September 2017

Forecasting Consumer Spending Using Ensemble Support Vector Regression

Jack Youldon

Supervisor: **Prof Philip Treleaven**

Supervisor: **Prof Paul Ormerod**

Abstract

In this dissertation we investigate using kernelised support vector regression on an array of macroeconomic variables and financial prices to forecast growth in consumer spending 1-4 quarters ahead in the United States.

Economic forecasts are generally made using a few standard econometric techniques, with manual expert adjustments made to incorporate forward knowledge which is difficult to express in the data. Support vector regression and ensemble methods are very rarely, if ever, used in economic forecasting due to a lack of familiarity of the techniques within the community. This thesis provides an introductory section, where we build a ground-up introduction to support vector regression, a review of econometric approaches to forecasting, and an analysis of the similarities and differences between the two approaches.

We investigate applying machine learning techniques to the relatively short data series that are available in macro-economics and discuss how the 'small-data' problem influences our solution.

We conduct an experiment comparing the performance of machine learning models with the track record of human experts, documented by the Survey of Professional Forecasters.

We show that an ensemble approach, with stochastic parameterisation and sub-space sampling performs well and gives an excellent configuration when cross validation is difficult.

Using an ensemble of linear regressors our model was competitive with or outperformed contemporary expert predictions, particularly over forecast horizons greater than 2 quarters.

All code for this experiment is provided at https://github.com/jy247/Thesis

Contents

List of Figures

List of Tables

Chapter 1: Structure of this Thesis

In our introduction we discuss both aspects of this problem, considering the importance of economics and particularly consumer spending in our day-to-day lives, and the rapid advances in the field of machine learning, which may enable us to have fresh insights into old problems.

We then present a literature review, which charts and describes the development of Support Vector Machines (SVMs) as a tool for regression analysis. In this analysis we also emphasize the link between support vector regression and more classical kernel ridge regression, which is something more similar to the current state of the art in typical econometric forecasting. In a separate part of the review we provide an analysis of some important work that has already been done by econometricians to date, hopefully with some useful insights into the many similarities between the disciplines and areas where new perspectives may be beneficial.

After the literature review we provide a section on the methods used in the project, specifically referring to where and how the data was sourced, how the models were specified in python and what steps and precautions we took around feature selection, data mining and validation.

Following the section on methods we provide our results and analysis. The first part of the results is focused on individual regressor models, and then the second part considers ensembles and particularly linear kernel SVR ensembles. The discussion is placed among the results and conclusions are drawn in a separate section at the end.

Finally we present further work, which we believe would be valuable to consider, and actual forecasts for the next four quarters, using a variety of our best models.

Chapter 2: Introduction

2.1 Applying Machine Learning Advances to Econometrics

Econometrics is a well developed field of study in which participants take a rigorous statistical approach to analysis and forecasting in problems relating to aggregate decisions of economic agents. Economic agents can be loosely defined as entities (people, companies and governments), who contribute their labor, capital and resources to produce and consume goods and services.

Machine learning is of course at its heart also a statistical science, with a particular emphasis on the practical implementation of algorithms to best utilise modern hardware. Support vector regression is a type of regularised regression defined by the epsilon-insensitive loss function. This choice of loss function leads to several desirable properties including sparseness and stability and corresponds to a noise model, which we show to be appropriate for economic data. In this paper then, we seek to take the tool-set of machine learning and apply it to the specific domain of macro-economic forecasting.

Although the fields have a lot in common they have largely developed apart and so there is the potential that techniques commonly in use in machine learning circles are unused in econometrics simply due to unfamiliarity. That said, there have been numerous attempts to share knowledge between the disciplines, including notably by Google chief-economist Hal Varian (Varian [2014]), who focuses on the applications of big data to economics, and by Nyman and Ormeord (Nyman and Ormerod [2017]) whose work on predicting recessions is a direct precursor to this dissertation. We will provide a more rigorous overview of current work in the literature review, but it is apparent that the techniques which we explore in this paper are, if not unheard of, extremely rarely used in the econometrics community and it is our hope that at least some of what we do is entirely new.

