Title

A Method for machine based learning for the purpose of classifying items based upon already learned information from corpi of known classified items.

Abstract

This invention, called the Evolution Classifier(TM) or Evolution Spam Filter(TM), is a method of taking a stream of items and classifying the items based upon corpi containing learned information to test the new items against the already classified items. The item is broken down into information tokens and the tokens are combined into fingerprints which are compared to the already learned fingerprints in the corpi.

An example where this would be used is a computerized filter to remove junk email (known as spam) from good email (known as ham). The examples used here will be in the context of spam filtering but is not limited to that particular function. This method of classification could be used to identify any items for which there are known classified items already learned for comparison.

If the new item is classified with a high degree of confidence the new item is added to the corpus of the learner where its tokens and fingerprints are used to help identify future items to be classified. Such a system would likely include a smart forgetting mechanism for cleaning out wrongly learned fingerprints and expiring old data.

Description

There are many ways to classify spam that already exists in prior art. Most of these systems involve starting with a corpus of known spam and ham and performing commonly known methods for matching it to the ham and spam corpi to see which one it most closely matches. A well known practice called Bayesian Filtering uses this to create a statistical analysis of words in an email to generate a probability spectrum as to how much it is like the spam and ham corpi respectively. It is also common to write rules to compare text in the item being tested to know text strings looking for matches.

This invention is different in that while it does do some matching, the “secret sauce” however is in what it *doesn’t match*. “**Not Matching” is the one of the novel ideas as to how this classifier works.** Instead of comparing fingerprints to a finite set of things that are known, this method compares fingerprints to an infinite number of things that are not known.

There are an infinite number of words and phrases that are never used in spam. For example, technical terms. If I’m in the electronics business I talk about capacitors, transistors, integrated circuits, switching power supplies. A doctor talks about disease names, treatments, procedures, medical devices. People in ordinary conversation talk about, “Do you want to go to lunch”, “How was your commute this morning?”, lots of ordinary phrases that are never seen in spam.

Spammers talk about other stuff. They talk about “Lonely Brazillian Girls looking for boyfriends”. “Cheap Canadian Medstore”, “Elon Musk’s secret to free electricity”. Although both use a lot of words and phrases in common, the world of words and phrases they don’t use in common is infinite.

So if I get an email from my patent lawyer talking about “independent claims” I know that email is ham because, and this is the secret sauce, no spammers have ever used the phrase “independent claims”. After a billion spam emails not once was “independent claims” mentioned in any spam. I don’t have to know anything else about the email because mentioning “independent claims” tells me the message is good. Similarly if the subject line says, “Meet Horny Russian Brides online!” I know that’s spam because the phrases “meet horny”, meet horny Russian”, Russian brides online” are used only in spam message and never used in ham.

Definitions:

In order to talk about the Evolution Classifier terminology will need to be defined.

Item - An item is the thing that is being classified. In this description of how the system with we will be using email messages as the item example. However an item is anything that can be classified. It could be blog posts. It could be photographs. It could be the list of things a person bought at a grocery store.

Stream - Is the line of items coming into the system for classification. A stream of items come into the system, they are classified, and they they are output to their respective destinations. When items are classified with high confidence as to if they are (for example) ham or spam then these items would be run through the learner where the items fingerprints are stored. These learning streams might be referred to as the “ham stream” or the “spam stream” which feed the ham and spam corpi.

Corpus - is a set or collection of fingerprints or a collection of sets of fingerprints (Figure 1A) that has been learned and is used as a reference for comparison and evaluating new items. There would generally be 2 or more distinct corpi. In this example the spam corpus and the ham corpus. Information from the input stream is classified and one of the results is new information is added to the corpus and old and wrong information is removed from the corpus by unlearning or forgetting. (See Figure 1A)

Token - A token is a unit of information that is either contained in the item or known about the item. In the case of email the words in the email are tokens. The message headers are also tokens. Other tokens can be generated by what’s not there or missing. “Missing Subject” is a token. “Failed to close the connection”, “Received on backup server”, “slow data rate”, all of these are pieces of information, which include behavioral information, are regarded as tokens. (See Figure 1C)

