Feature Selection for Image Data via Learning

Patricia G. Foschi[1], Deepak Kolippakkkam[2], Huan Liu[2]
[1]Romberg Tiburon Center for Environmental Studies
San Francisco State University, tfoschi@sfsu.edu
[2]Department of Computer Science and Engineering,
Arizona State University, {n.kolippakkam,hliu}@asu.edu

Abstract

Image mining requires the extraction of features from image data. Hundreds of features can be extracted, but only a subset of features is useful. Domain experts are usually required to determine relevant features in an ad hoc manner. We introduce a learning approach to feature selection for image mining. This approach allows domain experts to focus on tasks they do well: providing a labeled training image data set, instead of selecting relevant features ad hoc. Extensive experiments are conducted to verify the effectiveness of this learning approach.

1 Introduction

With the advent of the Internet and rapid advances made in storage devices, non-standard data such as images and videos have grown significantly [7]. Rich information is hidden in these data collections that is potentially useful in a wide range of applications and research. The conventional data mining process involves deriving novel and useful information from data [5]. The process of discovering valuable information from image data is called image mining, which finds its sources in data mining, content-based image retrieval (CBIR), image understanding and computer vision [8].

The tasks of image mining are mostly concerned with *classification* problems such as "labeling" regions of an image based on presence or absence of some characteristic patterns, and with *image retrieval* like determining "similar" images based on some criteria. In image mining, "training" images are used for learning purposes, and the results (knowledge obtained from training) are applied to a large number of new images to fulfill the required tasks.

In order to efficiently mine from images, we need to convert the image data into data that can be processed by image mining systems [10]. The conventional data format used is the *attribute-value* format (i.e., tabular format). The attributes (columns) are some characteristics representing an image object (instance) and rows form instances that have values corresponding to those attributes.

Attributes are called *features* in image processing. They are designed to describe the image pixel data. There could be hundreds of different features for an image. These features may include: *color* (in various channels), *texture* (directionality, linelikeness, contrast, roughness, coarseness), *edge, shape, spatial relations, temporal information, statistical measures* (moments - mean, variance, standard deviation, skewness, kurtosis). Domain experts can usually manually identify a subset of relevant features in an ad hoc manner. In general, no single feature can describe an image in its entirety [8].

2 Feature Selection via Learning

With many possible features for an image, often only a small number of features are useful in an application. Using all available features may negatively affect a mining algorithm's performance (e.g., time, accuracy). This is because many of the features may be redundant or irrelevant. With a limited number of instances but a large number of features, any mining algorithm will have difficulty learning properly - the so-called *curse of dimensionality*. Furthermore, with so many features, learning could become intolerably slow. Therefore, it is imperative to choose a set of relevant features. Choosing relevant features from a possible set is not a trivial problem [2, 3, 6]. In image mining, domain experts often play an important role in deciding relevant features. This work aims to find relevant feature-values by learning from training images. Therefore, domain experts are not required to provide relevant features, but only to identify or label instances of interest in an image. The latter is a much easier task than the former for experts, and its results are also more reliable because it directly deals with image classification.

However, for k features with binary values, the number of possible feature subsets for a given class is 2^k. With k being as large as hundreds, enumeration of all subsets is clearly an insurmountable task. Hence, we need to find an alternative approach. In the following subsections, we define the concepts of good feature subsets, and how a good subset can be obtained via learning.

In search of a good set of features, we need to find a suitable performance measure that can differentiate good features from the rest. The *goal* of this work is to find a *sufficient and necessary* subset of features. The necessity requirement indicates that the subset of features should be as small as possible; and the sufficiency requirement suggests that a good subset can guarantee the attainment of specified accuracy in image classification/detection.

In image mining, positive means the presence of an object of interest, negative otherwise. If an object of interest is to be detected in an image, one of the four possibilities can arise: we may *correctly* classify/detect the object's presence (True Positive - TP); we may *falsely* detect the object's presence (False Positive - FP); we may *miss* the detection of the object which should be *present* (False Negative - FN); we may *correctly* detect that the object is absent (True Negative - TN).

