SuperTML: Domain Transfer from Computer Vision to Structured Tabular Data through Two-Dimensional Word Embedding

Anonymous Authors[1]

Abstract

Structured tabular data is the most commonly used form of data in industry according to a Kaggle ML and DS Survey. Gradient Boosting Trees, Support Vector Machine, Random Forest, and Logistic Regression are typically used for classification tasks on tabular data. The recent work of Super Characters method using two-dimensional word embeddings achieved state-of-the-art results in text classification tasks, showcasing the promise of this new approach. In this paper, we propose the SuperTML method, which borrows the idea of Super Characters method and two-dimensional embeddings to address the problem of classification on tabular data. For each input of tabular data, the features are first projected into two-dimensional embeddings like an image, and then this image is fed into fine-tuned ImageNet CNN models for classification. Experimental results have shown that the proposed SuperTML method have achieved state-of-the-art results on both large and small datasets.

1. Introduction

In data science, data is categorized into structured data and unstructured data. Structured data is also known as tabular data, and the terms will be used interchangeably. Anthony Goldbloom, the founder and CEO of Kaggle observed that winning techniques have been divided by whether the data was structured or unstructured (Vorhies, 2016). Currently, DNN models are widely applied for usage on unstructured data such as image, speech, and text. According to Anthony, "When the data is unstructured, its definitely CNNs and RNNs that are carrying the day" (Vorhies, 2016). The successful CNN model in the ImageNet competition (Russakovsky et al., 2015) has outperformed human for image classification task by ResNet (He et al., 2016) since 2015.

On the other side of the spectrum, machine learning models such as Support Vector Machine (SVM), Gradient Boosting Trees (GBT), Random Forest, and Logistic Regression, have been used to process structured data. According to a recent survey of 14,000 data scientists by Kaggle (2017), a subdivision of structured data known as relational data is reported as the most popular type of data in industry, with at least 65% working daily with relational data. Regarding structured data competitions, Anthony says that currently XGBoost is winning practically every competition in the structured data category (Fogg, 2016). XGBoost (Chen & Guestrin, 2016) is one popular package implementing the Gradient Boosting method.

Recent research has tried using one-dimensional embedding and implementing RNNs or one-dimensional CNNs to address the TML (Tabular Machine Learning) tasks, or tasks that deal with structured data processing (Lam et al., 2018; Thomas, 2018), and also categorical embedding for tabular data with categorical features (Guo & Berkhahn, 2016; Chen et al., 2016). However, this reliance upon one-dimensional embeddings may soon come to change. Recent NLP research has shown that the two-dimensional embedding of the Super Characters method (Sun et al., 2018) is capable of achieving state-of-the-art results on large dataset benchmarks. The Super Characters method is a two-step method that was initially designed for text classification problems. In the first step, the characters of the input text are drawn onto a blank image. In the second step, the image is fed into two-dimensional CNN models for classification. The two-dimensional CNN models are trained by fine-tuning from pretrained models on large image dataset, e.g. ImageNet.

In this paper, we propose the SuperTML method, which borrows the concept of the Super Characters method to address TML problems. For each input, tabular features are first projected onto a two-dimensional embedding and fed into fine-tuned two-dimensional CNN models for classification. The proposed SuperTML method handles the categorical type and missing values in tabular data automatically, without need for explicit conversion into numerical

