Learning Migraine Triggers

CS9860 – Advanced Machine Learning

Mike Ghesquiere

1 ABSTRACT

Chronic migraines affect an estimated 1.4% to 2.2% of the population. These frequent migraines can cause significant pain and disruption to one's life. A dataset composed of one migraine sufferer's daily journals over the course of 16 months was analyzed for possible environmental causes or triggers to his migraines. Several models are considered and compared for migraine prediction. From these models, the most impactful triggers were extracted. Orthogonal Matching Pursuit was found to be a good model for both prediction and extraction. This model may prove useful in other self-recorded medical application. A novel approach to ordered-class classification is also proposed for possible applications in rating systems.

2 Introduction

2.1 Background

The cause of migraines, although not fully understood, is generally thought to be a combination of both genetic and environmental causes. For chronic migraine sufferers, knowledge of genetic causes provides very little practical, short-term relief. However, knowing the environmental triggers of one's migraines can be very useful in decreasing the rate, duration and severity of migraines. In addition, being able to predict the onset of a migraine could allow pre-emptive treatment through medication.

There exist known, common triggers but of course these vary between individuals. Migraine sufferers are often prescribed with tracking their migraines on a daily basis. From these journals, one hopes to find a pattern between environmental effects and their personal migraines.

2.2 Acquisition of Dataset

Sometimes finding these patterns are not so easily found. Around May 2013, I was approached by a family friend and chronic migraine sufferer regarding a journal which he had been keeping for the past 16 months. At the time I had no machine learning background but this course has enabled me to begin to intelligently analyse this dataset.

Stored as an excel spreadsheet, the dataset contained 490 rows of consecutive days of migraine severity. With the exception of five rows, each contained 24 columns of possible triggers along with the date and sporadic notes. Migraines were recorded as 0 (no migraine), 1 (mild migraine), or 2 (severe migraine). Each trigger was recorded as an integer from 1 (no presence) through 4 (strong presence). A full list of triggers can be seen below:

Cold air exposure
Physical exertion
Post-stress letdown
Smoked or cured meat
Citrus fruit or juice
Chocolate
Liquor or spirits
Dehydration
Nightshade vegetables
Overslept
Stress
Bananas
Beer
Red wine
Loud sounds
Changing weather
Perfume or strong odors
Lack of sleep
Missed a meal
Caffeine
Aged or blue cheese
Bright or flashing Lights
Sugar and Sweets
Hot and humid weather

Figure 1 List of all features included in the dataset.

2.3 Domain Knowledge and Pre-processing

Particularly since the subject tended to have migraines in the morning while the journal entries were made in the evening, triggers on the day of the migraine will not likely be the most important features. Thus historic days was left as a parameter to be tuned. Given a set of integer offsets: $\delta_1, \delta_2, \dots, \delta_k$ we then construct a feature vector $\widetilde{X_i}$ by:

$$\widetilde{X}_i = (1) \parallel X_{i-\delta_1} \parallel X_{i-\delta_2} \parallel \cdots \parallel X_{i-\delta_k}$$

Where ‖ denotes concatenation and X_j is the original unbiased vector. Note that a bias (1) is also added at this stage. Also, note this will also marginally reduce the number of samples:

$$\tilde{n} = n - \max(\{\delta_i | \delta_i \geq 0\}) + \min(\{\delta_i | \delta_i \leq 0\})$$

In addition to these 24 features, day of the week and time of year were also suspected to be possible triggers. Given that both are cyclic, any linear decision boundary would be heavily dependent on the codification used. To avoid this, I used a "one-hot" encoding for each. That is, each feature vector actually contained seven values corresponding to day of the week. Six of these would be zeroes while one would contain a one. Similarly month was encoded into 11 'cold' values and one 'hot' value. In total this adds 19 features to our input vectors which were not duplicated when including multiple days' worth of historic data as this would not contain any additional information. This leaves us with a final vector of length $24k + 20$. Preliminary findings suggested increasing the length of δ beyond $\delta = (1,2)$ would have no significant improvement in performance. $0 \notin \delta$ was a conscientious choice to avoid any possible triggers which would actually be caused by the migraine itself.

