Efficient Modeling of Analogy

Lars G. Johnsen and Christer Johansson

Dept. of Linguistics and Literature, University of Bergen,
N-5007 Bergen, Norway
{lars.johnsen, christer.johansson}@lili.uib.no

Abstract. Analogical modeling (AM) is a memory based model with a documented performance comparable to other types of memory based learning. Known algorithms implementing AM have a computationally complexity of $O(2^n)$. We formulate a representation theorem on analogical modeling which is used for implementing a range of approximations to AM with a complexity starting as low as $O(n)$.

1 Introduction

The algorithm for Analogical Modeling (AM) was first published in 1989 [1], and has since remained unchanged with the exception for some minor corrections [1, 2]. Known implementations of AM [2] suffer from having an exponential time complexity $O(n)$ where n is the number of features used to describe a particular example.

AM is a memory based model and constructs a classifier on the basis of a set of examples $\mathcal{D}$. The key computation for the analogical classifier is the construction of the analogical set $\mathcal{A}$ (defined below) associated with an exemplar τ in conjunction with $\mathcal{D}$. We will show that the effect of the analogical classifier is obtained by constructing a generally smaller set $\mathcal{M}$, which, together with a set of parameters C, has the same effect as the original. The aim of this article is to prove that there exists a simpler, yet (roughly) equivalent function to build an analogical classifier, which avoids building the full lattice $\mathcal{L}$ of the original model. The new function uses the set $\mathcal{M}$ and a set of parameters to compute a close approximation. Different parameter sets correspond to different approximations.

2 Background on AM

AM has been used as a simulation model of cognitive psycholinguistics, and it compares well with connectionist models [3, 4]. AM does not suffer from the problems associated with the delta-rule of connectionist learning [3–pp.62 ff], and at the same time it accounts for significant phenomena such as 'perceptual learning, latent inhibition, and extinction [...] within a single mechanism' (ibid. p.62). In fact, there are very few assumptions in AM; there are, for example, no assumption on the distribution of exemplars, nor are global weights calculated. Experience has a direct effect in AM in that only the database changes with

A. Gelbukh (Ed.): CICLing 2005, LNCS 3406, pp. 694–703, 2005.

added experience. The same unchanged mechanism is used to find support in the database for the categories of new input. Weights are not calculated, neither are there any fixed connections between any data. These ought to be very attractive features for a memory based model of categorization. In addition, AM reacts similar to noise as a naive k-NN model, and tend to fare slightly better than k-NN using information gain [5]. AM is also very robust for 'forgetting'. Randomly throwing out as much as half of the database is equivalent to standard statistical results [6–p.33], see also [3–pp.86–87] and [7–p.175].

AM has also been applied to practical tasks such as predicting morphological markings for past tense in Finnish [6, 1], Spanish gender, diminutive, and stress [4], Dutch word stress [5] and German plural [7] to mention a few studies. It has been suggested that outcomes could be found by means of analogy [8], which would be an unsupervised extension of the model.

AM typically beats a naive memory based (k-NN) model, and it performs similar to a k-NN model which uses the information gain of each feature [4, 5, 7]. Looking more closely at the results, AM estimates 'gang effects' of frequent formation patterns that are far from the input to be categorized [1, 6, 7]. This effect is harder to get in k-NN models, although clever feature design and information measures might help k-NN to discover such gang effects. AM has also inspired a clever way of database compression in k-NN models, using families of examples with homogeneous outcomes [9].

3 Preliminary Concepts and Definitions

The analogical classifier decides on the outcome for a particular instance τ based on the analogical support from a set $\mathcal{D}$ of examples. The examples in $\mathcal{D}$ have the same structure as τ, and are represented as indexed sets (see lattice definition below) over finite domains such as phonemes, words or morphemes. Each example $d \in \mathcal{D}$ has an atomic outcome classifying the example, notated $o(d)$, while the set of outcomes is notated $\mathcal{O}$. The analogical classifier now determines an outcome for τ from the set of outcomes $\mathcal{O}$. Whatever the outcomes actually are, we will use the fact that they form a finite domain of size n, so that each element of that domain can be associated with a unit vector in an n-dimensional integer space. This way the frequencies of different outcomes are computed by adding the corresponding unit vectors.

