Data Driven Feature Learning

by

saket.maheshwary , Ambika Kaul, Vikram Pudi

in

34th International Conference on Machine Learning Workshop
(*ICML-2017*)

Sydney, Australia

Report No: IIIT/TR/2017/-1

Centre for Data Engineering
International Institute of Information Technology
Hyderabad - 500 032, INDIA
August 2017

Data Driven Feature Learning

Saket Maheshwary [1] **Ambika Kaul** [1] **Vikram Pudi** [1]

Abstract

We present a regression-based feature learning algorithm that generates new features from a set of available features (raw data points). Being data-driven, it requires no domain knowledge and is hence generic. Such a representation is learnt by mining pairwise feature associations, identifying the linear or non-linear relationship between each pair, applying regression and selecting those relationships that are stable. Our experimental evaluation on 20 datasets taken from UC Irvine and Gene Expression, across different domains, provides evidence that the features learnt through our model can improve the overall prediction accuracy, substantially, over the original feature space across 8 different classifiers without any domain knowledge.

1. Introduction

In recent past, the impressive success of deep learning techniques (Hinton et al., 2012; Russakovsky et al., 2015) has substantiated the importance of feature learning. These techniques, however, are effective when extremely large amounts of training data and intensive computational resources are available. Generally, feature engineering requires substantial manual effort in designing and selecting features and is often tedious and non-scalable. A common practice, is to use all available features and leave the problem of identifying the useful feature sets to the learning model. An approach like this does not always work well. Most of the time, dealing with such large feature space proves to be ineffective as it raises computational complexity when only a small number of features are actually useful. Furthermore, most construction algorithms employ special-purpose construction methods and heuristics that are especially suited to their underlying representation. There are, however, several problems with this scheme. For instance, given a new classification problem, it is not obvious which of the various representations and associated algorithms should be selected. In many real-world classification problems, the target concept is best expressed by features constructed using domain-specific knowledge. The work in (Piramuthu & Sikora, 2009) presents TFC framework, an iterative operator based feature construction algorithm, exhausting all obtained features and then selecting the best ones using information gain. However, exhaustive search leads to combinatorial explosion of feature space making this approach non scalable. To avoid exhaustive search, learning models based on decision trees, such as (Pagallo, 1989; Schlimmer & Granger, 1986; Matheus & Rendell, 1989; Fan et al., 2010) have been proposed. The features generated by some of these methods have no generalization guarantee across different classifiers. The work in (Fan et al., 2010) proposed a model called FCTree, where new features were learned using decision trees as a number of several sequential transforms of the original feature space without any domain knowledge to improve classification performance.

In this paper, we propose a model to alleviate difficulties involved in feature engineering through automation. We develop a learning model that automates discovery of *underlying patterns* in the data by the way features are related to each other and selects a very small number of new features to create a significant improvement in predictive performance. We present a novel method for feature generation, that captures the prominent variations in feature pairs via regression, by picking only those data points that are relevant and leads to highly discriminative information. We experimentally demonstrate the effectiveness of our approach on datasets from diverse domains across multiple classifiers.

2. Proposed Approach

Rationale for proposed design. The overall behaviour of data is difficult to recognize by looking at isolated records. However, we can expect that each class of objects will have different pattern, from which it is much easier to identify the class. In some cases, the classes may be indistinguishable by their original features, and only apparent when

[1]Data Sciences and Analytics Center, Kohli Center on Intelligent Systems, International Institute of Information Technology, Hyderabad, India. Correspondence to: Saket Maheshwary <saket.maheshwary@research.iiit.ac.in>.

Proceedings of the 34^{th} International Conference on Machine Learning Workshop, Sydney, Australia, PMLR 70, 2017.

more discriminative information not obvious in the original feature space is generated. In earlier works, a set of domain dependent operators were identified to transform features, based on the understanding that highly informative features often result from manipulations of elementary ones. In this paper, instead of using operators, we use regression to discover underlying patterns by the way feature pairs are related to each other. *The precise manner in which a feature, say f_1 influences a feature f_2, might vary for each class*. While we cannot claim that it will vary for every feature pair, it is very likely, that for at least some feature pairs, this variation will be very *prominent* and would result in highly discriminative information. Our goal is to use regression as a means to discriminate between how features influence each other across classes and to mine feature relationships, linear or non-linear and their forecast. Our model has following 3 steps.