3 Classification Models

3.1 Ordered Classes

Given that the output features are exactly 0, 1, or 2, it is natural to think of this as a classification problem. However, most multi-class classification schemes take a "one-vs-all" approach which does not make sense in this context. This would create 3 classifiers:

(1) No migraine vs. Mild or Severe migraine
(2) Mild migraine vs. No or Severe migraine
(3) Severe migraine vs. No or Mild migraine

One would then classify a sample z using:

$$mclf_{naive}(z) = i \quad s.t. \quad clf_{(i)}(z) \text{ } is \text{ } maximal.$$

Although divisions (1) and (3) make logical sense, it is not expected for there to be any sort of cluster which contains non-migraine days as well as severe migraine days yet manages to exclude mild migraines. That is, $clf_{(2)}$ is *expected* to be very weak giving no better than chance predictions. Another formulation of this would be that severe migraines can be thought of as a *strict subclass* of mild migraines. Thus, we define our multi-class classifier as simply:

$$mclf_{ordered}(z) = clf_{(1)}(z) + clf_{(3)}(z)$$

Given perfect classifiers $clf_{(1)}$ and $clf_{(3)}$ this function will behave as desired (i.e. yielding 2 if severe, 1 if mild, or 0 if no migraine). I have no proven theoretical or statistical reasoning for non-ideal cases. To ensure that class 2 was indeed a subset of class 1, for testing I also used:

$$mclf_{test}(z) = clf_{(1)}(z) + 2clf_{(3)}(z)$$

By ensuring that $2 \notin Im(mclf_{test})$, it follows that the positive class of $clf_{(1)}$ contains the positive class of $clf_{(3)}$. This is a simple litmus test that can allow some classifiers to be dismissed immediately.

3.2 Generalization of Ordered Classes

This procedure can be generalized for any ordered sequence of classes. The most immediate extension would be for rating systems which use "star ratings". However it can technically (although it is not recommended for reasons I will discuss below) to also extend to general regression problems, even over unbounded ranges.

Given a strictly increasing sequence $\sigma_1, \sigma_2, \dots, \sigma_m$ we define:

$$\tilde{y}_i^j = \begin{cases} 0 & y_i < \sigma_j \\ 1 & y_i \geq \sigma_j \end{cases}$$

for $j \in \{1,2,\dots,m\}$ and all labels y_i in our dataset. Next, we train m binary classifiers such that:

$$clf_{(j)}(X) \mapsto \tilde{y}^j$$

Setting $\sigma_0 = 0$, we define the multiclass classifiers:

$$mclf_{ordered}(z) = \sum_{i=1}^{m} (\sigma_i - \sigma_{i-1})\, clf_{(i)}(z)$$

$$mclf_{test}(z) = \sum_{i=1}^{m} 2^{i-1} clf_{(i)}(z)$$

Again for the sake of validation, one should ensure the following holds:

$$Im(mclf_{test}) \subseteq \{0\} \cup \left\{2^j - 1 \,\middle|\, 1 \leq j \leq m\right\}$$

Example 1: The formulation from section 3.1 is equivalent to using the sequence (1,2).

Example 2: On a 5 star rating system, one could use the sequence (1,2,3,4,5).

In some ways this approach can be compared to the strategy of Lebesgue integration: slicing the image to create a stack of nested neighbourhoods. This also illustrates one downfall to this approach: poor extrapolation for outlier output values. Consider a one dimensional dataset whose output values are the identity.

Even in the best case (shown Figure 2), $mclf_{ordered}$ can make no extrapolations from the training points to values outside the original range (i.e. [0,1] in Figure 2). This will happen regardless of the choice of sequence to divide the range. With shorter sequences (or σ_i's poorly distributed), the steps will become more exaggerated. However, no amount of tuning the sequence will get rid of the plateaus at either end. Of course, for applications in which the output is bounded (e.g. rating systems), this problem disappears.

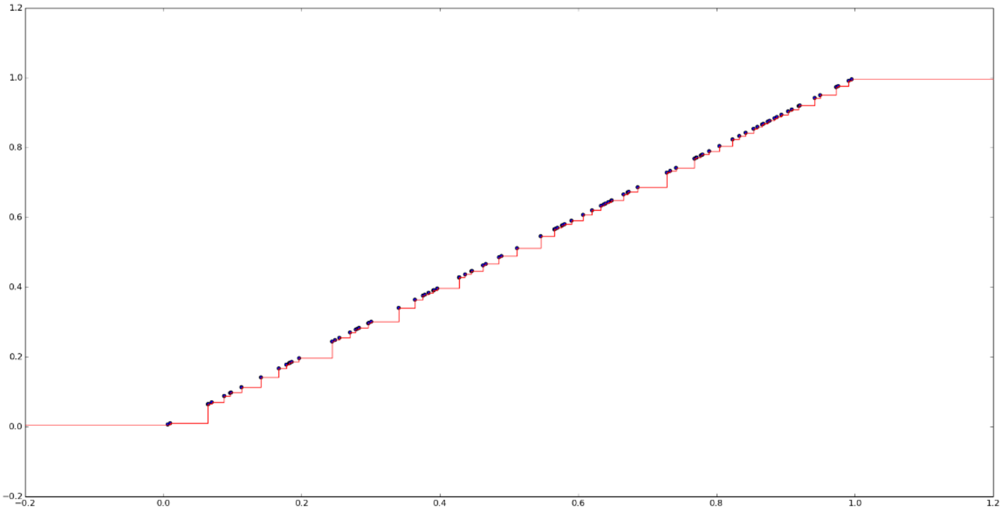

Figure 2: 100 points with identity output in blue. $clf_{ordered}$ *is shown in red using the original points as the* σ *sequence.*

