Hierarchical Clustering for Semi-Supervised Ground Truth Generation

Medha Baidya MEDHAB1@VT.EDU

Samuel Maddock SAMM@VT.EDU

Sazzadur Rahaman SAZZAD14@VT.EDU

Jason Ziglar JPZ@VT.EDU

Abstract

Supervised learning tasks can require a large collection of labeled data for accurate pattern recognition. For recognition of handwritten characters, manually producing ground truths can be very tedious. In this paper, we propose a semi-supervised hierarchical clustering method to reduce the necessary amount of human effort required for labeling a dataset of handwritten characters. The experimental results demonstrate that the approach can improve labeling accuracy over baseline methods.

1. Introduction

Collecting and labeling datasets consumes substantial resources in the application of machine learning techniques, due to wide success of supervised learning approaches. These techniques are often attractive as they allow control over the resulting output independent of input data; images of handwritten characters can be fed to a machine learning algorithm to classify them based on the the labels of similar character images already seen. These labels may correspond to the identification of the alphabet or digit from the image, the identification of where they originated, or the quality of the handwriting samples, etc. Assigning labels of potential interest grows more than linear in the amount of collected data. This is compounded by the explosion of data available for labeling; the internet has provided exponentially growing sources of images, video, and text—all of which can provide useful information ripe for machine learning. Automating the process of labeling data reduces the ability to specify the desired output, but partial automation can minimize human effort while maintaining control.

In this paper, we propose a semi-supervised learning technique for labeling large datasets of handwritten characters or digits. Our approach aims to discover clusters from one feature set where the labeling propagated to all members is in conflict with the labels proposed by other feature sets for the same data points, and splitting those clusters into sub-clusters to get more uniform clusters for voting. This approach leverages the human provided information to automatically identify inconsistencies between the feature sets, which ideally leads to a system which can start with low initial human involvement, and automatically request additional involvement as the data indicates confusion in labeling.

The overall proposed algorithm works as follows: for a set of training input examples, multiple feature representations of the inputs are extracted. These representations are clustered independently using k-means to produce a classification of inputs which appear similar. The centroids of each cluster are presented to a human expert to provide ground truth labelings for each cluster. The label for each cluster centroid is considered as a vote for the true label for all members of that cluster, resulting in every data point having one vote for the assigned label for each feature representation used, which can be used in a voting scheme to assign labels to every example. The primary novel contribution of this paper is to introduce an iterative hierarchical re-clustering step, in which every cluster is measured for how well the centroid label matches the votes for member points from other feature representations, with clusters with high conflict (e.g. votes from other representations do not agree with the label of the cluster under consideration) are subdivided into sub-clusters, with the new clusters annotated by the human expert. The technical details of the steps described are presented in the following sections of the paper, with experiments demonstrating the overall per-

Virginia Tech CSx824/ECEx424 final project report.

formance of the implemented approach.

Experiments included in this paper are based on results using the MNIST image database of handwritten characters from multiple authors (LeCun et al., 1998). This dataset contains a collection of 60,000 training images and 10,000 test images. Images are anti-aliased grayscale and are centered within a 28x28 pixel square.

2. Related Work

Several approaches to a semi-supervised method of automating data labeling exists in the literature.

(Vajda et al., 2011) presents an approach using an unsupervised clustering algorithm to group similar subsets of data. Using raw pixels, Principle Component Analysis (PCA), and auto-encoder as features, the unsupervised clustering provides cluster centroids for a human expert to label. Following this step, a voting scheme is used to decide the final label for samples with unanimous agreement.

(Vajda et al., 2015) uses a similar approach while adding additional image features and clustering methods. Local Binary Patterns (LBP) and Radon transform are among the features added to this approach. Additional clustering techniques are used such as Self-Organizing Map (SOM) and Growing Neural Gas (GNG) to compare performance in labeling and cluster compactness accuracies.

(Li et al., 2012) introduces an agglomerative hierarchical clustering approach for recognition of online handwritten digits. One or more strokes per symbol are used to generate a codebook mapping which is later used for labeling raw handwritten digits.

