Kirchner, Antje, and Curtis S. Signorino. 2018. "Using Support Vector Machines for Survey Research." *Survey Practice* 11 (1). https://doi.org/10.29115/SP-2018-0001.

ARTICLES

Using Support Vector Machines for Survey Research

Antje Kirchner[1], Curtis S. Signorino[2]

[1] Department of Sociology, University of Nebraska – Lincoln, RTI International, [2] Department of Political Science, University of Rochester

Keywords: support vector machines

https://doi.org/10.29115/SP-2018-0001

Survey Practice

Vol. 11, Issue 1, 2018

Recent developments in machine learning allow for flexible functional form estimation beyond the approaches typically used by survey researchers and social scientists. Support vector machines (SVMs) are one such technique, commonly used for binary classification problems, such as whether or not an individual decides to participate in a survey. Since their inception, SVMs have been extended to solve categorical classification and regression problems. Their versatility in combination with the fact that they perform well in the presence of a large number of predictors, even with a small number of cases, makes them very appealing for a wide range of problems, including character recognition and text classification, speech and speaker verification, as well as imputation problems and record linkage. In this article, we provide a non-technical introduction to the main concepts of SVMs, discuss their advantages and disadvantages, present ideas as to how they can be used in survey research, and, finally, provide a hands-on example, including code, as to how they can be used in survey research and how the results compare to a traditional logistic regression.

What are Support Vector Machines and How are They Constructed?

Support vector machines (SVMs) are commonly used for classification problems, such as predicting whether or not an individual chooses to vote or whether or not an individual decides to participate in a survey. The first SVM algorithms were developed in the mid-1990s and focused on predicting binary outcomes using machine learning theory to maximize predictive accuracy (e.g., Boser, Guyon, and Vapnik 1992; Vapnik 1995). They have since been extended to solve categorical classification and regression problems (Attewell, Monaghan, and Kwong 2015; James et al. 2013). As an introduction to SVMs in survey research, we focus on SVMs for binary classification.

Like many classification and prediction methods, SVMs classify binary outcomes (e.g., survey response versus nonresponse) by estimating a separation boundary within the space defined by a set of predictor variables. A given observation, defined by the values of the predictor variables — i.e., where it is located in that predictor space — is classified based on which side of the boundary it falls. Theoretically, there are an infinite number of ways such a boundary could be created. In the simplest case, SVMs create

a "maximal margin" in the predictor space — the largest buffer separating observations for one outcome from those of the other outcome. Cases that fall exactly on the margin are called support vectors because these specific cases alone define the unique boundary solution. If there are only two predictor variables, the separating boundary is a line; with three predictors, the boundary is a plane; and with more than three predictors, the boundary is typically referred to as a separating hyperplane. Predictions are obtained from SVMs by using the corresponding decision function that is a mathematical depiction of the boundary.

As an example, suppose a researcher wants to predict survey participation based on age (X_1) and income (X_2). Using these two predictors, the SVM attempts to classify cases as either survey respondents (red triangles) or survey nonrespondents (blue circles), as displayed in the left pane of Figure 1. The optimal hyperplane (i.e., boundary) separating respondents from nonrespondents is the line labeled H. The classification boundary is "optimal" in that it minimizes the classification error in the training data set. In the very simplest case, such as that depicted in the left panel of Figure 1, survey response in the training data is linearly separable by X_1 and X_2, and the estimated boundary H produces no in-sample classification error. It is easy to see in Figure 1 (left) that one could draw an infinite number of lines that would perfectly classify survey respondents and nonrespondents. This is where the maximal margin comes in. The maximal margin lines are depicted by the dashed lines labeled M. By definition, the separating boundary H bisects the region defined by the margin lines. The maximal margin classifier finds the maximal margins, such that the resulting separating hyperplane H is farthest from the training observations among all such hyperplanes (James et al. 2013). Observations lying along the margin are called support vectors. In Figure 1 (left), these are indicated by the open triangles and open circles. Moving the support vector observations only slightly would alter the margin and the resulting position of the separating hyperplane H. Once the boundary H has been estimated, one can apply it to in-sample (i.e., training) data, to test data, or to new data for forecasting. In Figure 1 (left), observations that fall below the estimated boundary H would be classified as survey respondents, while observations above H would be classified as survey nonrespondents.

