Intra-Feature Random Forest Clustering

Michael Cohen
mkc1000@gmail.com
767 Rhode Island St.
San Francisco, CA 94107
Tel: +1-415-963-2163

Abstract

Clustering algorithms are commonly used to find structure in data without explicitly being told what they are looking for. One key desideratum of a clustering algorithm is that the clusters it identifies given some set of features will generalize well to features that have not been measured. Yeung, et al. (2001) introduce a Figure of Merit closely aligned to this desideratum, which they use to evaluate clustering algorithms. Broadly, the Figure of Merit measures the within-cluster variance of features of the data that were not available to the clustering algorithm. Using this metric, Yeung, et al. found no clustering algorithms that reliably outperformed k-means on a suite of real world datasets (Yeung, 2001). This paper presents a novel clustering algorithm, intra-feature random forest clustering (IRFC), that does outperform k-means on a variety of real world datasets per this metric. IRFC begins by training an ensemble of decision trees of limited depth to predict randomly selected features given the remaining features. It then aggregates the partitions that are implied by these trees, and outputs however many clusters are specified by an input parameter.

Keywords

cluster analysis, random forest, unsupervised learning, ensemble, figure of merit

Acknowledgements

This research for this paper was supported financially by Galvanize, Inc.

Introduction

One of the central challenges for unsupervised learning has been the lack of a universally accepted validation metric (Giancarlo, et al., 2008). Giancarlo, et al. review several possible validation techniques, but they only evaluate those validation metrics by their ability to identify how many clusters a dataset should be partitioned into given some ground truth. Furthermore, many of those validation measures are only coincident with the key desiderata of a clustering algorithm. Singh, et al. (1988) articulate the aim of clustering as follows: “The purpose of cluster analysis is to place objects into groups suggested by the data such that objects in a given group have tendency to be similar to each other, and objects in different clusters tend to be dissimilar.” Of the seven metrics that Giancarlo, et al. (2008) review, three of them only consider the stability of cluster assignments under modified conditions: Clest (Dudoit, et al., 2002) considers stability of clusters after effectively subsetting the data, Consensus Clustering (Monti, et al., 2003), the stability after bootstrapping, ME (Model Explorer) (Ben-Hur, et al., 2002), the stability after adding random noise. While stability of assignments is certainly a nice feature, it is obviously not the primary aim of a clustering algorithm, or else we could create a perfect clustering algorithm that outputs the same assignments every time, completely arbitrarily. Three other metrics considered by Giancarlo, et al. only validate cluster assignments with data the clustering algorithm had access to: WCSS (Within Cluster Sum-of-Squares) (Kaufman, et al., 2009), Gap (Tibshirani, et al., 2001), and KL (Krzanowski, et al., 1988). These approaches have analogous risks to validating on the training set in supervised learning, as opposed to withholding a test set. Yeung’s Figure of Merit (Yeung, 2001), the remaining metric considered by Giancarlo, et al., directly measures the suitability of a partition according to Singh’s articulation of the purpose of clustering. To show this, we first review how the Figure of Merit is calculated, then we present a wide variety of examples from the literature of applications of clustering algorithms, and demonstrate that in all these applications, a clustering algorithm is useful insofar as it scores well according to Yeung’s Figure of Merit. Naturally, that is strong support for the utility of this metric.

To compute the Figure of Merit, features are withheld one by one, analogously to k-fold cross validation. Then, the clustering algorithm is applied to the remaining features, and the within-cluster variance of the withheld feature is divided by the total variance of that feature. Finally, that quantity is averaged over all features, again analogously to k-fold cross validation. An important distinction, however is that while k-fold cross validation withholds data points one at a time, this method withholds features. A lower Figure of Merit indicates “that objects in a given group have tendency to be similar to each other, and objects in different clusters tend to be dissimilar,” (Singh, 1998) even with respect to features the clustering algorithm did not have access to.