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

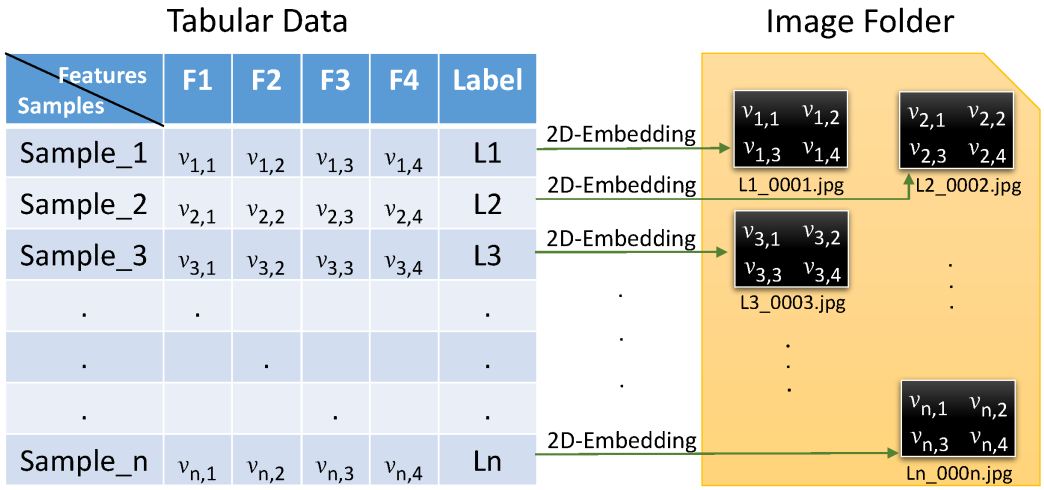


Figure 1. An example of converting training data from tabular into images with two-dimensional embeddings of the features in the tabular data. Therefore, the problem of machine learning for tabular data is converted into an image classification problem. The later problem can use pretrained two-dimensional CNN models on ImageNet for finetuning, for example, ResNet, SE-net and PolyNet. The tabular data given in this example has n samples, with each sample having four feature columns, and one label column. For example, assume the tabular data is to predict whether tomorrow's weather is "Sunny" or "Rainy". The four features F1, F2, F3, and F4 are respectively "color of the sky", "Fahrenheit temperature", "humidity percentage", and "wind speed in miles per hour". Sample_1 has class label L1="Sunny", with four features values given by $v_{1,1} =$ "$blue$", $v_{1,2} = 55, v_{1,3} =$ "$missing$", and $v_{1,4} = 17$. The two-dimensional embedding of Sample_1 will result in an image of "Sunny_0001.jpg" in the image folder. The four feature values are embedded into the image on different locations of the image. For example, $v_{1,1}$ is a categorical value of color "$blue$", so the top left of the image will have exactly the alphabets of "blue" written on it. For another example, $v_{1,2}$ is a numerical value of "23", so the top right of the image will have exactly the digits of "23" written on it. For yet another example, $v_{1,3}$ should be a numerical value but it is missing in this example, so the bottom left of the image will have exactly the alphabets of "missing" written on it. Other ways of writing the tabular features into image are also possible. For example, "blue" can be written in short as a single letter "b" if it is distinctive to other possible values in its feature column. The image names will be parsed into different classes for image classification. For example, L1 = L2 = "Sunny", and L3 = Ln ="Rainy". These will be used as class labels for training in the second step of SuperTML method.

type values.

2. The Proposed SuperTML Method

The SuperTML method is motivated by the analogy between TML problems and text classification tasks. For any sample given in tabular form, if its features are treated like stringified tokens of data, then each sample can be represented as a concatenation of tokenized features. By applying this paradigm of a tabular sample, the existing CNN models used in Super Characters method could be extended to be applicable to TML problems.

As mentioned in the introduction, the combination of two-dimensional embedding (a core competency of the Super Characters methodology) and pre-trained CNN models has achieved state-of-the-art results on text classification tasks. However, unlike the text classification problems studied in (Sun et al., 2018), tabular data has features in separate dimensions. Hence, generated images of tabular data should reserve some gap between features in different dimensions in order to guarantee that features will not overlap in the generated image.

SuperTML is composed of two steps, the first of which is two-dimensional embedding. This step projects features in the tabular data onto the generated images, which will be called the SuperTML images in this paper. The conversion of tabular training data to SuperTML image is illustrated in Figure 1, where a collection of samples containing four tabular features is being sorted.

The second step is using pretrained CNN models to finetune on the generated SuperTML images.

Figure 1 only shows the generation of SuperTML images for the training data. It should be noted that for inference, each instance of testing data goes through the same preprocessing to generate a SuperTML image (all of which use the same configuration of two-dimensional embedding) before getting fed into the CNN classification model.

Considering that features may have different importance for the classification task, it would be prudent to allocate larger spaces for important features and increase the font size of the corresponding feature values. This method,

Algorithm 1 SuperTML_VF: SuperTML method with Variant Font size for embedding.