For example, consider the database in table 1 below. There are two possible outcomes, x and y. Examples belonging to the classes are listed in the columns (6 exemplars of each class). For simplicity the sets in the database are written so that 039 is $\{0_1, 3_2, 9_3\}$. The test element in this case is $\tau = \{0_1, 2_2, 6_3\}$, and the column heading $\tau \cap x$ means the cell below comes from intersecting τ with the set with outcome x from the cell's row, and likewise for y. The outcomes x and y may be encoded, in a two-dimensional space, as $(1, 0)$ and $(0, 1)$ respectively. This encoding scheme is assumed in definitions below.

Note that an alternative form of representation for τ and the examples in $\mathcal{D}$ is the vector form. Given τ as in table 1, it may be written $\tau = (1, 2, 6)$. The

Table 1. An example adapted from Skousen [1–p.40]

x	$(\tau \cap x)$	y	$(\tau \cap y)$
027	$\{0_1, 2_2\}$	126	$\{2_2, 6_3\}$
039	$\{0_1\}$	137	$\emptyset$
046	$\{0_1, 6_3\}$	147	$\emptyset$
047	$\{0_1\}$	148	$\emptyset$
048	$\{0_1\}$	157	$\emptyset$
058	$\{0_1\}$	159	$\emptyset$

vector representation is used in [1], while we choose the set representation, as sets come equipped with the all the algebraic operations needed to combine τ and elements of $\mathcal{D}$.

The lattice $\mathcal{L}$ consists of elements generalizing τ, containing both the analogical set $\mathcal{A}$ and our set $\mathcal{M}$. $\mathcal{L}$ is the powerset of τ and represents all the possible feature matches that τ may yield. Vector notation represents elements of $\mathcal{L}$ as vectors, with an underscore denoting an undefined value, e.g. $(0, \_, 6)$, while in set representation they are subsets of τ, e.g. $\{0_1, 6_3\} \subseteq \{0_1, 2_2, 6_3\}$.

The set $\mathcal{M}$ consist of elements that represent shared features between τ and elements of $\mathcal{D}$. It is constructed by the function μ, which takes the test element τ and combines it with an element from $\mathcal{D}$, thus mapping τ onto a subset of $\mathcal{L}$:

$$\mathcal{M} = \mu(\tau) = \{(\tau \cap d) | d \in \mathcal{D}\ \} \subseteq \mathcal{L}.$$

For example, applying μ to the elements of table 1 results in the cell elements:

$$\mu(\{0_1, 2_2, 6_3\}) = \{\{0_1, 2_2, 6_3\} \cap d \mid d \in \mathcal{D}\}$$

$$= \{\{0_1, 2_2\}, \{0_1\}, \{0_1, 6_3\}, \{2_2, 6_3\}, \emptyset\}.$$

It is evident from this definition that the maximum size of the match set, $\mathcal{M}$, is either $||\mathcal{D}||$ or $||\mathcal{L}||$, whichever is smallest. For cases with many features, $||\mathcal{L}||$ is typically much larger than $||\mathcal{M}||$, the former growing exponentially with the size of τ. $\mathcal{M}$ may reach the size of $||\mathcal{D}||$ when all examples of the database are disjoint. Typically, $||\mathcal{M}||$ is significantly smaller. $||\mathcal{M}||$ for the database in table 1 with $\tau = 026$ is 5, namely $\{0_1, 2_2\}$, $\{0_1\}$, $\{0_1, 6_3\}$, $\{2_2, 6_3\}$ and the empty set, $\emptyset$. Here, the theoretical maximum is determined by the $8 = 2^3$ possibilities of $\mathcal{L}$.