Consider the following applications of clustering. Hilas (2008) et al. use clustering to detect telecommunication fraud. The clustering algorithm is therefore useful insofar as the clusters identified will have relatively low variance with respect to a feature the algorithm did not have access to: the Boolean value of whether the event was fraudulent. Masulli (1999) use clustering of medical images to support diagnosis. The algorithm is useful insofar as members of a given group have lower variance with respect to their diagnosis than the entire sample does. The true diagnosis, again, is a feature the algorithm did not have access to. Iliadis (2005) use clustering to identify forest types to assist in fire risk estimation. The algorithm is useful insofar as clusters are created in such a way that the variance of a new feature (whether there is a fire, in this case) is minimized within clusters. The utility of Li, et al.'s (2009) transcriptomic clusters is their ability to discriminate glioma subtypes, a feature not available to the clustering algorithm. Harrigan's (1985) use of clustering to identify "strategic groups" among competitors in an industry is relevant insofar as companies in the same clusters deserve strategic treatment that is more similar than that of companies in different clusters. Companies in the same cluster are expected to respond similarly to a broad range of treatments, and to a greater degree than two companies selected at random, so whatever metrics are used for this, there should be lower within-cluster variance than total variance. Becker et al. (2011) identify clusters in people flow using cellular data, and use categories of movement patterns to evaluate the comparative utility of different urban developments to members of different clusters. If the variance of individuals' utilities from Project X is no less for members of a particular cluster than it is for the population as a whole, the cluster assignments are unhelpful. When Chicco, et al. (2003) cluster electricity customers, they expect members of any given cluster to respond similarly to service regulations, but differently from members of other clusters. Again, this feature is not included in the original clustering, and the clustering algorithm is useful insofar as the clusters identified minimize within-cluster variance with respect to that new feature. Wang (2010) describes the utility of unsupervised market segmentation to the service industry. The utility arises from the expectation that customers in different market segments will respond uniquely to different sorts of targeting. Therefore, "customer response to Campaign X" needs to have a mean within-cluster-variance that is lower than the total variance in order for the clustering to be useful. Pham (1998) demonstrates the utility of clustering radar signals for the identification of aircrafts, aircraft identity being feature the clustering algorithm was not provided. Pavlidis, et al. (2003) use clustering in financial forecasting; their clustering algorithm obviously does not have access to future financial data, but it is tasked with making partitions that identify data points with similar future-behavior. Park's (2002) forecasting task is to predict freeway traffic with the assistance of unsupervised methods, and the case is analogous to financial forecasting. This is a miniscule sample of the applications of clustering, but they begin to support the following generalization: cluster assignments are likely to be useful when and only when novel features tend to have low within-cluster-variance compared to their total variance.

An extraordinary variety of clustering algorithms have been proposed (Xu, et al., 2005), as well as many cluster ensembling methods for cluster analysis (Vega-Pons, et al., 2011). IRFC represents a single clustering algorithm that implements the driving thesis of cluster ensembling: an ensemble of partitions, benefitting from a wisdom-of-crows effect, will generally outperform a single partition (Vega-Pons, et al., 2011). IRFC, unsurprisingly given the name, also borrows extensively from the supervised learning algorithm Random Forest, in that both make use of an ensemble of decision tree regressors (Breiman, 2001). The strong performance of IRFC with respect to Yeung's Figure of Merit is quite analogous to the strong performance of a random forest regressor with respect to the root-mean-square error of its predictions.

This paper first lays out the algorithm for IRFC, then describes its performance according to Yeung's Figure of Merit on a suite of real world data.

