Learning Feature Engineering for Classification

Fatemeh Nargesian, Horst Samulowitz, Udayan Khurana
Elias B. Khalil, Deepak Turaga

Outline

- **Introduction**
  - Automated Feature Engineering Problem
- **Methods**
  - Transformation Recommendation
  - Feature-Class Representation
  - Training
- **Experiments Results**
- **Conclusion**

Introduction

Feature Engineering:

- The process of extracting features from a raw dataset is called feature engineering.
- It is the practice of constructing suitable features from given features that lead to improved predictive performance.
- Coming up with features is difficult, time-consuming, and requires expert knowledge.

（Proposed） Automated Feature Engineering Problem

Motivation

meta-learning approach

- Automatically perform interpretable feature engineering for classification, based on learning from past feature engineering experiences.
- Learns useful patterns between features, transforms and target that improve learning accuracy.

Methods

Automated Feature Engineering Problem:

A dataset, D, with features, $F = \{f_1, \ldots, f_n\}$,

A target class, k

A set of transformations, $T = \{T_1, \ldots, T_m\}$,

A classification task, L

Objective:

Find q best paradigms for constructing new features such that appending the new features to D maximizes the accuracy of L.

- unary transformations $O(u * n)$
- binary transformations $O(b * P_2^n)$ (r-ary transformations)

Methods

Automated Feature Engineering Problem:

A dataset, D, with features, $F = \{f_1, \dots, f_n\}$,

A target class, k

A set of transformations, $T = \{T_1, \dots, T_m\}$,

A classification task, L

Objective:

Find q best paradigms for constructing new features such that appending the new features to D maximizes the accuracy of L.

Each paradigm consists of a candidate transformation $T_c \in T$ of arity r, an ordered list of features $[f_i, \dots, f_{i+r-1}]$ and a usefulness score.

Methods

Transformation Recommendation:

- Models the problem of predicting a useful r-ary transformation $T_c \in T_r$, for a given list of features $[f_1, \ldots, f_r]$ as a multi-class classification
- Input is a representation of features, $R[f_1, \ldots, f_r]$, and output classes are transformations in T_r.

$$c = \arg\max_k g_k(R_{[f_1,\ldots,f_r]})$$

$$\text{recommend}: \begin{cases} T_c, & \text{if } g_c(R_{[f_1,\ldots,f_r]}) > \gamma \\ \text{none}, & \text{otherwise} \end{cases}$$

$g_k(R[f_1, \ldots, f_r])$ be the confidence score of the MLP corresponding to transformation T_k

γ is the threshold for confidence scores

Methods

Feature-Class Representation:

- To convert feature values and their class labels to a fixed size feature vector representation that can be fed into LFE classifiers.
- LFE represents feature f in a dataset with k classes as follows(Quantile Sketch Array) :

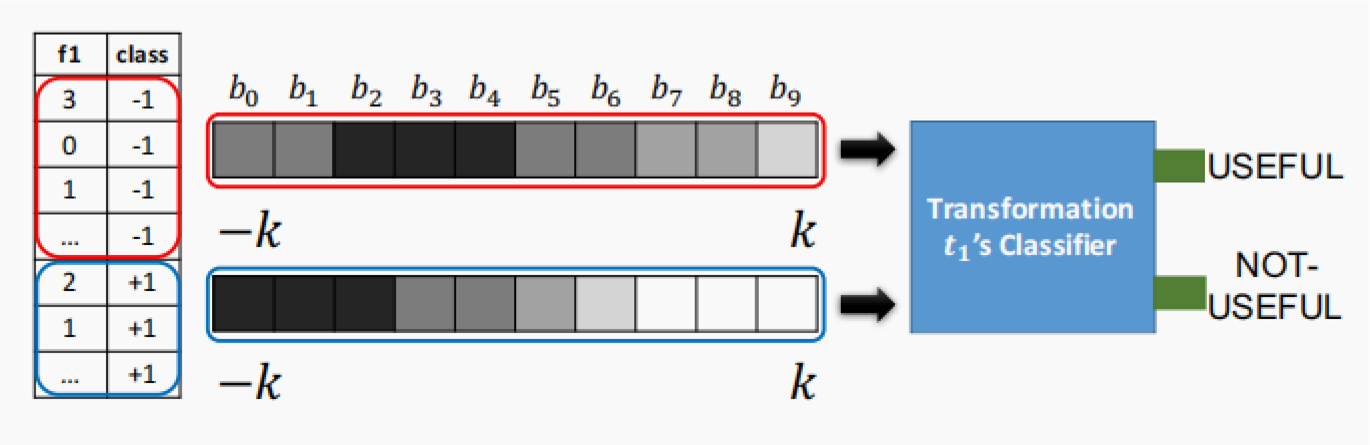
$$R_f = \left[Q_f^{(1)}; Q_f^{(2)}; \ldots ; Q_f^{(k)} \right]$$

where $Q_f^{(i)}$ is a fixed-sized representation of values in f that are associated with class i.