In image classification, we are concerned about the presence or absence of a particular object, a *two-class* classification problem. In building a good classification system, we wish to maximize the main diagonal elements, TP and TN, and minimize the off-diagonal elements, FP and FN (errors). Using TP, TN, FP, and FN, we can define measures like Accuracy *A*, Precision *P*, and Recall *R*.

$$A = \frac{TP + TN}{TP + FP + TN + FN}$$

When we are mainly concerned with correct detection of objects of interest, a high value of accuracy may not truly reflect what we want. A high TN can boost the value, although TN is only an indicator that objects of non-interest are truly detected. In order to detect the presence of an object of interest, we want to focus more on TP. Therefore, we turn to another two relevant measures, Precision and Recall, as follows:

$$P = \frac{TP}{TP + FP},$$

$$R = \frac{TP}{TP + FN}.$$

With the performance measures defined, we can systematically generate subsets of features and estimate their P and R based on training data. With the same Precision and Recall, we would prefer a subset with the smallest number of features. Therefore, we can start evaluating subsets of 1 feature, increasing the number of features one at a time until all features are considered. However, exhaustive search of a good subset is impractical when the number of features is large (e.g., $k > 20$) as the search space grows exponentially.

There exist many learning algorithms such as clustering, classification, association rules, that can learn efficiently from data. We consider them as candidates for feature selection via learning. Since class labels are available, we can rule out clustering algorithms. Classification algorithms try to maximize predictive accuracy while association rules [1] aim to maximize Support and Confidence. As we argued above, in the context of image mining, accuracy is not an ideal measure; hence we examine the application of association rules. Association rules are mined according to two measures: *Support* which measures generality of a rule and *Confidence* which measures precision of the rule. For a given class C and feature-values $f'_1 = v_1, f'_2 = v_2, ..., f'_m = v_m$, where $m \in \{1, ..., k\}$ and $f'_1, ..., f'_m$ are a subset of k features; let fv'_j be $f'_j = v_j$. Support and Confidence are defined as P$(fv'_1, fv'_2, ..., fv'_m, C)$ and P$(C|fv'_1, fv'_2, ..., fv'_m)$, respectively. An association rule with high support and confidence is both general and precise. This is consistent with the performance measures of Recall and Precision. Thus, a good association rule in the pre-specified form of $fv'_1, fv'_2, ..., fv'_m \Rightarrow C$ can define a good subset.

3 Experimental Evaluation

In the empirical study, we investigate the following issues: (1) Can we successfully learn good subsets of features from a training image? How well do these learned subsets compare with the full set? (2) Can these learned good feature subsets from a training image be successfully applied to testing images?

We first describe the application domain used in the experiments, next explain the ways of identifying features and detecting objects of interest, then present results of learning and evaluation.

3.1 Application - detecting Egeria

In this application, color infrared (CIR) images are aerially photographed and digitized. The object of interest is a weed (Egeria) that grows in water bodies. A set of 21 images is available. Each of these images is 300×300 pixels in TIF format (RGB). The weed is present in various parts of the image: clusters, patches, or sometimes as a 'mesh'. The problem is to automatically identify these weeds. A set of *cover* images corresponding to the 21 images is also provided in which the weeds are manually detected by experts for evaluation purposes. One image is used for training to obtain *good subsets* via learning and the rest are used for testing in experiments.

3.2 Feature identification and extraction

Following the advice from domain experts, we extract three types of features: color, texture, and edge. Color features represent Egeria's CIR colors in images. When Egeria occurs at various locations (water, shore, etc.), it displays different textures due to the characteristics of these locations. The edge feature is good for describing the boundaries of Egeria.

The raw image data are in pixels with three channels (NIR, R, G). Since a texture feature is a neighborhood property [8], it is defined for blocks of pixels. We extract features by dividing the image into *blocks* of 8×8 pixels starting from the upper left corner. The blocks overlap each other by 50% in both horizontal and vertical directions. For each block, we extract the three types of features.