3.3 Classification Results

Since the classes are unevenly weighted, simple accuracy of a model can be deceptive. For example, the zero function (classifier that always guesses '0') has 68.7% accuracy for the dataset. In addition, the distinction between mild and severe migraines is quite subjective. Errors between these two classes should be less heavily weighted than being confused with no migraine.

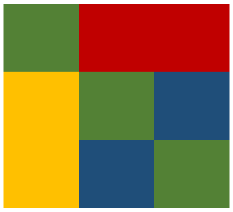

Figure 3 A confusion matrix partitioned into four scoring areas. Sum over the green region gives accuracy, red gives the number of false positives, yellow: the number of false negatives and blue yields 'false severity': the number of cases where a mild migraine is mis-classified as severe or vice versa.

Figure 3 shows the breakdown of a confusion matrix into four categories. We still desire accuracy to be maximal, however, false severity is less of an issue than false positive or false negative. Depending on application, relative penalty for false positive/negative/severity may vary but it is my opinion that false severity should be treated as "half right". That is, comparing relative performance among classifiers by the following key:

$$accuracy + 0.5(false\ severity)$$

	Accuracy	False Positive	False Negative	False Severity
Zero function	68.7%	**0.0%**	31.3%	0.0%
Density-weighted Random	52.6%	21.5%	21.5%	4.4%
$mclf_{ordered}$				
Logistic Regression	64.8%	10.6%	21.7%	2.8%
Linear SVC	67.3%	8.3%	21.9%	2.5%
Ridge Classifier	65.6%	10.4%	21.5%	2.5%
SVC – rbf kernel	68.5%	0.2%	31.3%	0.0%
SVC – polynomial kernel	65.8%	7.5%	24.8%	1.9%
kNN (k=10)	67.1%	4.3%	27.5%	1.0%
kNN (k=20)	67.5%	2.3%	29.6%	0.6%
kNN (k=30)	68.1%	1.4%	29.8%	0.6%
kNN (k=40)	**68.9%**	0.4%	**20.4%**	0.2%
$mclf_{naive}$				
Logistic Regression	67.3%	7.2%	23.6%	1.9%
Linear SVC	68.7%	5.4%	24.6%	1.2%
Ridge Classifier	68.7%	6.0%	23.8%	1.5%
SVC – rbf kernel	68.7%	**0.0%**	31.3%	0.0%
SVC – polynomial kernel	67.5%	3.5%	27.7%	1.2%
kNN (k=10)	68.3%	2.1%	29.2%	0.4%
kNN (k=20)	68.3%	0.6%	30.8%	0.2%
kNN (k=30)	68.7%	0.2%	30.8%	0.2%
kNN (k=40)	68.5%	0.2%	31.3%	0.0%

Figure 4 Comparison of various classifiers (with and without transformation from Section 3.1). No model gave results significantly above chance results.

As the reader may have noticed from Figure 4, classification results were not promising. At best, classification was achieving slightly greater than chance results. Using $mclf_{ordered}$ instead of $mclf_{naive}$ (as detailed in section 3.1) gave a *decrease* in accuracy however an improvement to false negative rates. False positive and severity rates were also increased. This can be traced to the fact that $mclf_{ordered}$ in some ways functions as a logical 'OR' of two classifiers, increasing the sensitivity of both. A 'balancing' between false positive/negative is not undesirable

4 Regression Models

4.1 Label Weightings

On the other hand, the problem could simply be thought of as a regression problem. This eliminates any need for $mclf_{ordered}$ or similar constructions. We can simply interpret the labels as the relative strength of a migraine. The original labels may not be completely trustworthy however. Although the data is labelled as 0, 1 or 2, it is quite natural to ask whether or not a severe migraine is *exactly* twice as strong as a mild one. Unrestricted re-labelling of the data points would leave three free variables to optimize. But under reasonable assumptions, we can constrain the optimization down to one variable.