We define two mappings between $\mathcal{L}$ and $\mathcal{D}$ which are generalized inverses of μ. One mapping called θ induces a partition over $\mathcal{D}$, and is used in conjunction with $\mathcal{M}$ to compute a score, while the other mapping, σ, gives the support for elements of $\mathcal{L}$, and is used in conjunction with $\mathcal{A}$.

The function θ yields all the elements d of $\mathcal{D}$ such that $a \cap d = \tau$, i.e.

$$\theta(a) = \{d \in \mathcal{D} \mid a = d \cap \tau\}.$$

We will prove the partition inducing property of θ is proved in proposition 1 in section 4 below.

The support function σ is defined for a particular element of $\mathcal{L}$ as the collection of elements in $\mathcal{D}$ that share a feature with it,

$$\sigma(m) = \{d \in \mathcal{D} |\ m \subseteq d\}.$$

The actual scoring of a subset of $\mathcal{D}$ is done by adding up all the outcomes for each element of the subset. The function *score* is defined by

$$score(Sub) = \sum_{y \in Sub} o(y).$$

As noted above, each outcome is considered to be a unit vector in n-dimensional space, where n is the total number of outcomes.

For instance, with reference to table 1, if *score* is applied to

$$\theta(\{0_1\}) = \{039, 047, 048, 058\}$$

we know that since $o(a) = (1, 0)$ for each of the 4 elements $a \in \theta(\{0_1\})$ that

$$score(\theta\{0_1\}) = (4, 0)$$

hence there are 4 cases of x's in this case. Compare this with

$$\sigma(\{0_1\}) = \{039, 047, 048, 058, 027, 046\}$$

which results in

$$score(\sigma\{0_1\}) = (4, 2)$$

meaning that there are 4 votes for x and 2 for y.

Note that $\theta(a) \subseteq \sigma(a)$ for any $a \in \mathcal{L}$. Given a pattern $a \in \mathcal{L}$, the $score(\sigma(a))$ returns the frequency of each outcome from $\sigma(a)$ in a vector $(outcome_1 ... outcome_n)$. In table 1, there are two outcomes x and y, so that $n = 2$ given that particular database.

Homogeneity is a key concept in analogical modeling. A formal definition is given below, but the content of homogeneity is that elements of $\mathcal{L}$ can only contribute unique scorings, scorings on the form $n\mathbf{i}$ where $n \in \mathbb{N}$ and $\mathbf{i}$ is a unit vector in the set of outcomes $\mathcal{O}$. In this case, the element is said to be deterministically homogeneous. Non-unique scorings are taken into account if it comes from an element which contains no other element with a non-void scoring.

Homogeneity is formally captured through the number of disagreements in a pattern. Disagreements in $m \in \mathcal{L}$ is measured by counting the number of different outcomes in $\sigma(m)$, notated $\delta(m)$. Following Skousen [1], we define disagreement within $\sigma(m) \times \sigma(m)$. Each point $< r, s > \in \sigma(m) \times \sigma(m)$ is an argument to the function κ, defined as

$$\kappa(r, s) = \begin{cases} 1 & \text{if } o(r) \neq o(s) \\ 0 & \text{otherwise} \end{cases}$$

Then $\delta(m)$ is defined as the sum of κ over all pairs in $\sigma(m) \times \sigma(m)$.

Definition 1 (Differences). *For a given $m \in \mathcal{L}$, the number of differences in m is defined as*

$$\delta(m) = \sum_{r,s \in \sigma(m)} \kappa(r,s)$$

A homogeneous pattern is required to have non-empty support and, if it is more general than a pattern with non-empty support, they must have exactly the same number of differences.

