Development of an Intelligent System for Organic Compound Analysis

Curtis Huttenhower
huttence@rose-hulman.edu
Dr. Andrew Kinley
kinley@rose-hulman.edu

Introduction

The Organic Compound Analysis system, or OrCA, has been developed to integrate artificial intelligence techniques in an effort to elucidate chemical relationships. The first of its two major design goals is to study the interaction and supplementation of disparate techniques in artificial intelligence; the second is to examine the ability of a computer to learn and manipulate the complex relationships between various properties of a chemical compound. From a chemistry standpoint, this means that OrCA will operate similarly to an organic chemist. Given simple data regarding a chemical such as is available in a laboratory setting, it will be able to determine more complex and intangible properties of the chemical. >From an artificial intelligence standpoint, it is of interest to study what relationships the system is able to determine, how these relationships develop, and in particular how they are supplemented by the coupling of different aspects of artificial intelligence. To these ends, OrCA was specified, designed, implemented, and tested. Although work on the system is still ongoing, tests to date have shown promising results and helped to elucidate the surprising abilities of even an incomplete version of the system.

Throughout the paper a slight knowledge of chemical compounds is assumed. Primarily, this revolves around the usage of terms such as functional group, functionality, solubility, and similar chemical properties. In brief, the functional group or functionality of a chemical is a particular atomic grouping which determines the majority of its characteristics. For example, all alcohols share an alcohol functionality; all acids share an acid functionality. The solubility data for a compound consists of a list of other chemicals in which it will or will not dissolve; sugar will dissolve in water, for example, but baking soda will not. Many other chemical terms which come up, such as melting or boiling point, should be apparent from context.

System Capabilities

The OrCA system has been designed to extract the most comprehensive and relevant chemical data from a system based on simple, easily-obtained physical data. The system deals with a wide range of qualities, and although each may serve as either a known or as an unknown, certain qualities are more conveniently obtained than others. Thus, for example, OrCA might be given a sample's melting point, boiling point, color, and odor; it would then attempt to provide the number of carbon atoms the compound contained and its major functional group, both with certainties.

The usefulness of OrCA spans both chemical and computational fields. For an organic chemist, such a tool would be invaluable in performing routine chemical identifications; for a computer scientist, the interplay of the system's components provides a basis for investigating the combinational capabilities of supposedly disparate facets of artificial intelligence. The strengths and weaknesses of the system provide important clues as to what relationships actually do exist between the various chemical properties, what information similar systems might be able to learn about other fields, and how well the individual components of the mechanism work together. All of these are important and relevant results which will be addressed

later in this document.

The specification of OrCA was reasonably simple and straightforward. The problem goal was specified as:

> "To design and implement a system which will correlate and interpret various types of organic chemical data. The design will allow for a close and useful integration of a number of different artificial intelligence techniques, and the system will utilize these to develop as much output from as little input as possible."

This goal allowed the query/response format of the system to be specified. For any given compound, a limited set of data may be provided, and a full or more complete description expected in return. The formal description of a compound is as follows:

· *Value := Numerical or other representation of a chemical datum*
This includes encodings of colors, temperatures, size, or other chemical properties. The specific encodings are discussed in Table 2.

· *Certainty := 0 - 1*
The certainty of a value is simply a number between zero and one, with zero indicating complete uncertainty and one indicating complete certainty. For example, a one should only appear for an input value, the certainty of which the system is completely sure.

· *State := [Value, Certainty]*
Any quality in a compound may have many states; each state represents a possible value for the quality coupled with the certainty for that value.

· *Quality := (State, State*)*
A quality represents a chemical property of a compound. Color, for example, is a quality which may take on the values red, blue, white, or others.

· *Compound := (Quality, Quality*)*
A compound is the internal representation of an actual chemical compound. It contains a collection of qualities just as a physical chemical has a number of properties of interest.

Similar structural specification of the system itself is given later in the architecture portion of this document.

