Rough Set Approach to Domain Knowledge Approximation

Tuan Trung Nguyen and Andrzej Skowron

Warsaw University
ul. Banacha 2, Warsaw, Poland
{skowron,nttrung}@mimuw.edu.pl

Abstract. Classification systems working on large feature spaces, despite extensive learning, often perform poorly on a group of atypical samples. The problem can be dealt with by incorporating domain knowledge about samples being recognized into the learning process. We present a method that allows to perform this task using a rough approximation framework. We show how human expert's domain knowledge expressed in natural language can be approximately translated by a machine learning recognition system. We present in details how the method performs on a system recognizing handwritten digits from a large digit database. Our approach is an extension of ideas developed in the rough mereology theory.

Keywords: Rough mereology, concept approximation, domain knowledge approximation, machine learning, handwritten digit recognition.

1 Introduction

Several decades of research on handwritten digit recognition have yielded significant success, with many efficient algorithms and recognition systems that provide impressive results. Extensive experiments however have shown that no matter how efficient and sophisticated the techniques employed are, a considerable set of samples remain difficult to deal with. This problem is subject to a recently growing trend in the community which stresses on the special treatment of difficult areas in the feature space.

We present a scheme for incorporating domain knowledge about handwritten digit samples into the learning process. The knowledge is provided by an hypothetical expert that will interact with the classification system during a later phase of the learning process, providing certain "guidance" to the difficult task of adaptive searching for correct classifiers.

In distinction to most popular domain knowledge based approaches widely used in recognition systems, ours concentrates on specific difficult, error-prone samples encountered during the learning phase. The expert will pass the correct classification of such cases to the system along with his explanation on how he made the decision on the class identity of the sample. The system then incorporates this knowledge, using its own descriptive language and primitives, to rebuild its classifiers.

G. Wang et al. (Eds.): RSFDGrC 2003, LNAI 2639, pp. 221–228, 2003.
© Springer-Verlag Berlin Heidelberg 2003

In this paper, we describe the expert's knowledge representation scheme, based on the rough mereology approach to concept's approximation, as well as the mechanism of interaction between expert and the classifier construction system presented in [6]. It is easy to observe that the approach is not limited to handwritten digits, but can be readily used in classifying other structured objects.

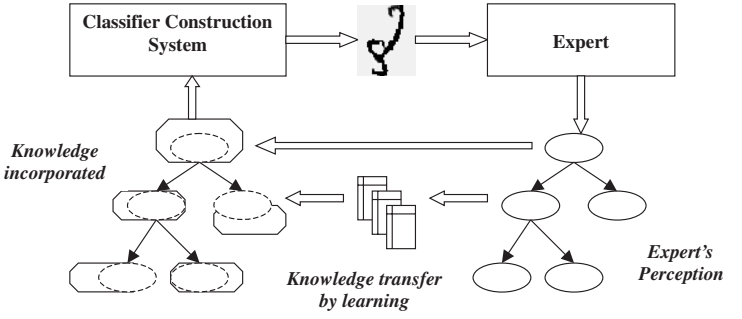

Fig. 1. General Outline

2 Adaptive Recognition System

The development of OCR in general and handwritten digit recognition in particular over the years yielded many highly effective description models for the analysis of digit images. For the research in this paper we have chosen the Enhanced Loci coding scheme, which assigns to every image's pixel a code reflecting the topology of its neighborhood. The Enhanced Loci algorithm, though simple, has proved to be very successful in digit recognition. For a detailed description of the Loci coding scheme, see [2].

Once the Loci coding is done, the digit image is segmented into regions consisting of pixels with the same code value. These regions then serve as *primitives* to build the graph representation of the image.

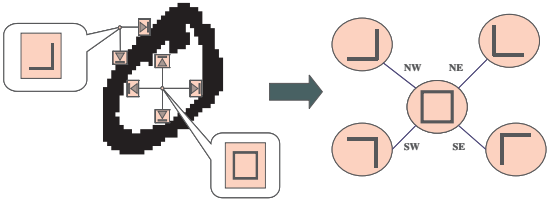

Fig. 2. Graph Model based on Loci Encoding.

3 "Hard" Samples Detection

Typical digit recognition system classifies an unknown sample by computing its "distance" or "similarity" to a collection of prototypes established during the training phase. The most popular, the k-nearest neighbor method assigns the sample to the class represented by the majority of its k neighbors samples. This and other traditional methods perform this task on an uniform basis irrespective to the "difficulty" of the investigated sample, whereas it is obvious that not all samples are equally easy to classify. Samples that are far from the "centers" of the class prototypes tend to fall on the boundaries between classes, are more error-prone and hence can be regarded as more "difficult". A straightforward criteria to detect such samples can be defined as follows:

Let PG_k be the prototype graph set constructed for class k during the training phase and d_k be the distance function established for that class. An unknown digit sample u_k of class k is considered "difficult", "hard" or "atypical" if:

$$d_k(u_k, PG_k) \geq \rho \max\{(v, PG_K) : v \in TR \wedge CLASS(v) = k\}$$

where $\rho \in (0, 1]$ is some cut-off threshold and TR is the training table.