Let $\phi: \{0,1,2\} \to \mathbb{R}$ be the function mapping original labels to new output labels. It should be uncontroversial to say that ϕ should preserve order (that is, $\phi(0) \leq \phi(1) \leq \phi(2)$). Moreover, this inequality should be strict so as to not disregard given information. Next we assume that any regression classifiers are invariant to affine transformations. If scaling and shifting the output features does not affect the trained model, any θ can be transformed into ϕ' such that $\phi'(0) = 0$ and $\phi'(2) = 1$. Thus we only need to 'move' $\phi'(1)$. We define $\phi_\alpha : \{0,1,2\} \rightarrow \mathbb{R}$ as follows:

$$\phi_\alpha(x) = \begin{cases} 0 & x = 0 \\ \alpha & x = 1 \\ 1 & x = 2 \end{cases}$$

for $0 < \alpha < 1$. Note that $2\phi_{0.5} = id$. As α approaches 1, mild migraines will be more likely to be confused with severe migraines. Conversely as α approaches 0, mild migraines will more likely be confused with severe migraines.

To optimize α, several classifiers were tried. There was no significant improvement in the separation of outputs for different classes for any of the classifiers. There is a subtle taper for $\alpha < 0.2$ (most exaggerated for using rbf SVR with $\alpha < 0.05$). However for $\alpha \geq 0.2$ very little variance is created. The trend caused can be mostly attributed to the heavier weighting of mild migraines increasing the average mean for the trained model. Overall, varying the label weights proved to be unsuccessful in improving results; thus $\phi_{0.5}$ was used.

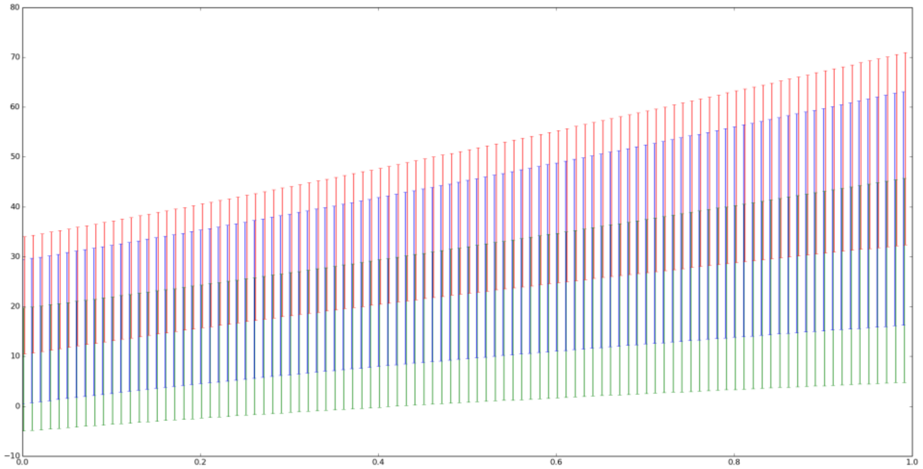

Figure 5 Along the horizontal axis, α is varied while linear regression predictions are shown on the vertical. Mean and standard deviation are shown for each 'no' (green), 'mild' (blue), and 'severe' (red) migraines.

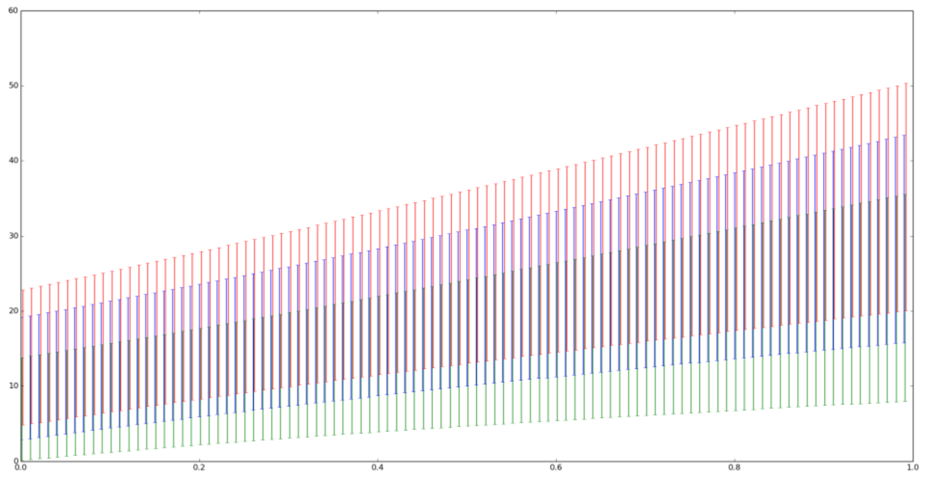

Figure 6 Same as Figure 3 but for kNN (k=20)

