Clustering using the Minimum Message Length Criterion and Simulated Annealing

Ian Davidson
CSIRO Australia, Division of Information Technology, 723 Swanston Street, Carlton, Victoria, Australia 3053, inpd@mel.dit.csiro.au

Abstract. Clustering has many uses such as the generation of taxonomies and concept formation. It is essentially a search through a model space to maximise a given criterion. The criterion aims to guide the search to find models that are suitable for a purpose. The search's aim is to efficiently and consistently find the model that gives the optimal criterion value. Considerable research has occurred into the criteria to use but minimal research has studied how to best search the model space. We describe how we have used simulated annealing to search the model space to optimise the minimum message length criterion. Simulated annealing can escape local minima and by changing some of its parameters we can tradeoff solution quality against computation time.

1. Introduction

Clustering often referred to as unsupervised classification or intrinsic classification has a long history in numerical taxonomy [1] and machine learning [2]. It has many applications such as the generation of taxonomies for flora and fauna, concept formation and data mining. Clustering systems work on a collection of objects each described by a set of n attributes. Clustering attempts to create classes and their descriptions and then assign the objects to one or more of these classes. As no training set of pre-classified objects exists, and the number of classes is unknown, clustering is unsupervised. The class descriptions form a model or hypothesis about the objects.

Clustering systems consist of three major parts. The knowledge representation scheme (KRS) defines the model space to search. The criterion provides a "goodness" value for each model and the search mechanism explores the model space attempting to find the optimal criterion value. The system attempts to find the best model for the given objects.

The KRS determines the type of classes and their possible interrelationships. The KRS can be amongst other things, probabilistic or deterministic, hierarchical or non-hierarchical and exclusive or non-exclusive. A dichotomy for clustering options which impact on the KRS has been defined elsewhere [3]. The choice of KRS determines the searchable model space.

The criterion evaluates the "goodness" for each of the models. It is usually a real value function that takes as parameters the objects and/or class descriptions and is the objective function of the search. The instance pair (H,C), can formally describe a clustering problem. H is the finite set of all models in the searchable space and C is the objective, cost or goodness function such that $C{:}H \rightarrow \Re$. That is the objective function assigns a real number to each model. Considerable research has occurred into finding criterion to use leading to many different types such as Euclidean, Manhattan, probabilistic, information theoretic and domain specific heuristics [3].

The search mechanism explores the model space attempting to find the best model by finding the optimal (either minimum or maximum) value of the objective function. For all but the most restrictive model spaces the number of possible models to evaluate is combinatorially large. Exhaustively evaluating each possible one is not even considered as a search mechanism. The search mechanism must consistently find the global optima or at least good local optima in a number of different application domains computational inexpensively. Fisher describes several search strategies currently used in clustering [4].

We believe that making use of the search techniques from combinatorial optimisation to search the model space has benefits. The simulated annealing (SA) algorithm has solved a large variety of NP hard problems in different domains such as the traveling salesman problem, circuit board layout and scheduling [12]. If used to search the model space the annealing algorithm's problem independence nature should allow application of the clustering system into different application domains with minimal changing. Furthermore by changing parameters of the annealing algorithm (namely the annealing schedule) we can tradeoff the quality of solution against convergence time. If implemented correctly without violation of requirements [11] SA is statistically guarantees to find the global optima. However this is computationally very expensive. To use simulated annealing to search the model space in finite time would require a different implementation.

In this paper we describe a clustering system that uses a non-hierarchical, probabilistic, exclusive knowledge representation scheme. That is, no relationship exists between classes, the properties of each class are probabilistic and assignments of an object is only to one class. We use the minimum message length criterion first described in [5] and search the model space using simulated annealing. Our results indicate the clustering system finds good local optima. We describe our application of the clusterer on Michalski's soybean collection.

2. The Minimum Message Length Approach to Induction

Most clustering systems are inductive in principle. Induction involves reasoning from few examples to a generalisation. The objects to cluster are a sample of objects from some population. A class description generated from the sample objects can make predictions about objects not in the sample but in the population.