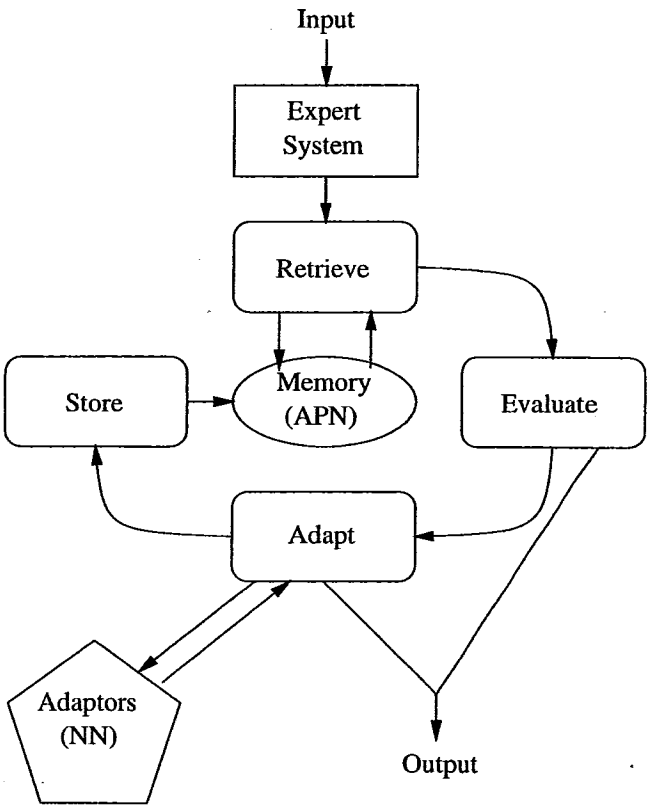

Figure 1: OrCA Architecture Overview

Architecture

With the original specification as described above, it became possible to plan the overall architecture of the system. The primary architecture was based on the case-based reasoning model (CBR) (3). CBR was selected as the mechanism for compound identification as chemical analysis while obeying no small set of hard and fast rules does have some regularity of attributes among different compounds. Aside from the CBR unit, it was decided that there were enough definite, unchanging rules to warrant the addition of an expert system to handle simple cases. With this in mind, the full architectural overview may be seen in Figure 1. The bulk of this diagram is simply standard CBR architecture, but it serves as a starting point and guideline for the construction of OrCA.

Adaptive Probability Network

Fundamentally, a belief or probability network represents a formal means of using incomplete information to generate conclusions with statistically precise uncer-

tainty. They represent causal relationships - if x happens, y has z probability of happening. If it is cloudy, there is a 60% chance of rain. If Bob is up to bat, he has a 30% chance of hitting the ball. However, their power lies in representing much more complex relationships than this. Any combination of events, represented as nodes, may be connected in terms of their causal relationships to form a directed acyclic graph. The development of belief network and Bayesian theory is presented in further detail in resources such as (3).

More relevant to OrCA is the issue of adapting a belief network as new data is provided. Certainly, if such a network is to be used as memory, it must be able to learn and thus remember new examples. Based on previous research ((1), (4), (2)), a number of algorithms were investigated. Belief network learning is based on adjusting the internal probabilities to maximize the likelihood of a given training case or cases. Using standard algorithms, a number of problems arose. Some algorithms were suited only for batch processing of training examples, or were constructed to use functional probability distributions rather than numeric probability data; these would not suit the purposes of OrCA. An algorithm provided in (1) proved adequate, save for a major computational flaw. It processed examples asymmetrically - a training example followed by a perfect counterexample would not return the system to a state of equilibrium. Since this was the only flaw, the algorithm proved to be modifiable. Rather than computing a local gradient descent, the version used in OrCA keeps a running average of the training data seen. This proved to be beneficial in two ways; it corrects the problem of imbalance and also tends to quickly lock on to initial patterns. Both of these served to produce more meaningful results from the system.

The final algorithm as implemented in OrCA may be abstracted as is seen in Table 1.

