U.S. DEPARTMENT OF ENERGY

PNNL-24142

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Measuring Semantic Dispersion in Zymurgy

Cliff A. Joslyn
David J. Haglin
Emilie A. Hogan
Alejandro Heredia-Langner
Patrick R. Paulson
Amanda M. White

May 2011

Pacific Northwest
NATIONAL LABORATORY

Proudly Operated by Battelle *Since 1965*

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

PNNL-24142

Measuring Semantic Dispersion in Zymurgy

Cliff A. Joslyn
David J. Haglin
Emilie A. Hogan
Alejandro Heredia-Langner
Patrick R. Paulson
Amanda M. White

May 2011

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99352

Measuring Semantic Dispersion in Zymurgy

Cliff A. Joslyn, David J. Haglin, Emilie Hogan, Alejandro Heredia-Langner,
Patrick R. Paulson, Amanda M. White

May, 2011

Contents

1 Executive Summary

We seek to develop a robust measure of the semantic extent of sets of Omega ontology nodes derived from Zymurgy analysis of input documents. The Omega 2 knowledge system [9] has a complex semantic architecture involving multiple ontological graphs structured as both hierarchical and non-hierarchical link types. Previously we have developed analytical techniques against the hierarchical portion of the Omega ontology [6], and a dispersion measure based on order metrics [5].

Here we describe a non-linear, decision tree classifier which uses the order metric statistics to distinguish Zymurgy output of real documents from artificial documents and random Omega node

sets [3]. Classification trees were applied separately to two sets of 76 Omega node sets, including Zymyurgy output from real and artificial documents, and entirely random nodes.

In many cases, average classification rates in cross-validation tests correspond to more than 70% of the documents being correctly identified. In general, average misclassification rates when using cross-validation remain under 40%. 10-fold crossvalidation was used to see if the same set of classification variables could be found to be best in separate tests. Even though the very best classification results for each set of 76 documents is not obtained using the exact same group of classification variables, there are many options that produce good results.

We discuss the relative values of different metrics, including experimental order theoretical metrics and more standard minimum path-based methods. While minpath is not recommended for many reasons, the extra computational burden of the upper order metric over the lower order metric is justified on the grounds of greater accuracy.

We conclude with a preliminary discussion about scoring methodology.

2 Background

As described previously [6], our work uses the Omega 2 ontology [9]. The Zymurgy system incorporates Omega 2 to create subsets $Q \subseteq P$ of Omega 2 nodes, together with their Zymurgy scores, which characterize a given input document. Our driving question has been, given such a ranked set $Q \subseteq P$, what is its dispersion, or coverage, within Omega?

We address this in the following technical approach:

- Based on our prior analysis [6], we address Omega 2 primarily in its context as a class hierarchy. We represent this as an ordered set $\mathcal{P} = \langle P, \leq \rangle$ of 121K nodes $a \in P$ arranged over 29 gross levels.[1]
- We establish multiple **metrics** d: $P^2 \to \mathbb{R}^+$ on $\mathcal{P}$ to measure distances $d(a, b) \in \mathbb{R}^+$ between concepts $a, b \in P$. These are justified and discussed in detail elsewhere [5, 8], but include the following:

  Upper Order Distance: Measures the amount of the ontology above the two nodes a, b, relative to the amount of the ontology above each one separately:

  $$d^*(a, b) = |\uparrow a| + |\uparrow b| - 2|\uparrow a \cap \uparrow b|,$$

  where for $a \in P, \uparrow a := \{b \geq a\}$ is its "up-set".

  Lower Order Distance: Measures the amount of the ontology below the two nodes a, b, relative to the amount of the ontology below each one separately:

  $$d_*(a, b) = |\downarrow a| + |\downarrow b| - 2|\downarrow a \cap \downarrow b|,$$

  where for $a \in P, \downarrow a := \{b \leq a\}$ is its "down-set".

  Undirected Minimum Path: $d_p(a, b)$ is the minimum path length between a and b in $\mathcal{P}$, where we take $\mathcal{P}$ as an *undirected* graph.

[1] We have noted elsewhere [4] that partial orders naturally admit to descriptions in terms of *interval-valued* levels, as we reported for Omega [6].

- For a metric d and any subset of nodes $Q \subseteq P$, we can define its **dispersion** $D_d(Q)$ as the measure of the average distance d amongst all the nodes in the set. Formally we have

$$D_d(Q) := \sum_{a,b \in Q} d(a,b), \quad \hat{D}_d(Q) := \frac{D_d(Q)}{m^2 - m} \tag{1}$$

  in both normalized and unnormalized form, respectively, where $m = |Q|$ is the total number of nodes in the set. The unnormalized form $D_d(Q)$ ranges in $[0, m^2 - m]$, while the normalized form $\hat{D}_d(Q)$ is relative to the size of the input set, and thus ranges in $[0, 1]$.

- Given a metric d, define the **segment** [1] $[[a,b]]_d$ as the set of all nodes which are "between" them in the metric sense:

$$[[a,b]]_d := \{c \in P : \ d(a,b) = d(a,c) + d(c,b)\}.$$

  Note that only when the minimum path length metric d_p is used does this translate into what we mean by "between" in the ordinary sense of looking at a diagram of the ontology.

- **Convexity** is the idea that any nodes between other nodes in a collection $Q \subseteq P$ are also in that collection, so that a subset of nodes $Q \subseteq P$ is convex if $\forall a, b \in Q, [[a,b]] \subseteq Q$. For a metric d, we can then define $C_d(Q)$, the **convex hull** of Q, by letting $C_d(Q)$ be the function

$$K_d(Q) = \bigcup_{a,b \in Q} [[a,b]]_d$$

  when iterated to convergence [5].

So finally, assume a given input set of Omega nodes Q and a metric d. We will deal with its normalized dispersion

$$\hat{D}_d(Q),$$

and call that its **normalized hollow dispersion**. This is in contrast to when we consider the convex hull $C_d(Q)$ of the node set, in which case we will refer to the dispersion

$$\hat{D}_d(C_d(Q))$$

as its **normalized solid dispersion**.

Concepts around these metrics can be difficult to understand at first. See further discussion and examples below in Sec. 6.

3 Experimental Setup

Our experimental design is shown in Fig. 1, and now described here.

- A set of 135 **input documents** were identified by the sponsor, from a set of documents collected by the Linguistic Data Consortium.[2] Each file pertains to exactly one topic; while we keep track of topcis, this notion is not used in our analysis of concept dispersion.