Definition 2 (Homogeneity). *A pattern m is homogeneous if $\sigma(m) \neq \oslash$ and, whenever $m \subseteq n$ and $\sigma(n) \neq \oslash$, $\delta(n) = \delta(m)$*

The stricter class of deterministically homogeneous elements captures those with unique scoring.

Definition 3 (Deterministic homogeneity). *A pattern m is deterministically homogeneous if m is homogeneous and $\delta(m) = 0$*

Two simple consequences of these definitions are the following two corollaries:

1. If m is deterministically homogeneous then $\sigma(m)$ has only one outcome.
2. If m is non-deterministically homogeneous and $n \subseteq m$ is homogeneous, then $\sigma(n) = \sigma(m)$

With these definitions in place, the analogical set $\mathcal{A}$ is defined to be the homogeneous elements of $\mathcal{L}$, i.e.

$$\mathcal{A} = \{a \in \mathcal{L} |\ a \textit{ is homogeneous}\}.$$

The analogical classifier assigns the outcome with highest frequency to τ based on the score for the analogical set. One effect of this is that in order to construct an equivalent to the analogical classifier it may not be necessary to compute the whole effect. Once the resulting outcome is determined the behaviour of the classifier is captured.

We saw above, with reference to table 1, how σ and θ are calculated. Let us enumerate some of the sets defined so far using the data in the table:

$$\begin{aligned}
\mathcal{L} &= \wp(\{0_1, 2_2, 6_3\}) \text{ where } \wp \text{ is the powerset operator} \\
\mathcal{A} &= \{\{0_1, 2_2\}, \{0_1\}, \{0_1, 6_3\}, \{2_2, 6_3\}\} \\
\mathcal{M} &= \mathcal{A} \cup \{\emptyset\} \\
\theta(\{2_2\}) &= \emptyset \\
\sigma(\{2_2\}) &= \{\{0_1, 2_2, 7_3\}, \{1_1, 2_2, 6_3\}\}
\end{aligned}$$

Note that $\mathcal{A}$ is a proper subset of $\mathcal{M}$ for this particular dataset. In general, the cardinality of $\mathcal{A}$ will be a much larger than that of $\mathcal{M}$. Since homogeneity is not a defining characteristic of $\mathcal{M}$, $\mathcal{M}$ will likely have members which are not members of $\mathcal{A}$. Homogeneity enters into play via the parameters used for scoring $\mathcal{M}$, as defined below, ensuring that non-homogeneous elements do not count in the overall score.

4 Results

In this section, we characterize the analogical set $\mathcal{A}$ in terms of the match set $\mathcal{M}$. We prove that it is only necessary to consider this match set, which formally contains all the possible support from the database for the outcome of the test pattern τ.

The following proposition ensures that θ induces an equivalence class over $\mathcal{D}$. In particular, sets $\theta(x)$ and $\theta(y)$ are disjoint whenever $x \neq y$.

Proposition 1. *For any $x, y \in \mathcal{L}$, if $x \neq y$ then $\theta(x) \cap \theta(y) = \emptyset$*

Proof. Assume that $\theta(x) \cap \theta(y) \neq \emptyset$, and pick an a such that $a \in \theta(x)$ and $a \in \theta(y)$. By definition we then have that that $x = a \cap \tau$ and $y = a \cap \tau$, so that $x = y$ contradicting $x \neq y$.

Proposition 2. *For any $p \in \mathcal{L}$, the support for elements $p \in \mathcal{L}$ is characterized by elements of $\mathcal{M} \subseteq \mathcal{L}$*

$$\sigma(p) = \bigcup_{\substack{p \subseteq x \\ x \in \mathcal{M}}} \{\theta(x)\}$$

Proof. Any element $\delta \in \mathcal{D}$ which is a member of $\sigma(p)$ contains p, i.e. $p \subseteq \delta$. Since $p \subseteq \tau$ it follows that $p \subseteq \tau \cap \delta$, and since $x = \tau \cap \delta \in \mathcal{M}$ we have that $\delta \in \theta(x)$. Conversely, if $\delta \in \theta(x)$ for $x \in \mathcal{M}$, $x = \tau \cap \delta$, and by the assumption that $p \subseteq x$, it follows that $p \subseteq \tau \cap \delta \subseteq \delta$, which means that $\delta \in \sigma(p)$.