(1) Mining Correlated Features: Distance correlation (Szekely et al., 2007) is used to determine if there exists a predictive relationship of interest for a given feature pair. We begin with original feature space F_{d} and find correlated feature pairs useful for feature construction. This process is iteratively carried for all the pairs in the feature space. After this step, independent (non-correlated) feature pairs are filtered out (Line 4 in Algorithm 1) and hence only the correlated data points (feature pairs) are picked up in this step for feature generation. The correlation between a given pair of feature vector can be linear (Line 8 in Algorithm 1) or nonlinear (Line 5 in Algorithm 1). We maintain a threshold parameter η_1 in order to segregate linear and nonlinear dependencies, respectively.

(2) Feature Generation: While *mining correlated features* detects the implicit patterns by the way features are related to each other, the *feature generation step* discovers and captures those prominent patterns and its variations that result in highly discriminative information. Here, given a training set of independent and dependent variables, we use regression algorithms to seek a connection between them for feature construction. Specifically, we use linear regression (LR) and support vector regression (SVR) techniques to determine linear and non linear relations respectively. We are using each feature to predict the values of other features by applying regression, and including those predicted values (regression forecast) to supplement the original feature space. For every correlated feature pair, the feature constructed is the forecast (prediction values) which we get by regressing F_i on F_j for every correlated feature pair (Line 6 and 9 in Algorithm 1) where F_i and F_j are independent and dependent variables respectively. Note that the feature constructed by regressing F_i on F_j is different from the feature built by regressing F_j on F_i. The features generated try to learn the underlying patterns between correlated feature pairs which models proposed in prior literature fail to capture, thus improving prediction accuracy.

A new feature is essentially the regression of one feature (independent variable) on another (dependent variable). This regression can be linear or non-linear. So, in a perfect regression result, the new feature simply replicates another feature which is already present. We filter out such type of regression results on feature pairs in our feature generation step.

Algorithm 1 Feature Generation

Input : Original Features F_d, η_1
Output: Newly constructed feature space F_N
$i = 0, j = 0, F_N = \phi$ **while** $i < d$ **do**
  while $j < d$ **do**
    if *($i \neq j$) and ($dcor(F_i, F_j) \neq 0$)* **then**
      if *($dcor(F_i, F_j) > 0$) and ($dcor(F_i, F_j) < \eta_1$)* **then**
        $F_N \leftarrow F_N \cup SVR(F_{\mathrm{i}}, F_{\mathrm{j}})$
      end
      if *($dcor(F_i, F_j) \geq \eta_1$) and ($dcor(F_i, F_j) \leq 1$)* **then**
        $F_N \leftarrow F_N \cup LR(F_{\mathrm{i}}, F_{\mathrm{j}})$
      end
    end
    j++
  end
  i++
end
return F_{N}

(3) Feature Selection: All the newly constructed features F_{N} might not be of equal importance. We use stability based selection (Meinshausen & Bühlmann, 2008) only on the newly constructed features to choose a subset of these features as it helps in increasing the classification accuracy by eliminating irrelevant features and prevents overfitting. In stability selection, data is perturbed several times (by iteratively sub-sampling the examples). For each perturbation, any learning model like regression, SVM etc. that produces sparse coefficients is applied to a sub-sample of the data. After a number of iterations, all features that were selected in a large fraction of the perturbations are chosen. Finally, a cutoff threshold η_2 ($0 < \eta_2 < 1$) is applied in order to select the most stable features. In this paper, instead of proposed lasso (Tibshirani, 1994) we use Randomized lasso (Wang et al., 2011) for stability selection.