Experiments indicate that parallel clusterings improve label accuracy when the cluster labels are unanimous, but performance suffers even under majority voting (Vajda et al., 2015). Increasing the number of clusters improves performance by generating more compact clusters, thus reducing the number of points for which conflicts occur—at the cost of requiring additional human labeling. Our approach attempts to minimize conflicted cluster labeling by introducing a hierarchical clustering method.

3. Image Features

Voting requires multiple parallel representations of the input image, so the core label propagation technique can be repeated in parallel on differing approaches. Features must vary in representing the original data in order to provide variety in votes, implying a need for complementarity between features. Multiple image features are implemented to attempt to provide variety - several of these are used based on previous experiments (Vajda et al., 2015). This experiment used five feature representations: raw data, an auto-encoder, Local Binary Profiles, the Radon Transform, and Profiles. The last proposed feature, Profiles, performed very poorly in previous experiments ($\sim$76% accuracy vs. $\sim$96% accuracy for other features), and did not provide much information on implementation details, which led to dropping this feature from future consideration. In order to increase the number of feature sets for voting, two additional representations known to perform well in many computer vision tasks were implemented - the Histogram of Oriented Gradients (HOG), and the GIST descriptor. This results in six features being implemented and used for clustering: Raw Data, Auto-Encoder, Local Binary Patterns, Radon Transform, HOG, and GIST.

3.1. Raw Data

The simplest representation of data to consider is the original data itself. Raw data has been used successfully in previous work, and is included here as well. The only transformations performed on the input data is to reshape the 2D image into a 1D feature vector (producing a 784 dimensional vector), and normalizing the raw pixel values to the $[0.0, 1.0]$ range.

3.2. Auto-Encoder

An auto-encoder (AE) is an early proposed deep learning method for performing dimensionality reduction, used as a data-driven approach to learning features (Hinton & Salakhutdinov, 2006). Creating an auto-encoder starts with training a neural network with the input layer and output layer have an equal number of nodes, and the hidden layer having fewer nodes. The training dataset is constructed with the inputs and labels both set to the raw data described previously, and with trainining performed with any neural network techniques desired. This results in a neural network learning some representation to reconstruct the original dataset. The hidden layer, having fewer nodes than the input, encodes the data in some lower dimensional feature space (e.g. encoding.) Dimensionality reduction comes from recording the hidden layer activations, as opposed to the final results of the output layer, which acts as the decoding step.

For these experiments, the auto-encoder uses 200 nodes in the hidden layer for dimensionality reduction, to match the reported configuration in (Vajda et al., 2015). The original experiments do not report the implementation details for the auto-encoder trained, limiting the ability to perfectly implement the original approach. The auto-encoder implemented here uses the tanh activation function for the hidden layer, and is trained over the entire training dataset using iRPROP algorithm (Igel & Hüsken, 2000) for updating weights during back-propagation.

3.3. Local Binary Patterns

Local Binary Patterns (LBP) encode local texture through binary comparisons of pixels and their immediate neighbors, and is part of the original experiments useful features (Ojala et al., 1994). This feature provides a rotation invariant representation of each image, and performed well in previous experiments performing the same task.

3.4. Radon Transform

The Radon transform simulates performing tomography over a 2D image (Radon, 1986). This means rotating the image through some number of orientations, computing line integrals through parallel lines in each configuration. These integrals represent the density of the image along different paths through the image. The radon transform is computed by rotation the image in 1° increments over $[0^\circ, 180^\circ]$.

3.5. Histogram of Oriented Gradients

Histogram of Oriented Gradients compute gradients in regions of the image, building histograms of how often gradients are oriented along certain directions in the image. Given that HOG measures the distribution of gradients in regions of an image, this can represent general shape in a compact and intuitive fashion, which has been shown to perform well in many classification tasks (Dalal & Triggs, 2005). For this experiment, HOG is configured to subdivide the image into four quadrants for computing gradients, with gradients divided into eight directions. These values were empirically determined to produce good clustering performance.

3.6. GIST Descriptor

The GIST descriptor computes a single feature to describe an entire image based on spectral frequencies transforms. This feature has been shown to perform very well in discriminating between images with different global properties, such as discriminating between natural and man-made scenes (Oliva & Torralba, 2001).