Chaitin [6], Kolmogorov [7] and Solmonoff [8] in varying forms independently proposed algorithmic information theory (AIT). AIT intuitively allows us to quantify the notion of complexity and compressibility of objects. Learning by induction is inherently about compressing observations (the objects) into a theory (the model). Boyle's law on ideal gases relates the number of molecules (N) in a measurable closed volume (V) to pressure (P), that is $P = k.N/V$. A table could store every possible combination of N and V and the resultant pressure. However, Boyle's law compresses this table into a much shorter description, the above equation.

Wallace and Boulton [5], extend this compressibility notion into their minimum message length (MML) approach to induction. They define a criterion which can be used to select the most probable model from a given set of mutually exclusive and exhaustive models, H^*, for the objects, D. The MML approach specifies that the minimal (in terms of length) encoding of the model and the objects given the model is the best. For each different model we can calculate the total encoding length. In terms of Bayes theorem, we wish to maximise the posterior distribution, *P(H_i | D,c)* where *c* is the background context :

$$P(H_i|D,c)=\frac{P(H_i|c).P(D|H_i,c)}{P(D)} \tag{1}$$

taking the logarithm of this expression yields

$$-\log P(H_i|D,c)=-\log P(H_i|c)+-\log P(D|H_i,c)+const \tag{2}$$

Our interest is in comparing relative probabilities so we can ignore *const.* Information theory [Shannon] tells us that -log *(P(occurrence))* is the minimum length in bits to encode the occurrence. Hence by minimising the equation described in (2) we inherently maximise the posterior distribution and find the most probable model. The expression to minimise has two parts, the first being the encoding of the model and the second the encoding of the objects given the model. The object collection is random if the size of encoding the model and the objects given the model is approximately equal to the size of directly encoding the objects. That is there is no way to compress the objects into a shorter description/theory. The two part message is precisely described for intrinsic non-hierarchical classification in [5] and [9].

The MML criterion only defines a goodness measure for a model with an inherent bias towards simple models. It does not indicate how to search the model space. To do that we use simulated annealing.

3. The Simulated Annealing Algorithm

The Metropolis criterion was first used as a Monte Carlo method for the evaluation of state equations in statistical mechanics by Metropolis et al. [10]. Kirkpatrick et al. [11] demonstrated how using the Metropolis criterion as a test in iterative optimisation can solve large combinatorial optimisation problems. They called their approach the simulated annealing technique as it mimics the annealing of a piece of metal to minimise the energy state of the molecules within the metal. SA is an iterative local optimisation technique. At any time there is only one current solution (unlike genetic algorithms where there are many) which is slightly changed at each iteration. As SA is a Markov process the current solution at time, *n*, $S_{n,}$ is a result of the perturbation of S_{n-1}. The algorithm continually perturbs the current solution to generate new candidate solutions. SA unconditionally accepts candidates of better quality than the previous solution and conditionally accepts those of a worse quality with a probability *p,* where:

$$p=e^{-\left(\frac{|Difference\ in\ Quality|}{System\ Temperature}\right)} \tag{3}$$

Allowing worse quality solutions to be accepted allows escaping from local minima which are common in most complex search problems [11]. We set the initial temperature T_0, so there is a 50% probability of accepting an increase in cost. This probability decreases as the temperature decreases. The cooling constant, *R* reduces the temperature such that, T_k = $T_{k-1}.R$.

$$T_0 = -\frac{C_0}{\log_e(0.5)} \tag{4}$$

C_0 is the goodness evaluation of the initial solution. The algorithm in its simplest form follows:

```
Choose an initial solution and set the initial temperature
Optimise:     Generate a new solution by perturbing the old solution
              If quality of new solution is better than of the old solution
                    old solution := new solution
                 Else with probability, $p$ (from equation 3)
                    old solution := new solution
              If no increase in quality is found after a long time(equilibrium
                 is found at this temperature) reduce temperature.
                 If temperature equals 0 Stop
              Goto Optimise
```

Simulated annealing statistically guarantees to find the global optimum, if the thermodynamic equilibrium is found at every temperature and the cooling schedule is slow enough [12]. However, this is an infinitely long process. We do not maintain these two requirements due to the need to find a solution in finite time. Instead after a fixed number of iterations at a temperature the temperature is reduced and the cooling constant provides discrete changes in the temperature. However non-ideal SA approaches such as ours still results in good solutions, though there is no guarantee of optimality [12].