3. Experimental Setup and Results

We compare our model with 2 other top performing models, namely, FCTree (FCT) and TFC on 20 datasets (shown in Table 1) from diverse domains. The performances reported are measured in terms of accuracy on the actual feature space (ORIG) and then on the supplemented fea-

ture space (i.e original features combined with the features learned) which is also the interpretation in prior literature (Piramuthu & Sikora, 2009; Fan et al., 2010). Our model results for the supplemented feature space are mentioned under **AL*** column in Table 2-3. The accuracy is reported after 5-fold cross-validation on 8 state-of-the-art classification (CLF) algorithms namely KNN, Logistic Regression (LR), SVM with linear kernel (SVM-L), SVM with polynomial kernel (SVM-P), Random Forest (RF), Adaboost (AB), Multi Layered Perceptron (NN) and Decison Tree (DT). Note that for both feature generation and feature selection only the training set in each fold of cross-validation was used. The accuracy of the classifiers is computed using the scikit-learn's (Pedregosa et al., 2011) default parameters. The default scikit-learn parameters were used for LR and SVR as well. For SVR, we use rbf kernel as it provides good generalization capabilities. The parameter value for η_1 was set at 0.7 since this value best segregates linear and non linear correlations as discussed in (Szekely et al., 2007). The value of parameter η_2 was determined by doing grid search on values $\{0.05, 0.1, 0.15, 0.2, ..., 0.7\}$. It is apparent from the results, demonstrated in Table 2-3 that for most of the scenarios, the classifiers can achieve much higher accuracy using our proposed model as compared to the original features, TFC and FCTree methods. We achieved an improvement of **12.17%** in accuracy over the original feature space across 20 datasets and 8 different classifiers. In order to verify that this improvement is mainly due to feature construction step rather than feature selection, we applied feature selection alone on the original feature space which resulted in an overall improvement of **3.93%**. This difference in overall improvement proves that the major improvement is due to our feature construction step.

Table 1. Characteristics of 20 datasets

Dataset Name	Features	Instances	Labels
Abalone	7	4177	3
Arcene	10000	200	2
Bank Note	4	1372	2
Colon	2000	62	2
Dermatology	34	366	4
E.coli	7	336	8
FeatureFourier	76	2000	10
FeaturePixel	240	2000	10
Ionosphere	34	351	2
Letter Recognition	16	20000	26
Leukaemia	7129	72	2
Libras	90	360	15
Lung Cancer	12600	203	2
Lymphoma	4026	96	9
Ovarian Cancer	15154	253	2
Poker	10	1025010	10
Prostate Cancer	5966	102	2
Shuttle	10	58000	7
Sonar	60	208	2
Wine	13	178	3

4. Scalability Analysis

Our scalability analysis experiments show the final number of new features after construction and selection. This en-