The design of the network itself proved challenging. As with standard probabilistic network modeling, each node represents an individual quality of a compound. Frequently, probabilistic networks deal only with binary values - yet this is impossible for attributes which span many values, such as color or temperature. The same algorithms and theory which deals with binary nodes, however, are able to deal with

function average-apn(N, D) returns a modified probabilistic network
inputs: N, a probabilistic network with table entries **w**
D, a data case

$\Delta \mathbf{w} \leftarrow 0$
set the evidence in N from D
for each variable i, value j, conditioning case k
if j==k
$\mathbf{w}_{i,j,k} \leftarrow (\mathbf{w}_{i,j,k} * ave_i + 1)/(ave_i + 1)$
else
$\mathbf{w}_{i,j,k} \leftarrow (\mathbf{w}_{i,j,k} * ave_i)/(ave_i + 1)$

return N

Table 1: APN Training Algorithm

multi-valued nodes. Each node encodes a set of values, such as temperature ranges or colors, as simple integers. In this manner, using almost entirely standard belief network propagation algorithms, known values may be entered into the network and unknown values predicted. The problem of dealing with full directed acyclic graphs as opposed to polytrees was dealt with through stochastic sampling. Similarly, cyclic graphs were adequately dealt with by limiting recursion depth within the standard DAG algorithm and assuming an even probability distribution after a particular depth. These solutions provided accurate results in a test environment and continued to perform satisfactorily in the testing implementation of OrCA.

The network design which was decided upon for OrCA is shown in Figure 2. As with any belief network, each node may assume one of many values; the range of values for each node (thus each quality) is shown in Table 2. The network architecture demonstrates the causal relationships which were determined for the qualities under consideration. Both these and the value encodings are based on empirical evidence available both experimentally and through introductory organic texts.

It is imperative when interpreting these tables to realize that the solubility encoding, as opposed to the others, does not correspond to the numerical values as given. Rather, each of the values listed corresponds to an individual node which may assume an output of either zero (is not soluble) or one (is soluble). Ideally, functional groups would follow a similar system, but

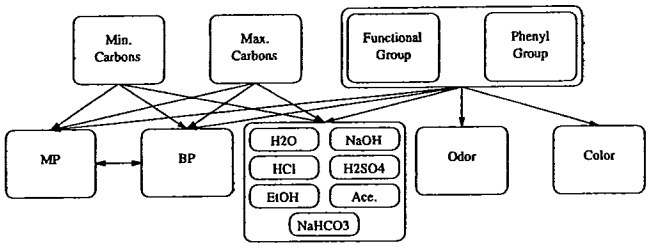

Figure 2: OrCA Belief Network Design

because of the arrangement of network dependencies this would require prohibitively large internal probability tables.

CBR Adapters

As with any case-based reasoner, the adapters in OrCA exist to provide a second chance when the central memory (the APN) fails to produce an answer with adequate certainty. For each property (thus each node in the belief network) there exists an adapter. These are made up of tiny back-propagation neural networks. Any one of them will take a small set of values as input, the values of those qualities designated as dependencies for the node being adapted, and return an output value and certainty for the node under investigation. They are much simpler and lightweight than the APN and effectively reduce each quality to a black box - if the proper values are input, a value and associated certainty are output with a minimum of fuss or specific calculation.

A key property of the adapters in any CBR is that they work to complement the main memory. Where one fails, the other is hoped to succeed. For this reason, only mechanisms significantly different than those used by probabilistic networks were considered for the OrCA adapters. Upon testing of the system, it was quickly seen that the APN which made up the main memory was generally much better at remembering specific examples and simple generalizations than it was at anticipating different yet related cases. Thus, any successful adapters were required to work by a mechanism which would allow them to generalize easily and work with tenuous or dissimilar data. The propagation algorithms used by network derivatives such as neural networks and Hopfield networks seemed a likely candidate for this feature, and it was

Val.	Min. C/Max. C	Func. Group	Phenyl Group
0	1	alcohol	no
1	2	aldehyde	yes
2	3	amide	
3	4	amine	
4	5	acid	
5	6	ester	
6	7 - 9	ether	
7	10 - 12	halide	
8	13 - 15	ketone	
9	16 - 19	nitrile	
10	20 -	nitro	
11		phenol	
12		thio	
13		none	

Val.	MP/BP	Solubilities	Odor	Color
0	-273 - -201	H_2O	no	red
1	-200 - -151	EtOH	bad	orange
2	-150 - -101	Ace.	egg	brown
3	-100 - -51	NaOH	fish	yellow
4	-50 - -1	$NaHCO_3$	medicine	white
5	0 - 49	HCl	fruit	green
6	50 - 99	H_2SO_4	good	blue
7	100 - 149		yes	purple
8	150 - 199			black
9	200 - 249			clear
10	250 - 299			
11	300 - 349			
12	350 -			

Table 2: Belief Network Value Encodings

indeed these structures which were examined for the OrCA system.