Alternatively, samples repeatedly misclassified during cross-validation tests in the training phase can as well be considered "difficult". Since class identity of all digit samples are known during the training phase, we can detect the "hard" ones beforehand and submit them to the expert for review.

4 Passing Domain Knowledge to Classifiers

Now suppose that at some point during the training phase, we detect a number of samples of class k that are misclassified. The samples are submitted to the expert, which returns not only the correct class identity, but also an explanation on *why*, and perhaps more importantly, *how* he arrived at his decision.

We will assume that the expert's explanation is expressed as a rule:

$$[CLASS(u) = k] \equiv \Im(EFeature_1(u), ..., EFeature_n(u))$$

where $EFeature_i$ represents the expert's perception of some characteristics of the sample u, while synthesis operator $\Im$ represents his perception of some relations between these characteristics.

It is assumed that the actual structure of $EFeature_i$ might not be flat, but can be multi-layered with various sub-concepts at subsequent levels of abstractions. For example, the expert may express his perception of digit '5' as (See Fig. 3):

$[CLASS(u) =$'5'$] \equiv a, b, c, d$ are parts of u; "Above_Right"(a,b); "HStroke"(a);
$b = Compose(c, d)$; "VStroke"(c); "WBelly"(d); "Above"(c, d) hold

where *Compose* is an assembling operator that produces a bigger part from smaller components.

The above means if there is a west-open belly below a vertical stroke and the two have a horizontal stroke above-right in the sample's image, then the sample is a '5'.

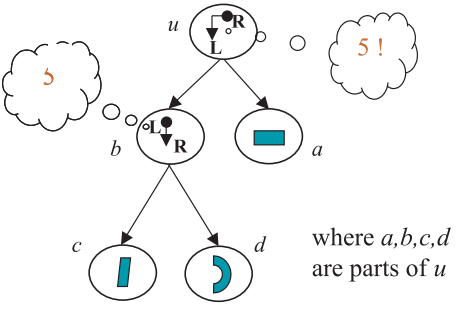

Fig. 3. Object Perception Provided by Experts

The main challenge here is that the expert explanation is expressed in his own descriptive language, intrinsically related to his natural perception of images and often heavily based on natural language constructs (a *foreign language* L_f), while classifiers have a different language designed to, for example, facilitate the computation of physical characteristics of the images (a *domestic language* L_d). For example, the expert may view sample images as a collection of shapes or strokes ("A '6' is something that has a neck connected with a circular belly") while the recognition system regards the samples as graphs of Loci-based nodes. The knowledge passing process hence can be considered as approximating of expert's concept by the classifier construction system.

It is essential here that the concept matching should not be "crisp", but expressed by some rough inclusion measures, determining if something is satisfying the concept to a certain degree [6]. For instance, a stroke at 85 degree to the horizontal can still be regarded as a vertical stroke, though obviously with a degree less than 1.0. The extent of such variations may be provided by the expert (e.g., by providing samples that represent "extremes" instances of a given concept).

Let us assume that such an inclusion measure is denoted by $Match(p, C) \in [0, 1]$, where p is a pattern (or a set of patterns) encoded in L_d and C is a concept expressed in L_f. An example of concept inclusion measures would be:

$$Match(p, C) = \frac{|\{u \in T : Found(p, u) \wedge Fit(C, u)\}|}{|\{u \in T : Fit(C, u)\}|}$$

where T is a common set of samples used by both the system and the expert to communicate with each other on the nature of expert's concepts, $Found(p, u)$ means a pattern p is present in u and $Fit(C, u)$ means u is regarded by the expert as fit to his concept C.

Our principal goal is, for each expert's explanation, find sets of patterns Pat, Pat_1,...,Pat_n and a relation $\Im_d$ such that

$$\textbf{if } (\forall i : Match(Pat_i, EFeature_i) \geq p_i) \wedge (Pat = \Im_d(Pat_1, ..., Pat_n))$$
$$\textbf{then } Quality(Pat) > \alpha$$

where $p, p_i : i \in \{1, .., n\}$ and α are certain cutoff thresholds, while the $Quality$ measure, intended to verify if the target pattern Pat fits into the expert's concept of digit class k, can be any, or combination, of the following criteria:

$$Support_{CLASS=k}(Pat) = |\{u \in U : Found(Pat, u) \wedge CLASS(u) = k\}|$$

$$Match_{CLASS=k}(Pat) = \frac{Support(Pat)}{|\{u \in U : Found(Pat, u)\}|}$$

$$Coverage_{CLASS=k}(Pat) = \frac{Support(Pat)}{|\{u \in U : CLASS(u) = k\}|}$$

where U is the training set.