Methods

Quantile Sketch Array:

- QSA uses quantile data sketch [Wang et al., 2013-SIGMOD] to represent feature values associated with a class label.
- QSA is a non-parametric representation that enables characterizing the approximate Probability Distribution Function of values.



Methods

Training:

- If the constructed feature shows performance improvement beyond a threshold, θ, the input features together with their class labels are considered as a positive training sample.
- Each classifier is trained with the samples for which the corresponding transformation has been found useful as positive samples and all other samples as negative.

Experiments

Evaluate three aspects of LFE:

- The impact of using Quantile Sketch Array representation on the performance of transformation classifiers compared to other representations.
- The capability of LFE in recommending useful transformations.
- The benefit of using LFE to perform feature engineering compared to other alternatives, in prediction accuracy and time taken.

Experiments Results

Transformations:

- unary transformations (10)
- binary transformations (2)

Transformation	log	sqrt	square	freq	round	tanh	sigmoid	isotonic-reg	zscore	normalize	sum	subt	mult	div
#Positive Training Samples	6710	6293	5488	73919	15855	70829	70529	624	664	31019	28492	36296	37086	19020
Classifier Performance	0.95	0.96	0.97	0.77	0.92	0.92	0.80	0.98	0.97	0.91	0.98	0.99	0.97	0.98

Table 1: Statistics of Training Samples and F1 Score of LFE Classifiers for 10-fold Cross Validation of Random Forest.

Feature Representation:

- Evaluate the efficacy of Quantile Sketch Array (QSA)

Hand-crafted Meta-features	Stratified Sampling	Meta-feature Learning	Quantile Sketch Array
0.5558	0.5173	0.3256	**0.9129**

Table 2: F1 Score of Transformation Classifiers.

Experiments Results

Dataset	#Numerical Features	#Data Points	Base Dataset	Majority	Brute-Force	Random (10 runs)	Evaluation based	unary LFE	binary LFE
AP-omentum-lung	10936	203	0.883	0.915	0.925	0.908	-	**0.929**	0.904
AP-omentum-ovary	10936	275	0.724	0.775	0.801	0.745	0.788	**0.811**	0.775
autos	48	4562	0.946	0.95	0.944	0.929	0.954	**0.96**	0.949
balance-scale	8	369	0.884	0.916	0.892	0.881	0.882	**0.919**	0.884
convex	784	50000	0.82	0.5	**0.913**	0.5	-	0.819	0.821
credit-a	6	690	0.753	0.647	0.521	0.643	0.748	**0.771**	0.771
dbworld-bodies	2	100	0.93	0.939	0.927	0.909	0.921	**0.961**	0.923
diabetes	8	768	0.745	0.694	0.737	0.719	0.731	**0.762**	0.749
fertility	9	100	0.854	0.872	0.861	0.832	0.833	**0.873**	0.861
gisette	5000	2100	0.941	0.601	0.741	0.855	-	**0.942**	0.933
hepatitis	6	155	0.747	0.736	0.753	0.727	**0.814**	0.807	0.831
higgs-boson-subset	28	50000	0.676	0.584	0.661	0.663	-	**0.68**	0.677
ionosphere	34	351	0.931	0.918	0.912	0.907	0.913	**0.932**	0.925
labor	8	57	0.856	0.827	0.855	0.806	0.862	**0.896**	0.896
lymph	10936	138	0.673	0.664	0.534	0.666	0.727	**0.757**	0.719
madelon	500	780	0.612	0.549	0.585	0.551	0.545	**0.617**	0.615
megawatt1	37	253	0.873	0.874	0.882	0.869	0.877	**0.894**	0.885
pima-indians-subset	8	768	0.74	0.687	**0.751**	0.726	0.735	0.745	0.76
secom	590	470	0.917	0.917	0.913	0.915	0.915	**0.918**	0.915
sonar	60	208	**0.808**	0.763	0.468	0.462	0.806	0.801	0.783
spambase	57	4601	**0.948**	0.737	0.39	0.413	0.948	0.947	0.947
spectf-heart	43	80	0.941	**0.955**	0.881	0.942	0.955	**0.955**	0.956
twitter-absolute	77	140707	**0.964**	0.866	0.946	0.958	0.963	0.964	0.964
Feature Engineering and Model Evaluation Time (seconds)		**geomean**	2.66	11.06	48.54	69.57	403.81	18.28	44.58
		average	19.23	1219.34	13723.51	2041.52	10508.75	97.90	188.17

Table 3: Statistics of Datasets and F1 Score of LFE and Other Feature Engineering Approaches with 10-fold Cross Validation of Random Forest. The best performing approach is shown in bold for each dataset. The improving approaches are underlined.