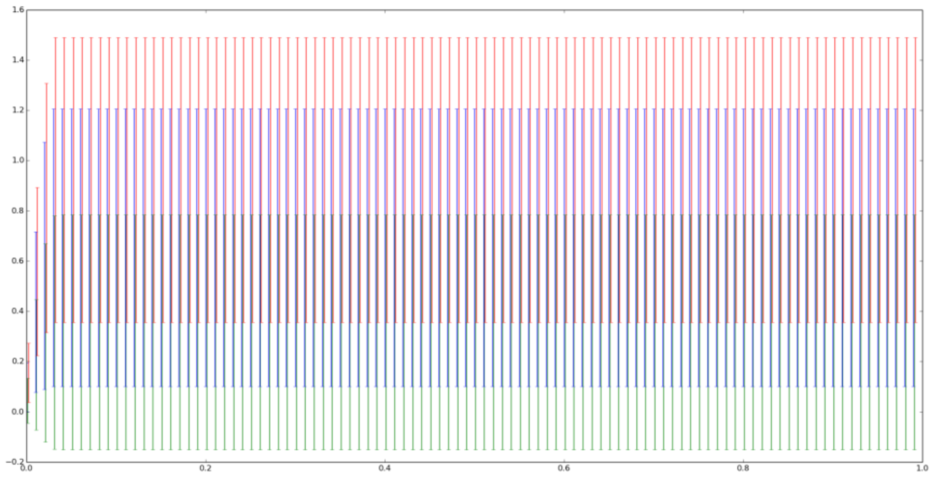

Figure 7 …and again, this time for SVM Regression using an rbf kernel.

4.2 Regression Results

Before comparing between models, some classifiers had parameters to be optimized. To optimize the variance between classes, the following function was used (plotted in black dots):

$$\sum_{i,j\in\{0,1,2\}} \left(\mu_i - \mu_j\right)^2 - \sum_{i\in\{0,1,2\}} \sigma_i^2$$

This function was designed to be optimal for when classes have a large difference between means (i.e. $\mu_i - \mu_j$) but a small standard deviation per cluster (i.e. σ^2). It was partially successful, although in practice seems to overemphasize classifiers with low standard deviation but low difference of means.

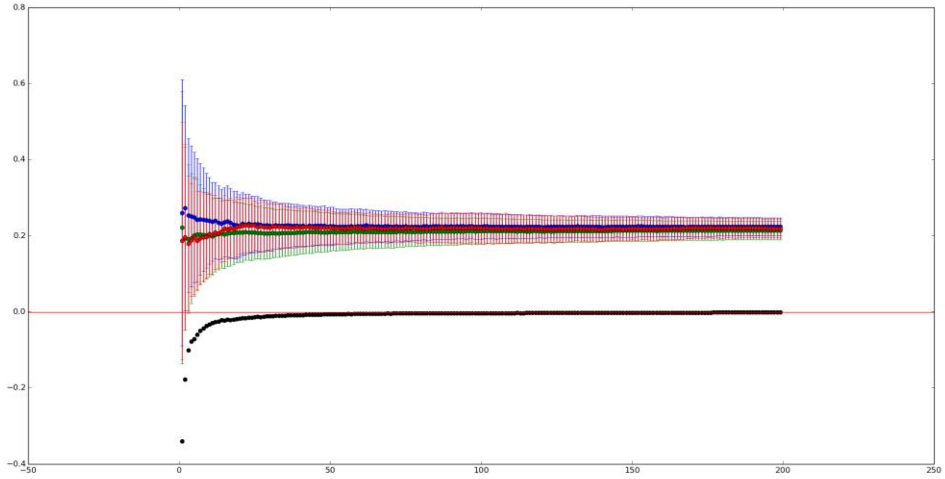

Figure 8 A kNN classifier was trained with varying k (along horizontal axis). Although increasing k *beyond 60 provides no effective increase in measured performance, severe migraines 'drift' to become closer to no migraine than mild.* $20 \leq k \leq 60$ *is recommended.*

