Disaster Agnostic Alerting System for Mitigation

Ankush A Mishra[1], Ganesh M Bhat[2]

[1]Department of CSE, PESIT – BSC, Hosur Road, Bangalore, India
ankushmishra9[at]gmail.com

[2]Department of CSE, PESIT – BSC, Hosur Road, Bangalore, India
ganeshmbhat96[at]gmail.com

Abstract: ***We propose that in the event that a disaster [Disaster is defined as a serious disruption of the functioning of a community or a society involving widespread human, material, economic or environmental losses and impacts, which exceeds the ability of the affected community or society to cope using its own resources. [Definition from International Strategy for Disaster Reduction], man-made or otherwise, has occurred, the fastest means of communication to any citizen in any part of the world is the media. Nothing travels as fast as the current media today, which is a direct consequence of the Internet. This print and online media can thus be used as a source of the fastest warning system to any authority present in the world. We thus, give a system, which can in effect be ac- curate enough to predict whether a media within a time frame is related to a disaster or not.***

Keywords: Machine Learning, disaster management, warning systems, media

1. Introduction

With the increase of disasters, man-made or otherwise, there is a need for fast responses towards the mitigation of any disaster which occurs. Thus, before the mitigation, there comes an early warning system for alerting the authorities [Authorities is interchangeable with any organisations or individuals which provide support during times of crisis] on whether a disaster has occurred in any region governing their territory. Currently, the United Nations uses the Global Disaster Alert and Coordination System (GDACS) [http://www.gdacs.org] which collects near real-time hazard information on earthquakes, tsunamis, floods, volcanoes, and tropical cyclones.. This system being robust in implementation, still is restrictive to only 5 natural disasters, it doesn't address other calamities like in the case of Man-Made disasters we had Japan's Nu- clear Reactor Failure or even Bridges breaking or Terrorist attacks and so on. Since, GDACS is meant to be only an early warning system, it can't address these issues as it needs to predict them. To this problem, we want to provide another solution which can be used as a disaster agnostic alerting system providing fast action requirements to the authorities, in case of any disaster.

Thus, we say that on the event of a disaster, man-made or otherwise, has occurred, the quickest means of communication to any citizen in any part of the world, is media. This source of information can be tapped and be used as a source of the fastest warning system to any authority present in the world. We analyse recent news that exists based on the region provided, and use Support Vector Machines(SVMs) (See also [5] which specifies that SVMs have the highest accuracy for text categorisation) as a classifier, to predict whether the article is a disaster or not. Depending on the out- come of the classifier we then send out alerts in any form to respective authorities. We use the GDELT (Global Data on Events, Location, and Tone)[4] open dataset for news articles or construct one of our own. On this dataset, we then run the classifier and create our classifier model, which can then be used for further predictions on whether an article processed is disaster related or not.

2. Background

Early Warning Systems that allows individuals exposed to hazard to be able to reduce risk and increase likelihood of a faster response towards mitigation of the hazard. There are three major operation aspects of a functional system: Monitoring and Predicting, Communicating alerts and Respond- ing. Each of these aspects are related to each other. [4]

There are two types of hazards:
Ongoing and Rapid/sudden-onset: These include hazards such as accidental oil spills, nuclear plant failures, and chemical plant accidents. Geological hazards and hydro-meteorological hazards.

Slow-onset: These hazards take time to build up, and thus have small current but long-term and cumulative environmental changes. These hazards go unnoticed in their early stages but over time, can cause grave environmental issues. These include acid rain, soil pollution, climate change, radioactive waste, and so on.

The system which we propose is that the method of 'Monitoring and Predicting' cannot be used in all cases, like for example during the cases of Droughts, there are too many factors which might become the cause and effect of the disaster. Thus, for a better and faster mitigation, we monitor different kind of data which can be over a large range of hazards. The system can also function as a multi-hazard global warning system. It's generally realised that there is a need for a warning system [Early Warning system (EWS) and Warning System are slightly different. EWS requires the monitoring and forecasting of future events, whereas a Warning System need not be restricted to forecasting, and acts like a generalised alert system] in place at the global level, the ability to identify the risk and occurrence of hazards and to better monitor how prone the population is at any given period.