4. Searching the Model Space Using SA

We discuss our model space and how the SA algorithm searches it. We define a taxonomy as a set of class description and the placement of objects into a class as assignment. For simplicity we only describe the situation where multi-state/nominal attributes represent objects. We can calculate the message length from a solution to the clustering problem. A solution consists of a proposed taxonomy and the assignment of objects to this taxonomy. The taxonomy consists of a number of classes. Each class provides probabilistic measures of attribute values if an object were to belong to that class. We assign the objects based on their probability of belonging to the classes. Suppose an object *b* has a 25% and 75% probability of belonging to classes *C* and *D* respectively. We generate a random number between 0 and 1 and assign *b* to *C* if the number is less than or equal to 0.25 otherwise to *D*. The probabilistic representation used assumes that each attribute is independent of each other. An example solution for a simple taxonomy follows.

Let: n = 2, s = 2, m = 2, d = 3
where
n is the number of classes.
s is the number of objects to classify.
m is the number of attributes each object possesses.
d is the number of values that each attribute can take on.

The two objects to cluster, O_1 and O_2 are described by two attributes M_1 and M_2:

	O_1	O_2
M_1	3	3
M_2	1	1

The two class taxonomy description follows:

	Class 1	Class 2
Pr(M_1 = 1)	0.8	0.1
Pr(M_1 = 2)	0.1	0.1
Pr(M_1 = 3)	0.1	0.8
Pr(M_2 = 1)	0.3	0.8
Pr(M_2 = 2)	0.6	0.1
Pr(M_2 = 3)	0.1	0.1

We treat each attribute as being independent from every other. The probability of object, O_i being in class T_j is the product of the probabilities for each attribute in the class having that object's values. For example the probability that O_2 belongs to Class 1 and 2 is (0.1 * 0.3 = 0.03) and (0.8 * 0.8 = 0.64) respectively. We assign objects exclusively to one class from these probabilities.

The initial solution is generated by selecting a random number of classes and generating uniform distributions for each attribute in each class. The following algorithm summarises our clustering system. Perturbation operators and application strategies follow in the next section.

```
Choose an initial taxonomy of a randomly generated number of classes with
    uniform distributions for each attribute.
Assign objects to classes.
OldLength := MessageLength(Initial Taxonomy and Object Assignment)
Temperature := - OldLength / $\log_e(0.5)$
Old taxonomy := initial taxonomy

While(Temperature > 0)
    Repeat Markov_cycle_length times (default is 30)
          New taxonomy = small perturbation of the old taxonomy
          Assign objects to classes.
          NewLength := MessageLength(New Taxonomy and Object
          Assignment)
          If NewLength < OldLength
                    old taxonomy := new taxonomy
          Else with probability, $p$
                    old taxonomy := new taxonomy
    End Repeat
    Temperature := Temperature.CoolingConstant
EndWhile
```

5. Perturbation Operators and Application Strategies

The perturbation operators change an existing taxonomy to generate a new one. The operators should fulfill the SA requirements [11] of being able to explore all models in the model space and making only local changes. To fulfill the first requirement, our perturbation operators can generate any taxonomy from any other over sufficient iterations. The changes made to taxonomies effect only a small local aspect of the taxonomy. This fulfills the second requirement.

Generating new class descriptions after assigning the objects into the current taxonomy is a natural perturbation operator. New probability distributions on attributes for each class can be derived from the objects currently within the class. Assigning objects to classes at iteration *n* leads to a new taxonomy for use at iteration *n+1*. This operator does not result in increasing the number of classes though they can be decreased if no objects are assigned to a class. Only the probability distributions on the attributes are changed and usually only slightly.