Fingerprint – A fingerprint is individual tokens or combinations of tokens that are extracted from the items tokens. Tokens are often extracted in a manner to increase the possibility that that they will be relevant to the classification process. Tokens can be used individually and/or combined into fingerprints by selecting 2 word, 3 word, 4 word, etc. phrases sequentially or by combining many combination of characteristics of the item into groups of 2 characteristics, 3 characteristics, 4 characteristics, etc. The fingerprints are then used to compare the item to the fingerprints in the corpi to classify the item. After classification the items fingerprints might be added to the matching corpi, or deleted from the non-matching corpi to unlearn the common fingerprints. (See Figure 1C)

Attributes - Attributes are fingerprint categories that separate item fingerprints into different classifications for separate comparison. For example in email the Subject would be an attribute. The message body would be a different attribute, The message headers would be yet another. Attached file names, the name part of the From address, The text inside of links within the message, the names and paths of the PHP script generating the message, as well as facts about the behavior of the email sender, are also attributes. (See Figure 1B)

Attributes contain different kinds of information and are often processed differently. An email subject, for example would be fingerprinted sequentially. (Figure 3A) A behavior attribute would be fingerprinted by generating fingerprints of all combinations of behavior looking for matching combinations stored in the corpi. (Figure 3B)

Generally attributes are fingerprinted separately, stored separately, and compared separately. Although the subject line and the body are both text they can be processed as separate independent sets. So the ham and spam corpi and not necessarily just one set but a collection of sets of fingerprints of different attributes. (Figure 1A)

Different attributes might be fingerprinted differently. For example - the subject line and parts of the body would be fingerprinted sequentially, the same order that the words appear, as 1, 2, 3, 4, ... word phrases. The number of tokens combined in the subject fingerprints might be different than in body fingerprints. Behavioral characteristics might be fingerprinted using all combinations of the tokens in groups of 1, 2, 3, 4, ... tokens. This allows different attributes to be tokenized differently and compared to their individual sets. (Figure 3A and 3B)

Sets - A set is a mathematical concept that is basically of collection of things. Sets can be compared in basically 3 ways. Set *union* is the sum of both sets (A or B). Set *intersection* are the set members are are in both sets (A and B). And set *difference* is one set minus the items in the other set. There can be 2 difference operations with 2 sets. Set A minus Set B. And Set B minus Set A. (Figure 5A and 5B)

Ham and Spam - In the spam filtering world spam refers to junk email and ham refers to good email. In this abstract however email is used as an example. Ham and Spam is also Good and Bad, Flowers and Weeds, Wheat and Chaff, Good Cells and Cancer. It could also be 3 or more states such as Republicans, Democrats, and Tea Party.

The Novel Methods

Like many spam filtering systems email comes in and is compared to spam and ham. Rules are applied, blacklists are queried, and at the end the good email is delivered to the user and the bad email is sent to the trash. Some of the high confidence results are cycled back into the learner to add to the accuracy of the filter. These prior art methods rely on matching things to the corpi or matching rules. They are all based on matching things.

The Secret Sauce

The novel element of this classifier is based on what it *doesn't match*. Although the Evolution Classifier does some matching like other filters the magic is in what it doesn't match. **Not matching** is the key to how this system works.

If an email message contains words and phrases that are used in other ham emails but never ever used in spam then it would be ham. But where do I get a list of every word never used in spam? I store every word used in spam and test to see if it's *not* in the list.

Imagine I have 2 sets (collections of sets technically) and these sets were generated from millions of samples. One set is every fingerprint used in ham and they other set is every fingerprint used in spam.

A new email comes in. It is separated into attributes, (Figure 4) tokenized, and hundreds of fingerprints are generated (Figure 3A and 3B). Through set intersection and set subtractions I derive a list of all fingerprint matches with ham and spam (Figure 5A and 5B). (An overview of the process can be visualized in Figure 2.)

The classifier is somewhat disinterested in what matches both sets or neither set. What it is looking for are fingerprints that match ham and **do not match** spam, or fingerprints that match spam that **do not match** ham.

Generally with email generating hundreds of fingerprints email with usually match one side very predominately over the other. Some email will not produce results and will need to be evaluated by other means.

The idea here is - if my email has characteristics that matched the ham sets and these characteristics were never seen - not even once - in millions of spams - then the message is ham. And if the message matches characteristics from the spam sets that was never seen - not even one - in ham, then it's spam.