Neural Networks

The application of neural networks to the OrCA problem proved much more successful than a number of previous efforts. The network implementation used was, uniformly, a simple three-layer, strongly connected back-propagation design. The number of nodes in any particular adapter was limited such that complexity was sacrificed in favor of speed. It was neither anticipated nor later seen in testing that this had any negative impact on performance. As with the particulars of APN implementation, the algorithms specific to neural networks will not be discussed here; see (3) for more information.

The interface of the adapters with the existing APN and substance specification is of greater interest. Each quality was associated with a list of dependencies and an adapter. The specific dependencies for each node are listed in Table 3. Each quality's adapter was constructed such that the network had an input for every possible value of every dependency and an output for every possibility of the quality being adapted. This will be made clear in the example below. We examine the process of adapting the odor for benzoic acid. >From the known chemical data for this compound and the encodings found in Table 2 we determine that the initial dependency values are four for functional group and one for phenyl group. Thus, the architecture and input values for the odor adapter are seen in Figure 3. The appropriately numbered inputs for functional group and phenyl group are set to one, while the rest are set to zero. Propagation through the network by standard algorithms will result in each output node being set to an activation between zero and one. Sample outputs are given in Table 4. From these, it may be seen that the adapted output would be zero. The certainty is calculated as the normalized network output as follows:

$$cert = \frac{0.66}{0.66+0.23+0.06+0.07+0.15+0.06+0.03+0.27} = 0.43$$

Thus, the final output of this adapter would be a guess of odor "none" with certainty 0.43. Calculations and processing in each adapter are identical to this save for differing dependencies and output ranges.

Quality	Dependencies
Min. C, Max. C	Func. Group, Phenyl Group, MP, BP
Func. Group, Phenyl Group	Solubilities, Odor, Color
MP, BP	Min. C, Max. C, Func. Group, Phenyl Group
Solubilities, Odor, Color	Func. Group, Phenyl Group

Table 3: Quality Dependency Specification

Output Node	0	1	2	3	4	5	6	7
Value	0.66	0.23	0.06	0.07	0.15	0.06	0.03	0.27

Table 4: Sample Adapter Output

Adapter Training

Training in the OrCA adapters is a much simpler process than it is in the APN memory. The neural networks rely only on standard back-propagation algorithms. In addition to the dependency data input into the network, a target quality value is provided to calculate network error. Continuing with the example seen above, to train the odor adapter using data from benzoic acid, the network would be configured as is seen in Figure 3. Back-propagation would then be performed to adjust the network weights, allowing for more accurate prediction if similar inputs (thus similar compounds) are encountered in the future. For more information on neural networks see (3).

Testing and Behavior

A key fact to realize before reviewing the testing of OrCA is that the system under consideration is not a full implementation of the OrCA specification. At the time of this writing, both the initial "guardian" expert system and processing of solubility data remains to be implemented. In some respects this has noticeably impeded the abilities of the system, but in others it appears to have had a significant impact. The portion of the system completed has been provided with a set of training data and queried using a variety of test sce-

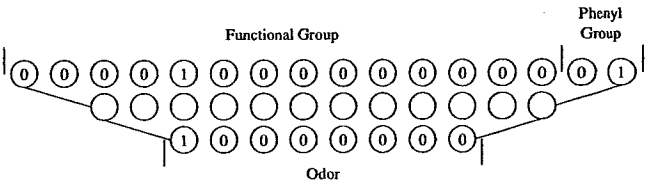

Figure 3: Adapter Training Values

narios. Both the training data and test cases have been designed to be as uniform and relevant as possible so as to provide an accurate sample of the system's capabilities. Both the source data, queries, and results are discussed below.

Test Scenarios

Testing was performed with a case base consisting of approximately sixty cases and utilizing some ten test queries. These queries, though few in number, were designed to test various portions of OrCA's behavior that would be of particular interest in a focused manner. Specifically, these queries consisted of compounds spanning a range of functionalities, quality values, initial data, and case base inclusion. This small but varied set of test cases allowed an in-depth study of a limited, yet sufficient, range of query possibilities. Most importantly, the test scenarios allowed investigation of system response to queries on compounds included in the case base, typical or common compounds not included in the case base, atypical compounds not included in the case base, queries employing very limited or very complete initial data, and queries requesting data on various compound qualities. The training data itself was collected in such a way as to maximize statistical regularity and allow for as regular a distribution of all qualities as possible. The outcome in each of these cases will be discussed below.