Input: Tabular data training set
Parameter: Image size of the generated SuperTML images
Output: Finetuned CNN model

1: Calculate the feature importance in the given tabular data provided by other machine learning methods.
2: Design the location and font size of each feature in order to occupy the image size as much as possible. Make sure no overlapping among features.
3: **for** each sample in the tabular data **do**
4:     **for** each feature of the sample **do**
5:         Draw feature in the designated location and font size.
6:     **end for**
7: **end for**
8: Finetune the pretrained CNN model on ImageNet with the generated SuperTML images.
9: **return** the trained CNN model on the tabular data

known as SuperTML_VF, is described in Algorithm 1.

To make the SuperTML more autonomous and remove the dependency on feature importance calculation done in Algorithm 1, the SuperTML_EF method is introduced in Algorithm 2. It allocates the same size to every feature, and thus tabular data can be directly embedded into SuperTML images without the need for calculating feature importance. This algorithm shows even better results than 1, which will be described more in depth later in the experimental section.

3. Experiments

The data statistics from UCI Machine Learning Repository is shown in Table 1.

3.1. Experiments on the Iris dataset

"This is perhaps the best known database to be found in the pattern recognition literature"[1]. The Iris dataset is widely used in machine learning courses and tutorials. Figure 2a shows an example of a generated SuperTML image, created using Iris data. The experimental results of using SE-net-154 shown in Table 2 is based on an 80:20 split of the 150 samples. It shows that the proposed SuperTML method achieves the same accuracy as XGBoost on this small dataset.

[1]https://archive.ics.uci.edu/ml/datasets/Iris

Algorithm 2 SuperTML_EF: SuperTML method with Equal Font size for embedding.

Input: Tabular data training set
Parameter: Image size of the generated SuperTML images
Output: Finetuned CNN model

1: **for** each sample in the tabular data **do**
2:     **for** each feature of the sample **do**
3:         Draw the feature in the same font size without overlapping, such that the total features of the sample will occupy the image size as much as possible.
4:     **end for**
5: **end for**
6: Finetune the pretrained CNN model on ImageNet with the generated SuperTML images.
7: **return** the trained CNN model on the tabular data

(a) SuperTML_EF image example for Iris data.

(b) SuperTML_VF image example for Wine data.

Figure 2. Examples of generated SuperTML image for Iris and Wine dataset.

3.2. Experiments on the Wine dataset

For this dataset[2] , we use SuperTML VF, which gives features different sizes on the SupterTML image according to their importance score. The feature importance score is obtained using the XGBoost package (Chen & Guestrin, 2016). One example of a SuperTML image created using data from this dataset is shown in Figure 2b. The results in Table 2 shows that the SuperTML method obtained a slightly better accuracy than XGBoost on this dataset.

3.3. Experiments on the Adult dataset

The task of this Adult dataset[3] is to predict whether a persons income is larger or smaller than 50,000 dollars per year based on a collection of surveyed data.

For categorical features that are represented by strings, the Squared English Word (SEW) method (Sun et al., 2019)

[2]https://archive.ics.uci.edu/ml/datasets/Wine
[3]https://archive.ics.uci.edu/ml/datasets/Adult

Table 1. Datasets statistics used in this paper from UCI Machine Learning Repository. The “**Missing**” in the table indicates whether there are missing values in the data set. The “NA” in the table denotes that there is no given split for the training and testing dataset.

Dataset	Classes	#Attributes	Train	Test	Total	Data Types	Missing
Iris	3	4	NA	NA	150	Real	No
Wine	3	13	NA	NA	178	Integer& Real	No
Adult	2	14	32,561	16,281	48,842	Integer & Categorical	Yes

Table 2. Model accuracy comparison on the tabular data from UCI Machine Learning Repository.

Accuracy	Iris(%)	Wine(%)	Adult(%)
XGBoost	93.33	96.88	87.32
SuperTML	**93.33**	**97.30**	**87.64**

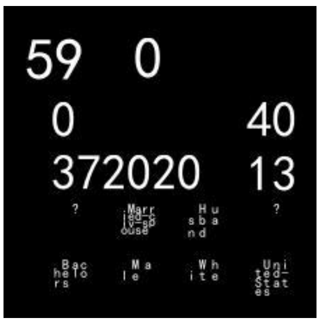
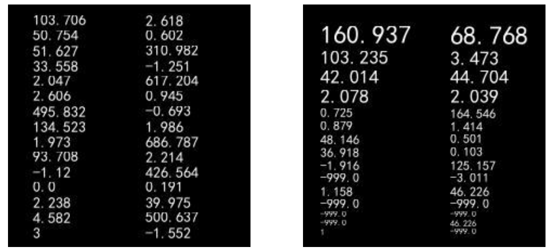