For color features, we use all three channels, and compare a block's *average* intensity to the weed's spectral values. For texture features, we define texture templates for cases such as weed alone, weed in water, weed near shores in the training image. These textures are chosen as per the domain expert's knowledge. *Discrete Cosine (DC) Transform* [4] is used for texture extraction and the DC values of the template are compared with those of the blocks. Edges are detected using *Sobel's Edge Detection Algorithm* from MatLab. Since this would detect *all* edges in the image, we combine it with color to define the boundaries of Egeria. Color and texture features are extracted in separate channels. We have *3* color features, *3* texture features with *3* color features for each texture (so a total of *9* texture features), and one edge feature. Hence, in total, there are *13* features.

The output of each feature extraction process is a *Block Matrix* of 73×73. Recall that a block is 8×8 pixels. A Block Matrix is a binary image where 1 means that the corresponding feature *is* present in that particular block and 0 denotes absence. Each block resulting from combining these block matrices is an instance in the training data with 13 features. There are 73×73 (5329) *instances* for an image. Each instance can have one of the class values: 0 or 1, which indicates absence or presence of the weed for that instance.

We are now ready to generate and validate a good subset from the attribute-value image data. In order to form a class-blanced training data set with/without Egeria, we select all the instances having class label 1, and an equal number of instances having class label 0. This ensures the prior probability of both class labels to be the same. For the training image, out of 5329 instances, 2272 instances are labeled 1, so we pick another 2272 instances randomly from those labeled 0 to form the training data.

3.3 Evaluation results of learning

Applying an association rule mining (ARM) algorithm [9] to the training data, we are able to find association rules that describe the relationships between feature-values with class label 1 (presence of Egeria). As we discussed in the previous section, the best subset is the rule with the highest Support and Confidence. The following is the *best subset* found by ARM:

color_1=1 & color_2=1 & color_3=1 &texture_2a=0 $\Rightarrow$ 1

This subset has only 4 features: 3 color features (all three channels) and the *first* channel of the *second* texture feature. This subset has a Support of 0.723, and a Confidence of 1. It has the highest Support among all subsets, indicating that three color features and one texture feature (absence of water body) are sufficient to detect the weed in the training data. This subset has effectively filtered the possible inclusion of water body in our coverage of the weed. Though 13 features are defined based on the domain, it suggests that these 4 features should be sufficient for detecting weeds.

We design experiments to verify if this subset indeed outperforms the full set in two parts: 1. for the training image; and 2. for the new images.

As mentioned earlier, each image has a manual cover image that indicates where Egeria blocks are. For the purpose of these experiments, this cover image is considered the perfect result achievable for Egeria detection, and its Precision and Recall are both 1.

The results for the training image are shown in Table 1 and in Figure 1 about the training image, cover image, full set, and best subset. We use PR gain as a single measure for evaluation. PR gain is defined as:

$$PRgain = \frac{P \times R - P_0 \times R_0}{P_0 \times R_0}.$$

The best subset has a PR gain of 30% over the full set for the training data.

Full Set		Best Subset		PR Gain
P	R	P	R	
0.7077	0.7935	0.7740	0.9488	0.3022

Table 1: Comparing results of Full Set vs. Best Subset

The best set is obtained from the training image. We now evaluate its generality for the 20 testing images. The performance is evaluated by comparing the P, R, and PR gains of the Full View and the Best View. Table 2 shows the results (including the training image lvi2). Though 4 cases have small negative gains, on average the PR gain is about 25%.

