Judging the Performance of Cascading Models: A First Look

Matthew Michelson MMICHELSON@FETCH.COM
Sofus A. Macskassy SOFMAC@FETCH.COM
Fetch Technologies, 841 Apollo St, Ste. 400, El Segundo, CA 90245 USA

Abstract

This paper explores different methods for interpreting the results of multiple, cascading machine learners, each of which performs a different task. For instance, the first learner may classify news as "sports," the second learner may extract the people from the sports articles, and the third learner may classify the extracted people as belonging to a certain team. We present a framework for modeling such cascading learners as a directed acyclic graph, which allows us to construct three-way contingency tables on which we can perform various independence tests. These independence tests provide insight into how the various learners' performance depend on their predecessor in the chain and/or the inputs themselves.

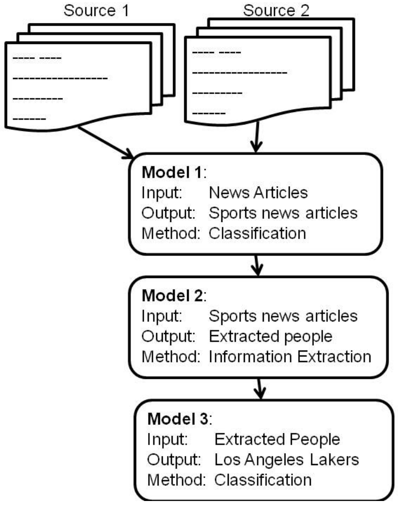

Figure 1. Cascading Models

1. Introduction

Chaining together multiple machine learners in a cascading process can yield powerful algorithms, but understanding the results of each learner poses problems. More concretely, consider three cascading machine learners, as shown in Figure 1. The ultimate goal is to process news articles and pull out mentions of anyone who belongs to the "Los Angeles Lakers" basketball team. To accomplish this goal requires three different machine learning models.

The first machine learner takes in news articles from two different sources and classifies the articles as sports or not sports. The second machine learner then takes all sports articles as inputs, and extracts all of the people in the news articles. Lastly, the third machine learner classifies each of the extracted people as players for the "Los Angeles Lakers" (or not). Since a model's input is the output of the model preceding it, we call this a "cascading" machine learning model.

Our notion of cascading learners here differs from previous work on combining machine learners where the goal is to group learners for improvement on the *same* task (e.g. (Gama & Brazdil, 2000; Dietterich, 2000; Kuncheva, 2004)). Instead, we focus on combining learners that each perform a different task, but tie together. This often happens in information extraction where it might be necessary to first discover the documents to process before extraction (Corney et al., 2004), or classify particular sentences to aid extraction (Riloff et al., 2005). In these cases, multiple processes affect each other toward the goal of extraction.

Difficulties arise in determining how well the results for each model relate to the inputs and the preceding model. For instance, Model$_3$ may classify Los Angeles Lakers players with 95% accuracy. However, Model$_2$ may only extract 50% of the people from the sports articles. Therefore, does Model$_3$ perform well because it is a great classifier, or because it only has to classify half of the extracted people? That is, how much does Model$_2$ depend on Model$_3$?

Further, a model might not only depend on the model preceding it, but the inputs themselves. Model$_2$ might only extract 50% of the people. However, this is not due to Model$_1$'s results, but rather the fact that it performs almost perfectly given articles from Source$_1$, but fails miserably on articles from Source$_2$. Therefore, input dependency is also something to consider.

Note that the input dependency, as we define it here, is based on disparate sources. However, this is easy to generalize to most machine learning exercises. The key is that the inputs need to be stratified and such that inputs can be tested for dependencies conditioned on the strata. For example, the traditional x-fold crossvalidation experimental framework fits this criterion. In this case, each "input strata" is the slice of test data for each trial, and users can see if the models are actually performing independently of the test data splits. So we believe that understanding both how each model depends on its predecessor *and* the input strata is important. Therefore, we need a mechanism by which to measure such dependencies.

To examine these dependencies, we propose using three-way contingency tables based on a directed acyclic graph (DAG) modeling of the cascade. Building three-way tables allows us to perform a number of statistical tests, such as mutual independence, joint independence, and conditional independence, which yields a better understanding of how each piece on the cascade fits together, along with the input strata. Mutually independent pieces means that we may assert that the performance of the current model does not depend on the previous model or the inputs. For instance, if it holds we could claim that Model_3 classifies Laker players with 95% accuracy, and this is not due to the limited people extracted by Model_2 or which source the articles come from.

