Fastfood Elastic Net: Combining Variable Selection with Kernel Expansion Approximations

Sonia Kopel(✉), Kellan Fluette, Geena Glen, and Paul E. Anderson

College of Charleston, Charleston, USA
{kopels,fluetteka}@g.cofc.edu, gmglen6@gmail.com, andersonpe2@cofc.edu
http://anderson-lab.github.io/

Abstract. As the complexity of a prediction problem grows, simple linear approaches tend to fail which has led to the development of algorithms to make complicated, nonlinear problems solvable both quickly and inexpensively. Fastfood, one of such algorithms, has been shown to generate reliable models, but its current state does not offer feature selection that is useful in solving a wide array of complex real-world problems that spans from cancer prediction to financial analysis.

The aim of this research is to extend Fastfood with variable importance by integrating with Elastic net. Elastic net offers feature selection, but is only capable of producing linear models. We show that in combining the two, it is possible to retain the feature selection offered by the Elastic net and the nonlinearity produced by Fastfood. Models constructed with the Fastfood enhanced Elastic net are relatively quick and inexpensive to compute and are also quite powerful in their ability to make accurate predictions.

Keywords: Kernel methods · Data mining · Algorithms and programming techniques for big data processing

1 Introduction

The value of effective prediction methods is self-evident: being able to predict future outcomes can be applied to nearly any field. The methods and algorithms that are used to generate these predictive models continue to evolve to handle large, complicated real-world datasets that have high dimensionality and large sample sizes. The fields that these datasets come from range from stock market and financial analysis to disease screening to weather prediction. For such datasets, many of the machine learning techniques that are commonly used to generate models either fail or are too expensive in either their required storage space or their runtime, rendering them inefficient.

More data offers the ability to train better, more realistic models, but the cost of generating these models is often computationally intractable because of the scope of the mathematical operations that a computer must perform during

P.M. Pardalos et al. (Eds.): MOD 2016, LNCS 10122, pp. 427–432, 2016.
DOI: 10.1007/978-3-319-51469-7_37

training. As a result, these more sophisticated models often cannot be computed in real-time, rendering them useless in many disciplines. Storage space is another concern. While the computer running the algorithms may be able to hold terabytes of data, it cannot hold a square matrix of corresponding dimensions. Thus, even in the rare cases when time is plentiful, the required computing resources to generate models from large datasets is unavailable to many researchers.

Linear techniques, such as Elastic net [6], tend not to succumb to these pitfalls because of their relative simplicity. However, complex datasets with hundreds, if not thousands, of features are unlikely to have linear decision boundaries. As a result, the linear models produced by these techniques are quite often unreliable or inaccurate.

One of the simpler methods for finding nonlinear decision boundaries is the kernel trick. The kernel trick transforms the data by implicitly mapping the features into a higher, possibly infinite, dimension and from there calculating a linear decision boundary [6]. For instance, if a dataset is not linearly separable in two dimensions, it can be transformed into a higher dimensional space. Depending on the function used to transform the data, it may be possible to find a linear decision boundary in this new feature space. The trick to this technique lies in the fact that the mapping function (ϕ) need never be explicitly defined. Rather, the individual points can be transformed by taking their dot product with a known kernel function (k), like the sigmoid function or the radial basis function [3]. The relationship between ϕ and k is as follows:

$$\langle \phi(x), \phi(y) \rangle = k(x, y) \tag{1}$$

$$f(x) = \langle w, \phi(x) \rangle = \sum_{i=1}^{N} a_i k(x_i, x) \tag{2}$$

Unfortunately, the kernel trick may also become intractable to compute as the computation and storage requirements for the kernel matrix are exponentially proportional to the number of samples in the dataset [3]. However, the Random Kitchen Sinks (RKS) algorithm chooses to approximate the kernel function more effectively by randomizing features instead of optimizing them [5]. It does this by randomly selecting these features from a known distribution. The authors show that with this method shallow neural nets achieve comparable accuracy to AdaBoost, a popular ensemble method which adjusts weak learners in favor of instances misclassified by previous classifiers [2]. Overall, RKS has comparable predictive ability to the commonly used AdaBoost, but does not require the same level of rigor on the part of the computer during training. However, RKS makes use of dense Gaussian random matrix multiplications which are computationally expensive. Le and Smola mitigate this problem by replacing these multiplications by multiplying several diagonal matrices in their algorithm: Fastfood [3].

