Change Point Support Vector Machine with L_1 Regularization

Ryan Lattanzi

Abstract

In this paper, we introduce the classification change point problem and attempt to solve it using an L_1 regularized linear support vector machine (SVM). Our choice classifier was determined primarily by its good performance in the increasingly relevant high dimensional setting. We explain and use an efficient algorithm that involves only one model update rather than an exhaustive search over many models, as is the current practice, to optimize over a grid of potential splitting values. Our algorithm is made robust in the sense that it includes detecting the case where no change point is present. Furthermore, it allows for a generalization over which the change point is determined; we label this as the change inducing variable. That is, we can detect a change point over any covariate included in the data, as we see in our application (Section 5). Once an estimate of the change point, $\hat{\tau}$, is obtained, we fit two regularized SVMs: one on all data points where the change inducing variable is less than $\hat{\tau}$, and one on all data that is greater than $\hat{\tau}$. To verify our proposed model's performance, we run a simulation (Sections 3 and 4) and compare our method with the standard linear SVM and nonlinear SVM (using the radial basis kernel) over several simulated cases and data. Although the nonlinear SVM performed best in terms of prediction loss on all cases, we find its main limitation to be the time to fit the model that grows quickly with more data. This is not practical in the boom of big data. However, our model's run time is approximately one-fifth of the nonlinear model in our largest simulation case, has much better performance (smaller prediction loss) than the standard linear SVM, and accurately obtains an estimation of the true change point especially as the data grows bigger.

1 Introduction

We are interested in the change point binary classification problem in which our true splitting point, τ_0, is unknown. In the case where a change point is present, the data would benefit from two different classifiers - one fit before, and one fit after the change point. However, we must keep in mind detecting cases in which no change point is necessary. This will be formulated in Section 2.3. Intuitively (and as current methods suggest), we could search over a grid of potential split values by updating the two models at every grid value. While this produces a globally optimal result, it is exhaustive and highly inefficient especially with bigger data. Hence, we propose an efficient method which approximates τ_0 with an estimate $\hat{\tau}$ that requires only one model update from some initialized model induced by a hypothesized change point, $\tau^{(0)}$. The algorithm and theory is based upon [1] that involves the regression change point problem using Lasso regression. The same ideas in the regression problem extend to the classification case which is why this approximation also yields near-optimal results.

The classifier chosen was an L_1 regularized linear support vector machine in order that we can apply our method to high-dimensional data in an interpretable manner. There are two points to clarify here. First, we refer to high dimensionality as the case when the cardinality our feature space, p, is comparable to or exceeds the number of training examples available, n. This situation is becoming more and more relevant as the era of big data continues to explode, and L_1 regularization is a popular method to handle it. Second, we say "interpretable" to mean that this type of regularization will simultaneously perform feature selection and reduce the dimension significantly by setting coefficients directly to zero.

For a simple illustration of the change point problem, consider the plot of simulated data below in Figure 1. A linear boundary would not suffice as a successful classifier since we see it would not separate the data adequately. If instead we examine Figure 2, we see the data split into two parts in which it is trivial to fit a linear SVM for each part. In other cases, we may see that a nonlinear boundary (induced by a kernel SVM) could successfully separate the data. However, a nonlinear SVM is difficult to construct with an L_1 regularization term, so our proposed model outperforms both in its simplicity to implement and the time it takes to fit the model (which we will see explicitly in Sections 3 and 4).

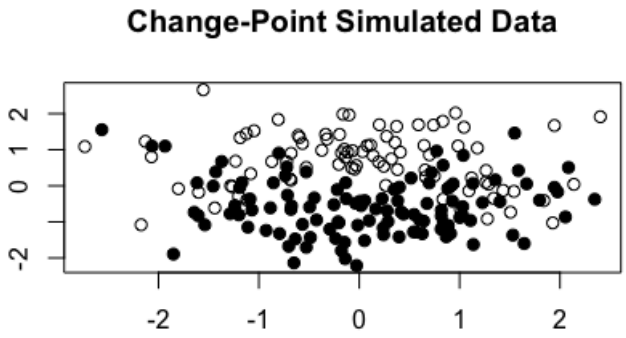
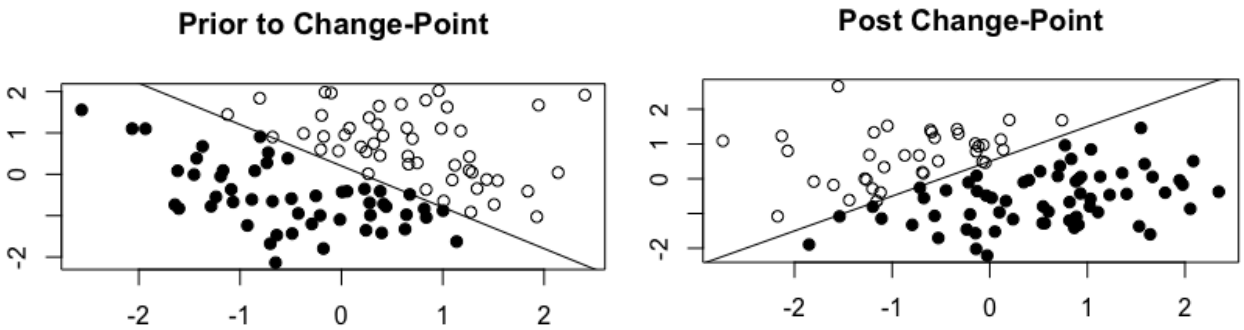


Figure 1: Simulated dataset with a change point present. A single linear classifier over this entire set would yield poor performance.