The Algorithm

IRFC consists of two stages: train an ensemble of limited-depth decision trees to derive an ensemble of partitions, then aggregate the partitions into a single one. For the first stage, the following parameters are used. nTrees is the number of trees to use. MaxDepth is the maximum depth for each decision tree. PredictionFraction is the fraction of features that are to be trained on for each tree. SetOfPoints is a matrix where each row is a data point and each column is a feature.

```
RandomForestTransform (SetOfPoints, nTrees, MaxDepth, PredictionFraction)

Output  := empty matrix
nFeatures := PredictionFraction * (number of features in SetOfPoints)
FOR i in nTrees DO
        TempY := ChooseRandomFeatures(nFeatures, SetOfPoints)
        // ChooseRandomFeatures returns nFeatures columns, selected randomly from SetOfPoints
        TempX := remaining columns from SetOfPoints
        Assignments := DecisionTreeRegressor(MaxDepth, TempX, TempY)
        // DecisionTreeRegressor returns the leaf_id of each data point.
        // DecisionTreeRegressor trains a decision tree as in a random forest: bootstrapping data, and randomly restricting which features can be split on at any given node.
        column i of Output := Assignments
END FOR
RETURN Output
```

For the second stage of the algorithm, k-medoids is employed (optionally Minkowski weighted), using the Jaccard distance between the rows in the matrix returned by RandomForestTransform. In this circumstance, the Jaccard distance between two points represents the fraction of trees for which the points are assigned to different leaves.

The computational complexity of the algorithm is $O(nk + n^2)$, where n is the number of data points, and k the number of features. nk is the complexity of the random forest stage, and n^2 the complexity of creating a Jaccard distance matrix between the transformed data points.

Performance Evaluation

The metric chosen for evaluation was Yeung's Figure of Merit, for the reasons discussed above. This metric was chosen before the algorithm was designed, to ensure that a positive result would not merely reflect the abundance of evaluation metrics for clustering algorithms. The greater the number of clusters that an algorithm outputs, the easier it is to have a small Figure of Merit. Therefore, when comparing algorithms, one must hold the number of clusters constant, and repeat over many values of the number of clusters.

Among existing clustering methods, k-means is perhaps best suited theoretically to perform well on this metric. K-means explicitly attempts to minimize the within-cluster variance of the features that it is trained on, which is plausibly an optimal heuristic for minimizing the within-cluster variance of the features it is not trained on. Indeed, Yeung, et al. (2001) found experimentally that no other algorithms they tested reliably outperformed k-means on real world data. Albaum, et al. (2011) confirmed this finding on different datasets.

IRFC was compared with k-means across four datasets and for many different values of k. For the iris dataset, the two algorithms performed equivalently, and across all datasets, the algorithms performed approximately equivalently for k equal to 2. In all other cases, IRFC outperformed k-means (Figure 1). For a description of the datasets, see Table 1.