Among the differences in various SVM classifiers, the two that we highlight here are (1) how they deal with classification errors and (2) whether the decision boundary H is a linear versus nonlinear function of the predictors. As the previous example illustrates, in the simplest scenario, we assume that the decision boundary is linear and that it perfectly separates the two outcomes in the predictor space. This assumption — e.g., that survey response versus nonresponse can be perfectly predicted by a linear function of X_1 and X_2 — usually does not hold in practice. An example of a decision

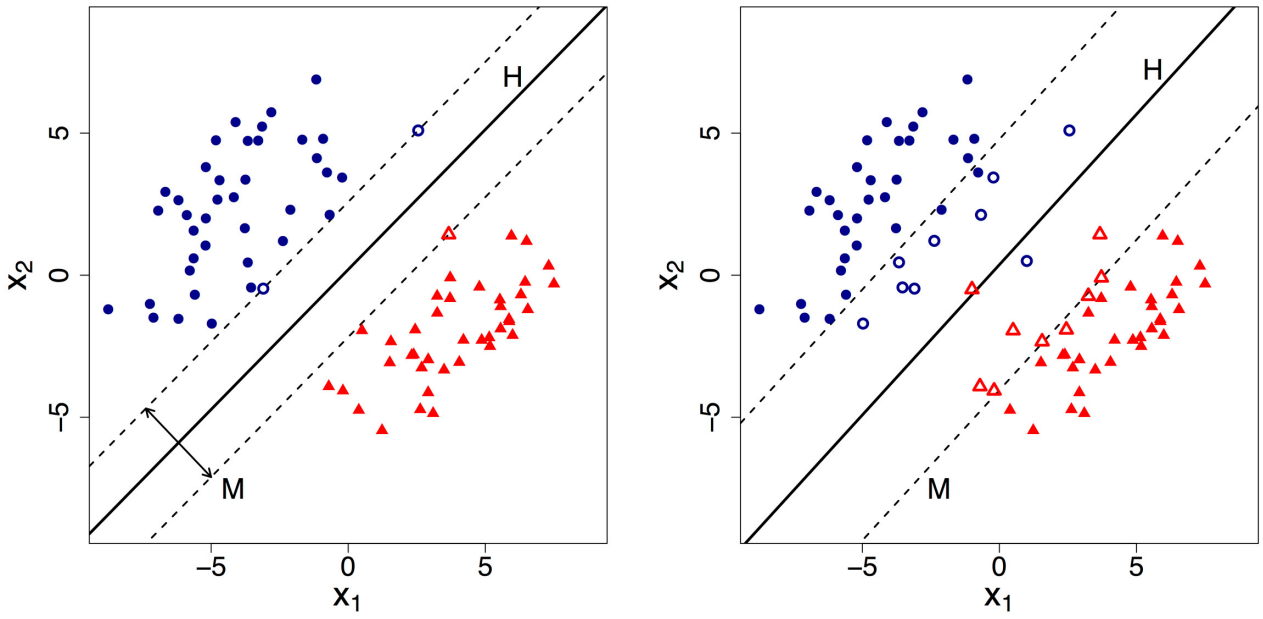

Figure 1 Linear SVM classification of survey response (red triangles) and nonresponse (blue circles), based on two predictors (X_1 and X_2). The classification hyperplane (H) and maximal margin (M) are shown in each plot. Support vectors are denoted by the open circles and triangles. LEFT: The decision boundary is linearly separable. RIGHT: Example of SVM classification using soft margins, when the decision boundary is linear but does not perfectly classify observations.

boundary that is not linearly separable is given in the right panel of Figure 1. As can be seen here, no (straight) line exists that would perfectly separate the survey respondents (red triangles) from the nonrespondents (blue circles).

One way to address this issue is to allow for a "soft margin" classifier (James et al. 2013). Contrary to a "hard margin" classifier, the soft-margin classifier allows for (1) observations that are correctly classified but lie between M and H, and (2) misclassified observations — i.e., those that fall on the "wrong" side of H. This technique employs slack variables, which keep track of the margin error for each observation. Finding the optimal boundary H now involves not just maximizing the margin, but also specifying a constraint on the total error *T* that will be allowed. Sometimes the soft-margin SVM optimization problem is recast to one where, instead of using *T*, a penalty *C* is used to weight the error, representing a trade-off between increasing the margin versus reducing misclassification in the training data. The total error *T* and penalty parameter *C* are inversely related to the optimal margin. Large penalties *C* (small allowances *T*), lead to smaller margins. Smaller penalties *C* (large allowances *T*), lead to larger margins, but also more misclassifications in the training data. In practice, the penalty *C* (or allowance T) is usually set through k-fold cross-validation.[1]