(a) Data before the change point with a separating boundary.

(b) Data after the change point with a separating boundary.

Figure 2: This split illustrates how the general direction of the data shifts before and after our change point. Clearly, fitting two classifiers is straightforward after observing this split and would yield favorable results.

1.1 Background of the Support Vector Machine

We will first discuss and formulate the maximal margin classifier and extend it to the support vector machine, which further extends to our proposed model. In the typical binary classification problem, we have class labelings $y \in \{-1, 1\}$ and want to find a separating linear hyperplane to accurately distinguish between the two classes. We can again refer to Figure 2 for a simple illustration of this concept, in which it is possible to successfully separate the two classes. The goal of the maximal margin classifier is to find the separating hyperplane that induces maximal distance between the data points and hyperplane. We define this distance to be the "margin" and denote it M. Intuitively, maximizing the margin will increase our confidence in the classifier since we will have more wiggle room, so to speak. The exact optimization problem is as follows:

$$\begin{aligned} &\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{maximize}} \quad M \\ &\text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 = 1, \\ &\qquad\qquad y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) \geq M \qquad \forall i = 1, \ldots, n. \end{aligned} \tag{1}$$

In words, this optimization problem finds model coefficients that ensure all observations are on the correct side of the hyperplane with at least a distance M from the hyperplane.

We would be naive to think a solution to (1) is always possible. In fact, in many cases it is not possible, in which we find ourselves in the non-separable case. Thankfully, we can extend the maximal margin classifier to the non-separable case by introducing slack variables, ϵ_i for $i = 1, \ldots, n$, and a nonnegative tuning parameter, C. These additions allow for some data points to lie within the margin strip or even on the wrong side of the hyperplane. The optimization becomes:

$$\begin{aligned} &\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{maximize}} \quad M \\ &\text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 = 1, \\ &\qquad\qquad y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) \geq M(1 - \epsilon_i), \\ &\qquad\qquad \epsilon_i \geq 0, \sum_{i=1}^{n} \epsilon_i \leq C. \end{aligned} \tag{2}$$

Ideally we would like to have $\epsilon_i = 0$, indicating that the particular observation is on the correct side of the margin. However, if we have $\epsilon_i > 0$, the observation is within the margin band, and $\epsilon_i > 1$ indicates the observation is on the wrong side of the hyperplane! Controlling how much slack (or violations) we allow in our classifier is precisely what our tuning parameter C sets out to do. The smaller C is, the less violations we allow, in turn decreasing our margin so as to minimize observations lying on the wrong side of it. Conversely, larger values of C allow for more violations and hence a larger margin. As per usual with a tuning parameter, the goal is to find the bias-variance sweet-spot via some model selection method such as cross-validation.

2 Problem Setup

2.1 Objective Equation and Some Notation

Section 9.5 in the well-written book *An Introduction to Statistical Learning* [2] introduces an alternative representation of (2) in a concise and elegant manner:

$$\underset{\beta_0,\beta_1,...,\beta_p}{\text{minimize}}\left\{ \sum_{i=1}^{n}(1-y_i f(x_i))_+ + \lambda \sum_{j=1}^{p}\beta_j^2 \right\}, \quad \lambda > 0, \tag{3}$$

where λ is a nonnegative tuning parameter and $f(x_i) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$. For the scope of this paper, we do not find it necessary to include its derivation, so we leave its verification to the reader. We only note that the tuning parameter λ has an effect very similar to C in (2), where it is proportional to both the number of violations tolerated and bias of the model. This representation follows the typical loss-penalty framework in which regularization is involved. Note that it is necessary to have a binary classification problem here with our classes corresponding to $y \in \{1, -1\}$. Hence, the loss function is simply the misclassification error. Also note that this formulation uses the ridge penalty term (L_2-norm). However, we are interested in solving this optimization problem using the lasso penalty term (L_1-norm) so that our formulation becomes

$$\underset{\beta_0,\beta_1,...,\beta_p}{\text{minimize}}\left\{ \sum_{i=1}^{n}(1-y_i f(x_i))_+ + \lambda \sum_{j=1}^{p}|\beta_j| \right\}, \quad \lambda > 0. \tag{4}$$

We prefer this method because it allows for a better interpretation of our model as it will set coefficients directly equal to zero (as mentioned earlier) whereas the penalty term in (3) does not. Especially in high dimensions, this becomes useful in obeying the famous notion of Occam's Razor to fit the simplest model possible in hopes of better generalization performance.

Now, it is necessary to introduce some notation that will be used throughout this paper. If we have n data points, Let $w = \{w_1, ..., w_n\}$ be some set of values (not necessarily ordered) which we will refer to as the change inducing variable. This allows for a generalization of our splitting measure so our change point can exist over values of some model covariate (we will see in our application that we set our change inducing variable to median income). Hence, even if the values contained in the set w are continuous, we have an inherent discretization of the interval $(\inf w, \sup w)$ based on the observed data points $w_1, ..., w_n$. It is also worth noting that τ_0 lies on the same scale as w. In turn, splitting the data at τ_0 will involve extracting two subsets of the original dataset $\{i | i \in \{1, ...n\}, w_i \leq \tau_0\}$ and $\{i | i \in \{1, ...n\}, w_i > \tau_0\}$.

