Chapter 14 Vector space classification

The basic hypothesis in using the vector space model for classification is the contiguity hypothesis.

Contiguity hypothesis. Document in the same class form a contiguous region, and regions of different classes do not overlap.

1. Rocchio classification

Rocchio classification uses centroids to define the boundaries. The centroid of a class c is computed as the vector average or center of mass of its members:

$$\overrightarrow{\mu}(c) = \frac{1}{|D_c|} \sum_{d \in D} \overrightarrow{v}(d) \tag{1}$$

where D_c is the set of documents in $\mathbb{D}$ whose class is $c : D_c = d :< d, c >\in \mathbb{D}$. $\overrightarrow{v}(d)$, the normalized vector of d, can be calculated by,

$$\begin{aligned} wf - idf_{t,d} &= wf_{t,d} \times idf_t \\ wf_{td} &= \begin{cases} 1 + logtf_{td} & if\, tf_{t,d} > 0 \\ 0 & otherwise \end{cases} \end{aligned} \tag{2}$$

Table 14.1 shows the tf–idf vector representations of the five documents. The two class centroids are $\mu_c = 1/3 \cdot (\overrightarrow{d}_1 + \overrightarrow{d}_2 + \overrightarrow{d}_3)$ and $\mu_{\overline{c}} = 1/1 \cdot (\overrightarrow{d}_4)$. The Euclidean distances (can also be alternatived by cosine similarity) of the test document from the centroids are $|\mu_c - \overrightarrow{d}_5| \approx 1.15$ and $|\mu_{\overline{c}} - \overrightarrow{d}_5| = 0.0$. Thus, Rocchio assigns d_5 to $\overline{c}$.

Table 14.1 Vectors and class centroids for the data in Table 13.1.

	term weights					
vector	Chinese	Japan	Tokyo	Macao	Beijing	Shanghai
$\vec{d}_1$	0	0	0	0	1.0	0
$\vec{d}_2$	0	0	0	0	0	1.0
$\vec{d}_3$	0	0	0	1.0	0	0
$\vec{d}_4$	0	0.71	0.71	0	0	0
$\vec{d}_5$	0	0.71	0.71	0	0	0
$\vec{\mu}_c$	0	0	0	0.33	0.33	0.33
$\vec{\mu}_{\overline{c}}$	0	0.71	0.71	0	0	0

2. k nearest neighbor

Unlike Rocchio, k nearest neighbor or kNN classification determines the decision boundary locally. For 1NN, we assign each Document to the class of its closest neighbor. For kNN, we assign each Document to the majority class of its k closest neighbors where k is a parameter.

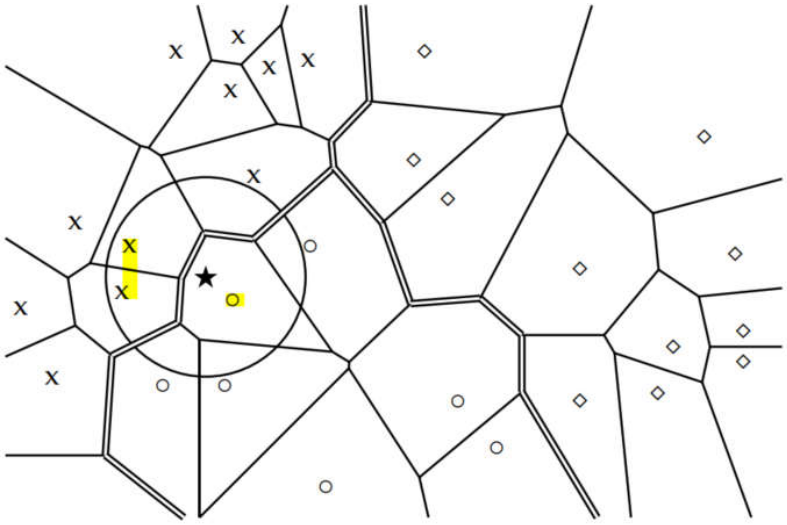

Figure 14.6 Voronoi tessellation and decision boundaries (double lines) in 1NN classification. The three classes are: X, circle and diamond.

There is a probabilistic version of this kNN classification algorithm. We can estimate the probability of membership in class c as the proportion of the k nearest neighbors in c. Figure 14.6 gives an example for $k = 3$. Probability estimates for class membership of the star are P(circle class|star) = 1/3, P(X class|star) = 2/3, and P(diamond class|star) = 0. The 3NN estimate (P1(circle class|star) = 1/3) and the 1NN estimate (P1(circle class|star) = 1) differ with 3NN preferring the X class and 1NN preferring the circle class.