In other words, we seek to translate the expert's knowledge into the domestic language so that to generalize the expert's reasoning to the largest possible number of physical digit samples. The requirements on inclusion degrees ensure the stability of the target reasoning scheme, as the target pattern Pat retains its quality regardless of deviations at input patterns Pat_i as long as they still approximate the expert's concept $EFeature_i$ to degrees at least p_i. This may also be described as *pattern robustness*.

Another important aspect of this process is its *concept approximation robustness*, meaning not only does it ensure that the target pattern Pat will retain its quality with regard to input patterns deviations in inclusion degrees, but it also should guarantee that if we have some input patterns Pat_i' equally "close" or "similar" to $EFeature_i$, then the target pattern $Pat' = \Im_d(Pat_1', ..., Pat_n')$ will meet the same quality requirements as Pat to a satisfactory degree. This leads to an approximation of $EFeature_i$ that is independent from particular patterns Pat_i, allowing us to construct approximation schemes that focus on inclusion degrees p_i rather than on a specific input patterns Pat_i.

One can observe that the main problem that poses here is how to establish the interaction between the expert who reasons in L_f and the classifier construction system that uses L_d. Here, once again, the system has to learn (with the expert's help) "what he meant when he said what he said." More precisely, the system will have to construct the measure $Match$ and the relation $\Im_d$.

In order to learn the measure $Match$, which essentially means we are trying to learn the expert's concept of $EFeature_i$, we will ask the expert to examine a given set of samples U and provide a decision table (U, d) where d is the expert decision whether $EFeature_i$ is present in a particular sample from U, for instance, whether a sample has a "WBelly" or not. We then try to select a set of features in the system's domestic language that will approximate the decision d, for example, number of pixels with the NES Loci code. For example:

	WBelly
u_1	yes
u_2	no
...	...
u_n	yes

$\Rightarrow$

	#NES	WBelly
u_1	252	yes
u_2	4	no
...	...	...
u_n	90	yes

In the above table, #NES is the number of white pixels that are bounded in all directions except to the West.

It is assumed that the set U will not be too large to ensure the feasibility of acquiring the expert's answers, which facilitates this feature selection task. Experiments have shown that for popular features such as the presence of a circle or a straight stroke, sometimes it is enough to employ some simple greedy heuristics. For more complex patterns, one can use some efficient evolutionary strategies.

Having approximated the concepts $EFeature_i$, we can try to translate the expert's relation $\Im$ into our $\Im_d$ by asking the expert to go through U and provide us with the additional attributes of whether he found the $EFeature_i$ and a decision d if the relation $\Im$ holds. We then replace the attributes corresponding to $EFeature_i$ with the characteristic functions of the domestic feature sets that approximate those concepts and try to add other features, possibly induced from original domestic primitives, in order to approximate the decision d. Again, this task should be resolved by means of adaptive or evolutionary search strategies without too much computing burden.

Here is an example how the concept of a "vertical stroke" "above" a "west-open belly" would be approximated:

	VStroke	WBelly	Above
u_1	yes	yes	yes
u_2	yes	no	no
...	...	...	...
u_n	yes	yes	no

$\Rightarrow$

	#V_S	#NES	$S_y < B_y$	Above
u_1	34	252	yes	yes
u_2	45	4	no	no
...	...	...	...	...
u_n	40	150	no	no

$\Downarrow$

	$Match$(#V_S,VStroke)	$Match$ (#NES,WBelly)	$Match$ ($S_y < B_y$)	$Match$(Above)
u_1	0.85	0.95	(yes,1.0)	(yes, 0.9)
u_2	0.95	1.0	(no, 0.1)	(no, 0.05)
...	...	...	...	...
u_n	0.90	0.70	(no, 0.3)	(no, 0.15)

In the above table, #V_S is the number of black pixels having the Loci code characterizing a vertical stroke and $S_y < B_y$ tells whether the median center of the stroke is placed closer to the upper edge of the image than the median center of the belly. The third table shows degrees of inclusion of these domestic features in the original expert's concept "VStroke", "WBelly" or "Above" respectively.

It is noteworthy that the concept approximation process should work under a requirement to the quality of the searched global pattern Pat, which should have a substantial support among other samples, not examined by the expert, from the training collection. This will ensure the knowledge passed by the expert on a particular example is actually generalized into more global concept.

5 Classifying Unknown Samples

Once we have established domestic language feature sets and constrain relations approximating the expert's reasoning on a particular type of digits, we essentially obtained a multi-layered reasoning scheme. An unknown sample can then be checked against this scheme to see whether it bears enough characteristic traits of this digit class. This can be done by decomposing the unknown pattern according to the structure of the reasoning scheme, checking its matching degree at each level and subsequent computing its matching degree at higher levels up to the root.