2.2 RMosek

We use RMosek in order to solve (4). Our notation is that $\beta = (\beta_0, \beta_1, ..., \beta_p)$ where β_0 is the intercept. Hence, it is necessary to define $x = (x_0, x_1, ..., x_p)$ where $x_0 = 1$. We must first alter our equation into a compatible formulation to be encoded with RMosek:

$$\begin{aligned} \underset{\beta_0,\beta_1,...,\beta_p}{\text{minimize}} \quad & \left\{ \sum_{i=1}^{n}(\ell_i)_+ + \lambda \sum_{j=1}^{p}(\beta_j^+ + \beta_j^-) \right\} \\ \text{subject to} \quad & \ell_i + y_i x_i' \beta \geq 1 \\ & \ell_i \geq 0, \end{aligned}$$

where

$$\ell_i = 1 - y_i x_i' \beta$$

$$\beta_j^+ = \begin{cases} \beta_j & \beta_j > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\beta_j^- = \begin{cases} -\beta_j & \beta_j < 0 \\ 0 & \text{otherwise} \end{cases}$$

With this construction, it is straightforward to implement our linear optimization problem into RMosek. We invite the reader to consult [3] for details.

2.3 Algorithm

Before we embark on the algorithm, we first must denote $\bar{\mathbb{R}}^* := \mathbb{R} \cup \{-\infty\}$ as the extended real line that includes negative infinity to account for the case of no change point. This will be our search space over which to estimate $\hat{\tau}$. We also define our loss function $Q(\tau, \beta, \gamma)$ to be similar to what we have seen in (4) but taking into account our change point:

$$Q(\tau, \beta, \gamma) = \sum_{i=1}^{n} (1 - y_i x_i^T \beta)_+ \, \mathbf{I}[w_i \leq \tau] + \sum_{i=1}^{n} (1 - y_i x_i^T \gamma)_+ \, \mathbf{I}[w_i > \tau]. \tag{5}$$

Algorithm 1: Detection and Estimation of Change Point and Classification Parameters

Step 0 (Initialize): Choose some initial $\tau^{(0)} \in \mathbb{R}$ that is marginally away from the boundaries of $\mathbb{R}$. Compute the initial classification parameter estimates,

$$(\hat{\beta}^{(0)}, \hat{\gamma}^{(0)}) = \underset{\beta, \gamma \in \mathbb{R}^p}{\arg\min} \Big\{ Q(\tau^{(0)}, \beta, \gamma) + \lambda_1 \left\| (\beta, \gamma) \right\|_1 \Big\}, \quad \lambda_1 > 0.$$

Step 1: Update $\tau^{(0)}$ to obtain the change point estimate $\hat{\tau}$ where,

$$\hat{\tau} = \underset{\tau \in \bar{\mathbb{R}}^*}{\arg\min} \Big\{ Q(\tau, \hat{\beta}^{(0)}, \hat{\gamma}^{(0)}) + \mu \left\| \Phi(\tau) \right\|_0 \Big\}, \quad \mu > 0.$$

Step 2: Update $(\hat{\beta}^{(0)}, \hat{\gamma}^{(0)})$ to obtain final classification model parameters $(\hat{\beta}^{(1)}, \hat{\gamma}^{(1)})$ where,

$$(\hat{\beta}^{(1)}, \hat{\gamma}^{(1)}) = \underset{\beta, \gamma \in \mathbb{R}^p}{\arg\min} \Big\{ Q(\hat{\tau}, \beta, \gamma) + \lambda_2 \left\| (\beta, \gamma) \right\|_1 \Big\}, \quad \lambda_2 > 0.$$

It is worth mentioning that in Step 0 and Step 2 we use a special development of the Akaike Information Criterion (AIC) seen in equation (8) from [4] in order to tune our hyperparameters λ_1 and λ_2, respectively. Furthermore, in Step 0, we typically choose $\tau^{(0)}$ to be the 50^{th} percentile of w, as this is the most logical starting point when no information about the true change point is available. It will also ensure we are marginally away from the boundaries of $\mathbb{R}$ as Step 0 indicates is necessary (the theoretical framework for this condition can be found in [1]). Figure 3 illustrates this requirement empirically by plotting the bias, $|\tau_0 - \hat{\tau}|$, over fifty different values of the initializer $\tau^{(0)}$. We can see

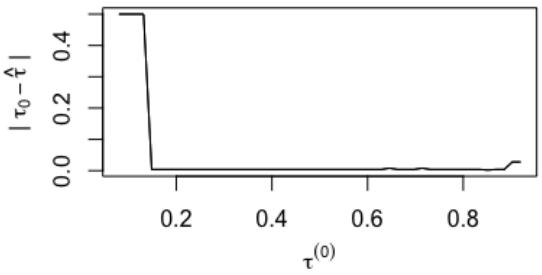


Figure 3: Plotting the bias of different values of our initializer $\tau^{(0)}$ for $\tau_0 = 0.7$, $n = 500$, $p = 25$.

our bias becomes a tad unstable as $\tau^{(0)}$ tends towards its boundaries, but for the most part, our result is irrespective of the initializer $\tau^{(0)}$.

