Weighted Hamming metric and KNN classification of nominal-continuous data

Aleksander Denisiuk
UWM, Olsztyn, Poland

Introduction

Consider data that have continuous as well as nominal features. Any additional structures are defined on the data. So, we use the Euclidean metric on continuous and the Hamming metric on nominal part of data.

The entire dataset is divided into classes. The main assumption is that each feature has different impact to the structure of classes. To model it, let's introduce appropriate multipliers in metric definition.

The problem is to define unknown multipliers. This is done by minimizing the total intra-class squared distance. Experiments show that using of the discovered weighted metric improves the standard KNN classification.

Data

The following data are considered:

$$\mathbb{X} = \{ \mathbf{X}_1, \dots, \mathbf{X}_M \}.$$

Each record consists of two parts

$$\mathbf{X}_i = (X_i, Y_i),$$

where $X_i = (x_i^1, \dots, x_i^n) \in \mathbb{R}^n$ are the continuous data, and $Y_i = (y_i^1, \dots, y_i^m)$ are the nominal data, $i = 1, \dots, M$.
Assume that $\mathbb{X}$ is divided into c classes,

$$\mathbb{X} = C_1 \cup \dots \cup C_c,$$

where $c < M$. This classes will be used for learning. The determined weights are used for classification of new records.

Metric

The Hamming metric on the set of nominal data is defined as follows:

$$\operatorname{dist}_h(Y_1, Y_2) = \frac{1}{m} \Big| \{ \beta = 1, \dots, m \mid y_1^\beta \neq y_2^\beta \} \Big| = \frac{1}{m} \sum_{\beta=1}^{m} \operatorname{diff}(y_1^\beta, y_2^\beta),$$

where $\operatorname{diff}(t_1, t_2) = \begin{cases} 1 & \text{if } t_1 \neq t_2, \\ 0 & \text{if } t_1 = t_2. \end{cases}$

Introduce the weights vector: $\mathbf{W} = (W, U) = (w_1, \dots, w_n, u_1, \dots u_m)$, where $w_\alpha > 0$, $u_\beta > 0$ for $\alpha = 1, \dots, n$, $\beta = 1, \dots, m$, and assume that classes are formed with respect to the *weighted distance*:

$$\begin{aligned} \operatorname{dist}^2_{\mathbf{W}}(\mathbf{X_1}, \mathbf{X_2}) &= \operatorname{dist}^2_{W,e}(X_1, X_2) + \operatorname{dist}^2_{U,h}(Y_1, Y_2) \\ &= \sum_{\alpha=1}^{n} w_\alpha^2 (x_1^\alpha - x_2^\alpha)^2 + \left(\sum_{\beta=1}^{m} u_\beta \operatorname{diff}(y_1^\beta, y_2^\beta) \right)^2. \end{aligned}$$

Total intra-class squared distance

To determine the weights vector $\mathbf{W}$ we minimize the total intra-class squared distance:

$$H(\mathbf{W}) = \frac{1}{M^2} \sum_{k=1}^{c} \left(\sum_{\mathbf{X}_i, \mathbf{X}_j \in C_k} \operatorname{dist}^2_{\mathbf{W}}(\mathbf{X}_i, \mathbf{X}_j) \right).$$

The objective function $H(\mathbf{W})$ is homogeneous with respect to $\mathbf{W}$. So, to workout an effective minimizing procedure, we assume that the generalized average of the weights is constant:

$$\left(\frac{1}{n+m} \left(\sum_{\alpha=1}^{n} w_\alpha^r + \sum_{\beta=1}^{m} u_\beta^r \right) \right)^{\frac{1}{r}} = 1, \quad r \in (0, 1).$$

Determining the weights

To solve constrained minimizing problem the method of Lagrange multipliers was used. The solution is as follows.