4. Clustering

To minimize the amount of human intervention, the data must be organized in some unsupervised manner in a fashion to allow any human annotation to apply to as many points as possible. In the limit, this intuition leads to the observation that the ideal approach would automatically group the dataset by labels, asking the human operator only to provide the desired token to represent each group. The more achievable solution for this approach is to use unsupervised clustering techniques to develop some segmentation of the data in which clusters are pure (e.g. all members of any individual cluster all belong to the same ground truth label.) To account for intra-class variation and improve the chances of clusters remaining pure, the initial clustering should over-segment the data, allowing multiple clusters to map to the same ground truth label.

Previous works compared several different clustering algorithms applied to the same problem and dataset. Throughout the comparisons performed, k-means consistently outperformed the other techniques presented (namely Self Organizing Maps and Growing Neural Gas.) In this work, k-means was used to perform clustering on each feature set computed for the data.

4.1. k-Means Clustering

k-means approach places k centroids over data forming k clusters of data points. In our project we applied k-means clustering on each feature set separately. These experiments use the Euclidean distance metric for all clustering operations.

5. Hierarchical Clustering and Voting

To address the impact of conflicting clusters on accuracy, we propose a hierarchical clustering approach: to selectively introduce additional clusters only in subsets of the data in which conflicts from coarser labels appear.

After clustering data points based on features, a label is assigned to the cluster centroids by a human expert. The label of the centroid is circulated to each of the data points in the corresponding cluster. Thus, each datapoint will inherit labels from each feature.

Following this step, a final label is assigned to a data point by calculating votes based on the labels inherited from individual features. If there exists k features and m labels, then a data point will inherit total k number of labels from k features. If a data point inherits jth label (where $j \in [1, m]$) from $\lfloor \frac{m}{2} \rfloor + 1$ number of features, the jth label wins *majority votes* for the data point from k set of features.

Naturally, not every data point considered will have labels with majority votes. The fitness of clusters is calculated based on a metric to determine how many data points from a cluster are not assigned labels with majority votes. Using the metric, we can decide which clusters are needed to be further subdivided into additional clusters and perform reclustering for a specific feature. This reclustering technique will replace a cluster centroid with additional cluster centroids.

Similar to the original clustering, the new cluster centroids are labeled by a human expert and continue the process of circulating centroid labels to the data points of the corresponding cluster in order to select a new set of clusters for subdivision. After running the whole process for several iterations, we learn a set of clusters with labeled centroids.

5.1. Metrics For Calculating Cluster Fitness

In order to decide which clusters to consider for reclustering, three metrics were devised to measure the fitness of clusters.

5.1.1. Metric 1

The reclustering aimed to improve the purity of initial clusters formed. Thus we measured cluster entropy as the metric of cluster quality. A pure cluster would have all data points with single label and the impure will have a random mix. In my metric to choose the victim cluster to disintegrate, I look for cluster which is confused bewteen smaller set of labels. Rather tha having an equal mix of all possible labeled data, it has near equal mix of smaller set of lables.

The algorithm iterates through each cluster per feature set. Each data points within the cluster are looked up into all the remaining feature sets to find their unique labels sets. We measure the probability of each such unique label within the feature set which measure the confidence of this feature about the data points being that labels.

Individual feature set label predictions may have some common intersection. Meaning there might be cases where say feature set gist and feature set hog predict a common label say 3 along with other labels. If this is different from the evaluating cluster predicted label then we think that the cluster is contradicted by other feature set for label prediction.
The metric calculation algorithm is described below:

```
**for** $f = 1$ to F **do**
    **for** $i = 1$ to K **do**
        Find unique labels of data points in cluster[i] in Feature Set $\neq$ f
        Find probability of individual Labels within the Feature Set $\neq$ f
        Find the number of times same labels predicted by Feature Set $\neq$ f. Call it freq
        cluster[i].entropy = freq * [sum of probablity for each predicted Labels different from cluster prediction]
    **end for**
**end for**
return cluster.entropy
```

Where F are total number of features and K are total number of clusters.

The victim cluster is chosen the one with highest entropy value. k-means clustering is applied on this to form two cluster out of it. The whole process is reapeated for fixed iteration, we chose it to be 10.