System Response

After testing of the OrCA unit was performed, a number of parameters and data were made available for analysis. However, because of the nature of the system, it is difficult to present any such data in numerical format. Also complicating the matter is that the results from a chemical and from a computer science perspective are sometimes intertwined and sometimes distinct; only a heuristic analysis of system performance is obtainable. Thus, this style of general result analysis is provided to the reader in the following sections, based on the training data which has already been discussed.

Positive Results

Compound Type	General Accuracy	Certainty Range
contained in case base	high	0.3 - 0.8
typical unknown compound	medium to high	0.15 - 0.7
atypical unknown compound	low to medium	0.1 - 0.3

- Accuracy and certainty satisfactory for many compounds
- Adaptation seen for slightly incorrect input
- Degradation in quality occurs statistically, not randomly
- Beneficial interaction between CBR memory and adapters

The most pleasant surprise provided by the OrCA system was that, in many cases, it functioned correctly. When presented with a compound which was present in its case base, the system was frequently able to produce correct results with 30 to 80 percent certainty. Only a small number of incorrect results appeared in these cases, and these generally had a certainty of 15 percent or lower. This accuracy extended, for the most part, to poorly specified queries as well. When queried on a compound contained in the case base, yet given only functionality and number of carbons, the system was able to provide results which were either correct or differed only by one value, yet with certainties on the order of 15 to 20 percent. Interestingly, the system was also able to adapt to incorrect data. When given a query such that minor fields such as color or odor differed from any known compound, the system was still able to retrieve values from the most similar known compound with reasonably high certainties.

As was expected, results grew less solid as testing moved away from compounds clearly included in the case base. When presented with queries for compounds similar to those in the training data - compounds which could be considered typical - the system was often able to produce correct answers, but not consistently. Certainties varied widely, from 15 to 70 percent, although higher certainties were seen to correspond with correct answers. Certain qualities were analyzed with more consistency and correctness than others. The correctness of functionality determination, for example, degraded much more quickly than the correctness of odor or color determination, both when given otherwise complete data. Finally, when

provided with very little initial data in these cases, OrCA was able to provide results approximately as accurate as those for poorly-specified queries on known compounds. Returned parameters were often correct or differed only by one value, yet certainties fell to a range of 15 to 20 percent.

Finally, when asked about atypical compounds or substances very different from those appearing in the case base, the system degraded even further. However, it did so in a useful manner; answers were never random or baseless, but reflected the statistical distribution of the training data. For example, when queried regarding a compound differing significantly from all those appearing in the case base, results were effectively a statistical description of the training data itself. Certainties corresponded to how often a particular value appeared in the training data rather than how likely that value was for the system at hand - which is certainly a logical assumption for a compound for which the system has no basis of judgement. If asked something for which it has no specific experience, why not assume that the experience it does have is accurate and answer accordingly? Although this does rule out OrCA for some of its more exotic possible applications, it ensures that the system will degrade gracefully when used under anything resembling normal conditions. Certainties ranged from 10 to 30 percent, generally reflecting exactly what the statistical probability of a particular value was based on the training data.

Aside from query-specific results, interesting patterns in the system also became apparent through testing. Most interestingly, the interaction between the main APN memory and the neural network adapters became rapidly apparent. As the amount or relevance of the data provided with a query decreased, the use of the adapters, as well as the degree to which they disagreed with the APN, increased. Less certain or less familiar data produced more instances of adaptation, more differences between the APN output and the adapter output, and frequently a much higher certainty from the adapters than from the APN. This in and of itself is one of the most promising results of the OrCA experimentation, since it demonstrates a remarkable and extremely useful ability for completely disparate artificial intelligence techniques to complement and reinforce each other.

Negative Results

- Less generalization than hoped for
- Undesirable favoritism in quality analysis
- Tendency to choose multiple answers in favor of a single certain one

The most immediate and obvious disappointment of the OrCA system is its less-than-ideal ability to generalize. Although it performed adequately with compounds with which it was not explicitly familiar, both certainties and restriction of answers could both be improved upon. For unfamiliar compounds, certainties of 50 percent or higher could be expected, as well as providing one likely answer rather than two or three. This is possibly the cause of inadequate training data, but likely the structure and functionality of the APN is at fault. Although the adapters, to some extent, aid the APN in generalization, the probabilistic network algorithms utilized in OrCA are far from the best at generalization on their own.