These two propositions serves as the building blocks for the representation theorem on analogical modeling.

4.1 Representing the Analogical Set

Recall the definition for scoring a set $S \subseteq \mathcal{D}$:

$$score(S) = \sum_{x \in S} o(x)$$

The total score attributed to the analogical set, *tot*, is the sum of all the scores over all elements of the analogical set:

$$tot = \sum_{p \in \mathcal{A}} score(\sigma(p)).$$

Our version of the total score, with the elements from $\mathcal{M}$, employs coefficients.

$$tot = \sum_{x \in \mathcal{M}} c_x score(\theta(x)).$$

The coefficients c_x encode for each $x \in \mathcal{M}$ the number of homogeneous elements l of $\mathcal{L}$ that are subsumed by x, i.e. $l \subseteq x$. Note that this implies that if a particular element $m \in \mathcal{M}$ is not homogeneous its coefficient $c_m = 0$, effectively erasing m's contribution from the total score.

Theorem 1. *The result of analogical modeling can be characterized in terms of the set* $\mathcal{M}$ *so that*

$$\sum_{p \in \mathcal{A}} score(\sigma(p)) = \sum_{x \in \mathcal{M}} c_x score(\theta(x))$$

where

$$c_x = \|\{p \mid p \text{ is homogeneous and } p \subseteq x\}\|$$

Proof. The scoring of the analogical set is

$$\sum_{p \in \mathcal{A}} score(\sigma(p))$$

which by proposition 2 is the same as

$$\sum_{p \in \mathcal{A}} score(\bigcup_{\substack{p \subseteq x \\ x \in \mathcal{M}}} \{\theta(x)\})$$

Proposition 1 says that this is a disjoint union, and it is therefore equal to

$$\sum_{p \in \mathcal{A}} \sum_{\substack{p \subseteq x \\ x \in \mathcal{M}}} score(\theta(x)).$$

Interchanging the two sums while retaining the condition $p \subseteq x$ on the inner sum, yields

$$\sum_{x \in \mathcal{M}} \sum_{\substack{p \subseteq x \\ p \in \mathcal{A}}} score(\theta(x))$$

From this expression we see that the number of times the term $score(\theta(x))$ is summed up is equal to the number of homogeneous elements it dominates. With c defined as above, the scoring of the analogical set is then equivalent to

$$\sum_{x \in \mathcal{M}} c_x score(\theta(x)).$$

The implication of this theorem is that $\mathcal{M}$ contains *all* the results necessary for computing the overall effect of the analogical set, without actually building the whole set. But how much is lost? Is it possible to estimate how many extensions a homogeneous pattern has?

4.2 Rough Approximation

The match set $\mathcal{M}$, as we have seen above, contains the necessary support for analogical modeling. We are primarily interested in the homogeneous subset of $\mathcal{M}$, which can be found by sorting $\mathcal{M}$ and determine which patterns have

multiple outcomes. A pattern with multiple outcomes is non-deterministically homogeneous, if there is no pattern in $\mathcal{M}$ that is closer to the test pattern.

Recall our example in table 1. The sorted match set for test pattern 026 is $\{\{0_1, 2_2\}, \{0_1, 6_3\}, \{2_2, 6_3\}, \{0_1\}, \{\}\}$. $\{0_1, 2_2\}$ and $\{0_1, 6_3\}$ match only one example each, both with outcome x. $\{2_2, 6_3\}$ matches one example with outcome *o*. $\{0_1\}$ matches 6 examples of outcome x. So far we have had only homogeneous outcomes. The last pattern $\{\}$ matches everything, but since there were patterns before, closer to 026, it is heterogeneous and should not be counted. In this case, the approximation gives the exact results of AM.