5.1.2. Metric 2

If the number of data points with conflicting labels (not having any candidate label with majority votes) from cluster C_i is n_i, the total number of data points in cluster C_i is N_i, value of metric 1 for the cluster C_i is M_i and total number of clusters for candidate feature is j, where $i \in \{1, 2, \cdots j\}$. M_i is calculated as followed:

$$M_i = \frac{n_i}{N_i} \tag{1}$$

This metric for calculating cluster fitness exposes clusters having higher density of conflicting data points. But this metric fails to penalize the clusters with lower density of conflicting data points, but still the number of total conflicting data points remains high.

5.1.3. Metric 3

If the number of data points with conflicting labels (not having any candidate label with majority votes) from cluster C_i is n_i, the average number of data points in a cluster is N_{avg}, value of metric 2 for the cluster C_i is M_i and total number of clusters for the candidate feature is j, where $i \in \{1, 2, \cdots j\}$. M_i is calculated as followed:

$$M_i = \frac{n_i}{N_{avg}} \tag{2}$$

This metric always penalizes the clusters with relatively higher number of conflicting data points.

6. Results

6.1. Clustering Performance

Clustering forms the basis of these approaches, and using the clustering of a single feature set can be considered the baseline for comparison. The first metric to consider is how compactly a clustering represents the original data, measuring how much variance exists within clusters and between clusters, as presented in (He et al., 2004). Variance of some set of points X is mesaured using standard deviation:

$$\sigma(X) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{3}$$

Feature	# Dimensions	k=50	k=150
Raw	784	0.7816	0.7286
AE	200	0.7315	0.6718
LBP	784	0.8018	0.7616
Radon	7200	0.6568	0.5969
HOG	32	0.6090	0.5471
GIST	960	0.6510	0.5920

Table 1. Compactness metric for training data of each feature set with k-means clustering.

In order to normalize this against the variation inherent in the dataset, the standard deviation for an individual cluster should be normalized by the standard deviation of the entire dataset. In addition, the compactness of a clustering needs to be considered over every cluster in a clustering. Thus, for K clusters dividing a dataset X into $X = C_1 \cup C_2 \cup \dots C_K$, compactness can be computed by:

$$\text{Compactness} = \frac{1}{K} \sum_{i=1}^{K} \frac{\sigma C_i}{\sigma X} \tag{4}$$

The compactness of each feature set is reported in Table 1 for k-means clustering over the training dataset, providing a measure of how well the clusters are capturing the variance of the original dataset.

The second metric computed for individual clusterings measures the accuracy of labeling the dataset using label propagation. The entire dataset is labeled by having ground truth labels provided for the centroid of each cluster, and assigning the centroid label to every datapoint contained within that cluster. Accuracy is computed as a percentage of correct classifications over the entire dataset, and are computed over both the training and test datasets for comparison.

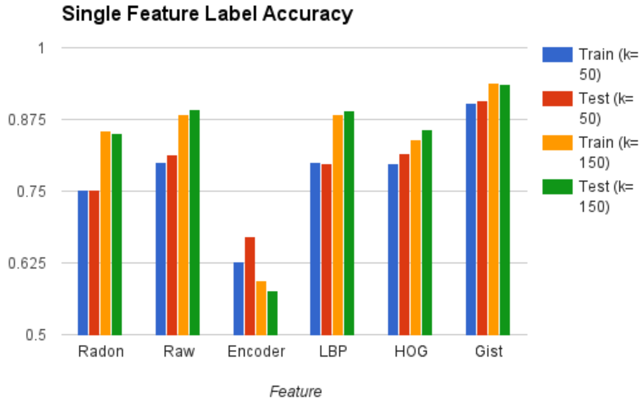

Figure 1. Accuracy of labels from considering only single feature sets.

Experiment	k	# Annotations	Acc_{train}	Acc_{test}
Voting	150	900	93.30	93.85
Metric 2	150	2896	94.37	94.60
Metric 3	150	1802	94.74	95.47
Voting	50	300	88.14	88.11
Metric 2	50	3542	91.30	93.82
Metric 3	50	1726	92.72	93.94

Table 2. Accuracy of label assignments in various voting/reclustering configurations.