The minimizing optimization found in Step 1 contains a lot of information. We see that this also follows the loss-penalty framework, in which our regularization accounts for the case where no change point is present. We define $\Phi(\tau) = P[w_i \leq \tau]$ so that when $\tau = -\infty$ (the case of no change point), $\Phi(\tau) = 0$ and $\Phi(\tau) > 0$ for all other τ. Then using the usual 0-norm defined by

$$\|x\|_0 = \begin{cases} 0 & x = 0 \\ 1 & \text{otherwise,} \end{cases}$$

we see that (for $\tau = -\infty$)

$$Q(\tau, \beta, \gamma) = \sum_{i=1}^{n} (1 - y_i x_i^T \gamma)_+$$

so only one classifier is used over the entire dataset. The tuning parameter μ of this step was obtained via Bayesian Information Criterion (BIC). We calculated $\hat{\tau}(\mu)$ that were computed over a grid of values of μ, and chose the value of μ that minimized the following,

$$BIC(\mu) = \log\left(Q(\hat{\tau}(\mu), \hat{\beta}(\mu), \hat{\gamma}(\mu))\right) + \frac{\log(n)}{n} \|\Phi(\hat{\tau}(\mu))\|_0 .$$

Here, $Q(\cdots)$ is defined in the same way as (2) and $\hat{\beta}(\mu), \hat{\gamma}(\mu)$ are coefficient estimates derived from the partition $\hat{\tau}(\mu)$ yields. There is one more technical note in this optimization step. Notice we define our search space for $\hat{\tau}$ to be the interval $\bar{\mathbb{R}}^*$. While this may seem infeasible, the implementation actually follows the discretization method briefly mentioned above in section 2.1 that depends on our change inducing variable w. Hence, even though w is holds continuous values, our implemented optimization is

$$\hat{\tau} = \underset{\tau \in \{-\infty\} \cup w}{\arg\min} \left\{ Q(\tau, \hat{\beta}^{(0)}, \hat{\gamma}^{(0)}) + \mu \|\Phi(\tau)\|_0 \right\}, \quad \mu > 0. \tag{6}$$

This optimization is computationally sound since it only requires computing n values of the objective function to locate the minimum.

3 Simulation

When simulating data, we wanted to ensure the change point elicited sufficiently distinct classifiers. To do this we worked backwards and initialized coefficients for two classifiers such that their decision boundaries had significant directions. Specifically, we set

$$\beta = (1, 1, 1, 1, 1, 1, 0_1, 0_2, ..., 0_{p-5})$$
$$\gamma = (1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0_1, ..., 0_{p-10})$$

where each vector is of size $p+1$ and the first element in each (always set to 1) represents the intercept coefficient. Two different feature matrices, $X_1 = [x_{ij}^{(1)}]$ and $X_2 = [x_{ij}^{(2)}]$ with with $i = 1, ..., n$ and $j = 2, ..., p+1$, were then constructed with elements $x_{ij}^{(1)} \sim \mathcal{N}(0, 1)$ IID and $x_i^{(2)} = (x_{i2}, ..., x_{i(p+1)}) \sim \mathcal{N}(0, \Sigma)$ in order to diversify our results. Our construction of Σ in X_2 is a $p \times p$ Toeplitz matrix as follows:

$$\Sigma = \begin{bmatrix} 1 & 0.5 & 0.5^2 & \dots & 0.5^{(p-1)} \\ 0.5 & 1 & 0.5 & \dots & 0.5^{(p-2)} \\ 0.5^2 & 0.5 & 1 & \dots & 0.5^{(p-3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0.5^{(p-1)} & 0.5^{(p-2)} & 0.5^{(p-3)} & \dots & 1 \end{bmatrix}$$

For simplicity, our change inducing variable was set to be the index of the dataset. That is, we set $w = \{\frac{1}{n}, \frac{2}{n}, ..., \frac{n-1}{n}\}$. Note that as explained before, we are not limited to this search space. Finally after setting a true τ_0, corresponding labels, y, for each data point were obtained by stacking

$$y[\text{pre}] = sign(X[\text{pre}]\beta)$$
$$y[\text{post}] = sign(X[\text{post}]\gamma)$$

where pre $= \{x_{ij} | i \leq \tau_0\}$ and post $= \{x_{ij} | i > \tau_0\}$.

In our simulation, we compared three different implementations of SVM's: a linear SVM over the entire dataset, a nonlinear SVM (radial basis kernel) over the entire dataset, and our change point SVM with L_1 regularization which we have constructed throughout this paper. For each feature matrix introduced above, we ran nine cases with $n \in \{150, 250, 350\}$ and $p \in \{25, 150, 250\}$ for each classifier. We also included different true change point values $\tau_0 = \{-\infty, 0.3, 0.7\}$ for each case. Each case was repeated 100 times in order that we achieve a sense of stability. In total, we ran simulations for 54 different cases to obtain our results. The information collected were averages over the 100 runs, including the bias of $\hat{\tau}$ (**Bias**$(\hat{\tau})$), mean squared error of $\hat{\tau}$ (**MSE**$(\hat{\tau})$), false-positive rate (**FP**), false-negative rate (**FN**), prediction loss over the same data (**Prediction Loss**), and time to fit the model (**Time**). Here we define **Bias**$(\hat{\tau}) = \hat{\tau} - \tau_0$ and **MSE**$(\hat{\tau}) = (\hat{\tau} - \tau_0)^2$. Since the first two classifiers did not compute any estimates $\hat{\tau}$, all columns pertaining to $\hat{\tau}$ are filled with dashes.