We could have ended here, since we have an approximation of AM, which needs to look at each example in the database only once to find the match set, and then needs a sorting of the match set (which can be done in $O(log(||\mathcal{M}||)||\mathcal{M}||)$, where $||\mathcal{M}||$ typically is much smaller than the size of the database). Finally, for each member in the match set, the algorithm goes through the examples once again. The time complexity of this algorithm in the worst case appears to be $O(log(N)N)$, though the worst case happens exactly when analogy is useless, i.e., when each example is best characterized as an atomic symbol. We expect typical time complexity to be close to linear. Furthermore, space complexity just depends on the size of the database. The search through the database is linear in the number of variables (whereas original AM is exponential, in both time and space complexity, for the number of features).

How often this approximation fails on practical language tasks is an empirical question. However, we will make an attempt at estimating the parameters c_x, which would make the approximation truly equivalent to full analogical modeling. The estimation can be done statistically by using Monte Carlo methods, presented in the next section.

4.3 Simulation of the Parameters

Monte Carlo simulation [10–pp.237–241] can be used to find the parameters c_x of theorem 7 above. When a is homogeneous, then $a \subseteq b$ implies that b is homogenous. We see that only the homogeneous subset of $\mathcal{M}$, notated $\mathcal{M}_\mathcal{A}$ is needed to estimate $\mathcal{A}$.

The collection of elements below any element $a \in \mathcal{M}_\mathcal{A}$ is $\wp(a)$. Removing the heterogeneous elements from $\wp(a)$ results in the desired collection of homogeneous elements p such that $p \subseteq a$.

The heterogeneous elements are found by counting the elements in $\wp(a)$ that are shared by other elements with a different outcome from a. The outcome, $o(p)$ for a pattern p is defined so that $o(p) = i$ if $o(x) = i$ for all $x \in \sigma(p)$. This will associate each deterministically homogenous element with a unique outcome, the outcome for the elements in its support. For non-deterministically homogenous elements the outcome is left undefined. The set $\mathcal{H}(p)$ is defined to be the intersection of p with all other elements of $\mathcal{M}_\mathcal{A}$ that have a different outcome

$$\mathcal{H}(p) = \{x \cap p \mid x \in \mathcal{M}_\mathcal{A}\}$$

under the constraint that $o(x) \neq o(p)$ if o is defined for p .

We are not aware of any algorithmically cheap (i.e. polynomial) way of determining the union of a family of powersets, but it is computationally cheap to determine whether a set belongs to that union: for a given candidate x, go through the elements of $\mathcal{H}_{limit(p)}$ one by one, and check whether x is a subset of it. This feature, together with accurate estimates of lower and upper bounds, makes $\mathcal{H}_{limit(p)}$ a candidate for Monte Carlo simulation.

The minimum number of elements in $\mathcal{H}_{limit(p)}$ is given by $\|\wp(x)\|$ where x is the largest element in $\mathcal{H}(p)$ so that $\|x\| = min(p)$. The smallest size, or lower bound $lb(p)$, of $\mathcal{H}_{limit(p)}$ is thus $2^{min(p)}$.

A candidate for the upper bound of $\mathcal{H}_{limit(p)}$ is

$$\wp(\bigcup_{x\in\mathcal{H}(p)}(x)).$$

This candidate limit can be improved by observing that no element of $\mathcal{H}_{limit(p)}$ has more members than $min(p)$.

Under the constraint that $\|\bigcup_{x\in\mathcal{H}(p)}(x))\| = max(p)$, the maximal number of sets in $\mathcal{H}_{limit(p)}$ is determined by how many ways we can construct sets from the $max(p)$ elements of p such that the sets have a cardinality smaller than or equal to $min(p)$. The upper bound, $ub(p)$ for $\mathcal{H}_{limit(p)}$ is then

$$ub(p) = \sum_{k=1}^{min(p)} \binom{max(p)}{k}$$

The size of $\mathcal{H}_{limit(p)}$ can now be estimated by sampling elements x_s from $\bigcup_{x\in\mathcal{H}(p)}(x) \subseteq p$ with cardinality less than or equal to $min(p)$.