[2] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2005T16

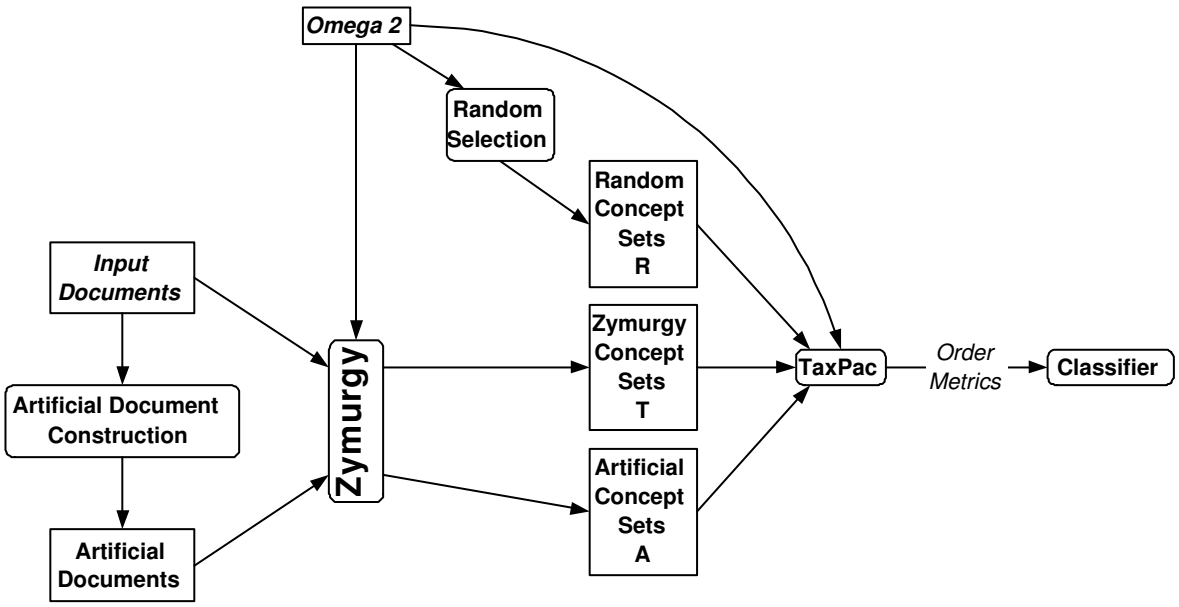

Figure 1: Experimental architecture.

- These are then run through **Zymurgy** to produce scored **Zymurgy concept sets** D.
- Additionally, we have an **artificial document construction** mechanism which creates **artificial documents**. To create the artificial documents, the average length of all the files in the `10_TOPCS` list was calculated along with the standard deviation in paragraph lengths. Each paragraph of these files was added to a collection of paragraphs, and then 50 synthetic documents were generated and assigned random topics, with size and number of paragraphs drawn from the same distribution as the input documents. These artificial documents are also run through Zymurgy to produce **artificial concept sets** A.
- We also have a **random selection** process to produce completely **random concept sets** R. Each concept set has 10 concepts selected randomly from concepts occurring in documents producing at least 10 concepts. The i'th concept in a random set is then the i'th concept of some randomly selected source concept list, using a uniform distribution.
- All three kinds of concept sets are fed into our Taxonomy Package (TaxPac) [7], where statistics shown in Table 1 are calculated. One additional variable is used, namely the number of concepts used in the set, which is m above in (1). Zymurgy typically return many concepts per document, weighted by significance, so that m ranges in $\{10, 11, \ldots, 582\}$ for one set of documents. For many reasons, including tractability and significance, we selected for $m \in \{2, 3, \ldots, 10\}$.
- Finally, we build statistical classifiers on the output metric variables to distinguish real concept sets D from random R and artificial A.

We performed two runs of 76 selected real input documents, each paired with 10 artificial documents and 15 random node sets, yielding 25 total non-real node sets, or 32.9%, per set. Figs. 2 and 3 show scatter plots of the decision variables, with points color-coded according to class membership for

Symbol	Statistic	Formula
`nlch`	Size of lower convex hull	$\|C_{d_*}(Q)\|$
`lsdn`	Lower solid dispersion, normalized	$\hat{D}_{d_*}(C_{d_*}(Q))$
`lhdn`	Lower hollow dispersion, normalized	$\hat{D}_{d_*}(Q)$
`nuch`	Size of upper convex hull	$\|C_{d^*}(Q)\|$
`usdn`	Upper solid dispersion, normalized	$\hat{D}_{d^*}(C_{d^*}(Q))$
`uhdn`	Upper hollow dispersion, normalized	$\hat{D}_{d^*}(Q)$
`nc`	Number of concepts	m

Table 1: Order statistics calculated for each concept set Q.

the two sets. It is clear that some of these variables are highly correlated, indicating that not all of them provide independent information. Also, it is clear that some of the variables included in the Figures are highly clustered around a single value (`usdn`, for example) with relatively few outlying observations present. In general, it does not appear that the distribution of points between the two sets of 76 documents differ in significant ways with the exception of a few outliers.

4 Decision Tree Models

Based on these results, we turned to non-linear classification systems, in particular a **decision tree** or **classification tree** [10] to predict document identity for either real documents D, or the combined category that includes generated (or synthetic) documents and randomly generated ones $N = A \cup R$. Classification trees are predictive models that seek to determine the value or identity of a target item using a series of simple binary decisions. Classification trees are made up of branching nodes, where a decision is made, and leaves, which indicate the value assigned to the target. In this analysis, a branching node splits based on the value of one of the descriptor variables. Classification is carried out by following the branches of the tree in an 'if - then' pattern. Fig. 4 shown an example using the single decision variable `nuch` for the case where nc = 2.

Fig. 4 indicates that the first split is carried out by asking if the value of `nuch` in a target document is less than 17, if this is true, then the document is classified as real D. Otherwise, the rightmost node in Fig. 4 is reached where the value of `nuch` for the target document is checked again. At this node, if `nuch` is greater than 988 the document is classified as real; otherwise the next node is reached. The process described can be followed for any other document until a leaf (and a class) is reached.

Classification trees can grow to unruly (and not practically informative) sizes and are constructed using a greedy algorithm with no guarantee of global optimality. Independent starts and cross-validations were used to ensure robustness and help avoid locally optimal solutions.

Because it is not known beforehand which variables will result in the most accurate classification of documents for the range of nc values from 2 to 10, all possible combinations of the six descriptor variables of interest were used for every value of nc available. This means that a classification tree is created using each one of the six descriptor variables at a time, then every combination of two descriptor variables at a time and so on until all six descriptor variables are used in a model simultaneously. The process of using multiple combinations of descriptor variables ensures that the problem space is explored thoroughly and helps avoid convergence to a suboptimal solution.