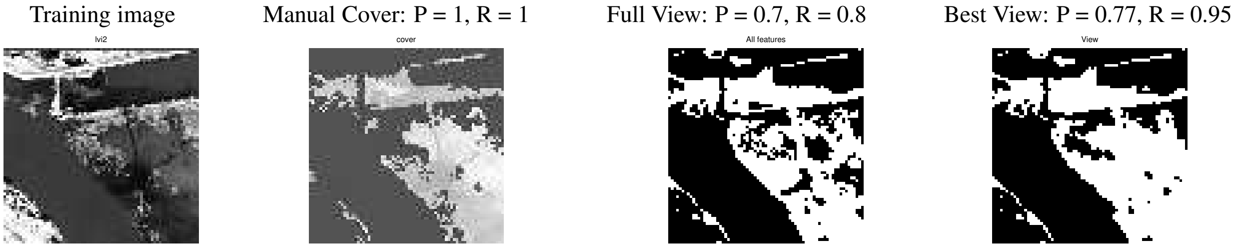

Figure 1: Results for the training image (lvi2)

Image	Full Set		Best Subset		PR Gain
	P	R	P	R	
lvi2	0.7077	0.7935	0.774	0.9448	0.3022
bb1	0.9420	0.5563	0.9700	0.6865	0.2707
di1	0.7685	0.5855	0.8151	0.6778	0.2278
ft1	0.4582	0.8421	0.4308	0.9570	0.0685
lps1	0.1135	0.8817	0.2342	0.9637	1.2553
ls1	0.3470	0.7462	0.4841	0.5648	0.0560
qi1	0.5874	0.8456	0.6833	0.9044	0.2442
vc1	0.8522	0.4287	0.9345	0.6217	0.5902
wi1	0.3193	0.7109	0.6009	0.8651	1.2901
ft1_2	0.7647	0.6658	0.8165	0.7385	0.1843
ft2	0.9807	0.6707	0.9840	0.6711	0.0040
ft3	0.8279	0.4211	0.9267	0.6058	0.6103
lvi1	0.9191	0.3695	0.9304	0.3043	-0.1663
ri1	0.5551	0.5013	0.6478	0.3599	-0.1622
vc1_2	0.4013	0.5706	0.4866	0.6718	0.4276
wi1_2	0.9489	0.5330	0.9632	0.5321	-0.0038
bb2	0.9161	0.4517	0.9261	0.4259	-0.0468
bb12	0.8358	0.5349	0.9470	0.5667	0.2004
ls1_2	0.4812	0.7764	0.5645	0.7105	0.0735
sl1	0.9605	0.6818	0.9733	0.6984	0.0380
sl2	0.9070	0.5579	0.9179	0.5672	0.0289
Avg	0.6949	0.6250	0.7624	0.6680	0.2471

Table 2: Testing Results. Avg denotes Average.

4 Conclusion

We identify the need for feature selection via learning, show that a full set can make image mining unnecessarily more complicated and less efficient and effective, then propose and verify a learning approach to feature selection. Our system can learn good subsets from a training image data set and then the subsets can be applied to unseen images for mass processing of image mining tasks. With this approach, domain experts do not need to determine what relevant features are, but only need to focus on the tasks they are best at (in this case, identifying if a block is Egeria or not in a training image).

Acknowledgements

We really appreciate the help rendered by Amit Mandvikar and Jigar Mody in this project.

References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499, 1994.

[2] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271, 1997.

[3] M. Dash and H. Liu. Feature selection methods for classifications. *Intelligent Data Analysis: An International Journal*, 1(3), 1997.

[4] R. Gonzalez and R. Woods. *Digital Image Processing*. Addison-Wesley, 1992.

[5] D. Hand, H. Mannila, and P. Smyth. *Principles of Data Mining*. A Bradford Book The MIT press, 2001.

[6] H. Liu and H. Motoda. *Feature Selection for Knowledge Discovery & Data Mining*. Boston: Kluwer Academic Publishers, 1998.

[7] A. Natsev, R. Rastogi, and K. Shim. WALRUS: a similarity retrieval algorithm for image databases. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 395–406, 1999.

[8] Y. Rui, T. S. Huang, and S. Chang. Image retrieval: Current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation*, 10(4):39–62, 1999.

[9] I. Witten and E. Frank. *Data Mining: Practical Machine Learning tools and techniques with java implementations*. Morgan Kauffmann, 2000.

[10] O. Zaane, J. Han, and H. Zhu. Mining recurrent items in multimedia with progressive resolution refinement. In *Proc. of ICDE*, 2000.