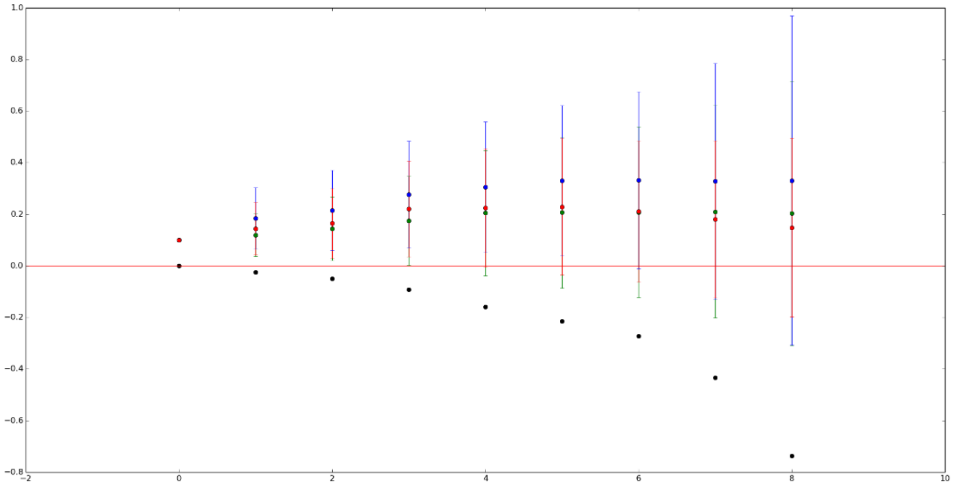

Figure 9 Support Vector Regression using a polynomial kernel of varying degrees. Degree 3 is recommended

Although initial optimization was not looking promising, two things are quite striking already from Figure 8 and Figure 9. Mild migraines (blue) tend to have a higher mean prediction than severe migraines. Particularly evidenced in Figure 8, mild and severe migraines have near identical output distributions suggesting that the primary factor for determining severe vs. mild migraine is not included in this set of triggers. This reinforces the school of thought that migraine triggers are an additive 'switch'. That is, once a certain threshold of triggers is reached, a migraine occurs. Migraine severity is independent of these triggers. However, designing a study to positively prove this claim would be very difficult if not impossible; it could always be argued that a particular, omitted feature would distinguish migraine severities.

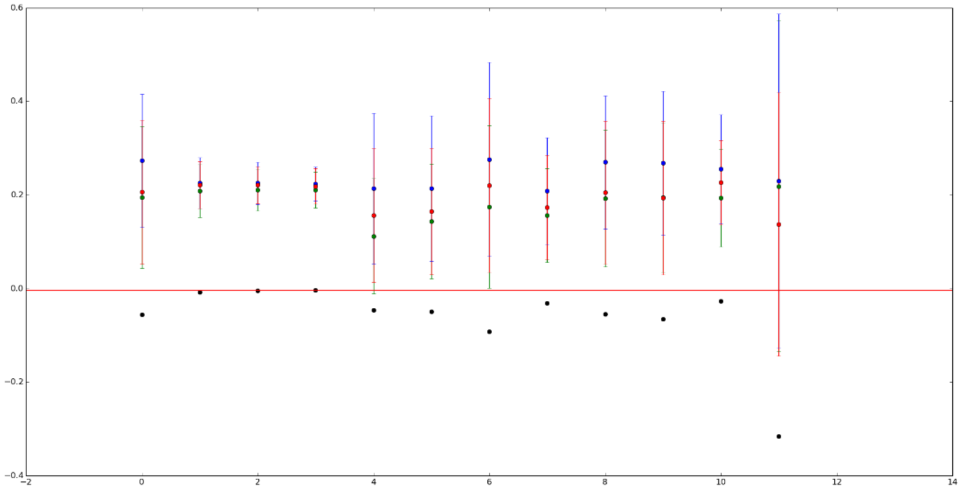

Figure 10 A comparison of performance for various regression models. (0) Linear Regression, (1) kNN (k=40), (2) kNN (k=60), (3) kNN (k=80), (4) SVR – linear kernel, (5) SVR – polynomial (deg=2) (6) SVR – polynomial (deg=3), (7) SVR – rbf kernel, (8) Linear ridge regression, (9) LARS, (10) Orthogonal Matching Pursuit, (11) Decision Tree. Best performance was given by the kNN regressors followed by OMP.

After looking into Orthogonal Matching Pursuit (OMP) further, it became apparent that this model was perfectly suited to this problem. It was designed to compensate for not only noisy measurements (journal features are highly subjective) but also high-dimension sparse signals (most features, especially when incorporating several days' triggers, are expected to be unrelated). Furthermore, OMP has easily interpreted coefficients for diagnosis which are themselves sparse. The sparse coefficient vector makes it trivial to accurately extract important features.