Experiments Results

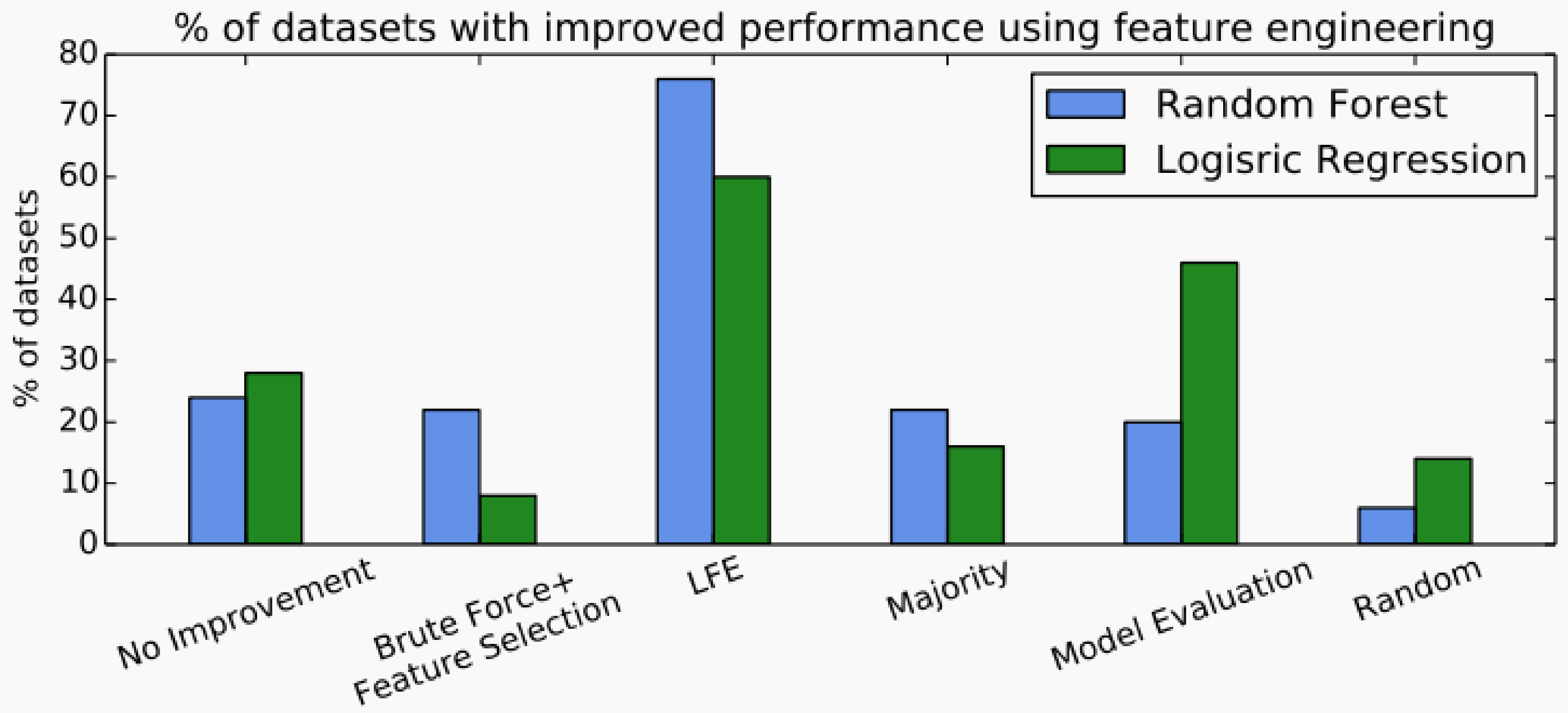


Figure 2: The percentage of datasets, from a sample of 50, for which a feature engineering approach results in performance improvement (measured by F1 score of 10 fold cross validation for Random Forest and Logistic Regression).

Conclusion

- Present a novel framework called LFE to perform automated feature engineering by learning patterns between feature characteristics, class distributions, and useful transformations, from historical data.
- Use a novel feature representation, called Quantile Sketch Array (QSA), that reduces any variable sized features to a fixed size array, preserving its essential characteristics

- The processing feature are only numeric data.