Satellite Anomaly Prediction using Survival Analysis and Machine Learning

Christopher Naughton, CS229 Final Project

1. Introduction

We live in an age of information where many facets of human life are influenced by the flow of data enabled by our space infrastructure. From commonplace utilities such as weather predictions and GPS, to critically important applications such as international communications, many incredible tools depend upon the proper functioning of satellite systems. Unfortunately, the space environment is inhospitable, even for machines. Energetic particles and micro-meteoroids continuously bombard the Earth. Although our atmosphere and ionosphere protect the Earth's surface, satellites continuously are exposed to these hazards. Many of these spacecraft threats are not random but are the result of a constant flux (e.g. meteoroids), or a cyclical flux (e.g. energetic particles from the sun) of dangerous particles, and therefore should be predictable. Since even minor anomalies can disrupt a spacecraft's control system enough to inhibit proper functioning, agencies such as NOAA and NASA have recorded hundreds of anomalous behaviors of different spacecrafts orbiting the earth to correlate these incidents with known space-weather phenomena [1] [2]. Though research has been done to analyze *why* spacecraft fail, this data has not been applied to predict *when* spacecraft will fail. In this paper we input space weather conditions and satellite attributes into linear regression, support vector regression, and modified Naïve Bayes predictors to predict the time until an anomaly occurs.

2. Related Work

Wilkinson et al. first used the anomaly data set to find correlations between space weather conditions and the observed satellite anomalies [2], but never tried to predict future anomalies based on the data. Although no previous work has tried to predict satellite anomalies, there have been many attempts to use machine learning to predict "Time to Event" problems. As explained by Wang in his survey [3], this form of problem has been traditionally solved using the subfield of statistics known as Survival Analysis and has had much attention from the machine learning community in recent years. Bellazzi and Zupan applied both a Naïve Bayes classifier and Decision Trees to predict a patient's long-term clinical status after surgery [4]. A study by Fard also successfully applied Bayesian methods to limited versions of several medical datasets [5]. A similar paper by Wolfson et. al focused on using Naïve Bayes with a prior determined by basic survival analysis to successfully predict cardiovascular risk [6].

3. Dataset and Features

The final dataset has 6 features for 726 anomalous events over 10 different satellites. The unprocessed data was taken from a NOAA database and consists of ~5000 anomalies on ~200 different spacecraft between 1976 and 1994 [1]. This data only includes a spacecraft name and a series of timestamps indicating the day that the anomaly occurred. Table 1 below shows an excerpt from this data set.

Satellite Name	GOES-6	GOES-6	GOES-7	GOES-7
Anomaly Date	1/28/1994	3/26/1994	9/14/1987	9/16/1987

Table 1: Sample Data from NOAA database [1]

To treat this data using a Survival Analysis approach, I split each anomaly into an independent "patient" and used the time between anomalies as a my "time to event" label. This is shown in Figure 1 below. The final processed data consists of a satellite name and the number of days until an anomaly occurred. Most anomalies occurred between 1 and 40 days of one another. I also include the time that the observation period started, which is the date of the previous anomaly.

Figure 1: Each anomaly was treated as an independent event

Unfortunately, most of the spacecraft names were classified or had gaps in their anomaly observation periods. After cross-referencing another satellite database [7] and removing the satellites with observational gaps, the data was reduced to 726 anomalous events occurring to 10 different satellites.

Each label has 6 features:

- *Starting Month of Observation*: determines the earth's relative position around the sun
- *Sunspot Number*: measures the solar activity at the start of observation, taken from [8]
- *X-ray flux:* count of high energy particles at the start of observation, taken from [9]
- *Mass:* size of the spacecraft
- *Perigee:* closest point to earth in spacecraft orbit, similar to "altitude"
- *Inclination:* tilt of orbit; determines path the satellite takes around the earth

This makes the final data matrix of dimension 726 x 6. The first 3 features act as a measure of space conditions at the start of observation. Due to the relatively old dates of some of the anomalies (1970s and 1980s), not much space-weather data is available. Additionally, the limited data required a *lot* of processing to interpret. The restricted nature of these features means that many space phenomena (such as meteoroid/debris impacts) are not being accounted for. The last 3 features account for differences in the type of satellite being observed. Note that since the data only observed 10 different satellites, the last three features are very sparse.

4. Methods

This paper compares traditional linear regression, support vector regression, and a modified Naïve-Bayes survival-analysis approach to predict the time to anomalous event.

4.1 Linear Regression: For preliminary results, I used the pre-programmed sci-kit learn linear regression predictor [10]. This predictor models the output as a linear function of the data. In vector equation form, the output hypothesis is:

$$h_\theta(x) = \vec{\theta}^T \vec{x}$$

where θ is a vector of weights found to minimize the error between the h(x) and the true value. These weights are typically calculated by performing the least mean squares algorithm on a set of training data. This model is limited as it can only describe linear trends.

