TF-IDF Weighted Similarity Estimates for Unseen Categories

David Bang[§], Feng-Chang Lin[†] , Michael R. Kosorok[†], Alex Comerfield [‡]

[§]Ancestry

[†]Department of Biostatistics, Gillings School of Public Health, UNC Chapel Hill

[‡]Bloomberg

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Background

The majority of machine learning models are static meaning once a model has been fitted; they are not able to dynamically adjust their decision paths, hyperparameters, etc. This is problematic when the test and train data differs in distribution especially when there are unseen categories. Modern approaches rely on preprocessing techniques such as imputation by treating unseen categories as missing values or by assigning them predetermined clusters. TF-IDF Similarity Weighted Estimates (TIWS) is a novel framework by treating categorical data in an NLP context. TIWS assigns the unseen category a linear combination of seen categories with weights based on similarity measures.

Method

Suppose we have train and test data $\mathcal{D} = \{(x, y)\}_{i=1}^{n}$, $\mathcal{D}^* = \{(x^*, y^*)\}_{i=1}^{n^*}$, respectively. Let $x_i = (x_{i1}, \ldots, x_{ip})^T \in \mathbb{R}^p$ consists of only categorical features where x_{ij} represents the j^{th} feature of x_i, and $j = 1, ..., p$. Let c_{jk} represent the k^{th} category of the j^{th} feature, and let $k = 1, .., l_j$ with l_j denoting the number of known categories in the j^{th} feature. WLOG, we assume there is only one unseen category in the feature j. We let $c_{jl_{j+1}}$ represent the unseen category. We compute a target estimate $\hat{y}_{jk}$ [2] for the target parameter $\mathbb{E}(y|x_{ij} = c_{jk})$ using the training label y. An empirical average of y with the same category c_{jk} can be used.

$$\hat{y}_{jk} = \frac{\sum_{i=1}^{n} I(x_{ij} = c_{jk}) \cdot y_i + ap}{\sum_{i=1}^{n} I(x_{ij} = c_{jk}) + a}$$

where $a > 0$ is a parameter. A common setting for p is the average target value in the dataset. This assigns categories a numeric value. To construct documents of categories it is important to combine only the feature data $\mathcal{D}_x \bigcup \mathcal{D}^*_x = \{(\tilde{x},)\}_{i=1}^{n+n^*}$. A document of a specific category is defined by

$$D_{jj'} = \begin{array}{c} \\ c_{j1} \\ c_{j2} \\ \vdots \\ c_{jl_{j+1}} \end{array} \begin{array}{c} \begin{array}{ccc} c_{j'1} & \cdots & c_{j'l_{j'}} \end{array} \\ \begin{pmatrix} f(c_{j1}, c_{j'1}) & \cdots & f(c_{j1}, c_{j'l_{j'}}) \\ f(c_{j2}, c_{j'1}) & \cdots & f(c_{j2}, c_{j'l_{j'}}) \\ \vdots & \vdots & \vdots \\ f(c_{jl_{j+1}}, c_{j'1}) & \cdots & f(c_{jl_{j+1}}, c_{j'l_{j'}}) \end{pmatrix} \end{array}$$

where $f(c_{jk}, c_{j'k'}) = \sum_{i=1}^{n} I(\tilde{x}_{ij} = c_{jk}) I(\tilde{x}_{ij'} = c_{j'k'})$ for $j' \neq j$, $j' = 1, ..., p$, $k' = 1, ..., l_{j'}$, and $k = 1, ..., l_j$. In short, frequency counts where $\tilde{x}_{ij} = c_{jk}$ and $\tilde{x}_{ij'} = c_{j'k'}$. We consider $\mathcal{D}_{jj'}$ as a text corpus for the j^{th} category, where its words are represented by each c_{jk}. For $j' = 1, \ldots, p$ and $j' \neq j$, we augment the $D_{jj'}$ matrix to the full document matrix D_j that combines matrices by categories of feature j. The document matrix D_j can be defined by $D_j = (D_{j1}| \cdots |D_{jp})$ where D_j is a $l_{j+1} \times \sum_{j'=1, j' \neq j}^{p} l_{j'}$ matrix. Now we create the term frequency (TF) and inverse document frequency (IDF) matrices as follows:

$$T_{D_j}(c_{jk}, c_{j'k'}) = \frac{f(c_{jk}, c_{j'k'}) - \mu_{c_{jk}}}{\sigma_{c_{jk}}}$$