Meanwhile, joint independence testing yields insights such as Model_3 may (or may not) depend on Model_2, but the models are both independent of the input strata. This isolates pieces with respect to the input since it says the dependency is only on the cascade, and has nothing to do with the various inputs.

Conditional independence tests, conditioned on the input strata, examine the role that each input strata plays in influencing the results. If a current model and its predecessor are conditionally independent based on the input strata, this means that each model is somehow dependent some given input. Note, they may not be dependent on each other, but only through the input. This is interesting because it would show that some input strata (say Source_2) is affecting the models and therefore affecting each result. Lastly, independence tests can yield more compact mathematical expressions for the whole cascading procedure (e.g., by factorization of conditionally independent pieces, multiplication of mutually independent pieces, etc.), which can allow for easier computation of the expected probabilities of seeing a certain path through the cascade (e.g. the path: article classified as sports, person is extracted, person is classified as Laker).

The rest of this paper is as follows. Section 2 describes how to turn cascading machine learners into a DAG which translates into a three-way contingency table, and reviews the independence tests for three-way tables. Section 3 presents our conclusions and future directions of this research.

2. Judging Cascading Learners

As stated above, the goal is to model our cascading learner process in such a way that we can build three-way tables for independence testing. To do this, we make the simplifying assumption that a model's performance depends on either (or both) the preceding model's performance in the cascade, and/or the inputs themselves. Although our assumption ignores the many possible complex interactions, it gives us a framework for generating three-way table tests. Further it is not clear how to model arbitrarily complex dependencies. Using this assumption we can generate a directed-acyclic graph from the cascade where the direction of edges follows the flow of inputs through the whole cascading process. So the nodes of the graph are the models (plus one node for the inputs), and the edges follow the flow of inputs through the models. Since each model's performance can also depend on the inputs themselves, we also add an edge between the input node and each model node. Figure 2 (a) shows the the DAG constructed from the cascade of Figure 1, and Figure 2(b) shows the generalization of the framework for arbitrary cascading models.

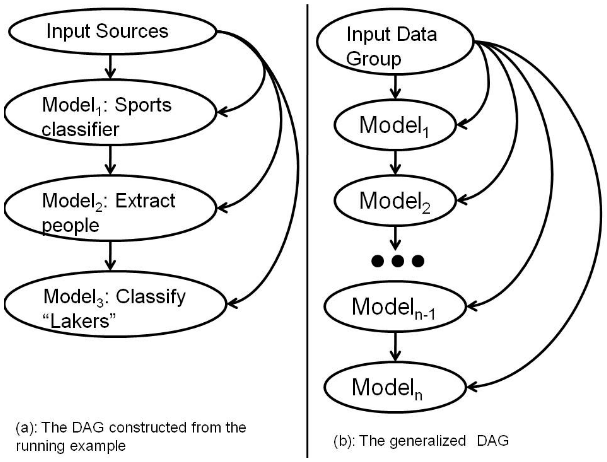
Input Sources
Model$_1$: Sports classifier
Model$_2$: Extract people
Model$_3$: Classify "Lakers"
(a): The DAG constructed from the running example
Input Data Group
Model$_1$
Model$_2$
Model$_{n-1}$
Model$_n$
(b): The generalized DAG

Figure 2. DAG for Cascading Models

Given the DAG, we next construct three-way tables for analysis. Note that for each current model (minus the first), we have three participating components: the current model M_x, the previous model M_{x-1}, and the inputs I. (For notational convenience, we will refer to predecessor M_{x-1} as M_p.) Further, our goal is to examine the performance, therefore for each model we create two categories: whether an extraction is correct (e.g. labeled "corr") or not ("incorr"). For inputs I, we define each category as the strata's label (e.g. Source 1, Fold 1, etc.).[1] Therefore, we construct tables by considering triples of the form I X M_p X M_x. For instance, given Model_3 of Figure 1, we come up with the following three-way table, shown in Table 1, which we condition on the input stratas. (Note, Model_2 gets almost 50% correct while Model_3 gets roughly 95% correct). Given this three-way table, we can now test a given model, its predecessor, and the inputs for the various independence tests.

[1] Note that three-way table analysis is not unique to nominal or categorical data. We make this assumption here to keep our discussion focused.

Table 1. A Three-Way Table: I X M$_2$ X M$_3$

Input Strata		Model 3	
	Model 2	Corr	Incorr
Source 1	Corr	474	26
	Incorr	466	24
		Corr	Incorr
Source 2	Corr	471	27
	Incorr	470	29

Table 2. A Three-Way Table for Joint Indep.