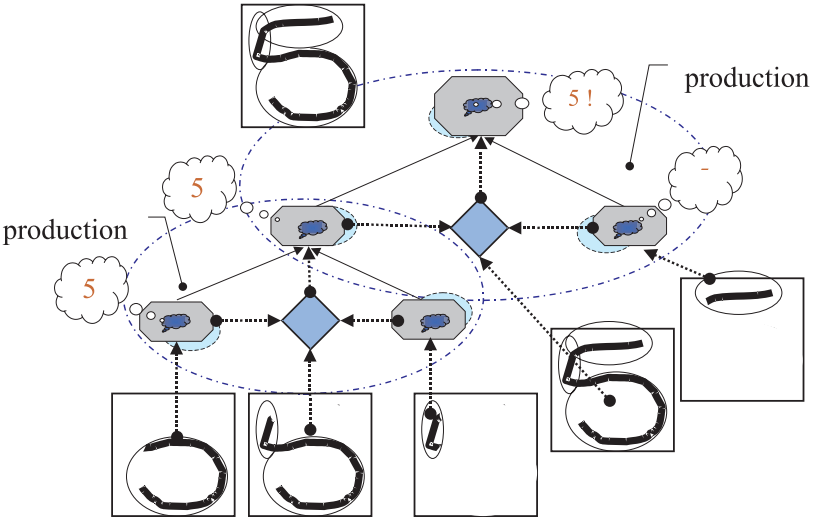

Fig. 4. *AR*-scheme Recognizing New Samples

It can be observed that each pattern and constrain relation set at a particular level of reasoning, called production (see Fig. 4.), determines a cluster of samples matching it. Based on the similarity measures developed during the training phase that correspond to the pattern sets, we can derive the distance of unknown samples to each cluster and, in consequence, develop the inclusion measure of a new sample in the concept approximated by the cluster. Such productions can be composed into approximate reasoning schemes (*AR*-schemes) under constraints [6] expressing that the quality of required input pattern by a production in *AR*-scheme is lower than delivered by production sending such pattern.

It is also essential to note that the quality requirement imposed while searching for the target patterns ensure that the obtained classifier is stable, i.e. resistant to certain derivations in the input sample. It is enough for the new input to match the basic patterns at the lowest level Pat_i to a degree greater than the satisfactory threshold p_i, and the outcome classification decision will be guaranteed to at least a satisfactory degree of accuracy.

6 Conclusion

A method for incorporating domain knowledge into the design and development of a classification system is presented. We have demonstrated how approximate reasoning scheme can be used in the process of knowledge transfer from human expert's ontology, often expressed in natural language, into computable pattern features. Developed schemes ensure stability and adaptability of constructed classifiers. We have shown how granular computing, equipped with rough mereology concepts can be effectively applied to a highly practical field such as OCR and handwritten digit recognition. Developed framework might as well be used in classifying other objects including, but not limited to, fingerprints, mugshots, iris scans as well as for more complex tasks like project WITAS (http://www.ida.liu.se/ext/witas/).

Acknowledgment. This work has been supported by Grant 8T11C02519 from the State Committee for Scientific Researches of the Republic of Poland (KBN) and partially by the Wallenberg Foundation Grant.

References

1. J. Geist, R. A. Wilkinson, S. Janet, P. J. Grother, B. Hammond, N. W. Larsen, R. M. Klear, C. J. C. Burges, R. Creecy, J. J. Hull, T. P. Vogl, and C. L. Wilson. The second census optical character recognition systems conference. *NIST Technical Report NISTIR 5452*, pages 1–261, 1994.
2. K. Komori, T. Kawatani, K. Ishii, and Y. Iida. A feature concentrated method for character recognition. In Bruce Gilchrist, editor, *Information Processing 77, Proceedings of the International Federation for Information Processing Congress 77*, pages 29–34, Toronto, Canada, August 8–12, 1977. North Holland.
3. Z.C. Li, C.Y. Suen, and J. Guo. Hierarchical models for analysis and recognition of handwritten characters. *Annals of Mathematics and Artificial Intelligence*, pages 149–174, 1994.
4. L. Polkowski and A. Skowron. Towards adaptive calculus of granules. In L.A. Zadeh and J. Kacprzyk, editors, *Computing with Words in Information/Intelligent Systems*, pages 201–227, Heidelberg, 1999. Physica-Verlag.
5. R.J. Schalkoff. *Pattern Recognition: Statistical, Structural and Neural Approaches.* John Wiley & Sons, Inc., 1992.
6. A. Skowron. Towards intelligent systems: Calculi of information granules. *Bulletin of the International Rough Set Society*, 5(1–2):9–30, 2001.