1 Readers should be aware that the letter "*C*" is frequently used in various SVM articles, books, tutorials, and statistical packages to represent both of the tuning parameters *T* and *C* mentioned above. This can be confusing, because the total error *T* and the penalty *C* are inversely related in their effect on the estimated margin. As an example, James et al. (2013, 112:346) use *C* to represent the total "budget" for errors (which we denote as *T*). On the other hand, the R kernlab package, which we use for our analysis and highlight in Table 1, uses *C* to

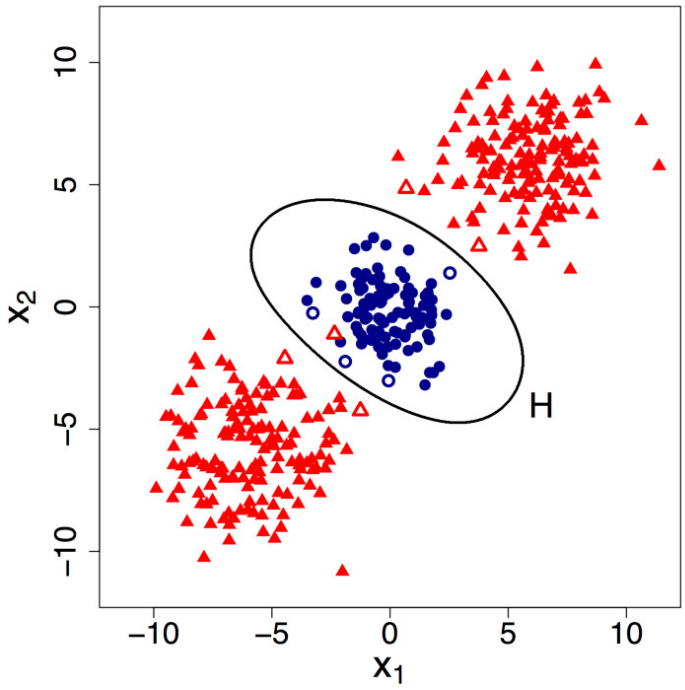

Figure 2 Nonlinear SVM classification of survey response (red triangles) and nonresponse (blue circles), based on two predictors (X_1 and X_2). Here, the SVM classifier employs a radial kernel and soft margins, producing the classification boundary H. Support vectors are denoted by the open circles and triangles.

Figure 1 (right), shows an example that is exactly the same as in Figure 1 (left), but with two observations (toward the middle of the plot) that do not allow for linear separability. An SVM model with C=1 (not shown) produces the same margins M and separating plane H as in the left panel. Figure 1 (right) shows the margins and separating plane when C=.01. In this case, there is a smaller penalty for margin violations, so the margins are wider than in the left panel. In addition to observations that lie on the margin, those that fall on the wrong side of their margin are considered support vectors, since changing any of these would change the margin and resulting boundary hyperplane.

A second issue that arises in SVM classification concerns whether the classification boundary is a linear function of the predictor variables. In Figure 1, it seems reasonable to assume a linear boundary line H. However, the linear assumption is restrictive and may not be appropriate for every application. Figure 2 illustrates the case where survey responses are nonlinearly related to the predictors X_1 and X_2. It is clear here that no (straight) line can be drawn that reasonably separates respondents from nonrespondents, even if we allow for misclassifications. Most SVM routines now allow for nonlinear transformations of the predictor variables. Typically, one can choose from linear, polynomial, spline, and radial basis functions of

represent the penalty for errors. Because the SVM optimization problem can be presented in two equivalent ways, but with inversely related tuning parameters, researchers applying (or reading about) SVMs should be particularly careful in determining which specification has been used and how to interpret the tuning parameter in that context.