For example, suppose I filter email for a machine dealer who sells "Machine A". An email comes in from a trusted source and "Machine A" is learned as ham. And in millions of spams no one has ever mentioned "Machine A". Then one second later someone else we filter for mentions "Machine A". They are classified as ham based on that one match. And if there are several matches on the ham side that are *not matched* on the spam side then the message fingerprints can be added to the ham sets.

Examples:

Let's take 2 subject lines and see how this works.

"Meet hot Russian Brides Online!"
"I read an article about Russian Brides in a magazine"

A traditional spam filter using Bayesian or hard coded rules about "Russian Brides" might determine that only 1 out of 500 emails mentioning the phrase "Russian Brides" is a good email. Thus the second line would have points assessed against it in the classification process using these traditional methods.

Using the Evolution Filter the phrase "Russian Brides" is in both sets and therefore has no influence on the results. But the first subject matches these phrases in the Spam Only set.

“Meet hot”
“Meet hot Russian”
“Meet hot Russian Brides”
“hot Russian Brides Online!”
“Russian Brides Online!”
“Brides Online!”
“Online!”

The second subject matches these phrases on the ham only set that are never used on the spam set.

“I read an article”
“read an article”
“read an article about”
“about Russian”
“an article about”
“in a magazine”
“Brides in a”

So even though the phrase “Russian Brides” has no influence each subject hits either ham or spam many times where the same phrase was never used in the subject line in the opposite set. And the number of hits is significant enough just from these subjects to cause the fingerprints to be learned, and that’s just looking at the Subject attribute. When this is combined with testing all attributes the messages usually come out strongly on one side or the other.

In rule based systems one would not normally build a white list rule to to allocate points based on seeing the phrase “read an article about”. That’s where the Evolution Filter is different. It didn’t need to have that rule because since it is comparing to the infinite set of what is *not matched* on the other side, it dynamically create billions of rules automatically.

Friends and Family

Because the comparison method is based on matching one set and not matching the other the learning feedback system is a lot faster and has different characteristics than a traditional Bayesian filter.

In my previous example, “Machine A” has been learned as ham and never seen in spam. Someone sends an email inquiring about “Machine A” and “Machine B”. Because “Machine B” was never ever used in a spam then “Machine B” also becomes a blessed phrase. Anyone who uses “Machine B” in their email is passed as ham. (Unless spammers start spamming about “Machine B” which would revert it to neutral.)

Once a few phrases in one email message are matched to a new email then all the fingerprints of the new email are learned as ham. And the new fingerprints that are not already in the ham sets and are not in the spam sets become effectively new rules for identifying ham. The system learns how you talk, what you are interested in, and people in your life that are interested in the same things have their email passed and learned. Then their friends interact with them and the learning continues.

Consider this example, Email Subject Only. Brackets enclose a phrase learned. Parens enclose the match phrase from previous email learned.

“Do you want to get some [lunch today]?”
“Going to (lunch today) and [to see a movie] afterwards”
“I want (to see a movie) about time travel, [are you interested]?”
“(Are you interested) in [getting together] after work at my place?”
“If more people were (getting together) to make a better world we would have [less poverty.]”
“[Better education] leads to (less poverty.)”

Even though the subject wanders the matches keep feeding the learner because fingerprints are learned on one side the do not match fingerprints on the other side. In reality the subjects in the above example might produce 5 to 10 fingerprints each that were never used in spam.

People who communicate by email usually have some sort of relationship and they talk about things they have in common. It's the things that they have in common that causes not only the email to pass, but the differences in the messages to be learned as well.

Similarly on the spam side, there are only so many ways you can misspell Viagra, and the first time it catches a message with it deliberately misspelled then that spelling is learned and every spam that misspells Viagra the same way is caught. Traditional rule based system encourages spammers to misspell words so they don't match the rule. With the Evolution classifier the misspelling is what gives them up because people who send good email never misspell, for example, Viagra as Viiaggra.

Spammers also want you to do something. it's a business model that there are just so many scams out there and so many ways to describe these scams. So as the system learns these phrases that only spammers use then it's easy to detect new scams based on older but similar scams.

As the recursive learning continues it separates out what is essentially 2 different cultures and languages. Things that only people who send good email talk about and things that only spammers talk about. As these sets grow the accuracy increases and less and less messages go unrecognized.