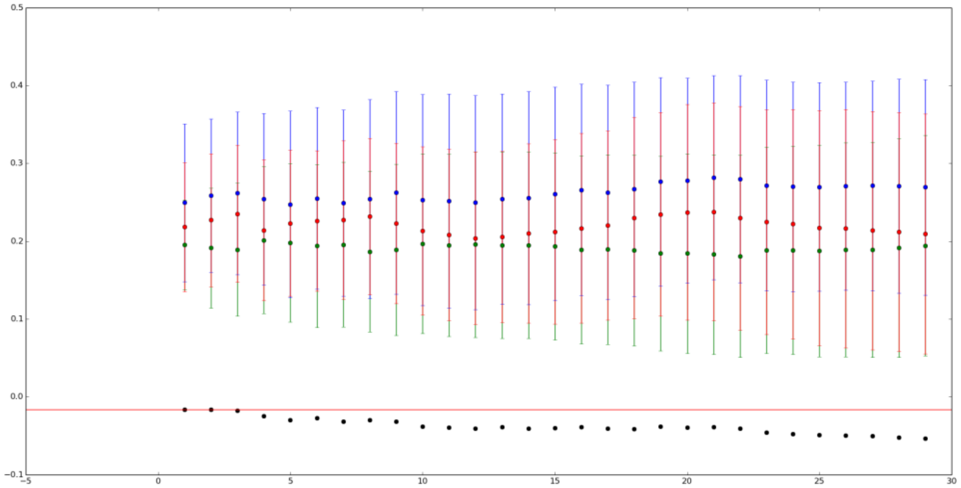

Figure 11 Performance of Orthogonal Matching Pursuit with varying number of non-zero coefficients. Peaks in inter-class variance occur at 3, 8 and 21. 8 non-zero coefficients provides a nice balance, of inter and intra class variance. This also approximates the recommended default of 10% of the number of features (68 total).

5 Feature Selection and Trigger Analysis

Of course, the real problem of interest is finding the most powerful migraine triggers. Ideally a model from Section 3 or Section 4 would have presented itself with good recognition rates. From this model, I would have extracted triggers. Since no model gave noteworthy classification or clustering, we must approach any feature selection results with some skepticism. Even the most powerful trigger can, at best, be expected to have moderate correlation.

5.1 Extracting Coefficients from Linear Models

Some learning models lend themselves much more easily to interpretation than others. kNN for example gives no indication which features might be most important to determining an output value. Linear models make feature extracting trivial however. By extracting the coefficients of each model, and taking those with the largest absolute value, impactful triggers can be found. The sign of each feature is then used to determine whether the trigger is correlated or anti-correlated with migraines. Figures 12, 13 and 14 show some of the top features according to several different models.

Linear Regression	Logistic Regression - L1 penalty	Logistic Regression - L2 penalty
['Aug', 0.2874],	['Saturday', 1.2485]	['Saturday', 1.267]
['Changing weather1', 0.2475]	['Changing weather1', 0.7946]	['Lack of sleep1', 1.0789]
['Lack of sleep1', 0.2448]	['Physical exertion1', 0.7538]	['Aug', 0.9476]
['Post-stress3', 0.1973]	['Lack of sleep1', 0.7115]	['Changing weather1', 0.8854]
['Loud sounds3', 0.1814]	['Sep', 0.6273]	['Physical exertion1', 0.8672]
…	…	…
['Nov', -0.1278]	['Physical exertion2', -0.7921]	['Physical exertion2', -0.9866]
['Cold air exposure1', -0.1426]	['Loud sounds2', -0.8135]	['Loud sounds2', -1.0058]
['Red wine2', -0.1670]	['Loud sounds1', -0.8412]	['May', -1.0742]
['Changing weather2', -0.1728]	['Red wine2', -1.1853]	['Red wine2', -1.3404]
['May', -0.2257]	['Monday', -1.1946]	['Monday', -1.5658]

Figure 12 Top 5 positive and negative coefficients as extracted from linear models. Features appended with a number denote how many days prior that trigger occurred. E.g. 'Changing weather1' means that the journal entry for yesterday recorded a change in weather.

OMP *(n_nonzero_coef= 8)*	OMP *(n_nonzero_coef= 21, top 8 by abs. value)*
['Saturday', 0.3566]	['Saturday', 0.3338]
['Monday', -0.2134]	['Changing weather1', 0.2059]
['Aug', 0.1992]	['Aug', 0.1972]
['Red wine2', -0.1464]	['Sep', 0.1739]
['Friday', 0.1265]	['Changing weather2', -0.1609]
['Red wine1', 0.1198]	['Monday', -0.1563]
['Physical exertion1', 0.1077]	['Lack of sleep1', 0.1561]
['Spirits1', 0.0714]	['Red wine2', -0.1297]

Figure 13 Top 8 features by absolute value according to Orthogonal Matching Pursuit. OMP gave consistently good triggers which tended to agree across all models.

SVM Regressor – Linear kernel	SVM Classifier – Linear kernel
['Monday', 0.1729]	['Monday', 0.7414]
['Changing weather2', 0.1328]	['Changing weather2', 0.6545]
['Mar', 0.1132]	['Tuesday', 0.5897]
['Sunday', 0.1112]	['Cold air exposure1', 0.5344]
['Cold air exposure1', 0.1102]	['Jul', 0.5095]
…	…
['Post-stress3', -0.1462]	['Nightshade2', -0.3320]
['Sep', -0.1507]	['Aug', -0.3550]
['Changing weather1', -0.1574]	['Sep', -0.6293]
['Aug', -0.2309]	['Changing weather1', -1.2779]
['Saturday', -0.5942]	['Saturday', -1.3102]

Figure 14 Top 5 positive and negative features as extracted from SVM models. Note that negative coefficients are correlated features unlike regression and OMP where they are anti-correlated features.