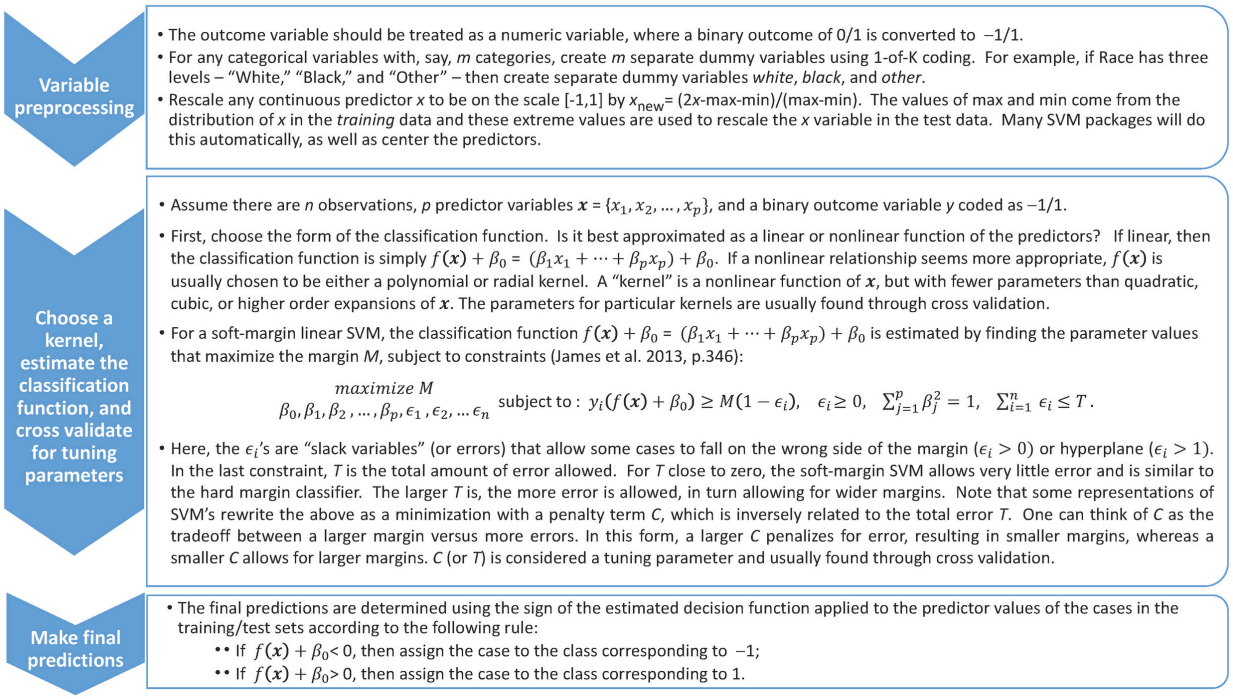

Figure 3 Steps in constructing a support vector machine

the predictors, among others. In Figure 2, we estimated an SVM with a radial kernel and allow for soft margins. Although the data is not perfectly separable even with the nonlinear kernel, we can see that the classification boundary H does a good job of classifying the training data. Any points within the ellipsoid H are classified as nonrespondents. Those outside are classified as respondents. The support vectors are again denoted by open circles and open triangles.

These are just a few simple examples of support vector classifiers/machines. Figure 3 provides more specific details of how SVMs work, including how they estimate the hyperplanes and decision functions, as well as how they can be used to generate predictions. Much of the notation relies on the very accessible introductions to SVMs that can be found in Bennet and Campbell (2000) and in James et al. (2013). An overview of software that can be used for estimating SVMs can be found at http://www.svms.org and http://www.kernel-machines.org/software. A list of R packages that can be used to estimate SVMs can be found at http://www.rdatamining.com. We highlight a few of the more popular R packages that can be used to estimate SVMs in Table 1.

Advantages and Disadvantages of Support Vector Machines

One of the most appealing aspects of SVMs is their versatility. Certainly, SVMs have been successfully applied to a wide range of problems including character recognition and text classification, speech and speaker verification, face detection, verification, and recognition, junk mail classification, credit rating analysis, and cancer and diabetes classification, among others (Attewell,

Table 1 Popular packages for implementing support vector machines in R

R Package	Brief description
e1071	This package provides R users with access to the very popular library of SVM tools and functions, libsvm, written by Chang and Lim (2016). Common kernels available include linear, radial, polynomial, and sigmoid and the SVM functions support both binary classification and regression applications. Multicategory outcomes can be used but only via a one-against-one approach in which SVMs are computed for each of the possible pairs of categories in the outcome. The package also includes functions for tuning support vector machines through cross-validation. https://cran.r-project.org/package=e1071
kernlab	This package offers a broad collection of kernel functions, many of which are not available in the e1071 package including hyperbolic tangent and LaPlacian among others. SVM functions support classification (binary and multicategory classes) and regression options. https://cran.r-project.org/package=kernlab
caret	The caret package provides a consistent framework for model calibration, cross-validation, and parameter tuning for SVMs and many other machine learning techniques. The actual SVMs are created by functions from the e1071 package for linear kernels and the kernlab package for all others. It also has the nice feature that it will automatically detect if a parallel environment has been initialized and then divide cross-validation across available computing threads, potentially greatly reducing computing time. https://cran.r-project.org/package=caret