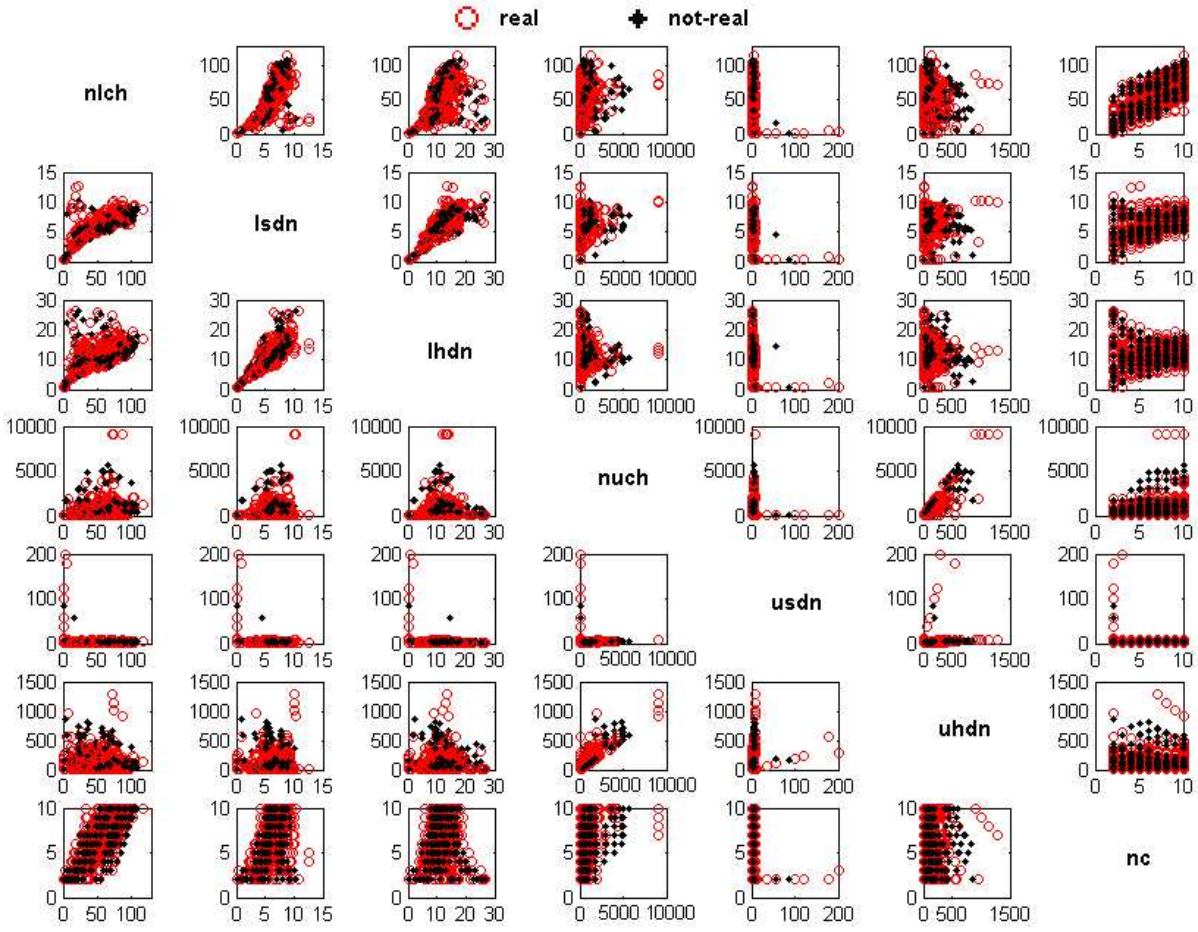

Figure 2: Scatter plot from first run of 76 documents.

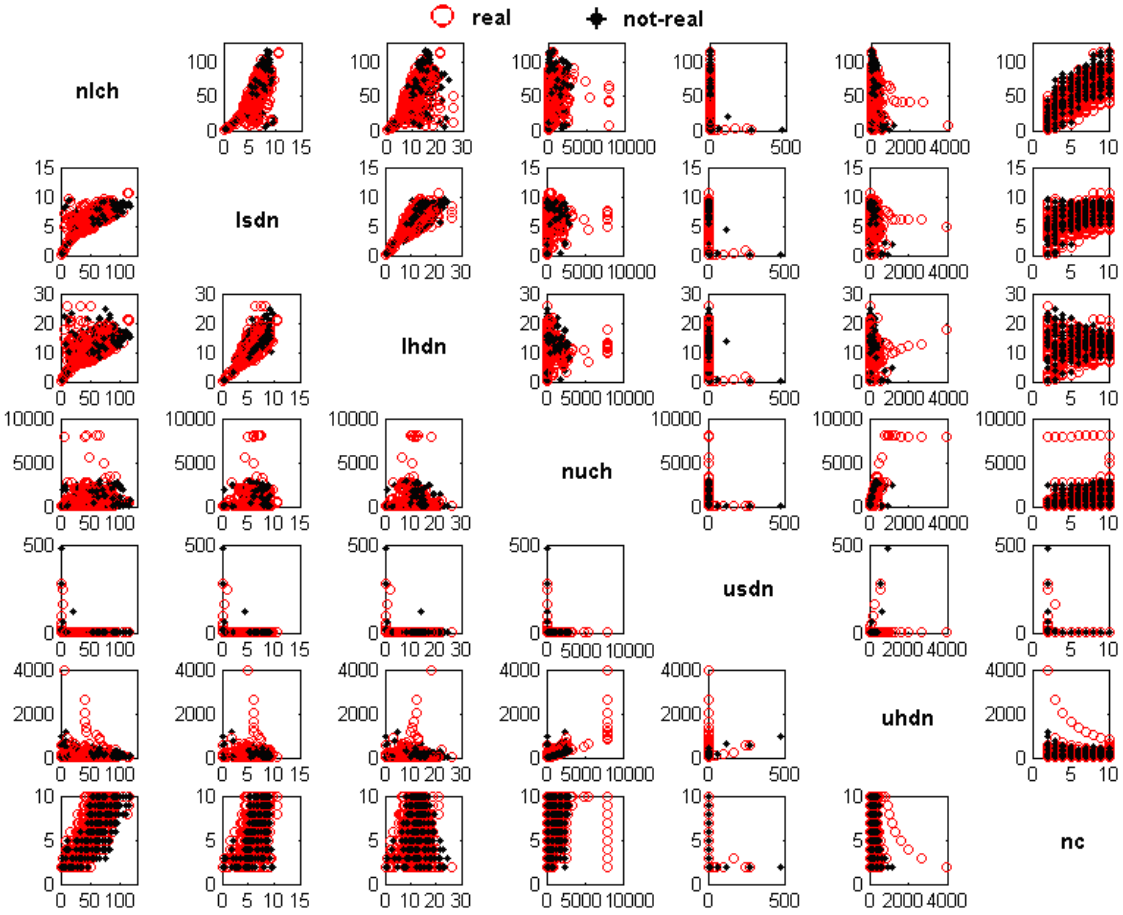

Figure 3: Scatter plot from second run of 76 documents.

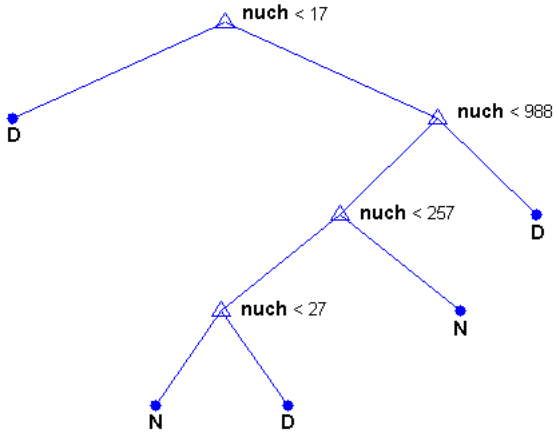

Figure 4: Classification tree for the Example where only the `nuch` variable was employed and nc = 2.