Continuous weights:

$$w_\alpha = \Lambda_r s_\alpha, \tag{1}$$

where

$$s_\alpha = \left(\frac{1}{M^2} \sum_{k=1}^{c} \sum_{i,j \in C_k} \left(x_\alpha^i - x_\alpha^j \right)^2 \right)^{-\frac{1}{2-r}}. \tag{2}$$

Nominal weights:

$$u_\beta = \Lambda_r z_\beta, \tag{3}$$

where z_β satisfies the following equation

$$z_\beta^{r-1} = \sum_{\gamma=1}^{m} A_{\beta\gamma} z_\gamma, \tag{4}$$

for matrix A defined as

$$A_{\beta\gamma} = \frac{1}{M^2} \sum_{k=1}^{c} \sum_{i,j \in C_k} \operatorname{diff}(y_\beta^i, y_\beta^j) \operatorname{diff}(y_\gamma^i, y_\gamma^j). \tag{5}$$

Multiplier Λ_r:

$$\Lambda_r = \left(\frac{\sum_{\alpha=1}^{n} s_\alpha^r + \sum_{\beta=1}^{m} z_\beta^r}{n+m} \right)^{-\frac{1}{r}} \tag{6}$$

Algorithm

To solve (4), a relaxation iterative method is applied:

$$z_{\beta,\text{next}} = z_\beta - \tau \left(z_\beta^{r-1} - \sum_{\gamma=1}^{m} A_{\beta\gamma} z_\gamma \right). \tag{7}$$

Algorithm 1. Determining of the metric weights

Require: $\mathbb{X} = \{ \mathbf{X}_1, \dots, \mathbf{X}_M \}$ is the set of records, $C_1, \dots, C_c$ — the set of classes
Ensure: $\mathbf{W} = (W, U)$ is the optimal weights vector
  Compute s_α, $\alpha = 1, \dots, n$ with (2)
  Compute matrix A with (5)
  Choose initial z vector as $\mathbf{z} = (1, \dots, 1)$
  while $\|z_{\text{next}} - z\| > \varepsilon$ **do**
    Compute z_{next} with (7)
  end while
  Compute Λ_r with (6)
  Compute w_α with (1) for $\alpha = 1, \dots, n$
  Compute u_β with (3) for $\beta = 1, \dots, m$
  return (W, U)

e-mail denisiuk@matman.uwm.edu.pl
homepage http://wmii.uwm.edu.pl/~denisjuk/

Numerical experiments

Australian Credit Approval **dataset**

Algorithm	**AUC**	$H(r)$
Weighted KNN, $r = 0.05$	0.942	166.53
Weighted KNN, $r = 0.15$	0.942	103.93
Weighted KNN, $r = 0.35$	0.938	52.12
Weighted KNN, $r = 0.55$	0.947	31.98
Weighted KNN, $r = 0.75$	0.947	21.15
Weighted KNN, $r = 0.95$	0.942	13.66
Unweighted KNN, normalized data	0.925	
Random forest	0.949	
Support Vector Machine	0.941	

Heart Disease **data set**

Algorithm	**AUC**	$H(r)$
Weighted KNN, $r = 0.05$	0.966	67.56
Weighted KNN, $r = 0.15$	0.969	54.62
Weighted KNN, $r = 0.35$	0.966	36.41
Weighted KNN, $r = 0.55$	0.974	25.68
Weighted KNN, $r = 0.75$	0.979	19.21
Weighted KNN, $r = 0.95$	0.946	14.55
Unweighted KNN, normalized data	0.953	
Random forest	0.941	
Support Vector Machine	0.966	

Artificial **data set**

Algorithm	**AUC**	$H(r)$
Weighted KNN, $r = 0.05$	0.997	10213.74
Weighted KNN, $r = 0.15$	0.997	1043.18
Weighted KNN, $r = 0.35$	0.997	127.58
Weighted KNN, $r = 0.55$	0.996	49.96
Weighted KNN, $r = 0.75$	0.995	26.68
Weighted KNN, $r = 0.95$	0.995	13.96
Unweighted KNN, normalized data	0.990	
Random forest	0.999	
Support Vector Machine, normalized data	0.990	

Conclusion

The proposed method is an interesting proposition for the classification problem. Moreover it could get further improvement. Specifically, we plan to consider alternatives to the standard Hamming metric for the nominal part.
Besides, the discovered metric can be used in other algorithms for analysis of nominal-continuous data that are based on similarity.

Further information

Analogous technique was used in case of non-supervised learning in [1].
The source code for experiment can be found at `https://gitlab.com/adenisiuk/weightedhamming`.
This poster is available for download at `http://wmii.uwm.edu.pl/~denisjuk/posters/praha2023.pdf`

References

[1]. Denisiuk, A., Grabowski, M.: Embedding of the hamming space into a sphere with weighted quadrance metric and c-means clustering of nominal-continuous data. Intelligent Data Analysis **22**(6), 1297001314 (2018).