4.2 Support Vector Regression: As the next step, I applied the pre-programmed sci-kit learn support-vector regression (SVR) predictor. SVR was chosen because it has been very successful as a nonlinear regression model for small sets of data [11]. This model tries to predict values that maximize the confidence in the data (known as the geometric margin) while minimizing the error. Mathematically, SVR solves the optimization problem:

$$\min_{W,b} \quad \frac{1}{2}\|W\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$$

Subject to the constraints:

$$y_i - (w^T\phi(x_i) + b) \le \epsilon + \xi_i$$
$$(w^T\phi(x_i) + b) - y_i \le \epsilon + \xi_i^*$$
$$\xi_i, \xi_i^* \;\; \ge \;\; 0, \;\; i = 1 \dots n$$

W is the weight vector (like θ in linear regression), b is the intercept term, $\phi(x)$ is a kernel to capture nonlinearities, ε is the desired error max error, ξ and ξ^* account for points outside the desired max error bound in the cost function, and C is a parameter that balances minimizing error (represented by the ξ terms in the cost function) and maximizing confidence (represented by the $\|W\|^2$ term).

The model in this paper uses a radial basis function kernel to capture nonlinear effects, and sets the hyperparameters to C = 10, ε = 0.1. Although larger C parameters (~100) fit the training set very well, the model then suffered from overfitting and did not generalize well to other data.

4.3 Modified Naïve Bayes: These time-to-event problems are often solved using a subfield of statistics known as Survival Analysis, in which a survival function S(t) is constructed to predict the probability of the event *not* occurring before time t [12]. Since Bayesian methods are commonly applied to survival analysis problems in the literature, I use a Naïve Bayes method like Wolfson et al [6]. Naive Bayes is a binary classifier, and so I use it to predict whether an event has or has not occurred given a time value and a feature vector. The true label of times is a series of discrete values, so to form a pseudo-regression model I iterate through every possible time value and determine at what time the anomaly is first likely to occur. Using the assumption that the features are all independent of one another, the classic Naïve Bayes formulation of Bayes rule is:

$$p(y=1|x) = \frac{\left(\prod_{j=1}^{d} p(x_j|y=1)\right) p(y=1)}{\left(\prod_{j=1}^{d} p(x_j|y=1)\right) p(y=1) + \left(\prod_{j=1}^{d} p(x_j|y=0)\right) p(y=0)}$$

If we define y = 1 to indicate that the event has not occurred, then for a given time t:

$$p(y=1) = S(t) \qquad\qquad p(y=0) = 1 - S(t)$$

Where S(t) can be thought of as the "prior" survival function. This is the estimated survival function based on purely on the time-to-event data and is independent of features. If the dataset has *n* independent observations with *m* recorded events before time *t*, then survival analysis typically predicts S(t) by splitting the event data up into time intervals *m* and applying the Kaplan Meier estimator [6] [12]:

$$S(t) = \prod_{i=1}^{m} \frac{r_i - d_i}{r_i}$$

Where r_i is the number of observations in which the event has *not* occurred by t_i, and d_i is the number of events that occurred at t_i for i = 1,…,m.

For the rest of the terms in Naïve Bayes, I model $p(x_j \mid y=1)$ and $p(x_j \mid y=0)$ as a series of gaussians whose parameters depend on the input prediction time. My code works as follows:

1) Split data into training and validation sets
2) For each possible prediction time t:
    a. Use training set to calculate S(t), as well as the means and variances for $p(x_j \mid y)$
    b. Use validation set to calculate probability of an event occurring within time *t*
    c. If the anomaly has occurred for the first time, set the predicted TTE = t

5. Results

The total dataset of 726 events is split into training and test datasets using k-fold cross validation with k = 10. This means that the dataset is partitioned into 10 equal sections, with 9 sections serving as the training set and 1 section serving as the test set. The predictor is trained and run 10 times, with each partition acting as the test set once. The mean error from each section is then averaged together to form an overall mean error for the model. For an output prediction vector $\hat{y}$ and a true label vector *y*, both of length n, the mean error is:

$$Error = \frac{1}{n}\sum_{i=1}^{n} \frac{|\hat{y}_i - y_i|}{y_i}$$

Table 2 below shows the results for each model. Unfortunately, no model adequately predicts the time to a satellite anomaly.

Model	Train Set Mean Error	Test Set Mean Error
Linear Regression	455%	475%
Support Vector Regression	65%	190%
Naïve Bayes	100%	167%

Table 2: All three models fail to accurately predict the time to anomaly

5.1 Linear Regression: Linear regression performs the worst on both the training and test sets, which is expected since it is the simplest algorithm, and is working with a sparse and noisy dataset.

