Exploiting Domain Structure with Hybrid Generative-Discriminative Models

Austen Kelly
University of Oregon
austenk@cs.uoregon.edu

***Abstract*—Machine learning methods often face a trade-off between the accuracy of discriminative models and the lower sample complexity of their generative counterparts. This inspires a need for hybrid methods. In this paper we present the graphical ensemble classifier (GEC), a novel combination of logistic regression and naive Bayes. By partitioning the feature space based on known independence structure, GEC is able to handle datasets with a diverse set of features and achieve higher accuracy than a purely discriminative model from less training data. In addition to describing the theoretical basis of our model, we show the practical effectiveness on artificial data, along with the 20-newsgroups and MediFor datasets.**

I. Introduction

Machine learning tasks often involve incorporating information from a variety of sources. For example, it is useful to consider network status information along with email text when attempting to identify spam emails, and when searching for fraudulent or inappropriate images posted on social media, one may want to incorporate both pixel information and metadata such as the caption and user information. A common approach to integrating these different feature types is to build complex data pipelines and custom infrastructure for the given dataset [20]. Such methods are often well tailored to the problem at hand but do not generalize well to changes in the input feature relationships or different datasets, which can lead to large overhead as code needs to be reorganized for new use over time [19].

In this paper, we present a general framework which can be used to efficiently and effectively integrate domain structure into arbitrary classification tasks. Our key observation into this problem is that multi-modal feature spaces create an inherent independence structure. Given a task with a known feature independence structure, we propose the graphical ensemble classifier (GEC), which leverages those independences in order to train smaller models, each over a subset of the original feature space.

As inspiration for the GEC design we look to a simple setting: logistic regression (LR) and naive Bayes (NB). These classic models form a well known generative-discriminative pair of probabilistic models. In particular, for a data set $X \in \mathbb{R}^{n \times m}$ with binary class labels Y, logistic regression discriminatively models the posterior distribution $P(Y|X)$ by learning weights for each feature in X. Naive Bayes, on the other hand, attempts to estimate $P(X, Y)$ under the assumption that the attributes in X are conditionally independent given the class label Y. Discriminative models are typically preferred because they tend to perform better at classification, but their generative counterparts reach asymptotic accuracy faster and thus can be useful when training data is limited [3]. We show that the GEC framework represents a class of models which span between generative and discriminative form depending on the true independence structure of the feature space.

The GEC method has a variety of useful properties. In particular, it is a linear combination of traditional models, making it simple to implement. Additionally, it is not limited to problems with multi-modal feature spaces; it conveniently generalizes to any task where independence structure between the features is known or can be well approximated. We also explore a variant based on creating an ensemble over randomized partitions for situations where domain knowledge of the feature space is unknown. Finally, we note that because GEC is a hybrid generative-discriminative model, it will have a lower sample complexity than a purely discriminative method would without loss in accuracy (given a perfect partitioning). This makes it particularly useful in domains where the amount of training data is low. As a real-world example, we apply GEC to the DARPA Media Forensics (MediFor) project dataset. The MediFor project aims to create an improved system for detecting if images have been altered. Teams from across the country have developed algorithms to detect a variety of manipulations; our contribution is to create an ensemble from the outputs of those models, using GEC to group models by similar methodology.

The rest of our paper is laid out as follows: In Section II we detail related work on In Section III we go on to describe the framework of the classification problem at hand and introduce our novel model in the context of clique trees. In Section IV we present results on a synthetic dataset, followed by results on the 20-newsgroup and MediFor datasets showing the real-world applicability and usefulness of our method, before making final conclusions and future remarks in Section V.

II. Related Work

The relationship between generative and discriminative models was first explored by Ng and Jordan [2], who showed that while discriminative models typically outperform generative ones, that is not true when training data is limited. Later work attempted to bridge this gap between generative and discriminative models in a variety of ways [1], [3], [6], [8], [9]. For example, in [3] Raina et. al. present an algorithm for text classification which learns naive Bayes models for sub-sections of the corpus and combines those predictions using weights learned discriminatively.

The work which is perhaps most closely related to ours is [1]. They propose a model called partitioned logistic regression (PLR) which combines the predictions of multiple logistic regression classifiers, each trained over an independent subset of the features, using principles of naive Bayes. Their method achieves greater accuracy than either logistic regression or naive Bayes across a varying number of training examples, and continues to perform surprisingly well even when their total independence assumption is weakened. We take that result and expand the model to explicitly account for some dependence between partitions, which we expect will allow for even better performance. Further, our model can be applied to a much wider class of problems as we are not limited to the case where features can be neatly partitioned into unrelated sets.