Table 2. Accuracy comparison across 8 classifiers on 10 datasets

Dataset	CLF	ORIG	TFC	FCT	AL*
Abalone	KNN	**23.27**	21.64	22.60	22.71
	LR	24.61	23.69	23.90	**25.50**
	SVM-L	25.71	25.64	25.72	**26.07**
	SVM-P	19.46	17.64	22.12	**22.77**
	RF	22.91	18.78	**23.02**	22.11
	AB	**20.61**	19.10	19.97	20.11
	NN	27.53	26.32	26.41	**27.81**
	DT	19.27	19.00	19.13	**19.41**
Arcene	KNN	80.5	80.5	80.5	**82.50**
	LR	**86.00**	84.00	84.00	85.50
	SVM-L	**88.50**	86.50	87.50	86.50
	SVM-P	**88.00**	87.00	86.00	85.50
	RF	76.50	76.23	76.92	**77.90**
	AB	72.50	74.00	75.00	**77.12**
	NN	65.50	68.97	69.95	**82.00**
	DT	69.00	69.00	69.00	**72.50**
Bank	KNN	99.92	99.27	99.52	**99.70**
	LR	98.90	97.95	98.68	**99.70**
	SVM-L	98.76	98.97	99.27	**99.92**
	SVM-P	98.90	98.27	99.16	**99.63**
	RF	99.05	98.27	98.75	**99.56**
	AB	99.63	99.27	**99.78**	99.48
	NN	**100.00**	99.02	99.02	99.92
	DT	98.25	98.12	98.56	**99.19**
Colon	KNN	78.97	79.34	79.56	**80.38**
	LR	75.38	74.68	75.12	**78.18**
	SVM-L	75.38	74.07	**76.02**	74.10
	SVM-P	**73.97**	70.16	71.29	70.69
	RF	70.64	71.37	71.73	**72.30**
	AB	72.56	71.23	73.06	**75.76**
	NN	62.82	65.67	69.12	**78.84**
	DT	75.89	75.16	**76.27**	73.97
Dermat.	KNN	89.11	90.46	92.89	**96.09**
	LR	97.21	97.76	97.97	**98.61**
	SVM-L	**97.21**	96.02	96.27	96.92
	SVM-P	**94.41**	94.00	94.12	93.56
	RF	**96.92**	96.45	96.61	95.81
	AB	54.13	57.12	**61.00**	54.96
	NN	98.04	97.13	97.22	**98.22**
	DT	**95.24**	95.06	94.96	94.68
E.coli	KNN	86.59	**88.42**	87.56	84.82
	LR	75.88	78.23	79.24	**87.19**
	SVM-L	85.71	85.71	85.71	**86.30**
	SVM-P	56.54	59.32	62.14	**80.33**
	RF	82.73	83.46	83.76	**86.59**
	AB	62.47	63.54	64.37	**65.75**
	NN	78.28	80.37	81.97	**86.90**
	DT	**79.74**	76.32	77.67	76.48
Fourier	KNN	83.85	82.17	**83.82**	83.55
	LR	79.45	79.97	80.00	**82.15**
	SVM-L	81.45	81.15	82.86	**83.05**
	SVM-P	8.70	42.25	57.97	**79.30**
	RF	**79.9**	78.90	79.16	79.31
	AB	48.65	46.66	49.29	**50.40**
	NN	81.90	82.34	83.12	**84.50**
	DT	74.00	74.00	**74.35**	**74.35**
Pixel	KNN	97.75	98.12	97.23	**97.95**
	LR	94.35	94.22	94.28	**95.75**
	SVM-L	92.9	92.57	93.26	**94.27**
	SVM-P	98.35	98.22	**98.66**	97.25
	RF	95.5	94.26	**95.12**	94.20
	AB	54.05	54.00	54.86	**55.60**
	NN	97.15	97.15	**97.75**	**97.75**
	DT	87.30	86.12	86.78	**86.95**
Ionosp	KNN	84.31	84.66	**84.87**	83.46
	LR	87.44	87.26	87.39	**87.95**
	SVM-L	87.44	86.71	**87.78**	84.30
	SVM-P	64.10	70.16	71.45	**74.63**
	RF	93.15	91.65	**93.16**	92.30
	AB	92.02	90.94	90.12	**92.43**
	NN	**93.14**	92.45	92.13	92.29
	DT	88.32	87.12	88.04	**88.59**
Letter	KNN	95.02	95.00	**95.96**	**95.96**
	LR	71.66	77.42	80.07	**83.71**
	SVM-L	54.96	57.98	58.81	**61.23**
	SVM-P	95.01	95.12	95.26	**96.52**
	RF	93.96	93.74	93.79	**94.14**
	AB	27.82	28.00	29.07	**30.38**
	NN	91.67	92.45	93.45	**95.22**
	DT	87.78	88.03	88.34	**90.12**