Another deficiency lies in the system's inability to analyze certain qualities as well as others. For reasons which are not fully understood, qualities at the roots of the causal graph seen in Figure 2 (number of carbons and functional groups) are frequently provided with lower certainty and less accuracy than qualities at the bottom. A number of causes for the phenomenon are possible. Most simply, the two qualities in question are those which would be more difficult to analyze for any chemist who was only given physical data. For functionality in particular, the physical data which is known to most directly correlate with this is solubility - precisely the data which was in no case provided to OrCA. Hopefully further exploration and refinement of the system to include this data will correct these problems. However, other causes are possible as well. The two could simply be less correlated to the physical data than is expected. There could conceivably be deficiencies in the Bayesian network algorithms as implemented in OrCA which prevent causal roots from being as accurately analyzed as other nodes, since they are indeed treated differently. Least likely, yet also least helpful, is the possibility that analysis of these qualities in terms of the problem domain is simply not a situation for which the current OrCA architecture is well-suited. This final possibility is quite unlikely,

though, in light of the fact that the system has been seen to function and adapt well in other situations and with all other query arrangements.

A minor disappointment in the system's performance has been its frequent inability to choose one strong answer in favor of one primary answer accompanied by a few weak answers. For example, typical results for functionality might include acid at 50 percent certainty, yet also none and alcohol at 10 and 20 percent, respectively. This tendency towards what a human might term indecisiveness likely stems from a design propensity for allowing the human chemist controlling the program to make the final decision. Many tunable parameters in the system have been adjusted so that if an answer is at all suspected, it will be provided in the final output. While this may be useful in some cases where a correct answer might otherwise be missed, it also allows OrCA to provide a combination of less certain answers rather than a single, strong answer in cases in which it "knows" it is correct.

Further Work

There are undeniably many routes which could be pursued for further work with the OrCA system. Most obviously, the highest priority lies in completing compilation of a full set of training data, including solubilities, and in prepending the initial expert system to the implemented architecture. This would allow the system to be tested under the full conditions for which it was designed and specified - and would hopefully increase the quality of results. Other obvious improvements include expanding the size of the case base and examining more test cases; both of these would immediately provide a more accurate and reliable characterization of the system.

A route which could easily provide interesting and easily obtainable information about the system lies in modifying internal parameters. Embedded within almost every portion of the system, from neural networks to high level CBR algorithms, are adjustable parameters which control a variety of processes within the system. These could easily be modified in order to study the response of the system; for testing purposes they were almost uniformly set to values which the author considered reasonable, yet little investigation has been performed along these avenues. There exist parameters which control the neural network response, APN learning rate and associativity, and CBR retrieval and adaption characteristics, among others - all of these could be modified to improve system behavior.

Embedded within the runtime environment of OrCA itself lies a combination of numerical values which provide a further course of research regardless of system performance. In the process of training the system, and the APN in particular, the training data and the relationships between qualities become abstracted into a set of numerical patterns and probabilities. Investigation of the post-training configuration of the APN and, to some extent, the neural networks provides a means to investigate the specific relationships between various chemical qualities. Some such relationships may be determined mathematically - yet heuristic evaluation of all such relationships in the manner which OrCA makes possible might allow simple, graspable relationships between chemical parameters to be isolated and described. Additionally, this analysis would provide an interesting project in artificial intelligence research as well to determine what suspected relationships the system was capable of perceiving and which it failed to discover.

References

(1) Binder, John et al "Adaptive Probabilistic Networks with Hidden Variables" 1996

(2) Heckerman, David et al "Learning Bayesian Networks: The Combination of Knowledge and Statistical Data" 1994

(2) Kolodner, Janet "Case-Based Reasoning" 1993

(3) Russel, Stuart and Norvig, Peter
Artificial Intelligence: A Modern Approach
Prentice-Hall, New Jersey 1995

(4) Thiesson, Bo et al "Learning Mixtures of Bayesian Networks" 1993