Although, there are several in place at a global scale, such

DOI: 10.21275/SR211127133130

as, World Food Programme's Humanitarian Early Warning Early Warning Service; AlertNet, the humanitarian information alert service by Reuters; ReliefWeb, the humanitarian information alert service by United Nations Office for the Coordination of Humanitarian Affairs (UN-OCHA)); GDACS (Global Disaster Alert and Coordination System), a joint initiative of the UN-OCHA and the European Commission Joint Research Center (EC-JRC). [4]

These systems, even though they cover a range of hazards such as ReliefWeb supplies information on earthquakes, tsunamis, severe weather, volcanic eruptions, storms, floods, droughts, cyclones, insect infestation, fires, and technological hazards and health; AlertNet provides information with relations to food insecurity and conflicts, whereas GDACS provides information on earthquakes, tsunamis, volcanic eruptions, floods, and cyclones. HEWS is used for earthquakes, severe weather, volcanic eruptions, floods, and locusts. Even with these vastly specific disasters a lot of mitigation value occurs, but for a global system, they should not be as restrictive, as shown. This problem is then addressed by our proposed system.

3. The System

A basic model of the system is illustrated in figure 1.

All these aspects of the system are handled on a centralised server, which can be running with access to either the local network or internet. A basic working of the system, happens as follows:

Once, the user using any device including a phone or an embedded system creates a HTTP GET query, with a query string to the server, with either the location in words or the latitude and longitude of the current user, or allowing the location API's to be used. The server then first checks for new news articles within a specific period of time, around 2-3 days old, on the RSS feeds of the news websites, and other disaster monitoring websites. These preselected articles are then saved to the disk. These articles are then sent to our classifier, which analyses all the articles from as input and gives a positive output if any of the websites are talking about disasters or not.

The way the classifier learns if an article is about disaster or not, is by depending on the training data given to it. If the training data given too has multiple articles which talk about Building Crashes, then the classifier will treat that as a positive output.

This positive output is then used by the server and is given as a response to the user on the web-page, or as a text file, which can be implemented as a HTTP GET request, so that embedded systems can use it as well. Other than a direct response, a specified number of other responses might be chosen by the user, when sending the initial request on the location.

Applications

This system being versatile, to the types of responses and inputs, makes it an important choice for being used as an alerting system. Like for example, other than the occasional end user wanting to know about the disaster in the near area, if this is needed to be implemented as a real-time, or almost real-time data on whether a specific disaster has occurred or not, then one can set up a system which sends a HTTP GET query string to the server, for a specific interval of time, like for example every 5 minutes. Using this as a base, any organisation can then implement an alarm system which acts as a client to our server, and processes the output produced to give any response to the individuals in the organisation.

Other than this, any user who has the ability to access the server, in any form possible can then make a personal device which acts like a custom made warning system.

4. Independent Features of the System

Having seen the system as a whole, we now analyse each specific feature that makes up the system. Our system consists of three major aspects:
1) Input from User
2) Processing through our Classifier
3) Output/Alerts to the authorities

All these aspects of the system should be accessible through our server URL, which depending on whether it is hosted locally or on the internet, allows the user to access the server. The server we tested this working on was "APACHE server". Once, we got it working, along with a suitable backend, to allow the running of our classifier model, and to give a response to the user.

1) Input from User

There are multiple ways to take the input location of the user. The first method is using the HTML5 Geolocation API, which if the user allows to be used, then is used to find the longitude and latitude of the user. Other than this method, we can use the Google Maps API to allow user input in the form of text. The user can provide the location to the web-page by either typing in the location, for example "Bangalore " or one can also input the latitude and longitude, for example "12.886, 77.664". This input is given on our webpage, which is accessed through the URL

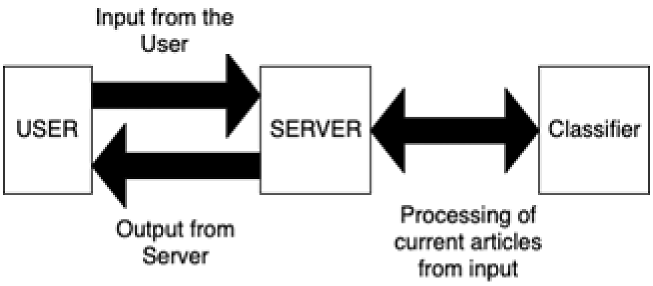

Figure 1: The three stages of the system, Input from the User, Processing through the classifier and Output from the User

2) Support Vector Machine (SVM) Classifier

We use Support Vector Machines as our classifier [1] [2]. It's also known as Large Margin classifiers or Optimal Margin Classifiers. They follow the Structural Risk Minimisation principle and along with that it also provides, high generalisation ability. They are universal leaders and by the use of a 'kernel trick' they can be used to learn polynomial classifiers, RBF's and neural networks. The optimal separating hyperplane and are the most difficult patterns to classify.