Finally, we note our treatment of τ_0 and $\hat{\tau}$ in both the finite and infinite cases to avoid ending up with infinite biases and MSE. When τ_0 is finite and we (incorrectly) obtain estimate $\hat{\tau} = -\infty$ to get a false negative, we set

- $\hat{\tau} = 0$ if $\tau_0 \leq 0.5$, or
- $\hat{\tau} = 1$ if $\tau_0 > 0.5$.

Otherwise, if $\hat{\tau}$ is finite, we use it as is. In the case that $\tau_0 = -\infty$, we set

- $\hat{\tau} = \tau_0 = 0$ if $\hat{\tau} = -\infty$,
- $\tau_0 = 0$ if $\hat{\tau}$ is finite and $\hat{\tau} \leq 0.5$, or
- $\tau_0 = 1$ if $\hat{\tau}$ is finite and $\hat{\tau} > 0.5$.

where the last two bullets represent false positives.

4 Results and Discussion

The results of our simulation can be found on the following pages. Specifically, Table 1, Table 2, and Table 3 shows comparisons for feature matrix X_1 and true change point values $\tau_0 = -\infty, 0.3, 0.7$ respectfully. Table 4, Table 5, and Table 6 show results for feature matrix X_2 and true change point values $\tau_0 = -\infty, 0.3, 0.7$ respectfully. An immediate observation from our results is that the prediction loss of the kernel SVM is zero through all cases. This is not extraordinarily abnormal since this particular SVM is notorious for fitting the data very well. So, why not stick with this method? The first and rather obvious reason is the time to fit this model. Compared to the linear and change point models, this SVM takes approximately five times longer to fit - a major disadvantage in the realm of big data. The second reason is the tendency to overfit with its intensive use of all parameters. This ties in with interpretability; we will not get the nice variable selection property that is provided with regularization.

If we look closely at Tables 1 and 4 ($\tau_0 = \infty$ case), our change point models perform similarly over all measures. They also perform similarly to the linear SVMs in terms of time and prediction loss, which is expected since our model reduces to a linear SVM when no change point is detected. Turning our attention to Tables 2 and 5 ($\tau_0 = 0.3$), we see both change point models have a similar runtime. However, there is a considerable difference in the other measures; our model performs significantly better with feature matrix X_2. The reason for this is unclear at this time, but brings up new questions to investigate relating our model to the nature of the data set at hand. Finally, we can see that the prediction loss gap between the change point and linear SVMs continues to grow in favor of our model especially as n increases. An analysis of Tables 3 and 6 ($\tau_0 = 0.7$) follows analogously.

Following the simulation, we also found it insightful to include a comparative ROC plot shown in Figure 4 where $\tau_0 = 0.7$, $n = 250$, and $p = 250$. This particular case was chosen to mimic high dimensions with n and p of comparable sizes. It is clear the linear SVM has the least performance power as its area under the curve is significantly less than the other two models. The impeccable performance of the radial SVM in our simulation is corroborated by Figure 4, showing the ideal curve, but we keep in mind its long run time. Although our change point model does not perform perfectly, we see that it improves upon the linear SVM with an almost undistinguishable run time, which leads us to prefer it over both models.

5 Application

Our application is based on the "Communities and Crime Data Set" (Redmond, 2009) pulled from the infamous UCI dataset repository [5]. This dataset includes a vast number of quantitative socio-economic attributes pertaining to both the community and law

Table 1: Simulation results for data generated from X_1 for Linear SVM, RBF SVM, and Change Point SVM with $n \in \{150, 250, 350\}$, $p \in \{25, 150, 250\}$, and $\tau_0 = -\infty$.

Method	n	p	Bias($\hat{\tau}$)	MSE($\hat{\tau}$)	FP	FN	Prediction Loss	Time
Linear SVM	150	25	-	-	-	-	0.012	0.502
	150	150	-	-	-	-	0.060	2.081
	150	250	-	-	-	-	0.079	2.703
	250	25	-	-	-	-	0.008	0.622
	250	150	-	-	-	-	0.043	3.499
	250	250	-	-	-	-	0.058	5.584
	350	25	-	-	-	-	0.007	0.844
	350	150	-	-	-	-	0.031	5.419
	350	250	-	-	-	-	0.038	8.733
RBF SVM	150	25	-	-	-	-	0	3.731
	150	150	-	-	-	-	0	13.929
	150	250	-	-	-	-	0	21.186
	250	25	-	-	-	-	0	5.718
	250	150	-	-	-	-	0	18.381
	250	250	-	-	-	-	0	31.159
	350	25	-	-	-	-	0	9.019
	350	150	-	-	-	-	0	26.851
	350	250	-	-	-	-	0	44.521
Change Point SVM	150	25	0.009	0.018	0.09	0	0.034	0.717
	150	150	0.003	0.003	0.05	0	0.056	2.284
	150	250	-0.006	0.001	0.06	0	0.073	2.847
	250	25	0.009	0.015	0.1	0	0.023	0.800
	250	150	0.001	0.008	0.04	0	0.036	3.705
	250	250	0.002	0.002	0.04	0	0.036	5.933
	350	25	0.015	0.019	0.1	0	0.018	0.959
	350	150	0.017	0.008	0.07	0	0.026	5.708
	350	250	0.0004	0.003	0.03	0	0.025	9.165

Table 2: Simulation results for data generated from X_1 for Linear SVM, RBF SVM, and Change Point SVM with $n \in \{150, 250, 350\}$, $p \in \{25, 150, 250\}$, and $\tau_0 = 0.3$.