Table 3. Accuracy comparison across 8 classifiers on 10 datasets

Dataset	CLF	ORIG	TFC	FCT	AL*
Leukae.	KNN	82.09	85.31	87.12	**94.27**
	LR	84.76	87.64	91.21	**93.75**
	SVM-L	84.26	86.83	88.92	**95.64**
	SVM-P	65.23	69.31	74.15	**81.90**
	RF	**90.28**	89.48	89.86	90.19
	AB	92.95	92.11	92.49	**93.15**
	NN	86.09	90.11	91.37	**93.81**
	DT	83.33	83.95	84.13	**86.00**
Libras	KNN	70.00	71.00	**71.18**	69.44
	LR	60.27	64.68	67.12	**70.00**
	SVM-L	68.61	69.88	**70.83**	67.22
	SVM-P	2.22	36.68	47.97	**49.44**
	RF	71.94	72.12	**73.07**	70.22
	AB	8.05	10.12	13.11	**18.05**
	NN	71.66	72.35	74.24	**76.33**
	DT	62.5	62.64	63.12	**65.55**
Lung	KNN	88.88	88.96	89.88	**92.61**
	LR	91.93	93.14	91.96	**93.92**
	SVM-L	91.95	92.64	92.66	**93.79**
	SVM-P	90.09	88.46	**90.32**	88.81
	RF	87.03	86.47	88.62	**90.03**
	AB	82.72	83.01	83.17	**83.33**
	NN	74.69	76.88	79.43	**90.44**
	DT	88.30	**89.17**	88.75	85.22
Lymp.	KNN	87.42	87.67	**88.12**	84.26
	LR	**87.52**	86.07	86.31	86.42
	SVM-L	85.47	85.37	85.31	**88.52**
	SVM-P	58.26	60.37	62.49	**68.73**
	RF	76.05	77.34	76.46	**79.05**
	AB	52.15	51.46	**52.71**	50.84
	NN	83.36	84.65	85.12	**85.36**
	DT	63.42	64.34	64.89	**68.73**
Ovarian	KNN	93.29	95.64	95.97	**97.12**
	LR	**100.00**	99.00	99.00	**100.00**
	SVM-L	**100.00**	99.24	99.41	**100.00**
	SVM-P	64.01	67.23	69.21	**91.66**
	RF	97.23	96.27	95.46	**97.62**
	AB	98.80	97.45	98.12	**99.27**
	NN	54.50	59.62	63.12	**88.41**
	DT	94.47	95.37	96.12	**97.63**
Poker	KNN	61.76	58.27	62.78	**64.14**
	LR	50.11	54.62	57.47	**59.46**
	SVM-L	49.74	46.54	47.44	**50.13**
	SVM-P	58.32	59.23	**61.17**	60.02
	RF	68.1	70.32	71.51	**72.26**
	AB	35.76	36.23	37.00	**39.46**
	NN	**99.72**	99.57	99.57	99.57
	DT	63.81	64.33	64.16	**66.17**
Prostate	KNN	87.23	88.25	89.32	**91.19**
	LR	90.19	90.89	91.46	**92.14**
	SVM-L	91.14	90.46	**91.19**	90.19
	SVM-P	**91.19**	88.12	89.97	88.47
	RF	86.28	87.16	88.30	**91.19**
	AB	88.19	88.30	89.12	**90.19**
	NN	50.90	54.67	59.12	**86.69**
	DT	77.61	77.21	78.37	**79.42**
Shuttle	KNN	99.78	99.02	99.57	**99.92**
	LR	92.90	93.31	93.76	**96.28**
	SVM-L	90.41	91.19	92.18	**96.57**
	SVM-P	**99.92**	99.81	99.81	99.88
	RF	**99.97**	99.72	99.72	99.81
	AB	87.50	89.17	90.42	**92.46**
	NN	99.71	99.52	99.71	**99.98**
	DT	**99.98**	99.18	99.52	99.67
Sonar	KNN	78.35	81.48	82.70	**83.19**
	LR	77.42	78.12	78.72	**79.00**
	SVM-L	73.54	74.54	75.75	**77.30**
	SVM-P	53.36	58.41	66.44	**81.71**
	RF	73.55	81.00	**82.54**	77.87
	AB	80.74	80.00	**81.04**	78.83
	NN	80.30	81.07	82.00	**84.09**
	DT	75.01	74.23	74.52	**75.02**
Wine	KNN	67.93	74.89	79.93	**90.12**
	LR	95.52	96.89	97.24	**98.30**
	SVM-L	83.03	88.14	89.94	**91.31**
	SVM-P	96.65	**96.68**	96.65	94.68
	RF	96.07	96.68	**97.12**	**97.12**
	AB	85.82	**88.12**	91.23	85.71
	NN	42.73	46.23	49.56	**87.19**
	DT	91.57	91.79	92.01	**93.02**

sures that classification algorithms that use these features as input are not burdened with the curse of dimensionality, and are hence scalable. The approach submitted by us selects a very small number of new features to reach the desired performance, thus making our model *scalable*. This is in contrast to the thousands of features considered by models suggested in prior literature. Figure 1 illustrates the number of features finally selected by our learning model against the number selected by FCTree and TFC on 20 datasets. Clearly, we can conclude that the number of features required by our model are significantly less. Our model on an average, uses at least *six times* lesser number of features than the best performing proposed techniques across all datasets.