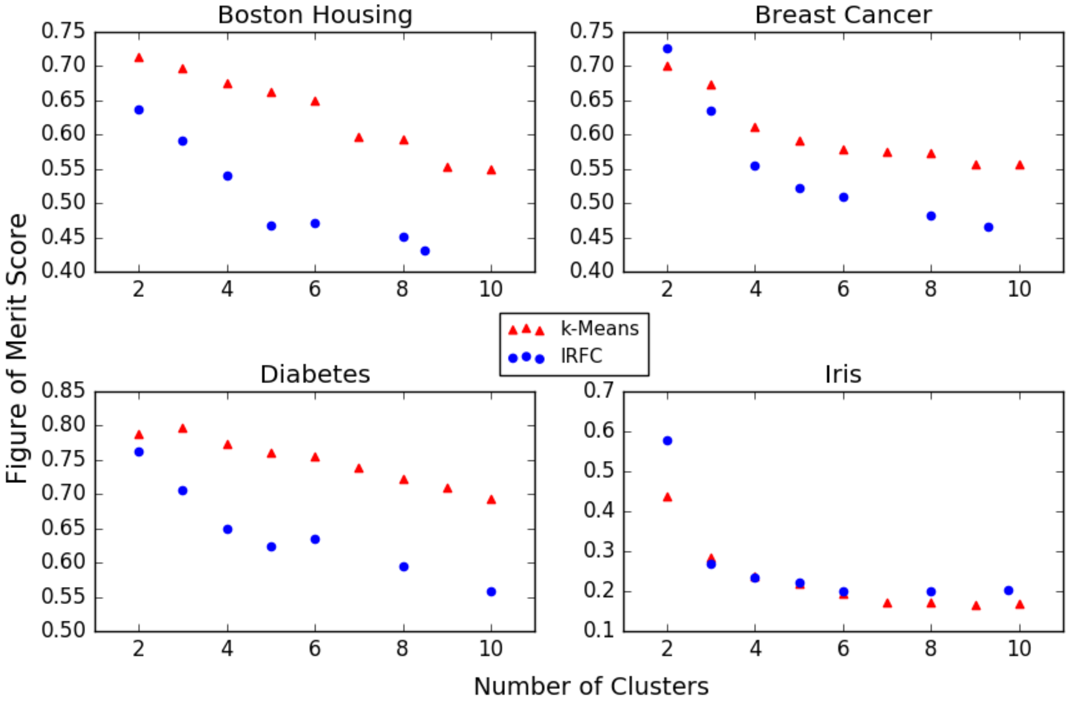

Fig. 1 *Figures of Merit for cluster assignments given the number of clusters. IRFC and k-means are compared across four standard datasets*

	Dimensions	Example Features
Boston Housing	506 x 13	Per capita crime rate (by town), average number of bedrooms per house, nitrogen oxides concentration, full-value property-tax rate per 10k
Breast Cancer	569 x 30	Mean radius (of tumors), mean area, mean concave points, worst radius, worst area, worst concave points
Diabetes	442 x 10	Age, body mass index, blood pressure
Iris	150 x 4	Petal length, petal width, sepal length, sepal width

Table 1. Descriptions of the datasets used. All features were scaled to a mean of 0 and a standard deviation of 1 prior to clustering

The roughly equivalent performance of k-means and IRFC on the iris dataset may simply reflect that the clusters in the iris dataset are very easily learned. The particularly small number of features in the iris data should also hamper IRFC. Since one feature at a time was withheld to validate the cluster assignments, both algorithms only had access to three features while clustering. In the other datasets, every decision tree in IRFC was trained on a unique subset of features, but for the iris data, there was significant redundancy, weakening the ability of IRFC to make use of the wisdom-of-crowds effect.

To acquire a single quantitative measure of the performance of an algorithm on a dataset, one can evaluate the area under the Figure of Merit curve, and divide by a normalizing factor. The resulting Integrated Figure of Merit simply compresses the information from Figure 1 (Table 2).

	K-Means	IRFC
Boston	0.658	0.531
Breast Cancer	0.627	0.570
Diabetes	0.766	0.660
Iris	0.274	0.301

Table 2. Integrated Figure of Merit for k-means and IRFC across four standard datasets

Conclusions

For k greater than 2, IRFC was found to generally outperform k-means according to Yeung, et al.'s Figure of Merit metric. While unsupervised learning may be used for other purposes, if one's goal is to predict which data points will have similar values for an unmeasured feature, IRFC is likely to be optimally effective for this task. Future research may consider the following extensions. This algorithm could be modified to output a hierarchical clustering model. Other methods besides Jaccard k-medoids could be used to aggregate the clusters. Vega-Pons, et al. (2011) review several worthy candidates for effective cluster aggregation. Other supervised learning models besides decision trees could be used in the first stage, in such a way that an implied data compression could be extracted. For example, if a neural network with a hidden layer of minimal width were used to predict one subset of features from another, the activations in the hidden layer would represent a continuous rather than discrete compression of the data. Hopefully, intra-feature random forest clustering can inspire a family of clustering algorithms that make similar use of supervised learning methods.