5.2 Support Vector Regression: By comparison, support vector regression (SVR) performs much better. In fact, out of all three algorithms it does the best job fitting the training set. This is most likely because any trends in the data are inherently nonlinear. Unfortunately, the model suffers from overfitting and does not predict the test set very well. As discussed in Section 4.2, the hyperparameter "C" is adjusted to mitigate overfitting. Figure 2 below plots the predicted label against the true label. Predictions with no error fall along the dashed black line.

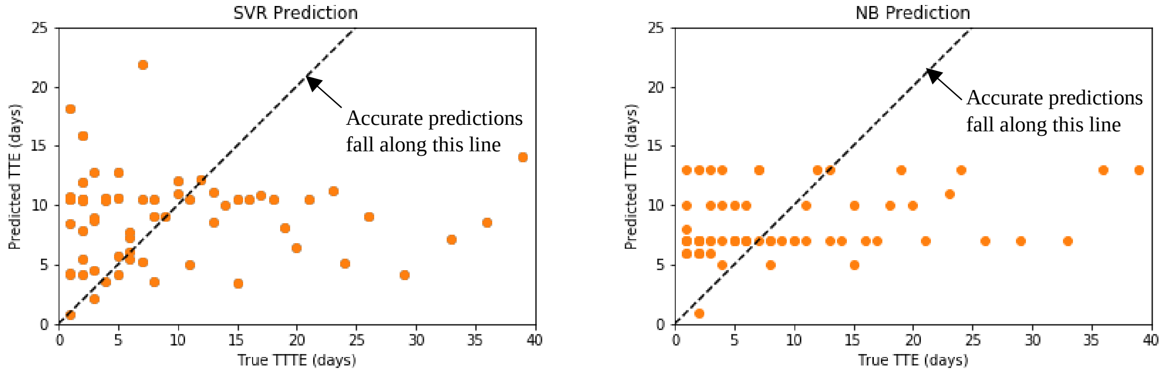

Figure 2: Comparison of SVR and NB predictions for a single test set

5.3 Naïve Bayes: Although the Naïve Bayes formulation does not fit the training set very well, it performs marginally better on the test set than SVR. Overall though, the distribution of predictions looks very close to the SVR output, as Figure 2 above illustrates. This poor performance is probably due to 1) The features are not distributed as gaussians, and 2) The features are not all independent, thus violating the fundamental Naïve Bayes assumption. Additionally, it is very likely that the features do not capture all the effects present in the space environment that are related to these anomalies.

6. Conclusion

In conclusion, none of the three algorithms accurately predict the time to satellite anomalies posed as a Survival Analysis problem. Although support vector regression appeared to fit the training data, it did not generalize well to the other test cases. The poor performance of the models is most likely due to the sparsity of the data set and the effect of missing factors. Aspects such as spacecraft geometry, space debris impacts, and earth's radiation belts contribute to spacecraft anomalies but are not properly captured by the modeling approach due to the lack of data during the time period of anomaly observation. Over the past 20 years, NOAA has greatly improved the diversity of space weather observations. Therefore, future work on this project would be to gather more recent records on satellite anomalies with a greater diversity of satellite features than those explored here.

Find my code online at: https://github.com/cwnaught/CS229-Final-Project.git

Contributions:

As the only member of my team, I performed all the data collection, processing, modeling, and testing.

References

[1]: Wilkinson (1993). *Satellite Anomalies*, electronic dataset, NOAA

[2]: Wilkinson et al. (1991). *TDRS-1 Single Event Upsets and the Effect of the Space Environment*. IEEE Transactions on Nuclear Science, Vol 38, No. 6

[3] Wang, P., Li, Y., & Reddy, C. (2017). *Machine Learning for Survival Analysis: A Survey*. ACM Computing Surveys.

[4] Bellazzi, R., & Zupan B. (2006). *Predictive data mining in clinical medicine: Current issues and guidelines*. International Journal of Medical Informatics

[5] Fard, M., Wang, P., Chawla, S., & Reddy, C. (2016). *A Bayesian Perspective on Early State Event Prediction in Longitudinal Data*. IEEE

[6] Wolfson et al. (2015). *A Naïve Bayes machine learning approach to risk prediction using censored, time-to-event data*. Stat Med

[7] Kelso, T. (2019). *Satellite Catalog*, electronic dataset, Celestrak

[8] *GOES SEM Data*, electronic dataset, NOAA

[9] *Sunspot Numbers*, electronic dataset, NOAA

[10] Pedregosa et al (2011). *Scikit-learn: Machine Learning in Python*, JMLR 12, pp. 2825-2830

[11] Smola, A., & Scholkopf, B. (2002). *A tutorial on support vector regression*. Statistics and computing, vol 14.

[12] Hosmer D.W., Lemeshow S., and May S., *Applied Survival Analysis: Regression Modeling of Time- to- Event Data*, Wiley, 2008.