Generalized Features. Their Application to Classification.

Svetlana Kiritchenko and Stan Matwin

SITE, University of Ottawa, Ottawa, Canada
{svkir, stan}@site.uottawa.ca

Classification learning algorithms in general, and text classification methods in particular, tend to focus on features of individual training examples, rather than on the relationships between the examples. However, in many situations a set of items contains more information than just feature values of individual items. For example, taking into account the articles that are cited by or cite an article in question would increase our chances of correct classification. We propose to recognize and put in use generalized features (or set features), which describe a training example, but depend on the dataset as a whole, with the goal of achieving better classification accuracy. Although the idea of generalized features is consistent with the objectives of relational learning (ILP), we feel that instead of using the computationally heavy and conceptually general ILP methods, there may be a benefit in looking for approaches that use specific relations between texts, and in particular, between emails.

Generalized features are the way to capture the information that lies beyond a particular item, the information that combines the dataset in some sort of structure. Different datasets have different structures, but we could guess what kind of information would be useful for classification. It is similar to the process of choosing relevant features. For example, we can guess that the references are relevant to the topic of an article, but the relative length is not.

There have been some attempts to include additional information about a dataset to the standard classification process based on plain features. One example is using references to classify technical articles and hyperlinks to classify web pages. This research shows that some links could be confusing while others are very helpful. Another example is character recognition. The recognition process can be based not only on the shape of a character, but also on preceding characters and even preceding words.

Our attention is focused on the email classification problem. Nowadays, when a typical user receives about 40-50 email messages daily, there is a great need in automatic classification systems that could sort, archive, and filter messages accurately. Typically, people work with emails as with general texts and base the classification decisions on the words that appear in the header and in the body of an email (the *bag of words* approach). But emails have other important sources of information, and one of them is particularly interesting for us: the time they are received. Time can be useful even as a plain feature. For example, a message received in the middle of the night is probably a junk message or has been sent from the other part of the world. Besides that, we could notice a pattern that a Java newsletter is sent every Friday morning. However, more important than plain time is a temporal sequence in which the messages arrive and/or are sent. Messages are not independent of each other. In fact, once a user has sent a message, he or she would expect to receive a reply. At the office when a working group is discussing a problem, users are likely to receive a bunch of messages on the same topic during a day or two. This information can help classification dramatically, though only a small part of it has been used in previous research. Messages that form threads "message – reply" have been investigated. We want to go further and extract all possible patterns that are present in a given email sequence and use these patterns to increase classification accuracy.

The proposed learning process can be divided into the following phases:

1. To discover all temporal patterns in data;
2. To analyze the patterns and choose the most predictive ones;
3. To employ the best patterns as generalized features in the classification process.

As the first phase, we have developed an algorithm MINTS (MINing Temporal Sequential patterns) that can find frequently occurring temporal patterns in an email sequence. The important feature of the algorithm is that it finds frequently occurring patterns consisting not only of event sequences, but also of the time intervals between the events. Therefore, the approach predicts not only the expected event in a sequence, but also when the event is likely to happen. The algorithm is general, so it can be applied to any domain where temporal relations are present. Having found the patterns, we choose the most predictive ones and discard the noise. Then, we develop the generalized features based on pattern predictions and incorporate them into the classical word-based classification.