Our approach also has strong connections to dropout [11]. Dropout was presented by Hinton et al. (2014) as a method of increasing the generalizability of deep neural networks, wherein nodes within the neural network are randomly omitted during training [11]. Many papers have gone on to use and explore variations of dropout, including by characterizing dropout as a form of regularization and expanding the method to apply to other models such as logistic regression and support vector machines [12], [13], [14], [15], [16]. However, little work has been done exploring the space of dropout where nodes are omitted non-randomly or the connections between dropout and ensemble methods. We argue that the GEC method may be thought of as a Bayesian approach to dropout, aimed at preserving sets of features based on their informativeness.

III. Ensemble Classifiers

We begin by briefly reviewing clique trees in the context of graphical models and go on to present our *graphical ensemble classifier* (GEC) framework which uses clique tree inference over subsets of features to create a general and statistically efficient approach to data classification.

As a tool for modeling the relationship between features, we look to Markov networks. Let X be a set of continuous or binary random variables, $X_1, X_2, \ldots, X_n$, with categorical class labels Y. A Markov network (MN), or Markov random field (MRF), is an undirected graph $G = (V, E)$ where each node V represents a feature [5]. Pairs of features (v_i, v_j) in a MN are dependent if there exists an edge e_{ij} between them, and are conditionally independent given a path of edges between them. MNs are thus a useful framework for describing arbitrary dependence between features.

For the remainder of this paper, we limit our feature dependence structure to be over MRFs whose factor graphs have tree structure, clique trees. A clique tree H over the feature space of $X \in \mathbb{R}^n$ is an undirected, singly-connected graph satisfying the following properties:

1) each node i in H is labeled with a clique of variables, $C_i \subset X$,
2) each variable $x_i \in X$ appears in at least one clique, and
3) if x_i appears in two cliques, C_i and C_j in the tree, it must also appear on all nodes in between them

[5]. Conveniently, clique trees have the property that they can be factorized to define the probability distribution:

$$P(X) = \frac{\prod_{c \in C} P(X_c)}{\prod_{s \in S} P(X_s)}$$

for a tree with cliques C and separator sets S [5]. Limiting our dependence graph to clique trees in this way ensures that we can do exact inference efficiently, but this work can be expanded to include arbitrary factor graphs with some approximations introduced.

A. *Graphical Ensemble Classifier (GEC)*

Using clique trees as our underlying graphical model for describing feature independence structure allows us to handle overlapping groups of variables, for we can then factorize the learning problem according to the factorization of the given clique tree. In particular, let $X \in \mathbb{R}^{n \times D}$ be a dataset with binary class labels $Y \in [0,1]^n$. Let the feature space of X satisfy a clique tree independence structure with cliques C and overlapping (separator) sets S. If X_c represents the data limited to features in clique $c \in C$ (or X_s for separator $s \in S$), then we have that:

$$P(X) = \frac{\prod_{c \in C} P(X_c)}{\prod_{s \in S} P(X_s)}.$$

For classification problems, we are interested in the discriminative task of predicting $P(Y|X)$. Conditioning the above equation on the class label Y and applying Bayes' rule we obtain that:

$$\begin{aligned} P(Y|X) &\propto P(X|Y)P(Y) \\ &= \frac{\prod_{c \in C} P(X_c|Y)}{\prod_{s \in S} P(X_s|Y)} P(Y) \\ &\propto \frac{\prod_{c \in C} P(Y|X_c)/P(Y)}{\prod_{s \in S} P(Y|X_s)/P(Y)} P(Y) \\ &= \frac{\prod_{c \in C} P(Y|X_c)}{\prod_{s \in S} P(Y|X_s)} P(Y)^{1+|S|-|C|} \end{aligned} \tag{1}$$

Using eq. (1), we fit linear sub-models $P(Y|X_c)$ and $P(Y|X_s)$ for each partition X_c and each overlapping set X_s, respectively.

