Harnessing Unlabeled Examples through Iterative Application of Dynamic Markov Modeling

Gordon V. Cormack

University of Waterloo, Waterloo, ON, N2L 3G1, Canada
gvcormac@uwaterloo.ca
cormack.uwaterloo.ca/cormack

Abstract. We describe the application of dynamic Markov modeling – a sequential bit-wise prediction technique – to labeling email corpora for the 2006 ECML/PKDD Discovery Challenge. Our technique involves: (1) converting the corpora's bag-of-words representation to a sequence of bits; (2) using logistic regression on the training data to induce an initial maximum likelihood classifier; (2) combining all test sets into one; (3) ordering the combined set by decreasing magnitude of the log-likelihood ratio; (4) iteratively applying dynamic Markov modeling (DMC) to compute successive log-likelihood estimates; (5) averaging successive estimates to form an overall estimate; (6) partitioning the combined estimates into separate results for each test set. Post-hoc experiments showed that: (a) the iterative process improved on the initial classifier in almost all cases; (b) treating each test set separately yielded nearly indistinguishable results.

1 Dynamic Markov Modeling

Recently we have shown that sequential adaptive data compression methods work well for classifying email messages into spam and non-spam [1, 2]. Our approach uses the DMC data compression model [3] to estimate the conditional likelihood of each successive bit in a message, under two separate prior assumptions: that the message is spam and that the message is non-spam. The log of the ratio of these two likelihoods is computed and the message is classified as spam if the average of these values is positive; otherwise the message is classified as non-spam. The average log-likelihood-ratio itself is used as the *decision function value* required by the 2006 ECML/PKDD Discovery Challenge[4].

The DMC model is a finite state machine with a binary label and a frequency count on each edge. As the message is processed – one bit at a time from left-to-right – the prior probability distribution for each bit is estimated using the ratio of the frequencies of the edges leaving the current state. Then the model is updated to take into account the bit's observed value. The update has three phases: first, the frequency of the edge labeled with the observed value is incremented; second, the target of this edge may be *cloned*; third, the edge is

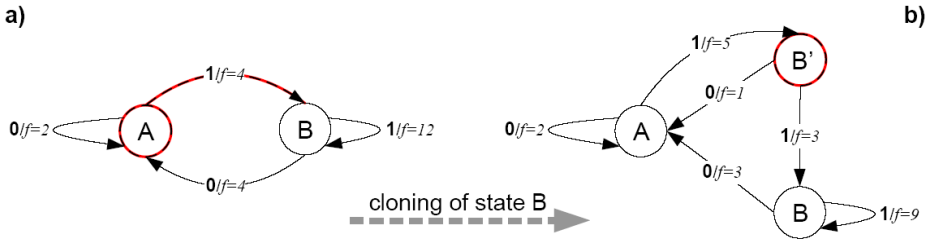

Fig. 1. DMC Cloning

followed to yield a new current state. Consider figure 1. The current state is A and the next bit has the value 1. The prior probability for this value is estimated to be $\frac{2}{3}$. Because B has been visited previously by this edge and also by some other edge,[1] it is cloned, resulting a new state B' to which the current edge is redirected. The outgoing edge frequencies of B and B' are divided in proportion to the number of times each has been visited (3:1 in this example). Finally the edge is followed resulting in a new current state of B'.

The precise criteria for cloning are as follows. Let f_1 be the frequency count (prior to incrementing) of the edge used to reach B. Let f_2 be the sum of the frequency counts of the edges leaving B. Cloning takes place if $f_1 > t_1$ and $f_2 - f_1 > t_2$ for two arbitrary threshold parameters t_1 and t_2. f_1 is the number of times that B has been visited within the current context; f_2 is the number of times it has been visited from other contexts. We used $t_1 = t_2 = 2$, default values which were known to work well in other applications.

To classify messages, we construct two DMC models – one for spam and one for non-spam. For each message in turn, we twice apply the adaptive process detailed above – once using the spam model and once using the non-spam model – to compute the likelihood of the message for each class. After each message is classified, updates to the model corresponding to the *incorrect* class (as determined by the label) are discarded, whereas updates to the model corresponding to the *correct* class are preserved.

The DMC model continuously grows, using more context for predicting common bit sequences. It is thus sensitive to bit, character, and word frequencies, as well as intra- and inter-word patterns as well as punctuation and formatting. Also, since the model is dynamic, it is sensitive to inter-message patterns and the order in which training examples are presented to it. These sensitivities appear to be advantageous in classifying real sequences of email messages.

