

# GSLT Machine Learning Course: Written Assignment

Magnus Sahlgren & Fredrik Olsson  
SICS, Swedish Institute of Computer Science

October 15, 2003

## 1 Semantic Categorization of Words as a Well-posed Learning Problem

The task we have chosen to define as a *well-posed learning problem* is the task of learning to which predefined semantic category a given word belongs. Following Mitchell (1997), we define our well-posed learning problem as a triple  $\{T, P, E\}$ , where:

- $T$  is the task of learning to which predefined semantic category a given word belongs,
- $P$  is the percentage of correctly classified words,
- $E$  is cooccurrence statistics.

It is important at this point to make clear that the definition of  $E$  is not uncontroversial; it is far from obvious what empirical evidence this kind of knowledge is based upon. The reason why we define  $E$  as such indirect evidence as cooccurrence statistics<sup>1</sup> is because it has a certain psychological veracity [1], [2], and because it has proven to be a viable methodology in other experiments pertaining to the learned acquisition of semantic knowledge [4], [1].

### 1.1 Data representation

As training experience, we have chosen indirect empirical evidence in the form of cooccurrence statistics. An obvious way of representing this information is to use a standard vector space representation of the data, which means that we represent the data in terms of vectors  $\vec{w}$  of cooccurrence values, such that each word in the data is represented by a vector  $\vec{w}_i = (w_1 \dots w_n)$  where  $w_n$  are the cooccurrence values of the words in the data in relation to the given word  $w_i$ . One way of defining these cooccurrence values is to determine a window of  $n$

---

<sup>1</sup>Direct evidence in this case would be a thesaurus or ontology.

words surrounding the given word  $w_i$ , and to specify a cooccurrence value as a function of the distance to  $w_i$  [2], [4]<sup>2</sup>.

## 1.2 Target Function

Having decided on the representational scheme, we can define the target function we wish to learn as a function  $Categorize : \vec{w} \rightarrow C$ , where  $\vec{w}$  is the cooccurrence vector for a word in the data, and  $C$  is a category from a set of predefined semantic categories. As an example, we use the 1 035 main categories specified in Roget's Thesaurus<sup>3</sup>. Examples of such categories are:

- Existence: existence, being, entity, esse, subsistence.
- Inexistence: inexistence; nonexistence, nonsubsistence; nonentity, nil.
- Presence: presence, occupancy, attendance, whereness, permeation.
- Absence: absence, inexistence, nonresidence, absenteeism, nonattendance.

## 2 Machine Learning Algorithms

The choice of representational scheme to a large extent determines which machine learning algorithms that are feasible. Given the vector representation of the data, some algorithms suggests themselves as more suitable and natural than others. For example, Artificial Neural Networks (ANNs) is an obvious choice. So is Memory-Based Learning (MBL; referred to as *instance-based learning* in Mitchell (1997)). As a third approach, we chose Decision Trees (DT).

### 2.1 Artificial Neural Networks

The ANN approach seems like a very suitable technique for the present type of learning problem. The chosen representational scheme — vectors with real-valued elements — seems particularly fitting as input for an ANN. We further imagine that the output of an ANN can be easily interpreted as a categorization decision of the provided input. For example, the ANN could have as many output nodes as categories (in this case 1 035), and the categorization decision would simply consists in the activation of one particular output node. In this case, the decision function is very simple:  $f(\vec{w}) = \vec{c}$ , where  $\vec{w}$  is the cooccurrence vector for a given word, and  $\vec{c}$  is the category vector consisting of 1 034 zeros and a single 1.

---

<sup>2</sup>A common function is to set  $w_n = 2^{1-d}$ , where  $d$  is the distance to  $w_i$ .

<sup>3</sup>Available from Project Gutenberg's website: <http://www.promo.net/pg/>

## 2.2 Memory-Based Learning

The MBL approach also seems like a natural choice for the present representational scheme. We define the MBL as a  $k$ -Nearest Neighbor ( $k$ -NN) algorithm, which is an instance-based learning method that classifies new examples based on their similarity to previously seen examples.  $k$ -NN is, in contrast to the ANN, a *lazy* learner, since it does not require an explicit training phase. Rather, it classifies new instances “locally” by looking at the category labels of the  $k$  most similar training examples:

1. For each new word, calculate the vector similarity between the vector for the new word  $\vec{w}'$  and the vectors for all the previously seen words  $\vec{w} \in W$ .
2. The categories  $C \in \mathcal{C}$  of the  $k$  most similar training examples (the “nearest neighbors”) are weighted with the similarity score between the training examples and the new word vector, and the highest ranking category is chosen as label for the new word.

The decision rule for this  $k$ -NN algorithm can be formalized as:

$$y(\vec{w}', C) \leftarrow \operatorname{argmax}_{C \in \mathcal{C}} \sum_{w_i \in kNN} \cos(\vec{w}', \vec{w}_i) y(\vec{w}_i, C_j)$$

where  $y(\vec{w}_i, C_j) \in \{0, 1\}$  is the decision for  $w_i$  with respect to  $C_j$ , and  $\cos(\vec{w}', \vec{w}_i)$  is the similarity between the new word vector  $\vec{w}'$  and training example  $\vec{w}_i$ .

This type of  $k$ -NN algorithm is generally known as *distance-weighted* [3], since the categories of the  $k$  nearest neighbors are weighted with the distance to the test example. An alternative approach would be to select the category label that is most frequent among the  $k$  nearest neighbors — i.e. a kind of majority vote. However, simply averaging among the  $k$  nearest neighbors does not take proximity of the training examples into account, which could potentially lead to faulty classification [3].

## 2.3 Decision Trees

As the last machine learning method, we chose decision trees, which is a series of machine learning algorithms that classify feature vectors by passing them down a decision tree from the root to a leaf that represents a category. This means that a decision tree for this example will have 1 035 leafs. At each node of the tree, a particular feature of the vectors is inspected to determine which branch the vector should travel down. When the vector reaches a leaf of the tree, it is labelled with the category that the leaf represents.

The decision tree for this task would be generated by statistically determining which feature of the training vectors that is most decisive for the categorization. A node is then formed to partition the training vectors into subsets based on this feature, and the process is repeated until all the elements belong to the same category, or until the amount of information in the subset is statistically insignificant. Note that the present representation of the data requires that the decision tree handles continuous-valued features.

## References

- [1] Landauer, T. K. and Dumais, S. T. (1997) “A Solution to Plato’s Problem: The Latent Semantic Analysis Theory of Acquisition, Induction and Representation of Knowledge.” *Psychological Review*, 104(2), pp. 211–240.
- [2] Lund, K., Burgess, C. and Atchley, R. A. (1995) “Semantic and Associative Priming in High-Dimensional Semantic Space.” *Proceedings of the 17th Annual Conference of the Cognitive Science Society* (pp. 660–665). Hillsdale, New Jersey: Erlbaum.
- [3] Mitchell, T. (1997) *Machine Learning*, McGraw-Hill.
- [4] Sahlgren, M. (2003) “Random Indexing of Words in Narrow Context Windows for Vector-Based Semantic Analysis” In Lenci, A., Montemagni, S. and Pirrelli, V. (Eds.): *Acquiring and Representing Semantic Knowledge*. Forthcoming.