Performance evaluation of the classification trees was carried out using 100 independently started 10-fold cross-validation runs for each of the 63 models under consideration. A 10-fold cross-validation, divides the dataset into 10 mutually exclusive subsets with roughly the same size and same composition. Then, each subset is used to test the classification tree obtained using the rest of the data to build the model. The number of incorrectly classified documents in each test set and every cross-validation can be used to obtain an error rate. In an effort to estimate actual average performance for these models, mean error rates over the 10-fold partitions and the 100 independent starts are reported. All classifications trees shown in this document were fitted using Matlab (2007).

Error rates (number of misclassified documents divided by the total number considered) produced by classification trees can depend on the proportion of documents of each type that are available (a tree will seek to correctly classify most of the documents of a category that dominates a given training set). This leads to the question of how to correctly interpret results from a classification tree.

To investigate this question, it is of interest to determine if the classification models created using the six descriptor variables and decision trees are better than a classification method that assigns document category simply as a function of the proportion of each class of document in the population. This simple, or naive, classification approach based on population proportions can be thought of as a lower bound on performance, since it requires practically no effort to implement and it does not make use of any other characteristic of the data in order to make a classification.

To measure the effect that the classification trees have compared to the naive classifier, we use the formulation of **lift rate**

$$L = 100\left(\frac{E-M}{E}\right),$$

where E is the expected error rate for the naive classifier and M is the measured error rate obtained with a given classification tree. L is maximized at $L = 100$ when $M = 0$, indicating a perfect classifier. $L = 0$ indicates no lift, that is, the classifier did no better than random. Finally, L is minimized at $L = 100(1 - \frac{1}{E}) < 0$ when $M = 1$. In our real runs, we determined that the expected error rate overall is $E = 0.5$, while for the class of real documents D we have $E = 0.329$, and for the class of non-real node sets N we have $E = 0.671$.

5 Classification Results

Complete classification results are very lengthy, as many models were run:

- For each of the two document sets;
- For each of the sets of node sets D (real node sets), N (non-real node sets), and $D \cup N$ (all node sets).
- For each number of variables $m \in \{2, 3, \ldots, 10\}$;
- For each model, namely each non-empty subset of the six decision variables, or $2^6 - 1 = 63$ models.

The lift rates for all of the resulting $2 \times 3 \times 9 \times 63 = 3,402$ models are shown in Fig. 5, and are documented completely elsewhere [3]. However, we can summarize the results as follows.

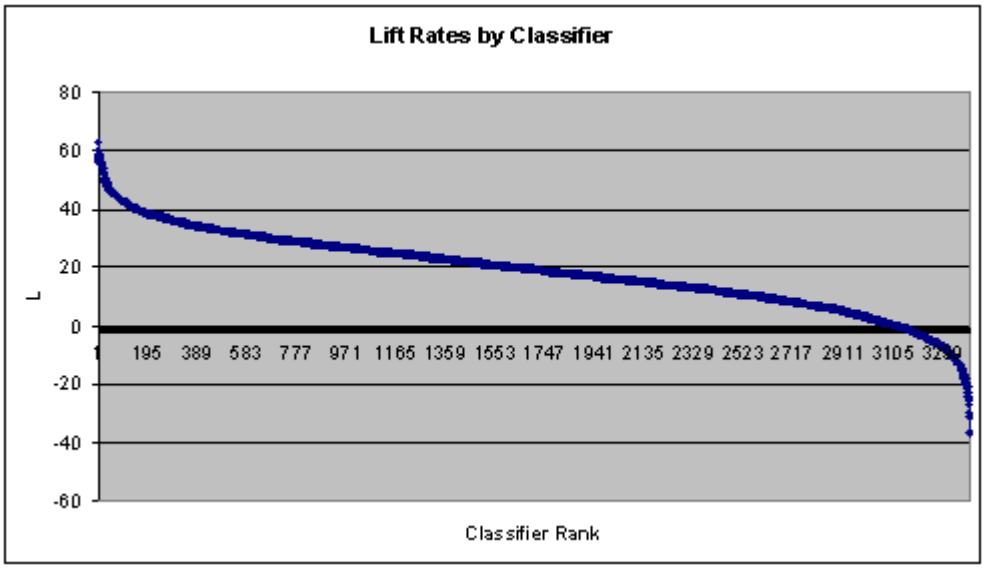

Figure 5: Lift rates for all classifiers.

m	2	3	4	5	6	7	8	9	10	Overall
Min	-20.05	-15.10	-9.38	-20.13	-27.14	-23.03	-29.77	-22.85	-36.98	-36.98
Mean	20.64	21.10	17.54	15.28	24.76	18.65	21.56	13.48	17.21	18.91
Max	56.96	56.67	47.32	47.37	62.92	42.34	45.71	41.58	48.62	62.92
Std	14.35	13.62	9.75	12.55	16.12	10.69	12.04	12.86	13.56	13.37

Table 2: Lift rates for $m \in \{2, 3, \ldots, 10\}$ and overall.

For only 280 of the 3,402 classifiers, or 8.2%, are the lift rates negative. This indicates that the classification trees provide superior average performance to the nave classifier and validates the usefulness of classification trees for this particular problem.

Minimum, mean, maximum, and standard deviations of lift rates for $m \in \{2, 3, \ldots, 10\}$ and overall are shown in Table 2, and then graphically in Fig. 6.

While mean lift rate is 18.8%, L is above 20% for most models, and there is a dramatic group for the 28 classifiers with $L \geq 50$. Table 3 shows the distribution of these classifiers by m and the number of variables in their models. $m = 2, 3$, especially for only 1 or 2 variables, are easily discounted. This leaves a strong result to favor $m = 6$, including the counts for the number of times the decision variables appear.

The models for those 14 classifiers with $L \geq 50$ and $m = 6$ are shown in Table 4. Fig. 7 then shows these models using a concept lattice representation [2] of the logical dependencies amongst the variables. The clear conclusion is that the key variables are `usdn` and `nlch`, that is, the normalized upper solid dispersion $\hat{D}_{d^*}(C_{d^*}(Q))$ and the size of the lower convex hull $|C_{d_*}(Q)|$.