References

Albaum, S. P., Hahne, H., Otto, A., Haußmann, U., Becher, D., Poetsch, A., ... & Nattkemper, T. W. (2011). A guide through the computational analysis of isotope-labeled mass spectrometry-based quantitative proteomics data: an application study. *Proteome science*, *9*(1), 1.

Becker, R. A., Caceres, R., Hanson, K., Loh, J. M., Urbanek, S., Varshavsky, A., & Volinsky, C. (2011). A tale of one city: Using cellular network data for urban planning. *IEEE Pervasive Computing*, *10*(4), 18-26.

Ben-Hur, A., Elisseeff, A., & Guyon, I. (2001, December). A stability based method for discovering structure in clustered data. In *Pacific symposium on biocomputing* (Vol. 7, pp. 6-17).

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

Chicco, G., Napoli, R., & Piglione, F. (2003, June). Application of clustering algorithms and self organising maps to classify electricity customers. In *Power Tech Conference Proceedings, 2003 IEEE Bologna* (Vol. 1, pp. 7-pp). IEEE.

Dudoit, S., & Fridlyand, J. (2002). A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome biology*, *3*(7), 1.

Harrigan, K. R. (1985). An application of clustering for strategic group analysis. *Strategic Management Journal*, *6*(1), 55-73.

Hilas, C. S., & Mastorocostas, P. A. (2008). An application of supervised and unsupervised learning approaches to telecommunications fraud detection. *Knowledge-Based Systems*, *21*(7), 721-726.

Iliadis, L. S. (2005). A decision support system applying an integrated fuzzy model for long-term forest fire risk estimation. *Environmental Modelling & Software*, *20*(5), 613-621.

Kaufman, L., & Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis* (Vol. 344). John Wiley & Sons.

Krzanowski, W. J., & Lai, Y. T. (1988). A criterion for determining the number of groups in a data set using sum-of-squares clustering. *Biometrics*, 23-34.

Li, A., Walling, J., Ahn, S., Kotliarov, Y., Su, Q., Quezado, M., ... & Fine, H. A. (2009). Unsupervised analysis of transcriptomic profiles reveals six glioma subtypes. *Cancer research*, *69*(5), 2091-2099.

Masulli, F., & Schenone, A. (1999). A fuzzy clustering based segmentation system as support to diagnosis in medical imaging. *Artificial intelligence in medicine*, *16*(2), 129-147.

Monti, S., Tamayo, P., Mesirov, J., & Golub, T. (2003). Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine learning*, *52*(1-2), 91-118.

Park, B. (2002). Hybrid neuro-fuzzy application in short-term freeway traffic volume forecasting. *Transportation Research Record: Journal of the Transportation Research Board*, (1802), 190-196.

Pavlidis, N. G., Tasoulis, D. K., & Vrahatis, M. N. (2003, December). Financial forecasting through unsupervised clustering and evolutionary trained neural networks. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on* (Vol. 4, pp. 2314-2321). IEEE.

Pham, D. T. (1998, July). Applications of unsupervised clustering algorithms to aircraft identification using high range resolution radar. In *Aerospace and Electronics Conference, 1998. NAECON 1998. Proceedings of the IEEE 1998 National* (pp. 228-235). IEEE.

Singh, C., & Kim, Y. (1988). An efficient technique for reliability analysis of power systems including time dependent sources. *IEEE Transactions on Power Systems*, *3*(3), 1090-1096.

Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *63*(2), 411-423.

Vega-Pons, S., & Ruiz-Shulcloper, J. (2011). A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*, *25*(03), 337-372.

Wang, C. H. (2010). Apply robust segmentation to the service industry using kernel induced fuzzy clustering techniques. *Expert Systems with Applications*, *37*(12), 8395-8400.

Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on neural networks*, *16*(3), 645-678.

Yeung, K. Y., Haynor, D. R., & Ruzzo, W. L. (2001). Validating clustering for gene expression data. *Bioinformatics*, *17*(4), 309-318.