The result is the GEC model. Given model weights W_c and W_s trained over each clique c and separator set s, respectively, we have that the log odds are:

$$\begin{aligned} \hat{lo}(X) &= \sum_{c \in C} W_c \cdot X_c - \sum_{s \in S} W_s \cdot X_s \\ &\quad + (1 - |C| + |S|) \log \hat{o} \end{aligned} \tag{2}$$

where $\hat{o} = \hat{P}(Y = 1)/\hat{P}(Y = 0)$ is the prior odds.

Relation to Existing Models: To put this into context, we note three interesting cases of this model:

1) If there is only one partition so that $|C| = 1$ and $X_1 = X$, then $S = \emptyset$ and eq. (1) trivially reduces to classic logistic regression.
2) If each partition X_i contains exactly one variable, then again $S = \emptyset$, $|C| = n$ and $X_i = x_i$ for all $0 < i \leq n$. Hence, eq. (1) reduces to:

$$\begin{aligned} P(Y|X) &\propto P(Y) \prod_{x_i \in X} \frac{P(Y|x_i)}{P(Y)} \\ &= P(Y) \prod_{x_i \in X} P(x_i|Y), \end{aligned}$$

which is simply naive Bayes.
3) If the X_C sets consist of a partition into k non-overlapping sets, then the equation reduces to:

$$\begin{aligned} P(Y|X) &\propto P(Y) \prod_{i=1}^{k} \frac{P(Y|X_i)}{P(Y)} \\ &= P(Y)^{1-k} \prod_{i=1}^{k} P(Y|X_i), \end{aligned}$$

which is the case of this problem handled by PLR [1].

Thus, depending on the nature of the underlying feature dependencies, the GEC model spans the space between being a generative and being a discriminative model, along with successfully encompassing a broad class of models.

B. *Randomized GEC*

While the above GEC framework is especially useful in a setting wherein the feature relationships are known, such information is often not known in practice. Inspired by the success of ensemble methods, we show that even when little to no structure is known, it is possible to leverage ensembles of randomly partitioned features and obtain accurate results without overfitting.

To accomplish this, we propose randomized GEC (Rand-GEC). Since no groupings are known or present, we instead randomly group features into k non-overlapping sets of equal size and run GEC. We then average over s such random groups to reduce bias. The procedure is described in Alg. 1.

Algorithm 1 Randomized GEC

procedure RANDGEC(N, d, k, s)
  Given data $X \in [0,1]^{(N,d)}$.
  for trial $t \in \{1, \ldots, s\}$ **do**
    Partition d features into k random groups, $f_1, \ldots, f_k$.
    Train LR models $M_{f_1}, \ldots, M_{f_k}$.
    Sum weights of sub-models using eq. (2) to make predictions on test set.
    Calculate test accuracy.
  end for
  Average accuracy of s trials.
end procedure

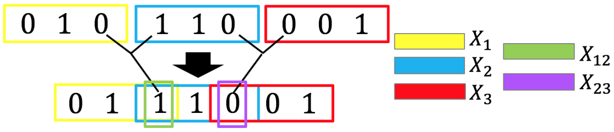


Fig. 1: A visual representation of the process of creating the overlapping sets in the artificial data generation process, where the size of overlapping sets, o, is 1. In this case, the model has $k = 3$ partitions (X_1, X_2, X_3) over 3 features each and ends up with two separator set models (X_{12}, X_{13}) over $o = 1$ feature each.

IV. Results

To display the capabilities of the graphical ensemble classifier, we present results on artificial data, 20-newsgroups, and MediFor. In each case we compare the accuracy of Bernoulli naive Bayes (NB), logistic regression (LR), partitioned logistic regression (PLR), graphical ensemble classifier (GEC), and randomized-GEC as a function of the number of training examples. Since GEC is designed specifically to break the independence assumption of PLR, we additionally consider a model (PLR-split) which takes the set of dependent features and divides them randomly between partitions so as not to count them twice.

A. Artificial data

We begin by creating an artificial data set to test the GEC model under a controlled setting. We expect to see GEC perform better relative to baseline models as the amount of overlap between partitions increases, as it is the only model which explicitly uses that structure. Additionally, we display the benefits of random partitions when there is no structure and few training examples.

Since our model assumes known clique tree structure over features, we design a structure where we can purposefully vary the feature independence tree and amount of dependence between partitions. We base our generation process on the methodology presented in [1].

