Report: Classification

Alex Berke and Matt Patenaude

1) Build the vector representation of the collection of documents for each of the provided features lists.

Done!

2) Run both the Naive Bayesian and Rocchio classifiers using training1.dat and each of the provided features sets. First, re-classify the documents in the training set and compute the classification error (% of misclassified documents). Then, classify the documents in test.dat and compute the classification error on this test set. (1 table) Report the results obtained using each of the features lists for both classifiers. Be sure to fill out the appropriate information in results.txt.

MNB performed better than Rocchio, but only really performed respectably with the training data, and feature set 1. For the most part, classification accuracy was considerably better with feature set 1 than with feature set 2.

Feature Set	Algorithm	Data Set	Error
features1.dat	MNB	training1.dat	10.45%
features1.dat	MNB	test.dat	39.46%
features1.dat	Rocchio	training1.dat	40.65%
features1.dat	Rocchio	test.dat	57.81%
features2.dat	MNB	training1.dat	37.75%
features2.dat	MNB	test.dat	45.05%
features2.dat	Rocchio	training1.dat	52.725%
features2.dat	Rocchio	test.dat	57.07%

3) Look at the results obtained in the previous two points, inspect the provided features lists and explain their impact on the classification accuracy.

Feature set 1 contains predominately long, unique, and highly technical and domain-specific terms, which aids in distinguishing documents of a highly technical nature among a number of closely-related categories. Conversely, feature set 2 is fairly generic, and contains what appears to be a more random sampling of dictionary terms. Because the terms in feature set 2 appear with closer-to-average frequency across the document corpus than those in feature set 1, the classification output using this feature set is less useful because it doesn't have much distinguishing power. Furthermore, feature set 1 is several orders of magnitude larger than feature set 2, which allows for much more fine-grained training of documents.

4) Repeat Point 2, using training2.dat, again including the information in results.txt.

As before, MNB performed better than Rocchio, and feature set 1 proved more discriminatory than feature set 2. However, on a whole, the data in training set 2 appeared to be less effective.

Feature Set	Algorithm	Data Set	Error
features1.dat	MNB	training2.dat	17.425%
features1.dat	MNB	test.dat	87.84%
features1.dat	Rocchio	training2.dat	55.45%
features1.dat	Rocchio	test.dat	84.92%
features2.dat	MNB	training2.dat	74.35%
features2.dat	MNB	test.dat	91.73%
features2.dat	Rocchio	training2.dat	76.10%
features2.dat	Rocchio	test.dat	89.97%

5) Inspect training2.dat, then look at the results obtained in the previous point: can you explain them?

An examination and comparison of training1.dat and training2.dat shows that, although both data sets contain the same set of documents, they differ in how they map those documents to categories. Specifically, documents that training1.dat puts into the same category, training2.dat scatters into different categories (e.g., training1.dat puts 181, 724, 1285, 1774, and 3084 all into category 0, while training2.dat puts them into categories 10, 4, 1, 3, and 6, respectively). This hinders the training process greatly by inhibiting the classification algorithms from finding commonalities among actually similar documents. Further, training1.dat and training2.dat differ wildly in categorization "opinions": the two training sets only agree on 426 of 4,000 categorizations. Presumably, the categorizations in test.dat are more consistent with those found in training1.dat.

6) Build your own list of features, named featuresADV.dat, with the goal of using it for classifying the document collection. Describe how you built it and the rationale behind your choices.

The classification process appears to work more effectively when the feature set more neatly "partitions" the set of documents into categories — i.e., the feature set is a union of terms that, for each category, occur with high frequency in that category, but low frequency in any other category. Therefore, to create our featuresADV.dat, we began with the classifications in training1.dat (to assign categories to documents), and identified, for each category, the top 10,000 terms weighted by incidence in the current category vs. incidence in other categories. We then computed the union of these terms, and the result was a feature list of over 58,000 unique terms. For further detail on how this works, see build_featuresADV.py.