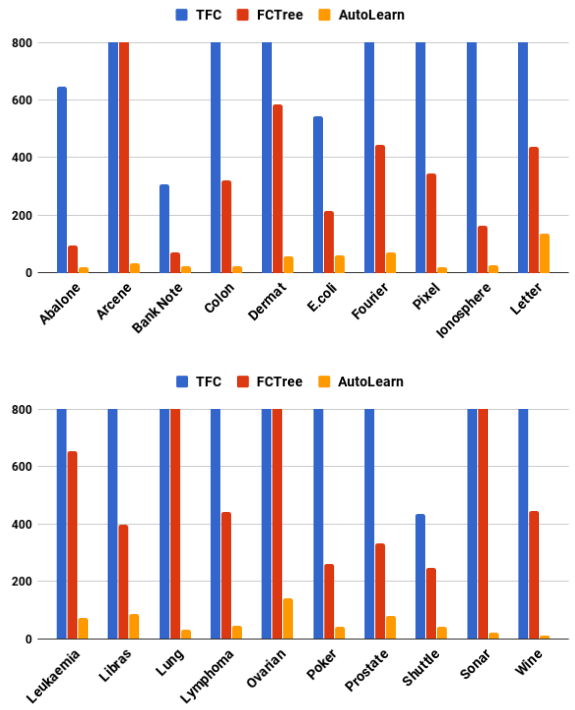

Figure 1. Scalability Analysis on 20 datasets

5. Conclusion

This paper studies how to efficiently and automatically generate and select highly predictive features that can be generalized over a number of classifiers on datasets from diverse domains by picking up only relevant data points. Our analysis shows that: (1) the distance correlation can effectively mine important pairwise associations; (2) correlated features assist the regression model to construct highly predictive features without domain related heuristics; (3) a set of features can be selected, without running into exhaustive search via stability based selection, that prevents overfitting and improves model generalization.

References

Fan, Wei, Zhong, Erheng, Peng, Jing, Verscheure, Olivier, Zhang, Kun, Ren, Jiangtao, Yan, Rong, and Yang, Qiang. Generalized and heuristic-free feature construction for improved accuracy. In *SDM*, 2010.

Hinton, Geoffrey, Deng, Li, Yu, Dong, Dahl, George E, Mohamed, Abdel-rahman, Jaitly, Navdeep, Senior, Andrew, Vanhoucke, Vincent, Nguyen, Patrick, Sainath, Tara N, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29 (6):82–97, 2012.

Matheus, Christopher J. and Rendell, Larry A. Constructive induction on decision trees. In *IJCAI*, 1989.

Meinshausen, Nicolai and Bühlmann, Peter. Stability selection. 2008.

Pagallo, Giulia. Learning dnf by decision trees. In *IJCAI*, 1989.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Piramuthu, Selwyn and Sikora, Riyaz T. Iterative feature construction for improving inductive learning algorithms. In *Expert Syst. Appl.*, pp. 34013406, 2009.

Russakovsky, Olga, Deng, Jia, Su, Hao, Krause, Jonathan, Satheesh, Sanjeev, Ma, Sean, Huang, Zhiheng, Karpathy, Andrej, Khosla, Aditya, Bernstein, Michael, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Schlimmer, Jeffrey C. and Granger, Richard. Incremental learning from noisy data. *Machine Learning*, 1:317–354, 1986.

Szekely, Gabor, Rizzo, Maria, and Bakirov, Nail. Measuring and testing dependence by correlation of distances. In *Annals of Statistics*, 2007.

Tibshirani, Robert. Regression shrinkage and selection via the lasso. 1994.

Wang, Sijian, Nan, Bin, Rosset, Saharon, and Zhu, Ji. Random lasso. *Annals*, 5(1):468–485, 2011.