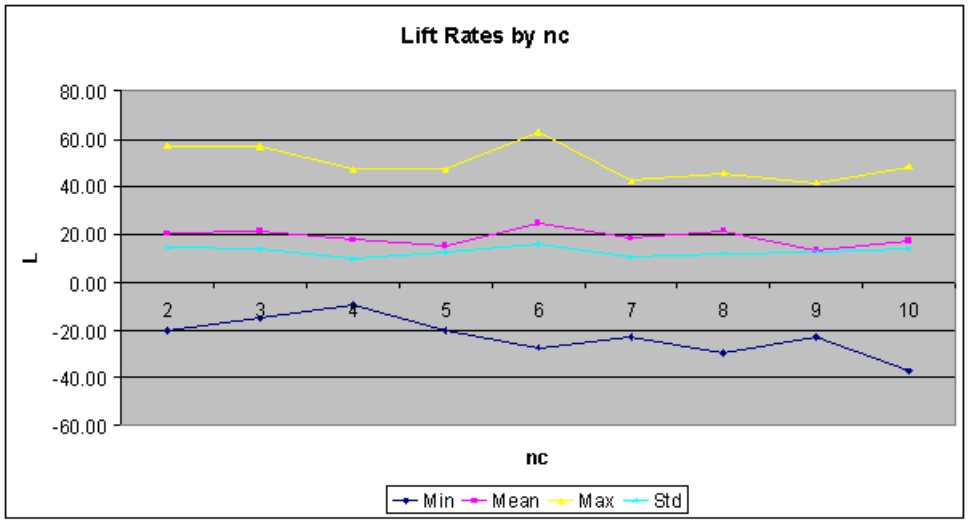

Figure 6: Min, mean, max, and standard deviation of lift rates by m.

m	# Variables	Count
2	1	4
2	2	1
2	3	1
3	3	1
3	4	3
3	5	3
3	6	1
6	3	1
6	4	5
6	5	6
6	6	2

Table 3: Count of the number of classifiers by number of concepts and decision variables for the top 28 classifiers with $L \geq 50$.

L	nlch	lsdn	nuch	usdn	lhdn	uhdn
62.92	√	√	√	√		
60.00	√	√	√	√	√	
59.11	√	√	√	√		
58.27	√	√		√		√
57.92	√	√	√	√		√
56.16	√	√		√		√
55.89	√	√	√	√		√
55.83	√	√		√	√	√
55.29	√	√	√	√	√	√
55.26	√	√	√	√	√	
52.82	√	√	√	√	√	√
52.76	√	√		√	√	√
52.13	√			√		√
50.05	√		√	√		√
TOTAL	14	12	9	14	6	10

Table 4: Models and lift rates for all classifiers with $L \geq 50, m > 3$.

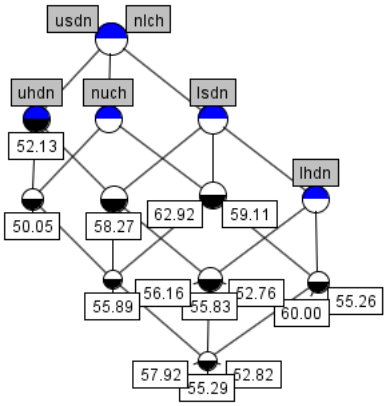

Figure 7: Concept lattice representation of the dependencies amongst the variables in the top 28 classifiers

	$d^*(\cdot,\cdot)$	$d_*(\cdot,\cdot)$	$d_p(\cdot,\cdot)$
Mammal, Bird	2	6	2
Dog, Cat	2	2	2
Dog, Eagle	4	2	4
Dog, Platypus	3	2	2

Table 5: Metrics for select pairs.

6 Metrics: Discussion and Pragmatics

As remarked above, our methods are dependent on the various metrics, including the upper d^*, lower d_*, and minimum path length d_p. This can affect accuracy, interpretability, and (significantly) computational performance. Additionally, properly understanding and interpreting especially the new metrics can be a challenge.

While the undirected minimum path length d_p is used commonly in ontology applications, sometimes in a weighted form, it is generally inadequate for a number of reasons. Additionally, the nature of real ontologies as upper-rooted, and very widely downward branching, means that the structure and distribution of the upper and lower distances d^*, d_* can vary dramatically.

Here we discuss these issues in more depth, including examples and illustrations. We will conclude that as a general matter, for both accuracy, interpretability, and computational performance reasons, we need to use combinations of upper and lower distance d^*, d_*, while excluding minpath distance d_p.

6.1 Examples

First we illustrate and consider the nature of each of these metrics. Fig. 8 shows a simple ontology fragment, including some slight multiple inheritance. Table 5 shows the distances between these nodes according to all three metrics.

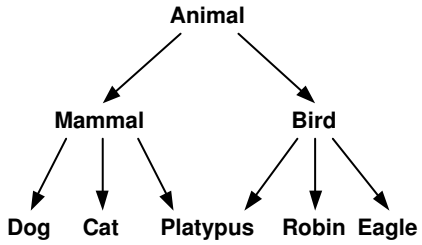

Figure 8: Ontology fragment for illustration.

As an illustrartion within Omega, Fig. 9 shows the top 5 Zymurgy scoring Omega concepts for document 15. The nodes are adorned with their interval ranks, indicating their position midway in the structure. Fig. 10 shows its upper convex hull $C_{d^*}(Q)$, and Fig. 11 shows its lower convex hull $C_{d_*}(Q)$.

Finally, Fig. 12 shows the segments $[[B,E]]_d$, $[[B,G]]_d$, and $[[E,G]]_d$ for the Boolean 3-cube, using

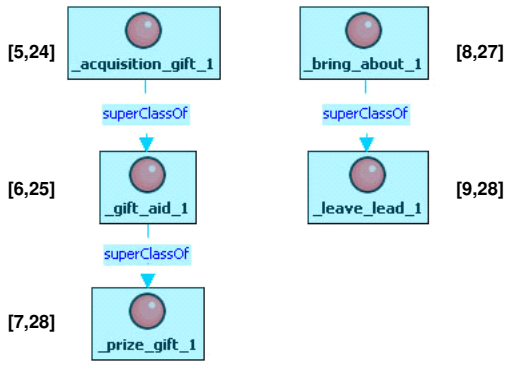

Figure 9: Top 5 concepts for document 15.

the upper distance d^* on the left, and the minpath distance d_p on the right. In each case the convex hull $C_d(\{B, E, G\})$ is the union of these segments.

6.2 Order Metrics

First, the upper and lower distances measure significantly different properties. While the upper distance d^* cannot distinguish the pairs $\langle$Mammal, Bird$\rangle$ and $\langle$Dog, Cat$\rangle$ in Fig. 8, the lower distance d_* indicates that $\langle$Mammal, Bird$\rangle$ are further apart than $\langle$Dog, Cat$\rangle$, which is semantically appropriate. However, the lower distance d_* is also not sufficient by itself, since it cannot distinguish leaves. In particular, it says that $\langle$Dog, Cat$\rangle$ and $\langle$Dog, Eagle$\rangle$ are the same distance, which is clearly inadequate.

Note the dramatic difference between the upper and lower hulls shown in Figs. 10–11. This is due to the vastly downward-branching nature of Omega, and indeed of most commonly used ontologies. Consequently the upper segments and hulls are orders of magnitude larger than the lowers, as shown in the `nuch` $\times$ `nlch` scatter plots in Figs. 2 and 3.

Since the upper and lower metrics are qualitatively different, but neither is sufficient alone, as a general matter, it may be preferable to use them in combination, and we have begun to examine this. On the other hand, if we were able to rely only on the lowers, this would be computational preferable. See further discussion below.