Let $Y \in \{0, 1\}$ be the class label of a random example $X \in \{0, 1\}^d$. For a given number of partitions k we generate random examples X using:

$$Y \sim \text{Bernoulli}(0.5)$$
$$\hat{X} = (\hat{X}_1, \ldots, \hat{X}_k) \sim N(\vec{\mu}_y, \Sigma_y)$$
$$X = (X_1, \ldots, X_k) = \left(f(\hat{X}_1), \ldots, f(\hat{X}_k)\right),$$

such that $N(\vec{\mu}_y, \Sigma_y)$ is a multivariate normal with parameters based on the class label $y \in Y$. In particular, $\vec{\mu}_0 = \{-\sqrt{d}\}^d$ and $\vec{\mu}_1 = \{\sqrt{d}\}^d$. To simulate k independent partitions, we generate each Σ_y by first creating a Gram matrix G_y and then zeroing out the covariance terms between the classes, as described below. For example, when $k = 2$:

$$G_y = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \text{ becomes } \Sigma_y = \begin{bmatrix} A & 0 \\ 0 & C \end{bmatrix}$$

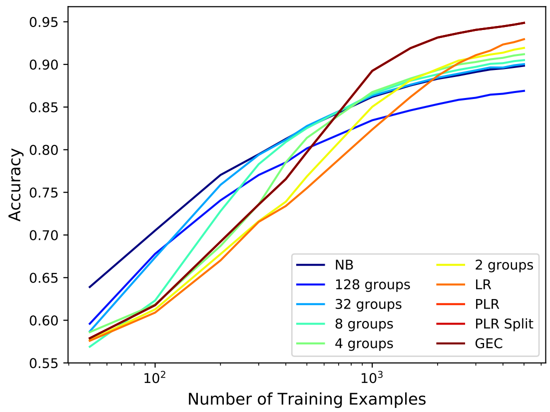
This process ensures that variables from each partition X_i are independent from variables in different partitions X_j for $i \neq j$.

Each G_y is formed from $k * d$ vectors of size 10 with entries drawn uniformly at random between -1 and 1. After zeroing out the covariance terms, this creates a unique, positive semi-definite $d \times d$ covariance matrix for each class label, which we use in the generation of $\hat{X} \sim N(\vec{\mu}_y, \Sigma_y)$.

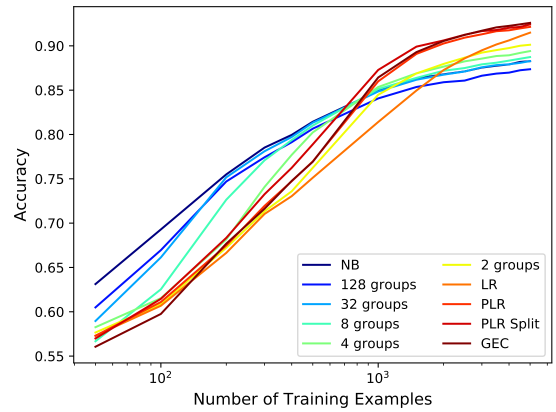
After generating $\hat{X}$ using the process described above, we expand the real valued samples from $\hat{X}$ into a binary representation. Using a sign bit and the bits corresponding to 2^2, 2^1, 2^0, 2^{-1}, and 2^{-2} we obtain the expanded samples $X = (X_1, \ldots, X_k)$ of size $6n$. This expansion process makes some features more informative than others. We use $f(\hat{X}_i)$ to denote this expanded set of features.

The above process results in a dataset consisting of k independent partitions, and mirrors the data generation process presented in [1]. We introduce a final step in the data generation to create a chain structure of dependence between the partitions. For each original pair of partitions (X_i, X_{i+1}) we pair o features from each and create dependence between them. Concretely, let $x_j^{(i)}$ denote the jth element in partition i. Then, we pair sets $O_i = \{x_{6n-o}^i, \ldots, x_{6n}^i\} \subset X_i$ and $O_{i+1} = \{x_1^{(i+1)}, x_2^{(i+1)}, \ldots, x_o^{(i+1)}\} \subset X_{i+1}$, to create the overlapping partition $X_{i(i+1)}$ where each element $x_j^{i(i+1)}$ is the maximum of the jth elements of O_i and O_{i+1}. This process is visually depicted in Fig. 1 for clarity. This allows us to control the level of dependence between partitions, varying from complete independence ($o = 0$) to full overlap ($o = d$).