If probabilistic assignment were the only perturbation operator, the number of classes would only decrease or stay constant and never increase as the search continued. Hence, we require additional perturbation operators. These include:

Splitting a large class into two. A randomly selected class containing more than 25% of the objects is split into two classes. Half the attributes for the existing class are randomly selected and copied into the new class along with their probability distributions. Attributes not copied have their distributions randomly generated.

Imperfect Duplication of a class. A randomly selected class has its description copied with a small fixed probability (5%) of changing the probability distributions for an attribute within a range of 10%. A repairer algorithm maintains that the sum of the distribution is 1.

We have three perturbation operators: probabilistic assignment, splitting and imperfect duplication. The strategies of applying these perturbation operators follow:

Fixed Random. A constant probability of applying the perturbation operators occurs during the optimisation process.

Adaptive Random. The probability of applying the perturbation operator changes during the optimisation process. The changes occur from feedback on how successful an operator has been at improving the quality of the solution.

Temperature matched. Perturbation operators are applied with probabilities dependent on the current temperature.

Fixed Sequence. Predefined sequences of perturbation operators are randomly applied during the optimisation process. For example one sequence could be: 30 iterations of assignment, 3 of split and 5 of duplication followed by 50 of iterations of assignment.

6. Experiments

We have tried our clusterer on Michalski’s large soybean disease collection [13]. The collection consists of 290 training examples each described by 35 attributes. We remove twenty four examples containing unknown values. Each example belongs to one of 15 classes of soybean disease. The reported best prediction results on the test set are 97% for supervised classification. We conducted applications of the cluster for varying annealing schedules. Reducing the cooling constant or the Markov cycle length allows faster convergence to a reduced quality solution (known as quenching). Higher cooling constants and Markov cycle length produces better quality solutions but convergence is slower. Table 1 shows our results at a cooling constant of 0.90 and a Markov cycle length of 30. The diseases were identified by experts and form the extrinsic categories whilst the single letter named classes were found by the cluster. The total number of instances (count column) for each disease and their distribution into the classes found by the cluster are shown.

	Classes														
Disease	**Count**	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**
diaporthestem-canker	10	0	0	0	0	0	0	0	0	10	0	0	0	0	0
charcoalrot	10	0	0	0	0	0	10	0	0	0	0	0	0	0	0
rhizoctoniarootrot	10	0	0	0	0	0	0	0	0	0	0	0	10	0	0
phytophthorarot	16	0	16	0	0	0	0	0	0	0	0	0	0	0	0
brownstemrot	20	0	0	0	0	0	0	0	0	0	0	0	0	20	0
powderymildew	10	0	0	0	0	0	0	0	0	0	0	10	0	0	0
downymildew	10	0	0	0	0	10	0	0	0	0	0	0	0	0	0
brownspot	40	37	0	0	0	3	0	0	0	0	0	0	0	0	0
bacterialblight	10	0	0	0	7	0	0	3	0	0	0	0	0	0	0
bacterialpustule	10	0	0	0	9	0	0	0	1	0	0	0	0	0	0
purpleseedstain	10	0	0	0	0	0	0	0	10	0	0	0	0	0	0
anthracnose	20	0	0	0	0	0	0	0	0	20	0	0	0	0	0
phyllostictaleaf-spot	10	0	0	0	0	0	0	0	0	0	9	0	0	0	1
alternarialeafspot	40	0	0	0	0	0	0	32	0	0	0	0	0	0	8
frogeyeleafspot	40	0	0	39	0	0	0	0	0	1	0	0	0	0	0
Total	266	37	16	39	16	13	10	35	11	31	9	10	10	20	9

Table 1: Results of Application of Clustering System to Soybean training set. Cooling constant 0.90, Markov cycle length 30.