If N_S denotes the total number of support vectors, then for n training patterns the expected value of the generalisation error rate is bounded

$$\xi_n[error\ rate] \leq \ \xi_n\,[N_n]/n$$

Another feature of SVM's is that they allow capacity control, which makes it possible to take into account the amount of training data.[7] They exploit the minimising the maximum loss, and features like capacity tuning off the function and distinctness of solution. They are in a way extensions to Logistical Regression, though they rely on preprocessing the data to represent patterns in a high dimension.[8] All this can be seen from figure 2.

The input to the training algorithm is a set of m samples x_j and labels y_j associated with them, where

$$(x1, y1), (x2, y2), (x3, y3), \ldots, (x_m, y_m) \qquad (1)$$

Here we assume each pattern x_k has been transformed to $y_k = \psi(x_{k},)$. For each of the n patterns, $k= 1, 2, 3 \ldots, n$, we let $z_k = \pm 1$ a linear discriminant in an augmented **y** space is

$$g(y) = a^{\mathrm{t}}\, y \qquad (2)$$

where the expectation is over all training sets of size n, taken from their distributions. This determines the test data error in the samples.

For training the SVM, the first step is to choose non-linear ψ-functions that map the input to a higher dimensional space, with the help of a Kernel, $K(x, x)$, given by

$$K(x, x) = \textstyle\sum_{\mathrm{i}} \varphi\,(x)\ \varphi\,(x')$$

In the absence of such a kernel function, one might choose to use polynomials, Gaussians or yet other basis functions. Thus, the problem can be recast to minimising of the weight vector constrained by separation into an unconstrained problem by method of Lagrange's undetermined multipliers, and our goal of minimising a, we construct the functional

$$L\,(a, \alpha) = 1/2||a||^2 - \log_{k=1} n\, a_k[Z_k a^t y_k - 1]$$

where both the weight vector and the transformed pattern vector are $ao = w_O$ and $yo = 1$. Thus, a separating plane ensures

$$Z_k g\,(y_k) \geq 1 \qquad (3)$$

The goal in training a Support Vector Machine is to find the separating hyperplane with the largest margin; we expect that the larger the margin, the better generalisation of the classifier. The distance from any hyperplane to a (transformed) pattern y is $g(y)/\ a$, and assuming that a positive margin exists

$$Z_k{}^*\ g(y_k)/||a|| \ \geq\ 1\ k= 1, \ldots n \qquad (4)$$

The support vectors are the training samples that define and seek to minimise $L()$ with respect to **a** and maximise it with respect to the undetermined multipliers $a_k \geq 0$[3].

For, our specific application with news article, we use a mixture of articles taken from GDELT and some manual selection of articles. During our preprocessing step, we first remove all the HTML specific tags from each of these articles. After that, we use the Porter Stemmer [6], to normalise each of the words by stripping the suffixes, in the cleaned file. We then find the words whose frequency is greater than or equal to 50 in all the articles.

The vocabulary list, defines our k feature input vector $X \in R^{m\,x\,k}$ for all m training samples, with each input and the associated labels with them $y \in R^m$. We, then train our classifier using libSVM, or any other suitable library. The Kernel we use, is the Gaussian Kernel or the Radial Basis Function given by:

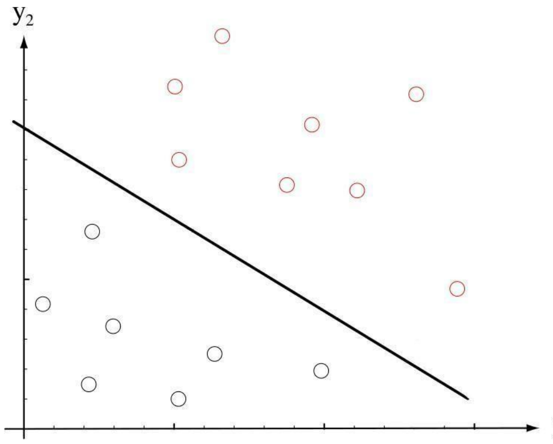

Figure 2: The linear decision boundary as generated by the optimal margin classifiers. Training a SVM consists of finding the optimal hyperplane with the maximum margin from the support vectors

$$K\,(x, x) = \exp(-||x - x||/2\sigma^2) \qquad (7)$$

where we give the σ value as $1/2$, i.e. γ value of 1. Upon training the SVM, we get around 0.045% error rate on the training data. Though, a greater accuracy could possibly be achieved using a Linear Kernel.