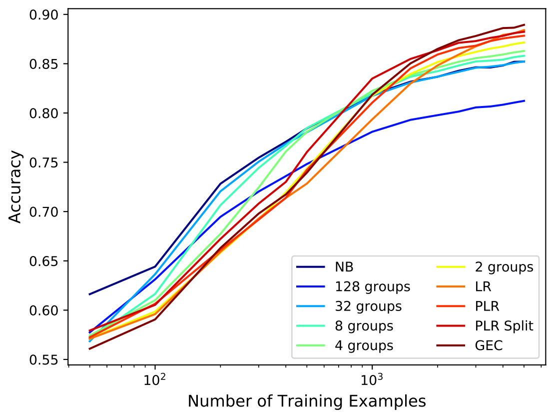
For each experiment we run 5-fold cross validation on the training data in order to choose the ℓ_2-regularization constant, C, for logistic regression. For simplicity, we choose $C \in [0.01, 0.1, 1, 10, 100]$ to be the same for each sub-model of a given model. (In preliminary experiments, we find that the results are not very sensitive to this tuning process.) For the randomized splits, we randomly divide features into k groups and average the results over s of these random splits to reduce bias. Results are averaged over 10 random datasets, with



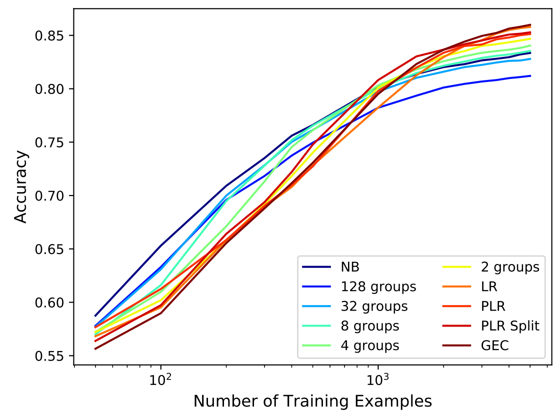
(a) 0 overlap.



(b) 60 overlap.



(c) 120 overlap.



(d) 180 overlap.

Fig. 2: Accuracy versus number of training examples on artificial data for varying dependence between groups on a semi-log scale. Each partition originally contains $d = 240$ features.

randomized groupings additionally averaged over $s = 3$ random splits per dataset.

Artificial Results: Our first experimental setting is designed to compare the effectiveness of GEC and PLR when there are known groups and some overlap (Fig. 2). In the case of no overlap (Fig. 2a), we observe as expected that PLR, PLR-Split, and GEC are all equivalent and significantly out-perform the other models in most cases. As the amount of dependence between the two groups grows, we see that PLR does not remain as competitive. Interestingly, PLR-split continues to succeed, particularly in the mid-range of number of training examples. However, in nearly all cases our GEC method obtains higher accuracy on held-out test data after seeing enough examples.

Next, we analyze the impact of the number of random partitions, k, on accuracy when an underlying group

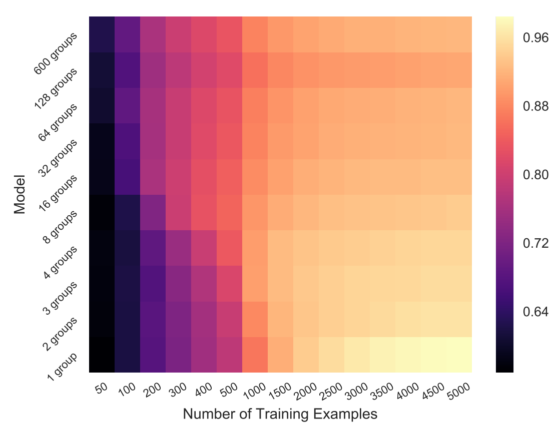
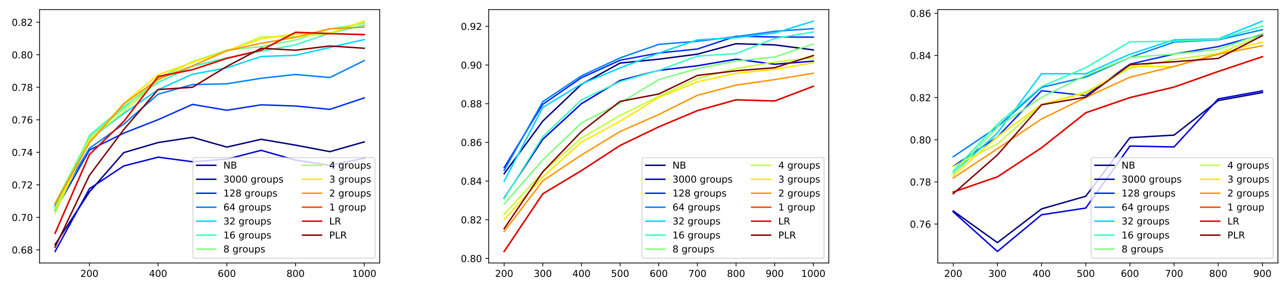