6.3 Minimum Undirected Path Length Metric

Consider now the minimum undirected path length (minpath) distance d_p. For the pairs $\langle$Mammal, Bird$\rangle$, $\langle$Dog, Cat$\rangle$, and $\langle$Dog, Eagle$\rangle$, the upper and minpath distances are equal. This is always the case when $\mathcal{P}$ is a tree, and while in this case $\mathcal{P}$ is not a tree, still the multiple inheritance is not involved for these particular pairs. But note in particular that this is *not* the case for the pair $\langle$Dog, Platypus$\rangle$, where the multiple inheritance occurs.

d_p is inadequate in that like d^* it also cannot distinguish the pairs $\langle$Mammal, Bird$\rangle$ and $\langle$Dog, Cat$\rangle$. In fact, it is generally inadequate in that it simply has fewer distinguishing values. When our hierarchy $\mathcal{P}$ is bounded, then either d_* or d^* can distinguish $N - 1$ values, for $N = |P|$ is the total number of nodes. But d_p lives in a much smaller range, approaching N only when $\mathcal{P}$ approaches a

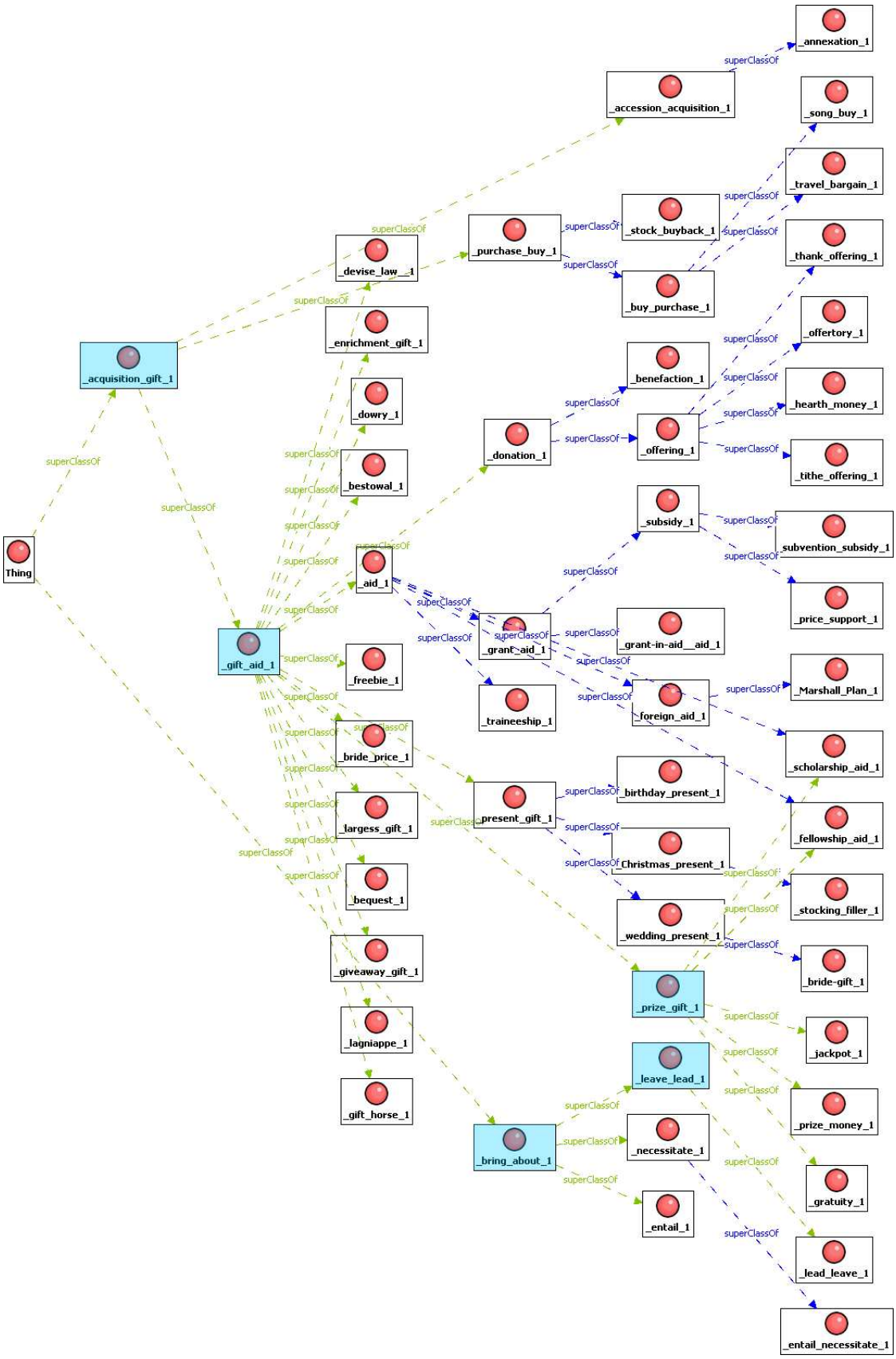

Figure 10: Upper convex hull $C_{d^*}(Q)$.

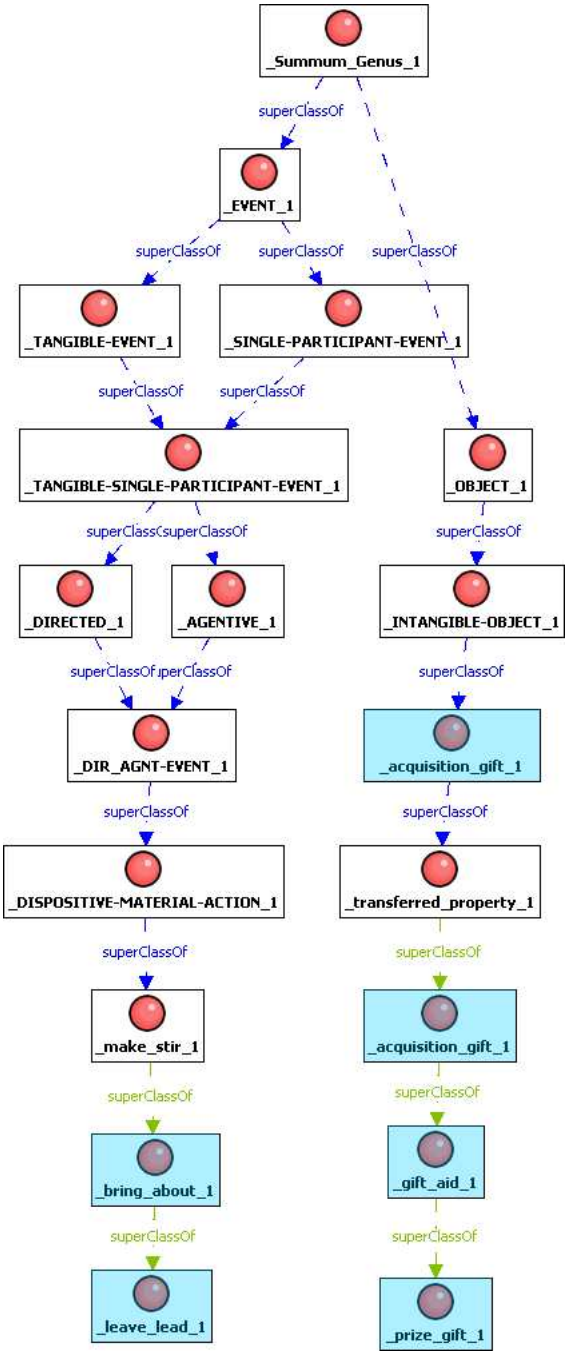

Figure 11: Lower convex hull $C_{d_*}(Q)$.