Like Fastfood, Elastic net can be combined with machine learning techniques. One implementation of Elastic net combines it with support vector machines (SVMs) yielding a substantial performance improvement without sacrificing accuracy [6]. Our work builds directly upon the Fastfood algorithm, which while

capable of generating models quickly does not provide a variable importance measure or feature section. Herein, we describe our research into developing a computationally efficient variable importance measure for Fastfood by leveraging the built-in feature selection of Elastic net [6]. Our results indicate that models generated with our improved variation of Fastfood retains the benefits of the Fastfood while also providing variable importance, which is not available in the standard Fastfood algorithm.

2 Methods

In order to generate nonlinear models with feature selection, we combine two existing algorithms: Fastfood and Elastic net. Our method, Fastfood Elastic Net (FFEN) retains the nonlinearity of Fastfood while incorporating feature selection of Elastic net to provide a variable importance measurement.

To reduce the already quick run-time complexity of RKS, Fastfood combines Hadamard matrices with Gaussian scaling matrices [3]. The algorithm replaces the Gaussian random matrices in RKS with this combination. Because of the relative ease of these computations in contrast multiplying to Gaussian random matrices, the complexity of the runtime is reduced from $O(nd)$ to $O(n \log d)$ [3]. This allows Fastfood to approximate the kernel feature map quickly. The main takeaway from this is that unlike in the kernel trick where ϕ is never defined, in Fastfood (and RKS) it is approximated.

Another appeal of Fastfood is that it has previously been combined with other machine learning techniques, most notably neural nets. By implementing the algorithm at each layer of the neural net, additional nonlinearity is added to the training of the neural net and the training process is also sped up [1].

Despite the successes of Fastfood in making accurate predictions in loglinear time, it is unable to inherently measure variable importance and perform variable selection because it relies on kernel approximation and projection into a new feature space. Elastic net, however, implicitly provides variable selection, but is a linear technique characterized in Eq. 3 [6]. Thus, Elastic net in isolation is not capable of creating a predictive model for complicated datasets with nonlinear decision boundaries.

$$\hat{\beta} = \arg\min_{\beta}(\|y - \mathbf{X}\beta\|^2 + \lambda\|\beta\|^2 + \alpha\|\beta\|_1) \tag{3}$$

where α and λ are parameters to be specified by the user, X is the data matrix, y are the labels or dependent variable, and β is a vector of coefficients representing the model.

Our method takes the original feature matrix (X) of size ($n \times d$) and applies Fastfood. This transforms the original features to a new feature matrix (F) of size ($n \times p$), where $p \geq d$. In this step, Fastfood approximates ϕ to transform X into a higher dimensional space.

$$\underset{n\times p}{F} = \underset{n\times d}{X} \times \underset{d\times p}{\phi} \tag{4}$$

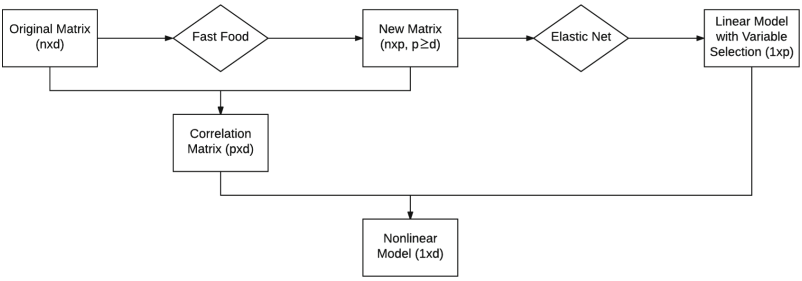

Fig. 1. Process

Elastic net is applied to this new feature matrix to generate a model with variable selection (L). Optimization of parameters λ and α within the Elastic Net algorithm are also optimized throughout the process. To reduce back to the original dimensionality, we calculate a correlation matrix (ρ) between the new features and the original features. Finally, these correlations are aggregated to reduce and relate the Elastic net coefficient vector of size $(1 \times p)$ to the original feature space of size $(1 \times d)$. The overall process of FFEN is shown in Fig. 1.

$$\underset{1\times d}{O} = \underset{1\times p}{L} \times \underset{p\times d}{\rho} \tag{5}$$

This method can then be run iteratively to remove extraneous variables or a single run can be used to integrate variable importance in the original dimensional space with the kernel approximation basis space.