Method	n	p	Bias($\hat{\tau}$)	MSE($\hat{\tau}$)	FP	FN	Prediction Loss	Time
Linear SVM	150	25	-	-	-	-	0.185	0.432
	150	150	-	-	-	-	0.281	1.798
	150	250	-	-	-	-	0.290	2.562
	250	25	-	-	-	-	0.164	0.552
	250	150	-	-	-	-	0.218	3.289
	250	250	-	-	-	-	0.243	5.040
	350	25	-	-	-	-	0.160	0.865
	350	150	-	-	-	-	0.191	5.317
	350	250	-	-	-	-	0.197	8.644
RBF SVM	150	25	-	-	-	-	0	3.453
	150	150	-	-	-	-	0	12.760
	150	250	-	-	-	-	0	20.414
	250	25	-	-	-	-	0	5.481
	250	150	-	-	-	-	0	17.771
	250	250	-	-	-	-	0	29.463
	350	25	-	-	-	-	0	9.187
	350	150	-	-	-	-	0	27.814
	350	250	-	-	-	-	0	45.766
Change Point SVM	150	25	-0.077	0.019	0	0.14	0.117	0.682
	150	150	-0.183	0.055	0	0.57	0.218	1.877
	150	250	-0.204	0.069	0	0.69	0.237	2.789
	250	25	-0.033	0.006	0	0.04	0.061	0.844
	250	150	-0.078	0.023	0	0.21	0.120	3.634
	250	250	-0.137	0.037	0	0.35	0.126	5.575
	350	25	-0.015	0.002	0	0.01	0.036	1.140
	350	150	-0.044	0.007	0	0.05	0.074	6.089
	350	250	-0.081	0.019	0	0.18	0.087	9.542

Table 3: Simulation results for data generated from X_1 for Linear SVM, RBF SVM, and Change Point SVM with $n \in \{150, 250, 350\}$, $p \in \{25, 150, 250\}$, and $\tau_0 = 0.7$.

Method	n	p	Bias($\hat{\tau}$)	MSE($\hat{\tau}$)	FP	FN	Prediction Loss	Time
Linear SVM	150	25	-	-	-	-	0.179	0.446
	150	150	-	-	-	-	0.266	1.796
	150	250	-	-	-	-	0.295	2.540
	250	25	-	-	-	-	0.160	0.582
	250	150	-	-	-	-	0.218	3.567
	250	250	-	-	-	-	0.248	5.001
	350	25	-	-	-	-	0.159	0.805
	350	150	-	-	-	-	0.186	4.869
	350	250	-	-	-	-	0.206	8.408
RBF SVM	150	25	-	-	-	-	0	3.539
	150	150	-	-	-	-	0	12.615
	150	250	-	-	-	-	0	20.359
	250	25	-	-	-	-	0	5.675
	250	150	-	-	-	-	0	19.409
	250	250	-	-	-	-	0	29.056
	350	25	-	-	-	-	0	8.759
	350	150	-	-	-	-	0	25.572
	350	250	-	-	-	-	0	44.192
Change Point SVM	150	25	0.036	0.018	0	0.16	0.102	0.700
	150	150	0.161	0.052	0	0.55	0.210	1.922
	150	250	0.196	0.069	0	0.71	0.251	2.631
	250	25	-0.001	0.002	0	0	0.055	0.907
	250	150	0.034	0.014	0	0.11	0.107	4.116
	250	250	0.077	0.025	0	0.25	0.127	5.586
	350	25	-0.011	0.001	0	0	0.036	1.063
	350	150	0.005	0.003	0	0.01	0.069	5.593
	350	250	0.032	0.011	0	0.09	0.081	9.235

Table 4: Simulation results for data generated from X_2 for Linear SVM, RBF SVM, and Change Point SVM with $n \in \{150, 250, 350\}$, $p \in \{25, 150, 250\}$, and $\tau_0 = -\infty$.