Table 2 Additional advantages and disadvantages of support vector machines

Major advantages of support vector machines	Major disadvantages of support vector machines
SVMs are robust to observations that are far away from the hyperplane and are efficient since they are based only on support vectors within the hyperplane.	SVMs can be computationally intensive and require a large amount of memory to perform the estimation, especially if the data sets are large (Horváth 2003).
SVMs perform well in the "big *p*, small *n*" scenario – in other words, SVMs can successfully generate classifications in the presence of a large number of predictors even with a small number of cases in the data set.	For nonlinear applications the user must select a kernel to be used by the SVM. The choice of kernel and any associated hyperparameters that are required by the kernel must be carefully chosen, and an incorrect choice of kernel, especially, can negatively affect performance of the SVM (Horváth 2003).
SVMs can adapt to nonlinear decision/classification boundaries using the various kernel functions and provide solutions even when the data are not linearly separable.	SVMs can seem like black boxes in that a final functional form or a table of coefficients for various predictors is not provided as part of the estimation.
SVMs provide a unique solution unlike other machine learning methods that rely on local minima such as neural networks.	
Because SVMs are constructed using only the support vectors they may have better classification performance when applied to data that are unbalanced with respect to the binary outcome (Attewell, Monaghan, and Kwong 2015).	

Monaghan, and Kwong 2015; Byun and Lee 2002). However, the versatility of SVMs does not come for free, as these models can take considerable time to run, depending on (1) the number of observations within the data set and (2) the granularity of cross-validation for tuning parameters. SVMs have traditionally been used for binary classification, but recent advances have extended classification to categorical outcomes with more than two classes using one-versus-one or one-versus-all approach (see James et al. 2013 for more details). Despite the extension to multinomial outcomes, SVMs applied to outcomes with more than two classes can be computationally intensive depending on the number of categories of interest and the size of the data sets. We highlight the other major advantages and disadvantages of SVMs in Table 2.

Table 3 Results from logistic regression and SVM

	Logistic regression model Prediction			SVM model Prediction		
Actual	Nonrespondent	Respondent	Total	Nonrespondent	Respondent	Total
Nonrespondent	2,144	433	2,577	2,275	302	2,577
Respondent	860	848	1,708	634	1,074	1,708
Total	3,004	1,281	4,285	2,909	1,376	4,285

How Have Support Vector Machines Been Used in Survey Research?

While SVMs have gained in popularity, the empirical applications within survey or public opinion research are few. In addition to the applications in other disciplines referenced above, examples include a study conducted by Cui and Curry (2005) that explores the use of SVMs for "robust accuracy" in marketing where the main goal is predictive accuracy rather than structural understanding of the contents of the model. More relevant for public opinion research, Malyscheff and Trafalis (2003) use SVMs for substantive analyses to investigate decision processes in the Electoral College while Olson, Delen, and Meng (2012) investigate bankruptcy using SVMs. Lu, Li, and Pan (2007) have used SVMs for imputation applied to student evaluations and Christen (2008, 2012) for record linkage purposes.

Classification Example

Using the National Health Interview Survey (NHIS) Example training dataset, we estimated two models of survey response: a "typical" logistic regression model – i.e., with no interactions or nonlinear functions of the regressors – and an SVM model. Each included demographic covariates, such as *age*, *sex*, *race*, *region of country*, *income, ratio of family income to the poverty threshold, telephone status, education level*, and *type of employment*. We employed a soft-margin SVM with a radial kernel. To determine the SVM tuning parameter C and radial kernel tuning parameter γ, we conducted 10-fold cross-validation applied to the training data. The resulting cross-validated values for these parameters were $\gamma = 0.0189$ and $C = 32$. The predictors were also preprocessed by centering them and scaling them as illustrated in the example R code displayed in the supplemental materials.