$$I_{D_j}(c_{jk}, c_{j'k'}) = T_{D_j^T}(c_{j'k'}, c_{jk})^T$$

where $\mu_{c_{jk}} = \frac{1}{n_j} \sum_{j'=1}^{p} \sum_{k'=1}^{l_{j'}} f(c_{jk}, c_{j'k'})$, $\sigma^2_{c_{jk}} = \frac{1}{n_j - 1} \sum_{\substack{j'=1 \\ j' \neq j}}^{p} \sum_{k'=1}^{l_{j'}} \{f(c_{jk}, c_{j'k'}) - \mu_{c_{jk}}\}^2$ and $n_j = \sum_{\substack{j'=1 \\ j' \neq j}}^{p} l_{j'}$. TF is the distribution of words that describe the document

Our proposed TIWS is the common standardization procedure for both TF and IDF due to its ubiquitious use in statistics but they can be uniquely defined depending on the context. Let $g(A, B)$ be a transformation function that aggregates two matrices A and B. Let $H_{D_j} = g(T_{D_j}, I_{D_j})$ be the aggregation matrix which uses the Hadamard (element-wise) multiplication. One can utilize a similarity metric $s(c_{jk}, c_{jk'})$ to describe the similarity between two row vectors of H_{D_j} at categories c_{jk} and $c_{jk'}$, and create a symmetric similarity matrix S that includes all pairwise similarities. Here, we choose a modified cosine similarity[1], which is defined as

$$s(c_{jk}, c_{jk'}) = \frac{1}{2} + \frac{1}{2} \cdot \frac{H_{D_j}(c_{jk}) H_{D_j}(c_{jk'})^T}{\sqrt{H_{D_j}(c_{jk})^{\otimes 2}} \sqrt{H_{D_j}(c_{jk'})^{\otimes 2}}}$$

where $H_{D_j}(c_{jk})$ is the row vector of H_{D_j} matrix at category c_{jk}, and $a^{\otimes 2} = aa'$ for a row vector a. It is recommended that $s(c_{jk}, c_{jk'}) \in [0, 1]$ since the similarity coefficients will be used as weights to find the predicted value of the target parameter of the unseen category. Finally, the unseen category $c_{jl_{j+1}}$ is defined by

$$\hat{y}_{jl_{j+1}} = \frac{\sum_{k=1}^{l_j} s(c_{jk}, c_{jl_{j+1}}) \cdot \hat{y}_{jk}}{\sum_{k=1}^{l_j} s(c_{jk}, c_{jl_{j+1}})}$$

Algorithmic Complexity

TWIS is very similar to the kNN [3] algorithm except the kNN algorithm sets the weights to be proportion of a category selected, $kNN_{c_{jl_{j+1}}} = \frac{\sum_{k=1}^{l_j} p_{jk} \cdot \hat{y}_{jk}}{\sum_{k=1}^{l_j} p_{jk}}$. However, kNN uses information within a local bound in an iterative fashion whereas TIWS uses information in a one-shot aggregate fashion. The time complexity for TIWS is $O((n + n*) \cdot l_{j+m})$ and space complexity of $O((n + n*) \cdot l_{j+m})$ for the temporary similarity matrix S.

Multiple Unseen Categories

Suppose there are multiple unseen categories in the test set for the jth feature. Then we only need to be working with the document matrix D_j and its similarity matrix S_j. When deriving the D_j it is important to note that we are taking frequency counts of all c_{jk} for $j = 1, ..., p$ and $k = 1, ..., l_j$ in $\mathcal{X}$. However, a TIWS estimate for an unseen $\hat{y}_{jl_{j+a}}$ for does not use any information from any other unseen category $c_{jl_{j+b}}$ for $a = 1, ..., m$ and $b = 1, ..., m$ where $a \neq b$ and for $m = 1, .., \infty$ unseen categories. The derivation is described in the following:

We set $diag(S_j)$ to 0 or $s(c_{ji}, c_{jk}) = 0$ for $i = k$ for $i = 1, ..., l_{jl_{j+m}}$ and $k = 1, ..., l_{jl_{j+m}}$.