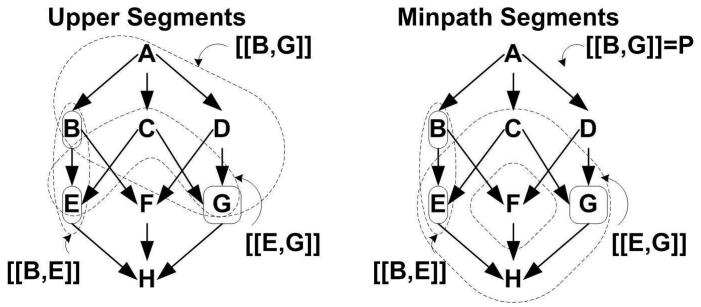

Figure 12: (Left) Segments in the Boolean 3-cube for upper distance d^*. (Right) Segments in the Boolean 3-cube for minpath distance d_p.

single linearly-ordered chain.

This is shown in the Table 5. It is also apparent in Fig. 12, where in each case the convex hull $C_d(\{B, E, G\})$ is the union of these segments. Note how the upper segments naturally focus on the upward-branching properties, and in particular the segment $[[B, G]]_{d^*} = \{A, B, C, D, G\}$ identifies a non-trivial set above the nodes in question. But the minpath segment $[[B, G]]_{d_p}$ is the *entire* structure P, since these are opposite pairs in the lattice. Again, minpath d_p is simply insufficiently distinguishing.

6.4 Computational Considerations

Computational performance varies dramatically for d^*, d_*, and d_p. First, we conjecture that both upper and lower segments are included within minpath segments, or formally

$$[[a, b]]_{d^*} \subseteq [[a, b]]_{d_p}, \quad [[a, b]]_{d_*} \subseteq [[a, b]]_{d_p}.$$

The question is *how much* bigger, and we have evidence that it is *much* bigger, so that

$$|C_{d^*}(a, b)| \ll |C_{d_p}(a, b)|, \quad |C_{d_*}(a, b)| \ll |C_{d_p}(a, b)|.$$

In fact, we have calculated minpath hulls sizes and computational times which are three orders of magniture larger in a medium-sized ($\sim 4K$) ontology. As a general matter, we believe that minpath hulls approach the size of the whole ontology. In particular, minpath hulls are not currently calculable on Omega.

Fig. 13 shows computational times for upper and lower convex hulls on our research platform, a 64 bit Linux server with 48 GB RAM, 16 TB disk, and two quad core 2.8 GHz processors. On a standard desktop, we observed run times about ten times longer.

6.5 Accuracy Considerations

So for computational reasons alone, we would wish to use only the lower order metric distance d_{l_*}. The question then becomes how this would affect overall classification accuracy.

Table 6 shows minimum, mean, maximum, and standard deviations of lift rates segregated by class of models. The far left column shows results for all models which include both upper and lower

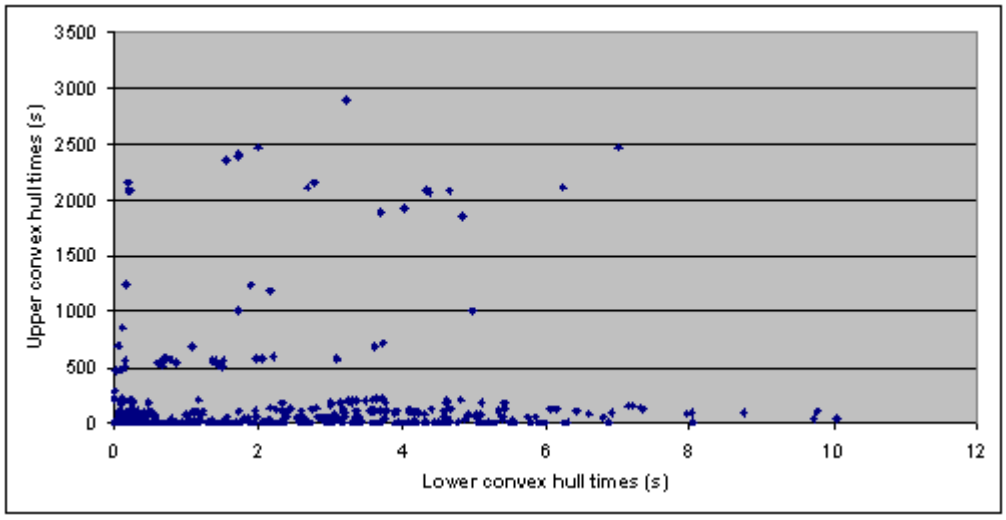

Figure 13: Lower vs. upper convex hull computational times.

	Uppers and lowers	All	Just uppers	Just lowers
Min	-24.95	-36.98	-26.92	-36.98
Mean	19.59	18.35	14.81	13.23
Max	62.92	62.92	44.68	48.62
Std	12.26	13.13	14.48	15.51

Table 6: Lift rates for: (left) all models which include both upper and lower variables; (center left) all models; (center right) all models including only upper variables; (right) all models including only lower variables.

variables (i.e. including at least one of `nuch, usdn` and `uhdn` and also one of `nlch, lsdn` and `lhdn`). Next to the right are all models all all forms. The next two columns show all models including only upper and lower variables. These are then again shown graphically in Fig. 14.

The clear conclusion is that for accuracy reasons, upper and lower distances should be used in combination. This is also born out by the arguments in Sec. 6.2 above, and the results shown in Table 4, which greatly favored the variable pair `usdn` and `nlch`.

7 Towards Scoring

We conclude this draft with a very drafty discussion of the desire to move from a binary classifier to actually producing a numerical score indicating to what extent a given node set can be clearly seen to be in the real set D, the non-real set N, or perhaps less confidently in between.