3 Results and Discussion

For evaluation, tests have been performed on both simulated linear datasets as well as real datasets from the UCI machine learning repository [4]. The simulated datasets had random variables masked, providing a known gold standard for feature selection and variable importance. Our Fastfood enhanced Elastic net (FFEN) was compared to lasso: a common implementation of Elastic net. To keep the comparisons fair, parameters λ and α for lasso were optimized in the same way as they were for FFEN. For the simulated linear data, the true β values for each variable were known, so the primary metric that was used to compare the FFEN model to lasso was average mean absolute error (MAE) for all of the β values. Table 1 depicts the results for simulated datasets of differing dimensions.

The results indicate that FFEN improves as the number of samples increases and the number of dimensions decreases. To justify this, we repeated our simulation 20 times on datasets of varying size (see Table 1). The average MAE for FFEN was 0.03097 as opposed to 0.0473 for lasso. Given the small standard

Table 1. Simulated linear data

n	d	FFEN MAE	Lasso MAE
400	100	0.2103	0.0401
1000	80	0.1162	0.0301
4000	10	0.0471	0.0300
5000	10	0.0234	0.0400

deviations, 0.007572 and 0.00725 respectively, we can with more than 99% certainty conclude that FFEN outperforms lasso on simulated linear datasets with those dimensions which we confirmed by performing a paired T-Test.

To compare performance on real-world datasets, we generated 20 different training and test sets, and ran both FFEN and lasso on each set. We then compared R^2 values and ran a paired T-Test to test to see which model was better. Table 2 depicts the average R^2 values for the 20 runs as well as the standard deviations (σ) for these runs. FFEN performed slightly worse than lasso for the KEGG Metabolic Reaction Network Dataset though the performance dropped slightly from approximately 0.90 to 0.88. The high value of R^2 for lasso indicates that this is a linear dataset, and therefore, FFEN provides marginal benefits. However, FFEN significant outperforms lasso on the Physicochemical Properties of Protein Tertiary Structure Dataset (Protein) and also outperformed lasso on the UCI million song dataset. Because UCI provided a preferred training and test set partition for the UCI million song dataset, only one run was needed to generate the results. The R^2 values for FFEN and lasso from this run are also depicted in Table 2.

Table 2. Empirical data

Dataset	n	d	FFEN R^2	FFEN σ	Lasso R^2	Lasso σ
Protein	45729	9	0.3366	0.0268	0.2387	0.0038
KEGG	64608	27	0.8754	0.0120	0.8992	0.0023
Music	515345	90	0.2245		0.2098	

In conclusion, FFEN has been shown to offer promising results. The models generated by our algorithm are comparable or better than those of Elastic net on simulated linear datasets when measuring the accuracy of the resulting coefficients. Further, FFEN shows improved accuracy when appropriate complex nonlinear datasets while incorporating novel variable importance not available in standard Fastfood. The use of Fastfood allows us to generate these models both quickly and inexpensively.

In the future, we hope to adjust the algorithm more efficiently and accurately perform feature selection. In addition, we are working on testing a wider range

of datasets with varying sample sizes and dimensionality. Further, we would like to compare the runtime of this algorithm to other runtime optimized machine learning algorithms, such as Elastic net enhanced SVMs (SVEN).

Acknowledgments. The authors would like to thank the College of Charleston for hosting the NSF Omics REU which is funded by the National Science Foundation DBI Award 1359301 as well as the UCI machine learning repository [4].

References

1. Bowick, M., Neiswanger, W.: Learning fastfood feature transforms for scalable neural networks. In: Proceedings of the International conference on..., pp. 1–15 (2013)
2. Freund, Y., Schapire, R.R.E.: Experiments with a new boosting algorithm. In: International Conference on Machine Learning, pp. 148–156 (1996)
3. Le, Q., Sarlós, T., Smola, A.J.: Fastfood – approximating kernel expansions in loglinear time. Int. Conf. Mach. Learn. **28**(1), 1–29 (2013)
4. Lichman, M.: UCI machine learning repository (2013)
5. Rahimi, A., Recht, B.: Weighted sums of random kitchen sinks: replacing minimization with randomization in learning. Adv. Neural Inf. Process. Syst. **1**(1), 1–8 (2009)
6. Zou, H., Hastie, T.: Journal of the Royal Statistical Society. Series B: Statistical Methodology **28** (2013)