Method	n	p	Bias($\hat{\tau}$)	MSE($\hat{\tau}$)	FP	FN	Prediction Loss	Time
Linear SVM	150	25	-	-	-	-	0.015	0.484
	150	150	-	-	-	-	0.038	1.786
	150	250	-	-	-	-	0.048	3.487
	250	25	-	-	-	-	0.008	0.585
	250	150	-	-	-	-	0.027	3.454
	250	250	-	-	-	-	0.032	5.288
	350	25	-	-	-	-	0.006	0.800
	350	150	-	-	-	-	0.020	5.082
	350	250	-	-	-	-	0.026	8.195
RBF SVM	150	25	-	-	-	-	0	3.615
	150	150	-	-	-	-	0	12.077
	150	250	-	-	-	-	0	24.126
	250	25	-	-	-	-	0	5.512
	250	150	-	-	-	-	0	18.172
	250	250	-	-	-	-	0	30.034
	350	25	-	-	-	-	0	8.766
	350	150	-	-	-	-	0	25.820
	350	250	-	-	-	-	0	43.102
Change Point SVM	150	25	0.004	0.009	0.06	0	0.023	0.639
	150	150	0.018	0.016	0.1	0	0.034	1.948
	150	250	0.007	0.012	0.08	0	0.039	4.050
	250	25	0.001	0.020	0.15	0	0.018	0.755
	250	150	0.024	0.013	0.08	0	0.025	3.701
	250	250	0.012	0.016	0.09	0	0.024	5.657
	350	25	0.015	0.010	0.07	0	0.014	0.890
	350	150	0.014	0.010	0.07	0	0.017	5.166
	350	250	0.012	0.009	0.07	0	0.018	8.429

Table 5: Simulation results for data generated from X_2 for Linear SVM, RBF SVM, and Change Point SVM with $n \in \{150, 250, 350\}$, $p \in \{25, 150, 250\}$, and $\tau_0 = 0.3$.

Method	n	p	Bias($\hat{\tau}$)	MSE($\hat{\tau}$)	FP	FN	Prediction Loss	Time
Linear SVM	150	25	-	-	-	-	0.155	0.439
	150	150	-	-	-	-	0.165	1.809
	150	250	-	-	-	-	0.171	2.886
	250	25	-	-	-	-	0.151	0.579
	250	150	-	-	-	-	0.157	3.542
	250	250	-	-	-	-	0.160	5.404
	350	25	-	-	-	-	0.152	0.812
	350	150	-	-	-	-	0.154	5.124
	350	250	-	-	-	-	0.160	8.297
RBF SVM	150	25	-	-	-	-	0	3.371
	150	150	-	-	-	-	0	12.252
	150	250	-	-	-	-	0	21.868
	250	25	-	-	-	-	0	5.543
	250	150	-	-	-	-	0	18.449
	250	250	-	-	-	-	0	30.441
	350	25	-	-	-	-	0	8.788
	350	150	-	-	-	-	0	26.115
	350	250	-	-	-	-	0	42.064
Change Point SVM	150	25	-0.028	0.009	0	0.06	0.060	0.699
	150	150	-0.071	0.026	0	0.21	0.096	2.061
	150	250	-0.094	0.035	0	0.33	0.099	3.378
	250	25	-0.007	0.001	0	0	0.035	0.862
	250	150	-0.017	0.004	0	0.03	0.053	3.858
	250	250	-0.006	0.003	0	0.01	0.054	5.832
	350	25	-0.005	0.0001	0	0	0.026	1.050
	350	150	-0.002	0.0003	0	0	0.035	5.548
	350	250	0.003	0.001	0	0	0.040	8.692

Table 6: Simulation results for data generated from X_2 for Linear SVM, RBF SVM, and Change Point SVM with $n \in \{150, 250, 350\}$, $p \in \{25, 150, 250\}$, and $\tau_0 = 0.7$.

Method	n	p	Bias($\hat{\tau}$)	MSE($\hat{\tau}$)	FP	FN	Prediction Loss	Time
Linear SVM	150	25	-	-	-	-	0.142	0.519
	150	150	-	-	-	-	0.172	1.925
	150	250	-	-	-	-	0.182	2.690
	250	25	-	-	-	-	0.151	0.617
	250	150	-	-	-	-	0.158	3.459
	250	250	-	-	-	-	0.159	5.254
	350	25	-	-	-	-	0.152	0.833
	350	150	-	-	-	-	0.152	5.930
	350	250	-	-	-	-	0.152	8.352
RBF SVM	150	25	-	-	-	-	0	3.594
	150	150	-	-	-	-	0	12.785
	150	250	-	-	-	-	0	20.199
	250	25	-	-	-	-	0	5.507
	250	150	-	-	-	-	0	17.482
	250	250	-	-	-	-	0	29.187
	350	25	-	-	-	-	0	8.772
	350	150	-	-	-	-	0	29.494
	350	250	-	-	-	-	0	43.258
Change Point SVM	150	25	-0.029	0.008	0	0.04	0.057	0.872
	150	150	0.006	0.020	0	0.14	0.097	2.312
	150	250	0.011	0.020	0	0.13	0.095	3.249
	250	25	-0.026	0.002	0	0	0.040	0.972
	250	150	-0.020	0.005	0	0.01	0.056	3.778
	250	250	-0.003	0.008	0	0.06	0.052	5.784
	350	25	-0.019	0.001	0	0	0.028	1.146
	350	150	-0.019	0.001	0	0	0.039	6.605
	350	250	-0.024	0.003	0	0	0.042	8.785