7.1 Definitions

We assume that a decision tree has been formed and we can identify the l leaves of the tree uniquely as $\{t_1, t_2, \ldots, t_l\}$. We have n instances $\mathcal{I} = \{I_1, \ldots, I_n\}$, each of which is known to be in one of k

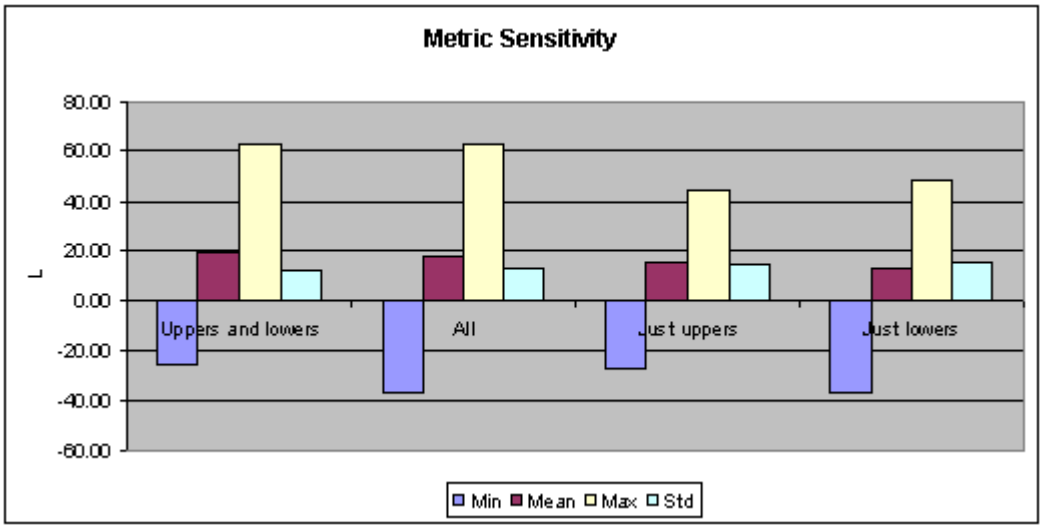

Figure 14: Lift rates for models by upper vs. lower.

classes: $\{c_1, \ldots, c_k\}$. Let $C(I_i) = c_j$ denote that instance I_i is known to be in class c_j.

Given one instance $I \in \mathcal{I}$, exactly one of the tree leaves will be "triggered" by satisfying conditions described by the non-leaf nodes of the decision tree. Call this leaf node $L(I)$.

Given a particular leaf node t_i, let $I_{t_i} = \{I \mid L(I) = t_i\}$. Let $|I_{t_i}|$ denote the number of instances that trigger leaf node t_i. Note that $\sum_{1 \leq i \leq l} |I_{t_i}| = n$.

Given a particular leaf node t_i, let $C^L(t_i) = c_j$ denote that tree leaf t_i asserts all instances in I_{t_i} are in class c_j. Note that some of the I_{t_i} instances may be "correct" in that they are known to be in class $C^T(t_i)$. Let $CI_{t_i} = \{I \mid I \in I_{t_i} \text{ and } C(I) = C^L(t_i)\}$ be the set of correct instances triggering leaf node t_i.

An obvious and simple method of assigning a confidence score for decisions by the tree is to say that for a given instance $I \in \mathcal{I}$, $conf(I) = \frac{|C^L(L(I))|}{|I_{L(I)}|}$. That is, the confidence is the proportion of all of the instances triggering a specific leaf that are correct relative to all of them that trigger the leaf. Note that each leaf has a computable confidence score. Thus, two instances in the same class may have different confidence scores.

Given a particular leaf node t, we call the conditions that trigger this node x, the number of correct instances TP, and the number of incorrect instances FP (for true positives and false positives, respectively). Note that $TP = |CI_t|$ and $FP = TP - |I_t|$. We can assign a probability estimate) or confidence score as:

$$P(y|x) = TP/(TP + FP) \tag{2}$$

where y is the outcome that an instance is correct. So, $P(Y|x)$ is the outcome that an instance is correct given that it triggers leaf node t. Let $\langle TP, FP \rangle$ denote a scenario for a leaf. So we can compute confidence by applying equation 2 to $\langle TP, FP \rangle$.

7.2 Problems

It is generally accepted that the approach to confidence using equation 2 is not a good representative. Indeed, $\langle 5, 0\rangle$ and $\langle 500, 0\rangle$ suggest different levels of confidence, yet equation 2 says they are the same. There are many known strategies of correcting this deficiency. One such strategy is to apply a Laplacian smoothing technique

$$P(y|x) = (TP + 1)/(TP + FP + C) \tag{3}$$

where C is a prior probability of $1/C$ for each class.

When a dataset has an extreme imbalance among its class membership—such as 98% of the instances in class one and 2% of its instances in class 2—this Laplacian smoothing may not be enough. We need to weight estimates toward the minority class. We let m be a tunable parameter for the amount of weighting toward the minority class. Let b denote the base rate of the majority class. Then we can use

$$P(y|x) = (TP + bm)/(TP + FP + m) \tag{4}$$

to reflect a confidence. It has been shown that for a fixed b, picking m so that $bm = 10$ works well.

7.3 Process

Given a decision tree T built from a training set $\mathcal{I}_r$ and a test set $\mathcal{I}_s$, we can build a scoring function using either the instance set $\mathcal{I}_r$ or $\mathcal{I}_s$, although we would expect a more accurate (less likely to be over-fit) scoring from the latter set.

References

[1] Bandelt, HJ: (1992) "Centroids and Medians of Finite Metric Spaces", *J. Graph Theory*, v. **16**:4, pp. 305-317

[2] Ganter, Bernhard; Stumme, Gerd; and Wille, Rudolf, eds.: (2005) *Formal Concept Analysis: Foundations and Applications*, Springer-Verlag

[3] Heredia-Langner, Alejandro; Joslyn, CA; White, Amanda M; and Paulson, PP et al.: (2011) "Analysis of Zymurgy Dispersion Experiment", *PNNL Technical Report*

[4] Joslyn, Cliff: (2004) "Poset Ontologies and Concept Lattices as Semantic Hierarchies", in: *Conceptual Structures at Work, Lecture Notes in Artificial Intelligence*, v. **3127**, ed. Wolff, Pfeiffer and Delugach, pp. 287-302, Springer-Verlag, Berlin, `ftp://ftp.c3.lanl.gov/pub/users/joslyn/iccs04.pdf`

[5] Joslyn, Cliff and Hogan, Emilie: (2010) "Order Metrics for Semantic Knowledge Systems", in: *5th Int. Conf. on Hybrid Artificial Intelligence System (HAIS 2010), Lecture Notes in Artificial Intelligence*, v. **6077**, ed. ES Corchado Rogriguez et al., pp. 399-409, Springer-Verlag, Berlin

[6] Joslyn, Cliff and Paulson, Patrick: (2009) "Hierarchical Anlysis of the Omega Ontology", *PNNL Technical Report PNNL-19041*

[7] Joslyn, Cliff and White, Amanda: (2009) "Taxonomy Package (TaxPac): An Experimental Mathematics Environment for Knowledge Systems Analysis", *PNNL Technical Report PNWD-4084*

[8] Orum, Chris and Joslyn, Cliff A: (2009) *Valuations and Metrics on Partially Ordered Sets*, in: *Discrete Mathematics*, `http://arxiv.org/abs/0903.2679v1`, submitted

[9] Philpot, Andrew; Hovy, Eduard; and Pantel, Patrick: (2005) "The Omega Ontology", in: *Proc. Ontolex 2005 - Ontologies and Lexical Resources*, `http://www.aclweb.org/anthology-new/I/I05/I05-7009.pdf`

[10] Quinlan, J.R.: (1986) "Induction of decision trees", *Machine learning*, v. 1:1, pp. 81-106

Pacific Northwest
NATIONAL LABORATORY

Proudly Operated by **Battelle** *Since 1965*

U.S. DEPARTMENT OF
ENERGY

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99352
1-888-375-PNNL (7665)
www.pnnl.gov