors $\mathbf{x} \in \mathbb{R}^d$ we have $\mathbf{x}^{\mathrm{T}} K \mathbf{x} \geq 0$.

Proof. Necessity:

$$
\begin{aligned}
&\because \forall \mathbf{x} \in \mathbb{R}^d, \mathbf{x}^{\mathrm{T}} K \mathbf{x} \geq 0 \\
&\therefore \forall v \text{ as eigenvector of } K, v^{\mathrm{T}} K v = v^{\mathrm{T}} \lambda v \geq 0 \\
&\therefore \lambda \geq 0 \ (\lambda \text{ corresponds to } v) \\
&\therefore \forall \lambda \geq 0 \text{ as eigenvalue of } K \\
&\therefore K \text{ has no negative eigenvalue} \\
&\because K \text{ is symmetric} \\
&\therefore K \text{ is positive semidefinite}
\end{aligned}
$$

Sufficiency:

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

$\therefore K$ has orthogonal eigenvectors for different eigenvalues

K can be diagonalized

$\therefore K$ can be represented as $K = Q\Lambda Q^{-1} = Q\Lambda Q^{\mathrm{T}}$

$Q = [v_1, v_2, \dots, v_d]^{\mathrm{T}}$: eigenvector matrix

Λ : diagonal matrix

$\because v_1, v_2, \dots, v_d$ are diagonal

$\therefore \forall \mathbf{x} \in R^d = (c_1 v_1 + c_2 c_2 + \cdots + c_d v_d)$

$\therefore \mathbf{x}^{\mathrm{T}} A \mathbf{x} = \mathbf{x}^{\mathrm{T}} Q \Lambda Q^{\mathrm{T}} \mathbf{x}$

$$= (c_1 v_1 + c_2 c_2 + \cdots + c_d v_d)\begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_d \end{pmatrix}\begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \cdots & \\ & & & \lambda_d \end{pmatrix}(v_1, v_2, \dots, v_d)(c_1 v_1 + c_2 c_2 + \cdots + c_d v_d)^{\mathrm{T}}$$

$$= (c_1 ||v_1||^2 + c_2 ||v_2||^2 + \cdots + (c_d ||v_d||^2))\begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \cdots & \\ & & & \lambda_d \end{pmatrix}(c_1 ||v_1||^2 + c_2 ||v_2||^2 + \cdots + (c_d ||v_d||^2))^{\mathrm{T}}$$

$$= \lambda_1 c_1^2 ||v_1||^4 + \lambda_2 c_2^2 ||v_2||^4 + \cdots + \lambda_d c_d^2 ||v_d||^4 \geq 0$$

$\therefore \mathbf{x}^{\mathrm{T}} K \mathbf{x} \geq 0$

□

3.2 Kernel properties

3.2.1 $k(\mathbf{x}, \mathbf{y}) = \alpha$

Proof.

$$\begin{aligned} k(\mathbf{x}, \mathbf{y}) &= \alpha \\ &= \sqrt{\alpha}\sqrt{\alpha} \\ &=< \phi(\mathbf{x}), \phi(\mathbf{y}) > \\ \phi(x) &= \sqrt{\alpha} \\ &\therefore K \text{ is kernel} \end{aligned}$$

□

3.2.2 $k(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) \cdot f(\mathbf{y})$

Proof.

$$\begin{aligned} k(\mathbf{x}, \mathbf{y}) &= f(\mathbf{x}) \cdot f(\mathbf{y}) \\ &=< f(\mathbf{x}), f(\mathbf{y}) > \\ &=< \phi(\mathbf{x}), \phi(\mathbf{y}) > \\ \phi(x) &= f(x) \\ &\therefore K \text{ is kernel} \end{aligned}$$

□

3.2.3 $k(\mathbf{x}, \mathbf{y}) = a \cdot k_1(\mathbf{x}, \mathbf{y}) + b \cdot k_2(\mathbf{x}, \mathbf{y})$

Proof.

$$\begin{aligned}
&K_{1i,j} = K_1(\mathbf{x}^{(i)}, \mathbf{y}^{(j)}) \\
&K_{2i,j} = K_2(\mathbf{x}^{(i)}, \mathbf{y}^{(j)}) \\
\therefore &K_{i,j} = a \cdot K_1(\mathbf{x}^{(i)}, \mathbf{y}^{(j)}) + b \cdot K_2(\mathbf{x}^{(i)}, \mathbf{y}^{(j)}) \\
&\quad\quad = a \cdot K_{1i,j} + b \cdot K_{2i,j} \\
&v^{\mathrm{T}} K_1 v = c_1 > 0 \quad K_1 v = (v^{\mathrm{T}})^{-1} c_1 \\
&v^{\mathrm{T}} K_2 v = c_2 > 0 \quad K_2 v = (v^{\mathrm{T}})^{-1} c_2 \\
\therefore &a K_1 v + b K_2 v = (v^{\mathrm{T}})^{-1} c_1 a + (v^{\mathrm{T}})^{-1} c_2 b \\
&(a K_1 + b K_2) v = (v^{\mathrm{T}})^{-1} c_1 a + (v^{\mathrm{T}})^{-1} c_2 b \\
&v^{\mathrm{T}} (a K_1 + b K_2) v = c_1 a + c_2 b \\
&v^{\mathrm{T}} K v = c_1 a + c_2 b \geq 0 \\
\therefore &K \text{ is kernel}
\end{aligned}$$

□

3.2.4 $k(\mathbf{x}, \mathbf{y}) = \frac{k_1(\mathbf{x}, \mathbf{y})}{\sqrt{k_1(\mathbf{x}, \mathbf{x}) k_1(\mathbf{y}, \mathbf{y})}}$

Proof.

$$\begin{aligned}
k(\mathbf{x}, \mathbf{y}) &= \frac{k_1(\mathbf{x}, \mathbf{y})}{\sqrt{k_1(\mathbf{x}, \mathbf{x}) k_1(\mathbf{y}, \mathbf{y})}} \\
&= \frac{< \phi(\mathbf{x}), \phi(\mathbf{y}) >}{\sqrt{< \phi(\mathbf{x}), \phi(\mathbf{x}) >} \sqrt{< \phi(\mathbf{y}), \phi(\mathbf{y}) >}} \\
&= \frac{< \phi(\mathbf{x}), \phi(\mathbf{y}) >}{|\phi(\mathbf{x})||\phi(\mathbf{y})|} \\
&=< \frac{\phi(\mathbf{x})}{|\phi(\mathbf{x})|}, \frac{\phi(\mathbf{y})}{|\phi(\mathbf{y})|} > \\
\therefore \phi\text{'}(x) &= \frac{\phi(x)}{|\phi(x)|} \\
\therefore K &\text{ is kernel}
\end{aligned}$$

□