Fig. 3: Heatmap of accuracy of number of random splits versus number of training examples, with $d = 600$ features.



(a) alt.atheism vs soc.religion.christianity (b) rec.sport.hockey vs rec.sport.baseball (c) comp.windows.x vs comp.graphics

Fig. 4: Accuracy vs number of training examples on the 20-newsgroups dataset.

structure is not present. Fig. 3 shows a heatmap of accuracy for training set size versus number of partitions, k. Results are again averaged over 10 random datasets of dimension $d = 600$ with no forced independence structure (one group). Each model is averaged over 3 random groupings. It is clear from this figure that given enough training data, logistic regression (1 group) is the superior choice. When the number of training examples gets smaller ($\sim 100 - 1500$), it becomes progressively better to use larger numbers of partitions. This reflects the notion that logistic regression over n features needs $O(n)$ samples to converge to asymptotic accuracy [2]; splitting into more groups allows each partition to be over a smaller number of features, decreasing the effective sample complexity to $O(n/k)$ for each partition.

B. 20 Newsgroups

The 20-newsgroups text dataset, as used in [4], consists of thousands of text documents relating to 20 different topic groups. We consider the task of distinguishing the topic pairs *alt.atheism* vs. *religion.misc*, *rec.sport.hockey* vs. *rec.sport.baseball*, and *comp.windows.x* vs. *comp.graphics*. Since this text data has no particular natural grouping, we choose to group words by their parts of speech for PLR. We only consider randomized GEC for this set of experiments, and omit PLR-split since there is no overlap in part of speech tagging.

In Fig. 4 we select the top 3000 most common words in the given training set as features and average all results over 10 random train-test splits. An important takeaway from this set of figures is that the answer of which model is best is very dependent on the data. Looking at Fig. 4a, atheism versus Christianity, we observe that using smaller numbers of groups is preferable to using more groups, but in Fig. 4b, hockey versus baseball, we see the opposite trend. The reasoning behind this stark difference comes from the fact that the task of differentiating the topics atheism and Christianity is harder than of hockey and baseball, as seen in an about 10% lower accuracy for the former. Some of the most common words for atheism versus Christianity include "god" and "believe," which on their own do not give much information to distinguish between the categories. For hockey and baseball, however, standalone words such as "pitcher" for baseball or "goalie" for hockey can alone be strong evidence for classification. As a result, datasets which require modeling of more complex relationships between features will often see better results with smaller numbers of groups, so that those interactions are not lost. It is thus important to be aware of the expected properties of a dataset before choosing what number of random GEC groups to use.

C. MNIST

Next, we explore the classic hand-written digit recognition task, MNIST [18]. The MNIST dataset consists of 60000 training examples of black-and-white handwritten digits 1-9, sized to 28x28 pixels each. For our experiments, we consider the binary task of differentiating two given numbers. We choose the pairs 1&7 and 4&9, which are classically more difficult to differentiate than most other pairs due to their visual similarity. Our hypothesis for this dataset is that partitioning the image into smaller regions will allow our model to out-perform logistic regression. Since the numbers are centered in each image, we focus on the middle region as our overlapping set. We consider three different partitioning schemes: focal, diagonal, and 9-grid, as depicted in Fig. 5.

An interesting point in the results of Fig. 6 is that PLR does not get much higher accuracy than random splits

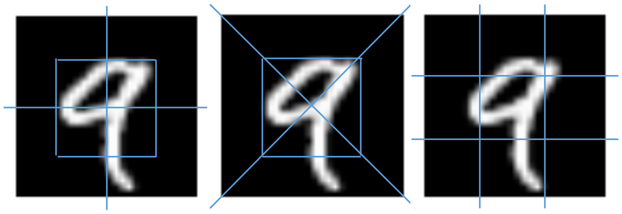


Fig. 5: Partitions used for PLR and GEC on MNIST dataset. From left to right: focal, diagonal, and 9-grid.