Our clustering system divided the examples into 14 classes instead of the 15 discovered by experts. It placed four diseases (charcoalrot, rhizoctoniarootrot, phytophthorarot and brownstemrot) into four separate homogenous classes and placed the majority of two other diseases (brownspot and frogeyeleafspot) into homogenous classes but misclassifed some members into other classes. It merged all examples of the diaporthestemcanker and anthracnose diseases into one class and the majority of examples of the bacterialblight and bacterialpustule into one class. The remaining diseases examples were mostly placed into separate classes but quite a few misclassifications occurred. Slightly faster convergence times resulted in more misclassifications. Extremely fast convergence (cooling rates of less than 0.5) resulted in the merging of multiple diseases into a single class. This could have also been achieved by weighting the first part of the message to be more greater than the second, thus penalising complex models.

7. Discussion

Understanding the effect and order of application of the perturbation operators are vital when searching the model space. For example, we found in our trials the lack of frequently applying the imperfect duplicate or split operator meant the search became trapped in a poor local minimum. The temperature matched strategy of applying perturbation operators was the best performing for this domain. We match the probability of applying the split and imperfect copy perturbation operators with the annealing schedule. The probability of applying these operators at the initial temperature was 10% and decreased in linear accordance with the temperature. Their application caused cost increases when initially applied, however after several iterations of probabilistic assignments the cost improved. Applying these operators at higher temperatures allowed the changes made to have a greater chance of being accepted.

There are many aspects of the classifier we would like to further study. The annealing schedule heuristic is not ideal. The annealing principal relies on finding equilibrium at each temperature in the annealing schedule. We do not check for this. We assume we find equilibrium after a fixed number of iterations at a temperature. We would like to follow a more adaptive approach that could stay at a temperature longer or even increase the temperature to satisfy the equilibrium requirement.

8. Conclusion

We describe our implementation of a clustering system that uses the MML criterion to evaluate the worth of taxonomies and simulated annealing to search the model space. We feel SA is a good search mechanism to explore the model space. It can escape local minima that exists in complex search spaces and is robust across a variety of problems. This should allow the clustering system's application in different domains without significant refinement. By changing the cooling constant and Markov cycle length we can tradeoff quality of solution against convergence time.

9. References

1. Dunn, G., Everitt, B.S., An Introduction to Mathematical Taxonomy, Cambridge University Press (1982)
2. Michalski, R.S., and Stepp, R., Learning from observation: conceptual clustering, In Michalski, R.S., Carbonell, J.G., Mitchell, T.M., editors, Machine learning : an artificial intelligence approach, CA: Morgan Kaufmann (1983)
3. Clifford, T., Stephenson, W., An introduction to numerical classification, Academic Press (1975)
4. Fisher, D., Iterative Optimization and Simplification of Hierarchical Clustering, Technical Report CS-95-01 Department of computer Science Vanderbilt University (1995)
5. Wallace, C.S., Boulton D.M., An Information Measure for Classification, Computer J Volume 11 No. 2, (1968) 185-194
6. Chaitin G., On The Difficulty of Computations, IEEE Information Theory, IT-16 (1970) 5-9
7. Kolmogorov A., Logical Basis for Information Theory and Probability Theory, IEEE Transactions of Information Theory and Control IT-14 (1965) 662-664
8. Solomonoff, R., A Formal Theory of Inductive Inference: Part 1, IEEE Information Theory and Control 7 (1964) 1-22

9. Wallace C.S., An Improved Program for Classification, 9th Australian Computer Science Conference (ACSC 9) Volume 8 No. 1 (1986) 357-366
10. Metropolis, N. et al, Equation of State Calculations by Fast Computing Machines, The Journal of Chemical Physics Volume 21 #6 June (1953) 1087-1092
11. Kirkpatrick, S., Gelati C., Vecchi, M., Simulated Annealing, Science 220, (1983) 671-680
12. Van Laarhoven, L., Theoretical and Computational Aspects of Simulated Annealing, CWI Tract (1988)
13. Michalski R.S., Chilausky, R.L., Learning by Being Told and Learning from Examples: An Experimental Comparison of the Two Methods of Knowledge Acquisition in the Context of Developing an Expert System for Soybean Disease Diagnosis, International Journal of Policy Analysis and Information Systems Vol. 4 No. 2 (1980)