7) Repeat Point 2, using training1.dat and featuresADV.dat. (1 table) Compare your results with those obtained with the provided features lists: did you improve the previous classification results? Report the confusion matrix for both MNB and Rocchio.

Our featuresADV.dat demonstrated some classification improvement, but for the most part fell somewhat squarely between features1.dat and features2.dat in terms of reliability on training data. Worthy of note, though, is that the results of test.dat classification were more accurate under both algorithms (3% better for MNB, 10% better for Rocchio) using featuresADV.dat than either features1.dat or features2.dat.

Feature Set	Algorithm	Data Set	Error
featuresADV.dat	MNB	training1.dat	19.025%
featuresADV.dat	MNB	test.dat	36.84%
featuresADV.dat	Rocchio	training1.dat	43.425%
featuresADV.dat	Rocchio	test.dat	47.82%

The confusion matrices for our results looked a bit like this:

MNB with training1.dat

		Desired Category										
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Actual Category	**0**	100	3	0	1	0	2	0	1	3	0	0
	1	2	524	9	15	9	7	7	12	16	0	24
	2	1	7	139	6	0	1	1	3	4	1	2
	3	6	17	13	345	2	6	4	8	10	1	18
	4	3	28	13	14	525	14	7	19	9	0	4
	5	0	0	0	0	0	24	0	0	0	0	0
	6	1	7	0	1	0	6	246	4	4	1	11
	7	5	7	8	8	1	7	10	155	5	1	3
	8	4	23	8	10	1	18	4	7	307	7	28
	9	3	3	0	7	1	15	1	2	21	148	8
	10	4	30	3	31	3	20	30	13	25	8	726

MNB with test.dat

		Desired Category										
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Actual Category	**0**	128	13	2	0	1	14	2	3	4	0	3
	1	13	1089	73	158	96	25	58	39	98	15	135
	2	10	18	232	18	5	22	1	20	7	0	1
	3	6	71	22	637	13	15	9	18	50	12	101
	4	7	130	37	90	1124	35	51	86	36	6	17
	5	0	0	0	0	0	10	0	0	0	0	0
	6	1	42	1	3	5	31	533	26	13	2	48
	7	44	25	47	38	3	35	37	303	32	13	26
	8	10	67	32	42	15	104	35	27	528	101	88
	9	13	6	5	16	3	46	6	7	102	219	45
	10	45	66	10	151	11	115	125	70	109	74	1513

Rocchio with training1.dat

		Desired Category										
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Actual Category	**0**	49	5	13	7	0	3	9	37	7	2	2
	1	26	455	43	78	45	4	46	41	69	5	125
	2	3	5	66	6	1	0	3	9	12	0	1
	3	7	28	10	228	6	6	6	17	12	8	63
	4	6	49	19	31	475	16	34	24	14	0	18
	5	9	10	17	14	2	64	6	9	17	9	50
	6	2	17	1	5	0	5	161	13	7	2	34
	7	12	2	2	8	0	3	7	50	2	1	8
	8	2	33	16	21	6	6	10	9	179	25	63
	9	7	19	4	13	2	10	4	3	56	105	29
	10	6	26	2	27	5	3	24	12	29	10	431

Rocchio with test.dat

		Desired Category										
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Actual Category	**0**	80	17	32	19	2	6	19	98	16	5	7
	1	37	1057	119	278	159	39	133	94	179	18	322
	2	13	15	133	13	5	6	3	18	10	2	7
	3	18	71	14	537	20	18	12	37	40	21	152
	4	11	136	51	81	1048	29	68	80	31	5	36
	5	20	32	37	22	15	154	52	25	57	19	126
	6	5	35	7	11	4	26	434	43	29	7	107
	7	38	12	17	24	0	15	20	117	21	4	25
	8	14	72	34	72	12	65	35	24	380	63	134
	9	20	35	10	27	6	48	18	18	156	273	56
	10	21	45	7	69	5	46	63	63	60	25	1005