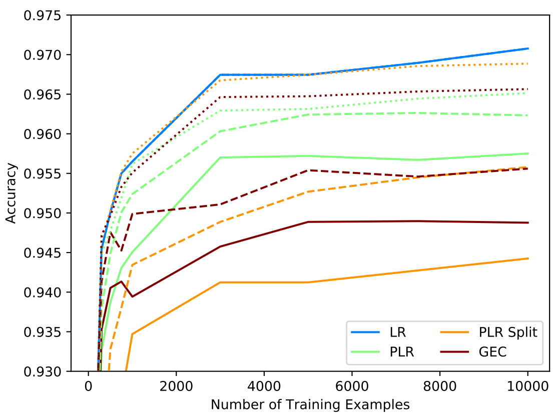


Fig. 6: Accuracy versus number of training examples on MNIST. Solid, dashed, and dotted lines represent focal, diagonal, and grid groupings, respectively.

into a few groups, and GEC does worse than either. This indicates that in many cases it is better to use a series or random groups than to attempt to use this method with a poorly designed group.

D. MediFor

The Media Forensics (MediFor) project is an ongoing effort into improving our capability of detecting manipulations to images. This issue has become of critical importance over the past few years as social media becomes increasingly prevalent and influential across the world, making it easier to spread false information and images. The MediFor project consists of a group of industry and university teams who have been independently developing methods for detecting certain image manipulations, such as crops, recaptures, blurs, and splices. Rather than develop another algorithm for directly detecting alterations, we are interested in the task of synthesizing the outputs of the existing algorithms, with the hope of getting better overall accuracy than any individual algorithm. We thus create a dataset whose features are the output confidence scores of each algorithm

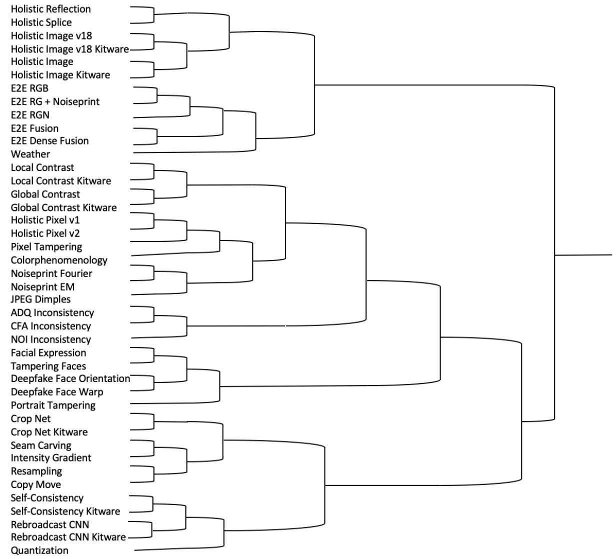


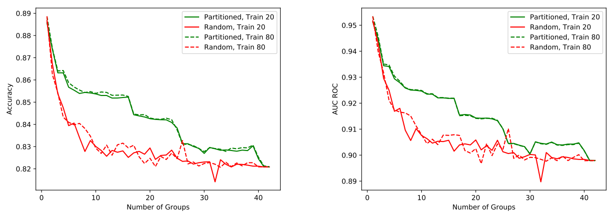
Fig. 7: Dendrogram describing groupings chosen for MediFor dataset.

for the given image. Since our inputs are the outputs of each algorithm, we then group features using domain knowledge of algorithm similarity.

Since current information about each algorithm is limited, we have manually chosen a sequence of groupings to represent a variety of possible relationships. We begin with each algorithm in it's own group (naive Bayes) and sequentially choose the two most similar groups to merge until all of the algorithms make up one group (logistic regression). The groupings we have chosen are displayed in the dendrogram in Fig. 7.

In Fig. 8 we present results of accuracy and area under the ROC curve (AUC ROC) as we increase number of partitions (following the groupings described in the dendrogram) when training on 20% and 80% of the dataset. As before, random groupings are averaged over 5 trials. All values represent an average over 5-fold cross validation.