It is not obvious that the DMC model should be applicable to the vector-space representation of messages in the ECML/PKDD Challenge. This representation discards lexical and formatting information, the order of words, and the order of messages in the test and training sets. Only the frequency of word occurrences is explicitly preserved. We used this information to create a sequential rendering of

[1] We know the total number of visits to B by summing the frequencies of its two outgoing edges.

each message in the following manner: we represented each feature in the vector space as a 16-bit[2] integer. A feature occurring k times in a message was rendered as k adjacent occurrences of the corresponding 16-bit integer. The renderings for all features were concatenated to render the message. This representation aptly captures word frequencies, but not intra- or inter-word patterns[3].

Experiments with the tuning data showed that the DMC model classified the training data very well, and classified the test data fairly well, but with high sensitivity to factors such as the order of the training data. Based on these experiments, we chose a hybrid approach, using logistic regression to build an initial classifier from the labeled training data, followed by several iterations of DMC to harness the unlabeled test data.

2 Iterative Classification

We applied Goodman's adaptive logistic regression implementation [5] to the training and test messages to yield initial log-likelihood-ratio estimates for each of the test messages. Based on these estimates we computed tentative labels for the test messages: those with a positive log-likelihood ratio were labeled as spam; the rest as non-spam.

A sequence of labeled messages was created by concatenating

- the training messages and labels, in the order given
- the test messages, labeled as described above, in decreasing order by the magnitude of the log-likelihood-ratio estimate (i.e. by diminishing confidence in the label).

Our DMC classifier was applied on-line to this sequence, classifying each message before training on its label, to yield a new sequence of log-likelihood-ratio estimates. A new sequence of labeled messages was created from these estimates as described above, and the process repeated five times.

Finally, the six log-likelihood-ratio estimates for each message (initial plus five iterations of DMC) were averaged to yield the final classifier output.

In summary, we use the output from one stage of classification to synthetically label training examples for the next. Since the training examples are ordered by confidence, we have reason to believe that early examples are likely to be correctly labeled, so that the adaptive DMC model will grow to include patterns from correctly classified messages that may not be present in the initial training

[2] There are fewer than 2^{16} distinct features in the ECML/PKDD Challenge data, but they are numbered discontiguously resulting in values greater than 2^{16}. We renumbered the features – preserving the original order but eliminating unused values – to achieve a 16-bit representation.

[3] We note that the feature numbers were assigned by ECML in order of first occurrence. We expect if two common features frequently occur adjacent, there is a good chance that their first occurrences are adjacent. Therefore, some small reflection of the inter-word patterns may be preserved in our rendering.

data. Similarly, by combining together the separate tests, we have reason to believe that the model may grow to incorporate other features representative of spam or non-spam. Finally, prior experience gives us reason to believe that combining classifiers – even combining weak and strong classifiers – by summing log-odds-ratios would provide accurate and reliable results [6].

3 Results

Test Set	Initial LR	Combined mailboxes	Separate mailboxes
task_a_u00	15.1	19.6	19.7
task_a_u01	12.4	11.7	11.2
task_a_u02	7.1	2.6	1.3
all task_a	12.0	11.4	10.9
task_b_u00	23.5	1.7	3.3
task_b_u01	21.3	2.5	2.9
task_b_u02	7.1	3.2	2.1
task_b_u03	3.8	1.2	1.0
task_b_u04	11.3	3.2	2.9
task_b_u05	12.0	10.1	11.9
task_b_u06	20.2	11.7	4.3
task_b_u07	9.5	3.6	4.3
task_b_u08	12.4	2.4	1.5
task_b_u09	16.6	4.0	4.7
task_b_u10	11.5	6.2	7.1
task_b_u11	9.5	5.3	5.1
task_b_u12	17.4	11.1	12.8
task_b_u13	12.7	7.4	8.0
task_b_u14	18.9	6.7	7.3
all task_b	13.7	5.2	5.1

Table 1. Per-mailbox error rates [1-AUC (%)]

Test Set	Initial LR	DMC1	DMC2	DMC3	DMC4	DMC5	Final average
all task_a	12.0	17.9	13.6	11.0	9.7	9.2	10.9
all task_b	13.7	10.0	6.1	5.8	5.5	5.6	5.1