6.2. Reclustering Results

In order to test the improvement in labeling performance, the accuracy of labeling using voting between multiple features, as well as voting after reclustering with alternate metrics were computed. For each trial, the number of human annotations is also reported in order to provide insight into the relative return on investment for different reclustering algorithms.

7. Conclusions

The new features introduced compare favorably to previous implemented features, both in clustering compactness and raw label propagation. These features provide information which provides strong discriminative information under unsupervised clustering, crucial for reducing the amount of labeling required. The experiments demonstrate that several different metrics for build the hierarchical clustering, which improve performance results of label propagation. Voting alone shows no significant difference compared to the best single clustering approach in either the train or test dataset. With reclustering, accuracy of labels improves over single clustering or voting baselines in all datasets, with the third metric producing the best results overall. The improvements are modest, given the increase in the number of human annotations required; the best single clustering requires 150 annotations for 93.64% accuracy on the test dataset, while the best reclustering approach requires 1802 annotations to increase that accuracy to 95.47%.

When considering the number of clusters to request human annotation, fewer clusters result in a more automated system. However, fewer clusters result in less accurate labelings across the dataset, resulting in a tradeoff between human effort and accuracy. Experiments tested re-clustering with two different initial clustering values, to represent lower and higher amounts of initial human involvement in the labeling. For the highest performing metric, the number of human annotations ends up at roughly the same levels (1726 annotations when $k = 50$, 1802 annotations when $k = 150$) while better performance with more initial clusters. The current re-clustering approach subdivides all clusters where the metric exceeds a fixed threshold, which leads to similar numbers of subdivisions. In this situation, a higher initial number of clusters allows clusters to start with fewer points per cluster on average, resulting in fewer clusters that are inconsistent.

Future work for this approach would focus on further refining the reclustering metrics to improve accuracy while minimizing the number of additional annotations. One particular approach which had been considered late in the project involved developing a metric based on the conditional entropy of labels for examples in a single cluster given the label distributions from other feature sets. This would provide a metric more rigorously founded, and can exploit the information present in human annotations (where the ground truth label is known with 100% certainty) as opposed to label propagations. This approach requires reworking much of the clustering approach to provide probability distributions for labels given a single feature set, which ran into time constraints for the project.

References

Dalal, Navneet and Triggs, Bill. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pp. 886–893. IEEE, 2005.

He, Ji, Tan, Ah-Hwee, Tan, Chew-Lim, and Sung, Sam-Yuan. On quantitative evaluation of clustering systems. In *Clustering and information retrieval*, pp. 105–133. Springer, 2004.

Hinton, Geoffrey E and Salakhutdinov, Ruslan R. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

Igel, Christian and Hüsken, Michael. Improving the rprop learning algorithm. In *Proceedings of the second international ICSC symposium on neural computation (NC 2000)*, volume 2000, pp. 115–121. Citeseer, 2000.

LeCun, Yann, Cortes, Corinna, and Burges, Christopher JC. The mnist database of handwritten digits, 1998.

Li, Jinpeng, Mouchere, Harold, and Viard-Gaudin, Christian. Reducing annotation workload using a codebook mapping and its evaluation in on-line handwriting. In *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*, pp. 752–757. IEEE, 2012.

Ojala, Timo, Pietikainen, Matti, and Harwood, David. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, number 1, pp. 582–585, 1994.

Oliva, Aude and Torralba, Antonio. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3): 145–175, 2001.

Radon, Johann. On the determination of functions from their integral values along certain manifolds. *Medical Imaging, IEEE Transactions on*, 5(4):170–176, 1986.

Vajda, Szilárd, Junaidi, Akmal, Fink, Gernot, et al. A semi-supervised ensemble learning approach for character labeling with minimal human effort. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pp. 259–263. IEEE, 2011.

Vajda, Szilárd, Rangoni, Yves, and Cecotti, Hubert. Semi-automatic ground truth generation using unsupervised clustering and limited manual labeling: Application to handwritten character recognition. *Pattern recognition letters*, 58:23–28, 2015.