Table 3 presents the confusion matrix for predicting response status computed by applying both the logistic regression and SVM models to the test data. The correctly classified cases fall along the main diagonal of the confusion matrix, while the misclassified cases fall along the off-diagonal. As can be seen from Table 3, both models correctly predicted the response status for the majority of cases.

Table 4: Various statistics of model accuracy for predicting response by applying the respective models, constructed using the training sample, to the test sample

Statistic (estimated using a 16% holdout test sample)	Main effects logistic regression model	Final SVM model
Accuracy (i.e. percentage correctly classified)	69.8%	78.2%
Sensitivity (i.e. true positive rate)	49.6%	62.9%
Specificity (i.e. true negative rate)	83.2%	88.3%
Balanced accuracy (mean of sensitivity and specificity)	66.4%	75.6%
Area under the ROC curve	74.2%	83.6%

As indicated by the various performance statistics presented in Table 4, the SVM model outperforms the main effects logistic regression model considerably. Specifically, the SVM model correctly classifies 78% of all cases compared to only 70% for the logistic regression model. Table 4 also shows that the true positive rate is considerably higher for the SVM model (62.9%) compared to the logistic regression model (49.6%). Smaller differences between the true negative rate were noted between the two models with the SVM model correctly classifying 88% of the nonrespondents compared to 83% for the logistic regression model. The overall area under the ROC curve was a full 10 percentage points higher for the SVM model compared to the logistic regression model as well.

REFERENCES

Attewell, P., D.B. Monaghan, and D. Kwong. 2015. *Data Mining for the Social Sciences: An Introduction*. Oakland, CA: University of California Press.

Bennet, K.P., and C. Campbell. 2000. "Support Vector Machines: Hype or Hallelujah?" *ACM SIGKDD Explorations Newsletter* 2 (2): 1–13. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.605.1310&rep=rep1&type=pdf.

Boser, B.E., I.M. Guyon, and V.N. Vapnik. 1992. "A Training Algorithm for Optimal Margin Classifiers." In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, edited by D. Haussler, 144-152. New York, NY: ACM Press.

Byun, H., and S.W. Lee. 2002. "Applications of Support Vector Machines for Pattern Recognition: A Survey." In *SVM '02 Proceedings of the First International Workshop on Pattern Recognition with Support Vector Machines*, 213–36. London, UK: Springer.

Chang, C.C., and C.J. Lim. 2016. "LIBSVM – A Library for Support Vector Machines." 2016. https://www.csie.ntu.edu.tw/~cjlin/libsvm/.

Christen, P. 2008. "Automatic Record Linkage Using Seeded Nearest Neighbour and Support Vector Machine Classification." In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 151–59. New York, NY: ACM Press.

Christen, P. 2012. *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Berlin/Heidelberg, Germany: Springer.

Cui, D., and D. Curry. 2005. "Prediction in Marketing Using the Support Vector Machine." *Marketing Science* 24 (4): 595–615.

Horváth, G. 2003. "Neural Networks in Measurement Systems (an Engineering View." In *Advances in Learning Theory: Methods, Models and Applications*, edited by J.A.K Suykens, G. Horváth, S. Basu, C. Micchelli, and J. Vandewalle, 190:375-396. NATO Science Series III: Computer & Systems Sciences. Amsterdam, The Netherlands: IOS Press.

James, G., D. Witten, T. Hastie, and R. Tibshirani. 2013. *An Introduction to Statistical Learning with Applications in R*. Vol. 112. New York, NY: Springer.

Lu, C., X. Li, and H. Pan. 2007. "Application of SVM and Fuzzy Set Theory for Classifying with Incomplete Survey Data." In *Proceedings of the IEEE International Conference on Service Systems and Service Management*, 1–4.

Malyscheff, A., and T. Trafalis. 2003. "Support Vector Machines and the Electoral College." In *Proceedings of the International Joint Conference on Neural Networks Portland, OR*, 2344–48. New York, NY: IEEE Press.

Olson, D.A., D. Delen, and Y. Meng. 2012. "Comparative Analysis of Data Mining Methods for Bankruptcy Prediction." *Decision Support Systems* 52 (2): 464–73.

Vapnik, V. 1995. *The Nature of Statistical Learning Theory*. New York, NY: Springer.

SUPPLEMENTARY MATERIALS

R-code example

Download: https://www.surveypractice.org/article/2715-using-support-vector-machines-for-survey-research/attachment/9166.zip