Table 2. Iterative results – separate mailboxes [1-AUC (%)]

Table 1 shows the area *above* the ROC curve (as a percentage) achieved by the initial and final classifier for each mailbox (test set) and for all mailboxes combined in Task A and Task B. The first column is the score achieved by the LR classifier using only the labeled training data. The second column is the end

Test Set	Initial SVM	DMC1	DMC2	DMC3	DMC4	DMC5	Final average
all task_a	14.2	11.4	8.5	7.1	6.5	6.2	7.0
all task_b	49.3	42.4	37.5	38.0	38.2	38.3	24.8

Table 3. SVM initial classifier – separate mailboxes [1-AUC (%)]

result of the iterative process applied to the combined mailboxes. This column corresponds to our official results. The third column shows the result of iterating on each mailbox separately.

We see that in all but one mailbox the iterative process reduced the error rate, in most cases substantially. Unfortunately, the one case was the first mailbox of Task A, with the net result being that the iterative process made an insubstantial improvement overall for this task. For Task B, the iterative process worked remarkably well. For all mailboxes the result was improved; for most, substantially so. The overall effect was that the result on Task B was significantly better than any other.

Table 2 illustrates the convergence of the iterative process. In general, classifier error diminishes with the number of iterations but appears to be near asymptote after the fifth iteration. The final combined score is in general a small further improvement.

We investigated several alternatives in an effort to determine why our method worked well for Task B and not for Task A. We first repeated the iterative process separately for each of the eighteen mailboxes in both tasks. The results – seen in column 3 of table 1 – are nearly indistinguishable from and certainly not significantly different from those derived from combined mailboxes. We know from data compression experiments that DMC is particularly good at modeling heterogeneous data, because it simply "grows" distinct states to handle each category. Therefore we should perhaps not have been as surprised as we were to observe that combining mailboxes has no substantive effect.

We investigated the use of a different initial classifier, namely SVM^{light} with binary features and default parameters. As shown in figure 3 the results were mixed. For Task A, the initial SVM classification was inferior to that yielded by logistic regression, but our iterative process was much better able to improve on it, yielding a dramatic overall improvement. For Task B, the initial SVM classification was very poor, presumably due to over-fitting. The iterative technique was able to improve on it considerably, but the overall result is still not good.

We further investigated the effect of the size of the training and test sets on the result. Using a small sample of the Task A training data had an insubstantial effect – the results were very slightly worse. Similarly, results on a sample of the Task A mailbox were insubstantially different from those on the whole.

4 Conclusion

Our technique for iterative labeling using Dynamic Markov Modeling makes effective use of unlabeled training data to improve on an initial discriminative

classifier derived from labeled data. The technique provided an improvement for seventeen of eighteen mailboxes consisting of unlabeled messages, whether applied to the mailboxes separately or as a common set. For the majority of mailboxes, the improvement was substantial, in some cases reducing the area above the ROC curve by a factor of ten.

We are unable to find a reason for discrepancy between Task A and Task B results. In our pilot experiments the method performed as well on Task A as on Task B. We tried adjusting the training and test set sizes, and the number of combined test sets. We can only conclude that the data used in at least one of the Task A mailboxes is materially different from the rest; perhaps examining the unobfuscated data would yield insight as to the nature of the difference.

References

1. Bratko, A., Cormack, G.V., Filipic, B., Lynam, T.R., Zupan, B.: Spam filtering using statistical data compression. **Journal of Machine Learning Research** (in press; see http://plg.uwaterloo.ca/~gvcormac/jmlr.pdf)
2. Cormack, G.V., Bratko, A.: Batch and on-line spam filter evaluation. In: CEAS 2006 – Third Conference on Email and Anti-Spam, Mountain View (2006)
3. Cormack, G.V., Horspool, R.N.S.: Data compression using dynamic Markov modelling. The Computer Journal **30**(6) (1987) 541–550
4. Bickel, S.: ECML/PKDD 2006 Discovery Challenge http://www.ecmlpkdd2006.org/challenge.html (2006)
5. Goodman, J., Yih, W.T.: Online discriminative spam filter training. In: The Third Conference on Email and Anti-Spam, Mountain View, CA (2006)
6. Lynam, T.R., Cormack, G.V.: On-line spam filter fusion. In: 29th ACM SIGIR Conference on Research and Development on Information Retrieval, Seattle (2006)