The estimate, $\hat{h_p}$, is computed from the ratio of sampled elements over the total number of samples by the equation

$$\frac{\|\{x_s \in \mathcal{H}_{limit(p)}\}\|}{\|\{x_s\}\|} = \frac{\hat{h_p}}{ub(p)}$$

which gives

$$\hat{h_p} = \frac{ub(p)\|\{x_s \in \mathcal{H}_{limit(p)}\}\|}{\|\{x_s\}\|}$$

If every sampled x_s is in $\mathcal{H}_{limit(p)}$ the estimate equals the upper bound.

Recall

$$c_x = \|\{p \mid p \textit{ is homogeneous and } p \subseteq x\}\|$$

in terms of $\hat{h_x}$; $c_x = \|\wp(x)\| - \hat{h_x}$. Thus we have an estimate of the required parameters c_x, which we can calculate within the time bounds of the Monte Carlo simulation, which depends mainly on the size of the match set, $O(\|\mathcal{M}\|^2)$. To estimate the variation of the simulations, we are forced to repeat the simulations a limited number of times, depending on the discrepancy between upper and lower bound for each particular pattern p. This increases complexity, but we are still within polynomial time.

5 Conclusions

We have shown a rough approximation of analogical modeling, which has many beneficial properties compared to the original model. AM is often correlated with k-NN learning, but the decisions of AM are based on the entire database, and not just a neighborhood of similar instances. This makes it possible for AM to find gang effects, which may be of both practical and theoretical interest [3]. The correctness of the approximation was formally proved above (section 4). Although there might still be some support that slips away, the approximation at least gives every instance its say on the final decision. In many cases, the rough approximation exactly matches full AM.

The discussion on Monte Carlo simulation, gives an outline of how the constants c_x could be approximated without calculating the full lattice (which would be exponential in both space and time complexity). The simulations could bring the approximation even closer to the theoretical model, even though it might not be practical in many cases, and probably should only be used in cases where the decision of the rough approximation is close to being undecided.

Acknowledgement. This work was supported by a grant from the Norwegian Research Council under the KUNSTI programme (project BREDT). We thank Viktor Trón and an anonymous reviewer for helpful comments and discussions.

References

1. Skousen, R.: Analogical Modeling of Language. Kluwer Academic, Dordrecht, the Netherlands (1989)
2. Skousen, R., Lonsdale, D., Parkinson, D.: Analogical Modeling: An exemplar-based approach to language. Volume 10 of Human Cognitive Processing. John Benjamins, Amsterdam, the Netherlands (2002)
3. Chandler, S.: Skousen's analogical approach as an exemplar-based model of categorization. In: Skousen et al. (2002) 51–105
4. Eddington, D.: A comparison of two analogical models: Tilburg memory-based learner versus analogical modeling. In: Skousen et al. (2002) 141–155
5. Daelemans, W., Gillis, S., Durieux, G.: Skousen's analogical modeling algorithm: A comparison with lazy learning. In: proceedings of NEMLAP, Manchester, UK. (1994)
6. Skousen, R.: Issues in analogical modeling. In: Skousen et al. (2002) 27–48
7. Daelemans, W.: A comparison of analogical modeling to memory-based language processing. In: Skousen et al. (2002) 157–179
8. Johansson, C.: The Hope for Analogical Categories. In: Skousen et al. (2002) 301–316
9. Bosch, A.v.d.: Expanding k-nn analogy with instance families. In: Skousen et al. (2002) 209–223
10. Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: Numerical Recipes in C. Second edition edn. Cambridge University Press (1992)