In this case we find that logistic regression (one group) is the best option for this task, with accuracy dropping off as we increase the number of groups. We can also see that the partitions made manually with knowledge of algorithm similarity achieve higher accuracy and AUC than random partitions do. These two observations both indicate that using the relationships between the algorithms is a better approach than assuming that they are all independent.



(a) Accuracy (b) AUC ROC

Fig. 8: MediFor results displaying accuracy and AUC ROC versus number of groups when training on 20% and 80% of the dataset. Green lines represent manual partitions based on domain knowledge, and red lines represent random groupings.

V. Conclusion

In this paper we presented the graphical ensemble classifier as a hybrid between logistic regression and naive Bayes and showed that it can obtain higher accuracy than baseline methods when an overlapping set structure is present in the data. In addition to requiring less data to fit an accurate model, GEC is simple to implement, for it is a simple linear combination of traditional logistic regression models. This means that it can be applied to a wide range of datasets with low overhead. We believe that this method shows promise and may be of use for domains with structured domain information is known.

Unfortunately, as seen in the 20-newsgroup and MediFor results GEC is very sensitive to the chosen partitioning. If a given grouping fails to capture key relationships in the features, accuracy may drop off substantially. Averaging over randomized partitions is a simple and surprisingly effective method for counteracting that property. Nevertheless, our artificial data results indicate that using a proper grouping with GEC is more effective than random splits. In future work, we hope to develop a strategy for automatically finding possible partitions based on feature correlations or other similarities. In addition, we aim to extend our method to accept different, nonlinear underlying discriminative models, such as support vector machines or even feed-forward neural networks. The super-linear complexity of such models should mean that reducing the size of the feature space will allow for even greater accuracy gains than we see with logistic regression.

References

[1] M. Chang, W. Yih, and C. Meek (2008). Partitioned logistic regression for spam filtering. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*

[2] A. Ng and M. Jordan (2001). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Proceedings of Neural Information Processing Systems 14*.

[3] R. Raina, Y. Shen, A. Ng, and A. McCallum (2003). Classification with hybrid generative/discriminative models. *http://ww.cs.standord.edu/˜rajatr/nips03.ps* .

[4] S. Wang, and C. Manning (2012). Baselines and bigrams: simple, good sentiment and topic classification. In *Proceedings of the ACL 12*.

[5] D. Koller and N. Friedman (2009). *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press.

[6] G. I. Webb, J. R. Boughton, F. Zheng, K. M. Ting, and H. Salem (2010). Decreasingly naive bayes: Aggregating n-dependence estimators. *KNOWSYS Lab Technical Report, Monash University*.

[7] V. N. Vapnik (1998). *Statistical Learning Theory*. John Wiley & Sons.

[8] J. Kittler, M. Hatef, R. P. Duin, and J. Matas (1998). On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[9] G. E. Hinton (2002). Training products of experts by minimizing contrastive divergence. In *Neural Computation*.

[10] T. Dietterich (2000). Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*.

[11] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014) Dropout: A simple way to prevent neural networks from overfitting. In *Journal of Machine Learning Research, Vol. 15*

[12] S. Wager, S. Wang, and P. Liang (2013) Dropout training as adaptive regularization. In *Advances in Neural Information Processing*.

[13] N. Chen, J. Zhu, J. Chen, and B. Zhang (2014) Dropout training for support vector machines. In *Association for the Advancement of Artificial Intelligence*.

[14] L. Ba, and B. Frey (2013) Adaptive dropout for training deep neural networks. In *Advances in Neural Information Processing Systems*.

[15] S. Wager, W. Fithian, S. Wang, and P. Liang (2014) Altitude training: strong bounds for single-layer dropout. In *Advances in Neural Information Processing Systems*.

[16] L. Maaten, N. Chen, S. Tyree, and K. Weinberger (2013) Learning with marginalized corrupted features. In *Proceedings of the International Conference on Machine Learning*.

[17] M. Abramowitz, and I. Stegun (1965) Handbook of mathematical functions: with formulas, graphs, and mathematical tables. *Dover Books on Mathematics*.

[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner (1998) Gradient-based learning applied to document recognition. *Proceedings of the IEEE*

[19] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips (2015) Hidden technical debt in machine learning systems. *Conference on Neural Information Processing Systems*

[20] D. Sculley, M. E. Otey, et. al. (2011) Detecting adversarial advertisements in the wild. *Conference on Knowledge Discovery and Data Mining.*