Model 2 Result		Input Strata	
	Model 3	Source 1	Source 2
Model 2: Corr	Corr	474	471
	Incorr	26	27
		Source 1	Source 2
Model 2: Incorr	Corr	466	470
	Incorr	24	29

2.1. Mutual Independence

If a mutual independence test holds, this means that the performance of current M$_x$ does not depend on either its predecessor M$_p$, or the inputs I. This is an important result, for instance, when we examine Model$_3$ and see its performance is 95%. If Mutual Independence holds, we can assert that Model$_3$ is performing well and did not depend on some configuration of Model$_2$ or certain inputs I. Note, however, that there could still be bias introduced by Model$_2$. For instance, Model$_2$ could only get 50% of the extractions correct, but these could be the easiest for Model$_3$ to classify. In this case, if these easy cases were independently distributed across sources, we could have mutual independence, but this could be misleading because of the bias. This is still preliminary work, and we need to determine how to isolate such bias in our framework. Nonetheless, a number of interesting results can come from mutual independence testing. For one, as we stated, we can isolate the pieces that perform well regardless of their predecessor and inputs. Second, if all models are mutually independent, then we can theorize about the eventual probability of any possible outcome through the process. That is, we have some likelihood of a given input strata member, and likelihoods for each category of each model, so if the whole chain is mutually independent, the probability of seeing some path through the chain is just the product of these likelihoods. This is an interesting result as it allows users to infer various probable outcomes. If this holds through the whole process, it becomes easy to determine the likelihood of all outcomes through the process (and at each step of the cascade).

To calculate the mutual independence, we first calculate all of the expected cell counts for the three-way table. That is, for some combination (ipx) $\in$ {I X M$_p$ X M$_x$}, we define the expected cell count as:

$$E_{ipx} = \frac{n_{i++}n_{+p+}n_{++x}}{n^2}$$

Where n_{i++} is the count for each i summed over the other variables (e.g. $n_{i++} = \sum_{p=1}^{M_p}\sum_{x=1}^{M_x} n_{ipx}$), (similarly for n_{+p+}, n_{++x}) and n^2 is the square of the total counts. For example, using Table 1, the total number of observations is 1,987, and so the value of $E_{Source\ 1,Corr,Incorr}$ = ((474 + 26 + 466 + 24)*(474 + 26 + 471 + 27)*(26 + 24 + 27 + 29)) / (1,987^2) = 26.53.

Once all expected frequencies are calculated, one compares them to the observed frequencies (e.g. the values in the table) using the chi-square statistic.[2] At a 95% confidence level we see that indeed, Model$_3$, Model$_2$, and the Inputs share mutual independence. That implies that the accuracy of Model$_3$ does not depend on how well Model$_2$ performs or what source delivers the inputs. Further, we could factorize this part of the chain into independent events, and if Mutual Independence holds for Model$_1$ and Model$_2$, the probability of being classified by any classifier along the way is just the product of those likelihoods. (Note, you need to test for dual independence between Model$_1$ and the Inputs).

2.2. Joint Independence

We use the Joint Independence test to examine whether the models M$_x$ and M$_p$ perform independently of the inputs I. If Joint Independence holds, although there may (or may not) be an association between M$_x$ and M$_p$, their performance does not depend on the various input strata I. This result would state that across the various inputs, the performance of M$_x$ and M$_p$ should hold (or to put it another way, the performance of M$_x$ and M$_p$ does not predict the input strata as a response). To test M$_x$ and M$_p$ jointly against I, we set up the three-way table slightly differently, as shown in Table 2.

Using this new view of the table, the test for Joint Independence is similar to testing for Mutual Independence (in fact it is a special case). However, we change the definition of the expected cell count to:

$$E_{pxi} = \frac{n_{px+}n_{++i}}{n}$$

Again, we compare these expected values to the observed table values using the Chi-square test for independence.[3]

Using Table 2, the Joint Independence holds for a Chi-square test with a confidence of 95%. Therefore, the performance of Model$_2$ and Model$_3$ does not depend on the input coming from either Source$_1$ or Source$_2$. Given how the performance is evenly distributed across sources, this is not surprising. So, the result indicates

[2] The degrees of freedom for this table are (I X M$_p$ X M$_x$ - 1) - ((I - 1)+(M$_p$-1)+(M$_x$-1)). So, in our example it is (2*2*2-1)-3 = 4 degrees of freedom.