$$\vec{c} = \begin{bmatrix} 0 & s(c_{j2}, c_{j1}) & \ldots & s(c_{jl_{j+1}}, c_{j1}) & s(c_{jl_{j+m}}, c_{j1}) \\ s(c_{j2}, c_{j1}) & 0 & \ldots & \ldots & \ldots \\ \vdots & \ldots & 0 & \ldots & \ldots \\ s(c_{jl_{j+1}}, c_{j1}) & \ldots & \ldots & 0 & \ldots \\ s(c_{jl_{j+m}}, c_{j1}) & \ldots & \ldots & \ldots & 0 \end{bmatrix} \begin{bmatrix} \hat{y}_{j1} \\ \hat{y}_{j2} \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Notice that $\vec{c}_{[l_{j+1}]} = \sum_{i=1}^{l_{j+m}} s(c_{jl_{j+1}}, c_{ji}) \hat{y}_{ji} = \sum_{\substack{i=1 \\ i \neq l_{j+1}}}^{l_{j+m}} s(c_{jl_{j+1}}, c_{ji}) \hat{y}_{ji}$ since we set $s(c_{ji'}, c_{ji}) = 0$ for $i = i'$ for $i = 1, ..., p'$ and for $i' = 1, ..., p$. Also notice that $\vec{c}_{[l_{j+1}]}$ does not contain $\hat{y}_{jl_{j+1}}$. To get the TIWS estimate for $\hat{c}_{jl_{j+1}}$ simply equate $\hat{y}_{jl_{j+1}} = \vec{c}_{[l_{j+1}]}$. So for multiple unseen categories $\{\hat{y}_{jl_{j+1}}, ..., \hat{y}_{jl_{j+m}}\}$ we can simply grab $\vec{c}_{[l_{j+1}:l_{j+m}]}$ for those TIWS estimates. Distribute the remaining weights that were set to 0 equally among all reference categories i.e. distribute 1 - $\frac{1}{m} \sum_{\ldots}^{l_{j+m}} s(c_{jl_{j+1}}, c_{ji}) \hat{y}_{ji}$. The case of multiple

Case Studies

Preliminary TIWS applications were used on the Titanic and Wake County Sudden Death data. However, due to limited spacing, only an interesting case in the Titanic data will be displayed. Here we set the unseen category for the $embarked = Q$ feature to be unseen.

Diverging performance for TIWS & kNN					
Unseen Categories $\{embarked_Q\}$ (n=77)					
CV1 (n=51)		CV2 (n=51)		CV3 (n=52)	
Method	Results	Method	Results	Method	Results
kNN (all)	Accuracy: 70.6% Precision: 60.0% Recall: 35.3%	kNN (all)	Accuracy: 68.6% Precision: 63.6% Recall: 36.8%	kNN (all)	Accuracy: 67.3% Precision: 81.8% Recall: 37.5%
TIWS	Accuracy: 80.4% Precision: 68.4% Recall: 76.5%	TIWS	Accuracy: 74.5% Precision: 63.6% Recall: 73.7%	TIWS	Accuracy: 78.8% Precision: 76.0% Recall: 79.2%

Table 1: General unseen categories for a specific feature 'embarked'

embarked consisted of only three categories $\{S, C, Q\}$. For every observation, $embarked_Q$ was most similar to $embarked_S$ for the kNNs, but TIWS determined that $embarked_Q$ was most similar to $embarked_C$. This may be due to the number of observations per class $\{S, C, Q\} \rightarrow \{644, 168, 77\}$. So there was a much higher chance that the majority of the nearest neighbors consisted of $embarked_S$. TIWS performed significantly better across all metrics and folds. This highlights a limitation of kNN since kNN can be very near-sighted. Across various cases in both data, TIWS performed similarly to kNN on average this was highlighted in the **Algorithmic Complexity** portion. However, there are clear benefits to TIWS since it is able to draw bootstrapped or jackknifed estimates, automate clustering, and penalize unimportant features.

References

[1] Li B. and Han L. "Distance Weighted Cosine Similarity Measure for Text Classification". In: *Intelligent Data Engineering and Automated Learning – IDEAL 2013* (2013).

[2] A.V. Dorogush et al. "CatBoost: Unbiased Boosting with Categorical Features". In: *In Advances in Neural Information Processing Systems* (2018).

[3] E. Fix and J.L. Hodges. "Discriminatory Analysis, Nonparametric Discrimination: Consistency Properties". In: *USAF School of Aviation Medicine, Randolph Field* (1951).