We can also weight the “votes” of the k nearest neighbors by their cosine similarity. In this scheme, a class’s score is computed as:

$$score(c,d) = \sum_{d' \in S_k} I_c(d') cos(\overrightarrow{(}d'), \overrightarrow{(}d)) \tag{3}$$

where S_k is the set of d's k nearest neighbors and $I_c(d') = 1$ iff d' is in class c and 0 otherwise. We then assign the Document to the class with the highest score.

3. Linear classifiers

$$\overrightarrow{w}^T \overrightarrow{x} = b \tag{4}$$

The assignment criterion then is: assign to c if $\overrightarrow{w}^T \overrightarrow{x} > b$ and to $\overline{c}$ if $\overrightarrow{w}^T \overrightarrow{x} \leq b$.

The two learning methods – Naive Bayes and Rocchio – are instances of linear classifiers, the perhaps most important group of text classifiers.

For Rocchio, observe that a vector $\overrightarrow{x}$ is on the decision boundary if it has equal distance to the two class centroids:

$$\begin{gathered} |\overrightarrow{\mu}(c_1) - \overrightarrow{x}| = |\overrightarrow{\mu}(c_2) - \overrightarrow{x}| \\ |\overrightarrow{\mu}(c_1)|^2 - 2\overrightarrow{\mu}(c_1)\overrightarrow{x} + |\overrightarrow{x}|^2 = |\overrightarrow{\mu}(c_2)|^2 - 2\overrightarrow{\mu}(c_2)\overrightarrow{x} + |\overrightarrow{x}|^2 \\ (\overrightarrow{\mu}(c_1) - \overrightarrow{\mu}(c_2))\overrightarrow{x} = 0.5 \cdot (|\overrightarrow{\mu}(c_1)|^2 - |\overrightarrow{\mu}(c_2)|^2) \end{gathered} \tag{5}$$

Tuse, Rocchio, a linear classifier with normal vector $\overrightarrow{w} = \overrightarrow{\mu}(c_1) - \overrightarrow{\mu}(c_2)$ and $b = 0.5 \cdot (|\overrightarrow{\mu}(c_1)|^2 - |\overrightarrow{\mu}(c_2)|^2)$.

For Naive Bayes,

$$\hat{P}(c|d) \propto \hat{P}(c) \prod_{1 \leq k \leq n_d} \hat{P}(t_k|c) \tag{6}$$

Consequently,

$$log \frac{\hat{P}(c|d)}{\hat{P}(\overline{c}|d)} = log \frac{\hat{P}(c)}{\hat{P}(\overline{c})} + \sum_{1 \leq k \leq n_d} log \frac{\hat{P}(t_k|c)}{\hat{P}(t_k|\overline{c})} \tag{7}$$

Therefore, $w_i = log \hat{P}(t_i|c)/\hat{P}(t_i|\overline{c})$, x_i is the number of occurrences of t_i in d, and $b = -log[\hat{P}(c)/\hat{P}(\overline{c})]$.

Table 14.4 A linear classifier. The dimensions t_i and parameters w_i of a linear classifier for the class *interest* (as in interest rate) in Reuters-21578. The threshold is $b = 0$. Terms like dlr and world have negative weights because they are indicators for the competing class *currency*.

t_i	w_i	d_{1i}	d_{2i}	t_i	w_i	d_{1i}	d_{2i}
prime	0.70	0	1	dlrs	-0.71	1	1
rate	0.67	1	0	world	-0.35	1	0
interest	0.63	0	0	sees	-0.33	0	0
rates	0.60	0	0	year	-0.25	0	0
discount	0.46	1	0	group	-0.24	0	0
bundesbank	0.43	0	0	dlr	-0.24	0	0

Table 14.4 defines a linear classifier for the category interest in Reuters-21578. We assign document $\overrightarrow{d}_1$ "rate discount dlrs world" to interest since $\overrightarrow{w}^T \overrightarrow{d}_1 = 0.67 \times 1 + 0.46 \times 1 + (-0.71) \times 1 + (-0.35) \times = 0.07 > 0 = b$. We assign $\overrightarrow{d}_2$ "prime dlrs" to the complement class (not in interest) because $\overrightarrow{w}^T \overrightarrow{d}_2 = -0.01 \leq b$. For simplicity, we assume a simple binary vector representation in this example: 1 for occurring terms, 0 for nonoccurring terms.