After our classifier is trained, we can then use this model to test if any news article is related to disaster or not. Specifically, the system will check the RSS feeds for any news aggregators and sources, on articles mentioning or specifying the region or the larger region enclosing the location the user specifies. These articles are then sent to our classifier model, which then predicts if any of those articles are disaster related, or not. We setup a threshold for the amount of articles for

Now, this system is disaster agnostic due to the fact that depending on the data we train the classifier with, it will treat that and related articles as disaster related. Say, for example we consider a specific subset of articles which talk about Terrorist attacks, and we say that these articles have the label of disaster. Thus, the classifier will then after training, consider articles on Terrorist attacks with some marginal errors, as disaster. Similarly, this can be extended to anything from Nuclear breakdowns to Terrorist Bombing. Thus, making it a very flexible and malleable system, independent of the type of disaster occurring.

Output to the user
Now, depending on the function output from the classifier prediction model, we can then choose what sort of output will be given. If the function returns a non-disaster output, then the user should be notified by either a HTTP GET of a file, which contains the output value, and also, on the web-page which the user accesses the server. If the output was a disaster output, then again the HTTP GET for the file out-put and also, a display on the webpage. Other than that, we also output to the various forms which the user specifies, like email to a group of people, sending out text messages, and so on.

5. Clients to the Server

Since, our system is based on a client-server model, we allow the ability for users to create clients to our server model. The client-server model, provides a lot of flexibility to the end user and removes a lot of restrictions which, if this was implemented on a system locally would be dependant on the server itself to handle specific input and output procedures, which no matter, how robust and complete, they may be, but will not be able provide for all cases. Thus, the client-server model is more applicable to a wider range of authorities and individuals, for their personal applications.

Our server, being based on the HTTP and HTTPS protocol, requires a client which can act as a client to HTTP/1.1 protocol as well, as defined by RFC 2068. For input it should give a HTTP GET request with a query string to the server. For example, a GET on http://server/?location=”Bangalore” or http://server/?lat=12.06&long=76.08. With a specific set location or any application with the help of Global Positioning System (GPS) can then be given as input to the server. The other way to input the location manually to the page, which produces a similar query string. But for automation, this would not be a feasible solution.

With respect to the output produced by the server after the processing has occurred on the server-side, it can occur in two ways. The first method is to display the output on the page we supply the input, along with the specific server-side alerting systems, such as sending out emails, or SMS’s. The second and more robust way is a HTTP GET response with a file containing the response of the classifier. This response is then analysed and depending on a positive or negative out- put, the alerting system can then perform a desired output specification.

6. Discussion

This system we proposed, we consider to be complementary to the current early warning systems in place al- ready. Most of the disaster systems, as mentioned, use multiple observing stations, such as Meteorological and earth resources satellites, aircraft’s and ground stations[4] monitoring different aspects of multiple factors which could be indicators towards the forecasting the future state of the specific disaster they monitor. These systems work in Real-Time and thus provide quick action and support but only for a specific major range of disasters. These systems in a way try to predict that in the near future will there occur, or try to monitor real-time data, which may change. Our system is not Real-Time, as it is media dependent. Being dependent on a source of information, which due multiple factors such as author biasing, and media exposure, limit our current system. As it is sensor-independent, the system is not restricted to specific disaster ranges, and thus can be applied to a large range of disasters, including man-made disasters. Though, it cannot be applied to ‘Slow-onset’ environment threats, whose progress is made much more slowly compared to the ‘Ongoing and rapid/sudden-onset’ environmental threats, need real time data and also depend on multiple factors, which may not be obvious at first glance. Thus, this system is suitable for an application only for ‘Ongoing and rapid/sudden-onset’ threats. Our system also suffers from problems of distance, where remote areas, with low media exposure might take time for the news to reach from and to the region. Thus, causing an insurmountable delay which can only be solved, with even more advanced communication and information technologies. Until that occurs, our system is restricted to major areas in the world.

7. Conclusion

In this paper, we proposed a system which can be used as a disaster agnostic alerting system, which can then be used as a client-server model. This model is then used as a method for a fast response system for mitigation for any sort of disaster, restricted to the ‘Ongoing and rapid/sudden-onset’ threats. Even though media does have its limitations for disasters affecting the society on a large scale, it should be sufficient to address most of the disasters the classifier is trained with.

References

[1] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[2] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[3] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. John Wiley & Sons, 2012.

[4] V. F. Grasso and A. Singh. Early warning systems: State-of-art analysis and future directions. *Draft report, UNEP*, 2011.

[5] T. Joachims. *Text categorization with support vector machines: Learning with many relevant features*. Springer, 1998.

[6] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[7] A. J. Smola, B. Scho¨lkopf, and K.-R. Mu¨ller. The connection between regularization operators and support vector kernels. *Neural networks*, 11(4):637–649, 1998.

[8] V. N. Vapnik and V. Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.