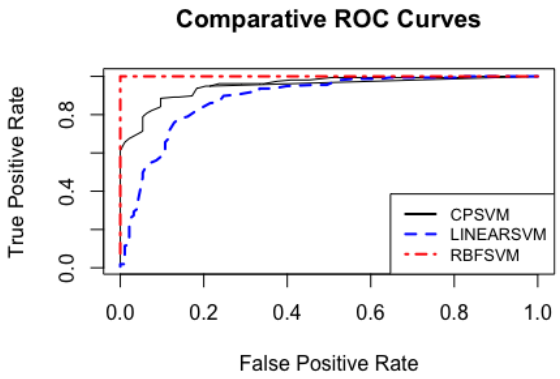


Figure 4: ROC Curves for the case $\tau_0 = 0.7$,$n = 250$, and $p = 250$ for change point, linear, and radial SVMs.

enforcement, such as the median family income and per capita number of police officers. These attributes are strung together in an attempt to explain the response variable, Per Capita Violent Crimes - the total number of violent crimes per one hundred thousand people. Here, violent crimes refer to murder, rape, robbery, and assault. The dataset contains 1994 observations and 128 attributes, where each observation represents a particular community. A full description can be found in [5].

Since we are dealing with a classification problem rather than regression, it was necessary to to bin the data in two classes. We defined a threshold to be the median of the response, so that anything over the median would have class label $\{1\}$ and anything under would be labeled $\{-1\}$. The interpretation of this binning is such that a community with positive labels is indicative of a higher-crime area.

In our study, we are interested in whether or not violent crime at a community level is is influenced by distinct socio-economic factors above and below a threshold of median income, and to estimate that threshold (if any) at which a change occurs. Hence our change inducing variable is the $medIncome$ attribute, and differing coefficients in our model before and after a median income threshold illustrate the different ways in which the factors effect per capita crime. We note the data set in interest came already normalized, so our raw output is uninterpretable unless we link it back to the unnormalized data.

Although the full dataset had $n = 1994$ and $p = 128$, there were an abundance of missing values. After removing all observations with missing points, we had only $n = 319$ communities remaining. We also removed those attributes which had a correlation with the median income attribute that was greater than 0.75. From the remaining factors, we further reduced our feature space by removing attributes that were highly correlated (with > 0.95 correlation) to other attributes in order to have a credible model. That left us with $p = 77$ predictor variables (excluding the median income variable).

Our change point SVM spit out the estimate $\hat{\tau} = 0.25$, meaning every community with median income less than 0.25 (on the scaled data) would have a different classifier than those communities above 0.25. This threshold lies at the 48^{th} percentile and Table 7 summarizes our results. We can see how much the dimension is reduced, as only a fraction of our variables are nonzero. Furthermore, it is interesting to note the near disjoint selection of attributes from $\hat{\beta}^{(1)}$ and $\hat{\gamma}^{(1)}$ that indicate a different set of influences depending on the median income value.

Table 7: Results from running the change point model over "Communities and Crime" data. The change inducing variable was taken to be median income. The estimated change point was $\hat{\tau}$=0.25, the 48^{th} percentile. Below all nonzero model coefficients truncated at 10^{-4}, where $\hat{\beta}^{(1)}$ are pre-change coefficients and $\hat{\gamma}^{(1)}$ are post-change coefficients.

Variable	**Description**	**Coefficient** ($\hat{\beta}^{(1)}$)	**Coefficient** ($\hat{\gamma}^{(1)}$)
racepctblack	% of population that is african american	0.2601	0.0000
racePctWhite	% of population that is caucasian	-0.2045	-0.0589
agePct12t21	% of population that is 12-21 in age	0.0000	0.0713
agePct65up	% of population that is 65 and over in age	0.2211	0.0000
pctWFarmSelf	% of households with farm or self employment income in 1989	-0.0411	0.0000
pctWInvInc	% of households with investment / rent income in 1989	0.0000	-0.1467
PctUnemployed	% of people 16 and over, in the labor force, and unemployed	0.0000	0.2027
PctIlleg	% of kids born to never married	0.4524	0.3501
PersPerOwnOccHous	mean persons per owner occupied household	-0.0699	0.0000
PctVacantBoarded	% of vacant housing that is boarded up	0.1472	0.1002
MedYrHousBuilt	median year housing units built	0.0125	0.0000
MedRentPctHousInc	median gross rent as a percentage of household income	0.1071	0.0000
NumStreet	number of homeless people counted in the street	0.0000	0.5309
PolicReqPerOffic	total requests for police per police officer	0.2404	0.0000
NumKindsDrugsSeiz	number of different kinds of drugs seized	0.0564	0.0000
PolicAveOTWorked	police average overtime worked	0.1505	0.0000
PolicCars	number of police cars	0.3597	0.0000
LemasGangUnitDeploy	gang unit deployed	0.1893	0.0000

References

[1] Kaul, A., Jandhyala, V. and Fotopoulos, S. (2019) *An efficient two step algorithm for high dimensional change point regression models without grid search.* Journal of Machine Learning Research 1.

[2] James, G., Witten, D., Hastie, T. and Tibshirani, R. (2013) *An Introduction to Statistical Learning.* Springer Science+Business Media New York.

[3] Mosek Aps. (2015) *Users Guide to the R-to-MOSEK Optimization Interface.* Mosek Aps.

[4] Claeskens, G., Croux, C. and Kerckhoven, J. (2008) *An Information Criterion for Variable Selection in Support Vector Machines.* Journal of Machine Learning Research 9.

[5] Redmond, M. (2009) *Communities and Crime Data Set.* Retrieved from `https://archive.ics.uci.edu/ml/datasets/Communities+and+Crime`