4. The bias-variance tradeoff

For classification, to find a classifier γ such that, averaged over documents d, $\gamma(d)$ is as close as possible to the true probability $P(c|d)$. We measure this using mean squared error:

$$MSE(\gamma) = E_d[\gamma(d) - P(c|d)]^2 \tag{8}$$

where E_d is the expectation with respect to $p(d)$.

Based on Equation (8), we adopt as our goal to find a function Γ that, average over labeled training sets $\mathbb{D}$, learns classifiers γ with minimal MSE:

$$learning - error(\Gamma) = E_D[MSE(\Gamma(\mathbb{D}))] \tag{9}$$

where $E_{\mathbb{D}}$ is the expectation over labeled training sets.

Besides,

$$E[x - \alpha] = Ex^2 - 2Ex\alpha + \alpha^2 = (Ex)^2 - 2Ex\alpha + \alpha^2 + Ex^2 - 2(Ex)^2 + (Ex)^2 = [Ex - \alpha]^2 + Ex^2 - E2x(Ex) + E(Ex)^2 = [Ex - \alpha]^2 + E[x - Ex]^2$$

Therefore,

$$
\begin{aligned}
learning-error(\Gamma) =& E_D[MSE(\Gamma(\mathbb{D}))] = E_{\mathbb{D}}E_d[\Gamma_{\mathbb{D}}(d) - P(c|d)]^2 = E_d E_{\mathbb{D}}[\Gamma_{\mathbb{D}}(d) - P(c|d)]^2 \\
=& E_d[[E_{\mathbb{D}}\Gamma_{\mathbb{D}}(d) - P(c|d)]^2 + E_{\mathbb{D}}[\Gamma_{\mathbb{D}}(d) - E_{\mathbb{D}}\Gamma_{\mathbb{D}}(d)]^2] \\
=& E_d[bias(\Gamma, d) + variance(\Gamma, d)] \\
bias(\Gamma, d) =& [P(c|d) - E_{\mathbb{D}}\Gamma_{\mathbb{D}}(d)]^2 \\
variance(\Gamma, d) =& E_{\mathbb{D}}[\Gamma_{\mathbb{D}}(d) - E_{\mathbb{D}}\Gamma_{\mathbb{D}}(d)]^2
\end{aligned} \tag{10}
$$

Bias is small if (i) the classifiers are consistently right or (ii) different training sets cause errors on different documents or (iii) different training sets cause positive and negative errors on the same documents, but that average out to close to 0.

Variance is large if different training sets D give rise to very different classifiers $\Gamma_{\mathbb{D}}$. It is small if the training set has a minor effect on the classification decisions $\Gamma_{\mathbb{D}}$ makes, be they correct or incorrect.

The learning error has two components, bias, and variance, which in general cannot be minimized simultaneously. Instead, we have to weigh the respective merits of bias and variance in our application and choose accordingly. This tradeoff is called the bias-variance tradeoff.

Conclusion

- To respecting contiguity, the classes in Rocchio classification must be approximate spheres with similar radii.
- Unweighted and unnormalized counts should not be used in vector space classification.
- Rocchio classification is simple and efficient but inaccurate if classes are not approximately spheres with similar radii.
- If the training set is large, then kNN can handle nonspherical and other complex classes better than Rocchio.
- A large number of text classifiers can be viewed as linear classifiers – classifiers that classify based on a simple linear combination of the features.
- Because of the bias-variance tradeoff, more complex nonlinear models are not systematically better than linear models.
- Nonlinear models have more parameters to fit on a limited amount of training data and are more likely to make mistakes for small and noisy data sets.
- For k nearest neighbor, it is desirable for k to be odd to make ties less likely. k = 3 and k = 5 are common choices, but much larger values, between 50 and 100, are also used.
- For k nearest neighbor, weighting by similarities is often more accurate than simple voting
- The error of 1NN is asymptotically (as the training set increases) bounded by twice the Bayes error rate. That is, if the optimal classifier has an error rate of x, then 1NN has an asymptotic error rate of $2x$. This is due to the effect of noise.
- Naive Bayes and Rocchio – are instances of linear classifiers, the perhaps most important group of text classifiers.
- Bias is small if (i) the classifiers are consistently right or (ii) different training sets cause errors on different documents, or (iii) different training sets cause positive and negative errors on the same documents, but that average out to close to 0.
- We can also think of variance as the model complexity or, equivalently, memory capacity of the learning method – how detailed a characterization of the training set it can remember and then apply to new data.