[3] For this test, the degrees of freedom changes to (M$_p$ X M$_x$ -1)(I -1). So, our example becomes (2*2 -1)(2 -1) = 3.

that the models do hold across the various input strata. To reiterate, the Joint Independence test allows us to examine whether the results of the two connected models depends on the strata of the inputs.

2.3. Conditional Independence

Conditional Independence testing allows us to examine the case where the models are related to each other, but only through the inputs. That is, M_x is related to I, and M_p is related to I, but M_x and M_p are not directly related, but rather only related to each other because of their relation to I. In other words, the relationship between the performance of the models is related to how they each perform on some *given* input strata. It is important to understand that this tests for independence *given* (i.e. fixing) the strata of the inputs. This is in contrast to Joint Independence where M_x and M_p are independent of *any* input strata. For this test, we use the original view of the example data provided by Table 1.

As with the previous two tests, we calculate the expected counts and compare them to the observed counts, using the Chi-square test. In this case, we define degrees of freedom as (I X (M_p - 1)(M_x - 1)), and the expected counts as:

$$E_{ipx} = \frac{n_{ip+}n_{i+x}}{n_{i++}}$$

In fact, using Table 1, Model_2 and Model_3 are conditionally independent given input strata I, at a confidence level of 95%.

Beyond examining how the models hold up against given input strata, if Conditional Independence holds we can factorize this piece of the cascade down to just the conditionally independent part. That is, rather than considering the whole probability of seeing a given classification chain up to the current model, we instead just use the factorization. This yields a more compact and easily calculated probability.

2.4. Brief discussion

Using our example, all three of the independence models hold. However, one must consider the relationships between these models. Essentially, Joint Independence is a special case of Mutual Independence, so if variables are mutually independent, then they are jointly independent. If the models are jointly independent of the input, that means that given any input, the models would still be independent. Therefore, since the models are jointly independent of the input, they are also conditionally independent of it too, since they would be independent for a *given* input if they are independent for *any* input. Essentially, the tests should be assessed in order. If Mutual Independence holds, there is no need to test for Joint Independence. However, if it does not hold, one can test for Joint Independence to see if the models are independent of all of the inputs. If they are not, then one can apply the Conditional Independence test, to find out if for some given input strata the models are independent.

3. Conclusions and Future Work

This paper presented a framework for understanding how cascading machine learners fit together to accomplish a certain task. In particular, we want to examine how the performance of some part of the chain is affected by the previous part and the inputs themselves. We show how to address this issue by formulating a DAG of the chain, which allows for three-way independence testing. Our independence tests allow for the analysis of 3 cases: Mutual Independence (MI) testing shows that a current model's performance is not affected by either its predecessor or the inputs. When MI does not hold, we test for Joint Independence (JI) between the models and the inputs. JI shows that although there may or may not be a dependency between the models, the models do not depend on any of the input strata. Lastly, when JI does not hold, we test for Conditional Independence (CI), conditioned on the inputs. CI shows that even though the models may appear independent, but they in fact depend on certain fixed values of the input (e.g. some fixed strata).

As stated, it is possible for bias to leak into this analysis. For instance, Model_2, Model_3, and the Inputs of Figure 1 could all be mutually independent. However, Model_3 could still do well because Model_2 only parses out the easiest people to classify. So, even though a mutual independence is established (i.e., the easy people are evenly distributed across sources and pulled out as likely as other people), this bias could still exist. Modeling, understanding, and even discovering this bias is part of our future research.

Acknowledgements

This work was sponsored in part by the Office of Naval Research under award number N00014-07-C-0923 and by the Defense Advanced Research Projects Agency (DARPA), through the Department of the Interior, NBC, Acquisition Services Division, under Contract No. NBCHD030010. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Office of Naval Research, DARPA, or the U.S. Government.

References

Corney, D. P. A., Jones, D. T., Buxton, B. F., & Langdon, W. B. (2004). Biorat: Extracting biological information from full-length papers. *Bioinformatics*, *20*, 3206–3213.

Dietterich, T. G. (2000). Ensemble methods in machine learning. *Multiple Classifier Systems* (pp. 1–15).

Gama, J. a., & Brazdil, P. (2000). Cascade generalization. *Mach. Learn.*, *41*, 315–343.

Kuncheva, L. I. (2004). *Combining pattern classifiers: methods and algorithms*. Wiley-Interscience.

Riloff, E., Wiebe, J., & Phillips, W. (2005). Exploiting subjectivity classiffication to improve information extraction. *Proceedings of AAAI*.