5.2 Impactful Triggers from Feature Selection

Feature Selection algorithms are also designed to reduce problems to only the most important features. Generally this is to reduce the required training set to a feasible size but they can also be used to identify the most impactful features. However, feature selection does not, by itself, tell us whether or not a feature is beneficial or detrimental to migraines.

K Best – f_regression	K Best – f_chi2	K Best – f_classif
['Saturday', 24.9200]	['Saturday', 38.1478]	['Saturday', 24.2303]
['Monday', 13.9969]	['Monday', 14.8099]	['Monday', 8.9263]
['Citrus1', 9.8427]	['Citrus1', 5.3079]	['Smoked or cured meat1', 6.1759]
['Spirits1', 8.3527]	['Friday', 4.7958]	['Caffeine1', 5.5047]
['Beer2', 6.858]	['Caffeine1', 4.2535]	['Red wine1', 5.2751]
['Red wine1', 6.8449]	['Smoked or cured meat1', 3.8947]	['Changing weather1', 5.1622]
['Smoked or cured meat1', 6.5979]	['Spirits1', 3.8467]	['Citrus1', 4.9115]
['Physical exertion1', 6.3198]	['Beer2', 3.0801]	['Spirits1', 4.2186]
['Caffeine1', 6.0175]	['Aug', 3.0314]	['Beer2', 3.4771]
['Citrus2', 3.7193]	['Red wine1', 3.0041]	['Physical exertion1', 3.1823]

Figure 15 K Best feature selection with various p-value functions.

After using K Best feature selection, the top 10 for each p-value function gave a list of 13 features which are likely to be impactful. Whether a given feature is positively or negatively correlated with a migraine was then determined by comparing coefficients for each. Some features had contradictory scores between models (e.g. smoked meat1, Friday, beer2). Others had very weak scores across all models (e.g. caffeine1). After dismissing these features, a list of features nearly identical to OMP10 remained; Citrus1 and Citrus2 being the only exceptions.

	Linear	L1	L2		SVR	SVC
Saturday	*	1.2486	1.2669		-0.6458	-1.310
Monday	*	-1.1955	-1.5658		0.1766	0.7414
Citrus1	-0.0521	-0.6081	-0.6720		0.0164	0.0535
Smoked meat1	0.0168	-0.0258	-0.0022		-0.0157	-0.1788
Spirits1	0.0680	0.5151	0.5814		-0.0185	-0.0482
Friday	*	0.3660	0.3237		0.0863	0.1769
Caffeine1	0.0183	0.0951	0.1411		-0.0017	-0.1231
Red wine1	0.1023	0.6257	0.7632		-0.0450	-0.1663
Beer2	0.0172	-0.03129	-0.0180		-0.0049	-0.0334
Physical1	0.1181	0.7534	0.8672		-0.0463	0.0097
Aug	0.2140	0.6060	0.9476		-0.6458	-0.3550
Citrus2	0.0156	0.0209	0.0792		-0.0212	-0.2088
Weather1	0.2483	0.7998	0.8854		-0.1419	-1.2779

*Figure 16 Top 13 features (selected from Figure 11) * denotes -8798673712910. Linear regression behaved unusually for the hot valued weekdays. More unusually was that months were unaffected. These terms were not considered.*

6 CONCLUSIONS

Although no model was able to able to achieve good prediction for the dataset, this is not too surprising. Migraines are not well understood. However, results suggest that migraine severity is not linked with these triggers.

Orthogonal Matching Pursuit proved to be a very strong model for this dataset. Not only did OMP provide comparatively good prediction but allowed for very straightforward trigger analysis. Other variants, namely Block-OMP, Group OMP, Stagewise OMP (StOMP) and Compressive Sampling MP (CoSaMP), also require attention. It is unknown whether pre-packaged versions of these algorithms are currently available. These models may also be suitable for other self-monitored symptom tracking, particularly diabetes.

In regards to features, although some triggers were already well known (Monday, Saturday, Changing Weather1), new triggers were discovered (Cold air exposure1). Cold air exposure is of particular interest as it is negatively correlated with migraines and can be controlled, especially in the cooler months. Lastly, while the subject has previously abstained from alcohol to control migraines, it appears that beer has no effect on his migraines. This does not transfer to all alcohol as spirits and red wine *are* correlated with migraines.