2.2 Motivations

Consumer spending, also referred to as personal consumption, is a number, in dollars, which equates to the total amount of money spent by households on goods and services in a year. It includes everything that individuals spend their money on from housing to transportation, groceries to consumer durables to healthcare and personal financial products such as insurance. The number is an estimate, generated by collating responses to interview and diary surveys from approximately 7,000 and 14,000 respondents respectively. Data is published quarterly and subject to backwards revision until the final confirmation of the number, 6 months after the initial release.

Trends in consumer spending are important to companies who want to anticipate future demand. If a car company anticipates robust personal consumption in the future then they might choose to expand production. Likewise financial analysts valuing companies' bonds and shares will have their projections affected by anticipated changes in consumer spending. Beyond that consumer spending is an important indicator of the health of the economy overall, in the USA it makes up over 70% of total GDP.

Gross Domestic Product (GDP) is a somewhat vague if useful 'yardstick' number which measures the total value of economic activity in a country annually. It is typically used by governments and analysts to

understand whether an economy is growing or shrinking, to estimate tax returns, to judge whether national debts are sustainable and to calculate countries' relative contributions to international endeavours. So if the goal of policy makers is to ensure a steadily growing economy, what conditions should they seek to put in place in order to maximize consumer spending and what steps should they take in order to bring those conditions about? If we are able to address these question we would want to know if any hypothesized policy changes would be harmful to other parts of the economy and if they are likely to lead to permanent increases in GDP, or a temporary boost that would experience mean reversion? In effect we are speculating about the short-term and longer-term effects of changes in economic conditions and recognizing that there may be non-linear dynamics, which link the two.

For all of these reasons, the question of whether we are able to forecast consumer spending is an interesting one, and if we are able to build a model, can we understand how changing conditions effect our result over different time frames and how we might be able to change policy in order to encourage a sustainable increase in our target variable.

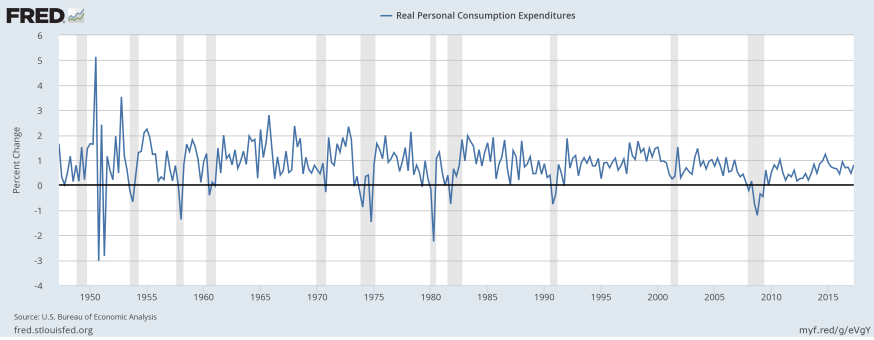


Figure 1: [1]Quarter on quarter percentage change in US personal consumption since 1950

2.3 The Failure of Forecasts

Despite the significant investment of time and effort that has been invested into the problem of economic forecasting the track record is relatively poor. The Survey of Professional Forecasters (SPF) allows us to analyze contemporary predictions and taking important observations from the literature (Ormerod and Mounfield [2000]) (Hendry and Clements [2003]) we note that:

- The mean SPF expert has never predicted a negative number more than one quarter ahead, and never until it was already known that the economy was in recession.
- The information content of the forecasts dwindles to virtually nothing, as little as 3 quarters ahead.
- Despite developments in the field and new models and techniques, the performance of forecasts has not improved over the length of the record.

All of these suggest that we must recognise there are likely to be limits to what can be achieved; Hendry and Clements quote Maxine Singer (singer [1997]) when they say:

Because of the things we don't know we don't know, the future is largely unpredictable.

[1]Sourced from Federal Reserve Economic Data Service (FRED)

They note the struggles between reality and theory in econometrics, whereby ensembles (or 'pooling' in econometric literature) seems to outperform single forecasts, despite their theory suggesting that the best individual model should be superior (Makridakis and Hibon [2000]).