Figure 3. SuperTML_VF image example from Adult dataset. This sample has age = 59, capital gain = 0, capital loss = 0, hours per week = 40, fnlweight = 372020, education number = 13, occupation = ”?” (missing value), marital status = ”Married-civ-spouse”, relationship = ”Husband”, workclass = ”?” (missing value), education = ”Bachelors”, sex = ”Male”, race = ”White”, native country = ”United-States”.

is used. One example of a generated SuperTML image is given in Figure 3. Table 2 shows the results on Adult dataset. We can see that on this dataset, the SuperTML method still has a higher accuracy than the fine-tuned XGBoost model, outperforming it by 0.32% points of accuracy.

3.4. Experiments on the Higgs Boson Machine Learning Challenge dataset

The Higgs Boson Machine Learning Challenge involved a binary classification task to classify quantum events as signal or background. It was hosted by Kaggle, and though the contest is over, the challenge data is available on opendata (Adam-Bourdarios et al., 2015). It has 25,000 training samples, and 55,000 testing samples. Each example has 30 features, each of which is stored as a real number value. In this challenge, AMS score (Adam-Bourdarios et al., 2014) is used as the performance metric. Figure 4 shows two examples of generated SuperTML images.



(a) SuperTML_EF background event example.

(b) SuperTML_VF signal event example.

Figure 4. Examples of SuperTML images for Higgs Boson .

Table 3. Comparison of AMS score on Higgs Boson. The first two rows are winners in the Higgs Boson Challenge.

Methods	AMS
DNN by Gabor Meli	3.806
XGBoost	3.761
SuperTML_EF(224x224)	**3.979**
SuperTML_VF (224x224)	3.838

Table 3 shows the comparison of different algorithms. The DNN method and XGBoost used in the first two rows are using the numerical values of the features as input to the models, which is different from the SuperTML method of using two-dimensional embeddings. It shows that SuperTML_EF method gives the best AMS score of 3.979. In addition, the SuperTML_EF gives better results than SuperTME_VF results, which indicates SuperTML method can work well without the calculation of the importance scores.

4. Conclusion

The proposed SuperTML method borrows the idea of two-dimensional embedding from Super Characters and transfers the knowledge learned from computer vision to the structured tabular data. Experimental results shows that the proposed SuperTML method has achieved state-of-the-art results on both large and small tabular dataset.

References

Adam-Bourdarios, C., Cowan, G., Germain, C., Guyon, I., Kegl, B., and Rousseau, D. Learning to discover: the higgs boson machine learning challenge. *URL http://higgsml. lal. in2p3. fr/documentation*, pp. 9, 2014.

Adam-Bourdarios, C., Cowan, G., Germain, C., Guyon, I., Kégl, B., and Rousseau, D. The higgs boson machine learning challenge. In *NIPS 2014 Workshop on High-energy Physics and Machine Learning*, pp. 19–55, 2015.

Chen, T. and Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794. ACM, 2016.

Chen, T., Tang, L.-A., Sun, Y., Chen, Z., and Zhang, K. Entity embedding-based anomaly detection for heterogeneous categorical events. In *Proceedings of International Joint Conferences on Artifical Intelligence (IJCAI)*, pp. 1396–1403, 2016.

Fogg, A. Anthony goldbloom gives you the secret to winning kaggle competitions, 2016.

Guo, C. and Berkhahn, F. Entity embeddings of categorical variables. *arXiv preprint arXiv:1604.06737*, 2016.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Lam, H. T., Minh, T. N., Sinn, M., Buesser, B., and Wistuba, M. Neural feature learning from relational database. *arXiv preprint arXiv:1801.05372*, 2018.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet LargeScale VisualRecognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Sun, B., Yang, L., Dong, P., Zhang, W., Dong, J., and Young, C. Super characters: A conversion from sentiment classification to image classification. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 309–315, 2018.

Sun, B., Yang, L., Chi, C., Zhang, W., and Lin, M. Squared english word: A method of generating glyph to use super characters for sentiment analysis. *arXiv preprint arXiv:1902.02160*, 2019.

Thomas, R. An introduction to deep learning for tabular data, 2018.

Vorhies, W. Has deep learning made traditional machine learning irrelevant?, 2016.