Forgive and Forget

The power of the Evolution Classifier doesn't just lie in how it learns. It also lies in how it unlearns and forgets. Other prior art systems like this one are based on learning. This one has a feature based on unique ways of unlearning.

If a spam is learned wrongly as good and then the same spam comes in that is correctly classified as spam, if both are learned on both sides then it becomes neutral and although the filter doesn't classify it wrongly - it doesn't classify it at all.

But if we introduce forgetting then in situations where there is conflict instead of just learning the item on one side - we also unlearn it on the other.

Learning involves adding a new items fingerprints to the side it matches. Unlearning means deleting the matching fingerprints from the other side.

Deleting fingerprints from the other set includes a certain level of error. For example, an email is wrongly classified as spam and the user checks the spam folder and finds the good message and “releases” it. This causes the message not only to be learned as ham - but to be unlearned as spam.

The problem is - the subject contains the word “the” in it and the word “the” is unlearned from the spam set. Then a spam comes in with “the” in the subject line, it matches “the” on the ham side and nothing on the spam side so it would generate an indicator it was ham. But in this case there are hundreds of other fingerprints and those fingerprints are likely to find other keys that would classify it as spam, or at least neutral so it isn’t learned too. With a word like “the” however is a few seconds spam with “the” in it will be relearned on the spam side making “the” neutral again, as it should be.

In addition to caught errors a small percentage of known spam or ham will be included in the unlearning process. This process tends to kill off wrongly classified data so that it might be relearned in a better way. So a wrongly classified spam might prevent similar spams from being caught for a while, but the data it churned and everything that is digested is eventually excreted. Unlearning improves accuracy just as learning improves accuracy.

Use it or Lose it

The learning system also logs the time when a fingerprint was last accessed and if something isn’t refreshed by a new learn every now and then it is eventually forgotten. Out with the old to make room for the new. The ham and spam dates are stored separately so that a common fingerprint might turn into a spam only fingerprint over time. For example, if it’s December the “Merry Christmas” is neutral. But if it’s the end of January then “Merry Christmas” becomes spam only.

Claims in General

These are some of the claims, but not necessarily all of, I intend to use in the non-provisional patent.

The Comparison Claimed Method

The learner creates 2 corpi; one containing sets of fingerprints off all items learned to be ham, and the other containing sets of fingerprints of all items learned to be spam. (Figure 2) Within the corpi are sets of fingerprints; each set for each attribute being tracked. (Figure 1A, 1B, 1C) The message is tokenized and the tokens are combined into hundreds of fingerprints. (Figure 4) These fingerprints create test sets, one set for each attribute. (Figure 3A and 3B)

The fingerprints in the test sets are compared to the corpi sets using set functions as follows:

The test set is intersected with the ham set to create a subset of fingerprints representing fingerprints in the test set that have been seem in ham. The same thing is done with the spam sets to create a subset of fingerprints that have been seen in spam. (Figure 5A)

The spam subset is subtracted from the ham subset leaving a new subset which is the fingerprints in the test item that are only found in ham. Then the ham subset is subtracted from the spam subset to form a new subset of fingerprints only found in spam. (Figure 5B)

The sets of fingerprints of the test set “only found in ham” and “only found in spam” are compared and the set that is bigger or better determines if the item is ham or spam.

The two subset groups that are ultimately compared are test sets intersect ham sets minus spam sets for the ham subsets, and test sets intersect spam sets minus ham sets for the spam subsets.

This process is repeated for each attribute creating individual results for each attribute. Those results are combined to form an ultimate evaluation not only as to the classification but also a confidence level in the result. The confidence level is used to determine if the new item is to be learned by the system. (Figure 2)

The Fingerprinting Claimed Method

While it is likely prior are to use words or some word combinations as tokens, my method of using combinations or sequential combination to make fingerprints out of tokens is novel. (Figure 3A and 3B)

The Unlearner Claimed Method

While the learner is about adding new fingerprints to the matched corpi the unlearner is about removing matched token from the unmatched corpi. Matching tokens in the unmatched side are either deleted or reduced in stature. This causes information wrongly learned to expire and in some cases also triggers a relearning process where information can be relearned better.

Combining these Methods

Even if the above methods are not novel unto themselves the combined use of these methods is novel. (Figure 2 and Figure 4)

Author/Inventor
Marc Perkel
7498 Chestnut St.
Gilroy CA. 95020
415-987-6272
marc@perkel.com