They also mention that simple models have been shown to perform better, which appeals to the machine learning idea that in an environment of limited information, a simple model will be less likely to overfit and have a better generalisation error. The econometricians believe, however, that this out-performance is NOT a direct result of the simplicity of the models, saying that it is instead the adaptability of the models and:

It just happens that, to date, many adaptive models have been simple.

It is clear then that there is a history of underwhelming performance for economic forecasting and some philosophical differences, which give us reason to suggest that a machine learning approach might outperform. Further analysis of these differences is provided in the Literature Review 4.2.1

Chapter 3: Background

Regression is the process of providing an estimate for the value of an unknown variable y, based on a vector of known variables x. In the context of this thesis y is consumer spending and x is the set of chosen variables, which describe the state of the world at a particular point in time, such as the inflation rate, the savings rate and the unemployment rate.

In this section we try and describe from first principles how we specify and parameterise a model. We build up from there to a derivation of the dual form of kernelised support vector regression and try to provide some insights into the importance of different aspects of the solution.

3.1 Building a Model

It is not possible for us to create a fully representative model of the economy as it firstly depends upon the individual decisions of every person in the country and secondly depends on fundamentally unknowable aspects of the future, from the weather to new technologies and international politics, trade and conflicts. As discussed, even the target of our investigation is derived from a survey of a small subsample of the economic actors. Our goal in this thesis is not to generate something that looks like the economy itself as a model but merely to try and generate a regression on some of the important macro-economic variables and financial asset prices, which may contain information about the likely path of future consumption.
In particular we are interested in variables which may have a causal impact on consumer spending, or a leading correlation, and therefore contain some information about the future state. In reality there may be highly non-linear relationships, particularly as dynamic effects come in to play. For example if the Bank of England has a target for long term inflation of 2%, then changes in the future path of short term interest rates might have something more like a step function around this level. Interest rates and inflation, would each have their own causal influence on consumption as well as correlations caused by indirect effects, such as that of inflation on interest rates.

As the complexity of the model increases we require more information in order to accurately specify it. If the system is in a state, which is well represented by the training set, we might expect a simple model to perform well and a more complex model would only be needed as we move away from the modal target value. As we move away from this value we would expect our prediction to become more uncertain and there to be less training examples to specify our model. Therefore we can see that trade-off between model error and generalisation error is particularly relevant for outlying values.

Most economic forecasting models restrict themselves to linear factors, although some consider polynomial or other more elaborate relationships. There is a preference towards simpler given the constraint of small datasets, although we investigate both simple and complicated models. It is a sensible proposition to consider that there might be different regimes, for instance a 'normal' economy and a 'distressed' one, and this naturally leads us to using decision methods or ensemble methods. A separable 2 part linear model might then result in a sensible trade-off, allowing us to fit a simple model to non-linear data.

3.2 Regression Models and Cost Functions

As explained, regression is the model by which we generate our target variable y from our known variables x. In its most general form we want to define some function, mapping input data to forecast:

$$y_i = f(x_i) \tag{3.1}$$

In one of the most simple examples of a linear regression, this function might look like a weighted sum of the parameters of the input variable:

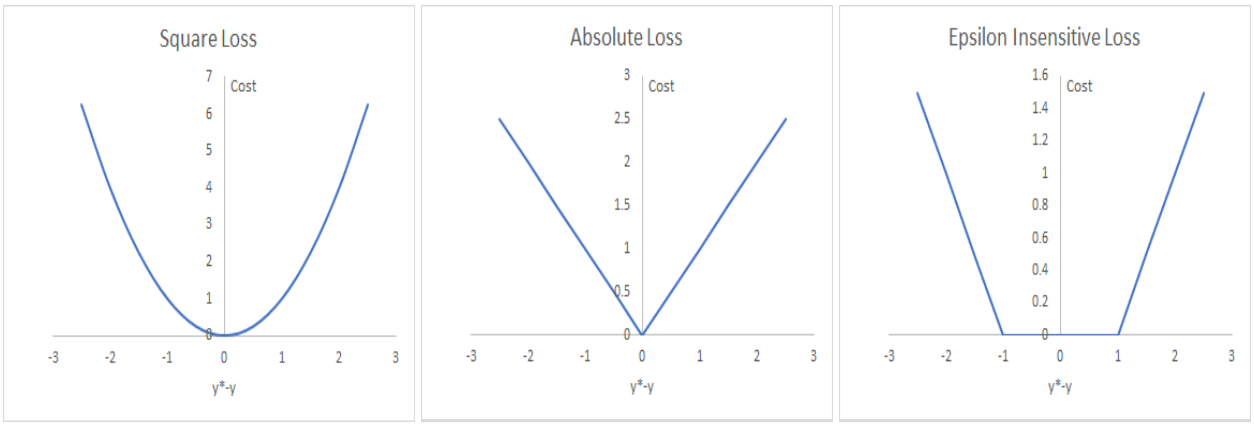
$$y_i = w^T x_i + b \tag{3.2}$$

In order to determine what is a good forecast and what is a bad forecast we need to define some kind of cost function. That is a function which takes the prediction and the true value and returns a number which is some indication of how far away your guess was.

$$c = g(y_i, y_i^*) \tag{3.3}$$

where y_i^* is the true value of the unknown variable for the i'th data point. In our example above the most typical cost function would be the mean squared error (MSE)

$$c = (y_i^* - y_i)^2 \tag{3.4}$$



(a) **(b)** **(c)**

Figure 2: Some Examples of Commonly Used Loss Functions

When we learn the parameters of the regression, in this case the coefficients in the weight vector, we try and do so such that it minimizes the cost function over the training set. This is called 'fitting our model'.

$$f(x) = \min_{f(x)} \sum_i^m g(f(x_i), y_i^*) \tag{3.5}$$

$$f(x) = \min_{f(x)} \sum_i^m (y_i^* - f(x_i))^2 \tag{3.6}$$

$$w = \arg\min_w \sum_i^m (y_i^* - w^T x_i + b)^2 \tag{3.7}$$

So f(x) is our model, it will define what parameters there are and how we apply them to the input data to get an output. And g(x) is our cost function, it will define how we optimise the parameters

There are two broad classes of model, different machine learning approaches, to solving this problem.

3.2.1 Curve Fitting

A set of weights w are generated during the training process. In the most simple case of a linear regression, these weights multiplied by the vector x provide an approximation to y. In more complicated regressions, the weight vector may correspond to some transformation or combination of the input vectors (see the section on kernel methods for more details), fitting a quadratic curve to a set of points might be the most simple example of this. A support vector regression, which this dissertation will focus on, is an example of curve fitting regression.
Advantages:

- All of the information is built into a prediction.

Limitations:

- Can be sensitive to outliers.
- Does not handle categorical data well.
- The relationship between the input dimensions and the target variable (e.g. linear, polynomial, exponential) needs to be specified as part of the model definition.

3.2.2 Decision Tree Methods

In this class of algorithms the prediction is generated by a series of discrete decisions as exemplified by the flow chart below. Once a sample has been classified into a particular branch by this process, a prediction might be generated by averaging training samples from the same class. A random forest is a good example of a decision tree method.
Advantages:

- Each dimension of the input data is evaluated independently and so this approach is resilient to features with very different scales or types, including a mixture of categorical and numerical data.
- Naturally non-linear, each decision is like a heavy-side step function acting on one variable in the input, as such it will be able to separate non-linearly separable data without extension.
- Fast, as the number of classes in a tree grows exponentially with increased depth, not many comparisons are typically needed in order to fully specify a test point. In addition, comparisons tend to be fast operations.

Limitations

- May struggle in very high dimensional or sparse datasets, which can lead to deeper trees with less populated leaves. This makes makes noise a significant factor as there are less values to average over.
- Not good at estimating out of sample test examples, the method has no concept of extrapolation, which a linear regressor would.

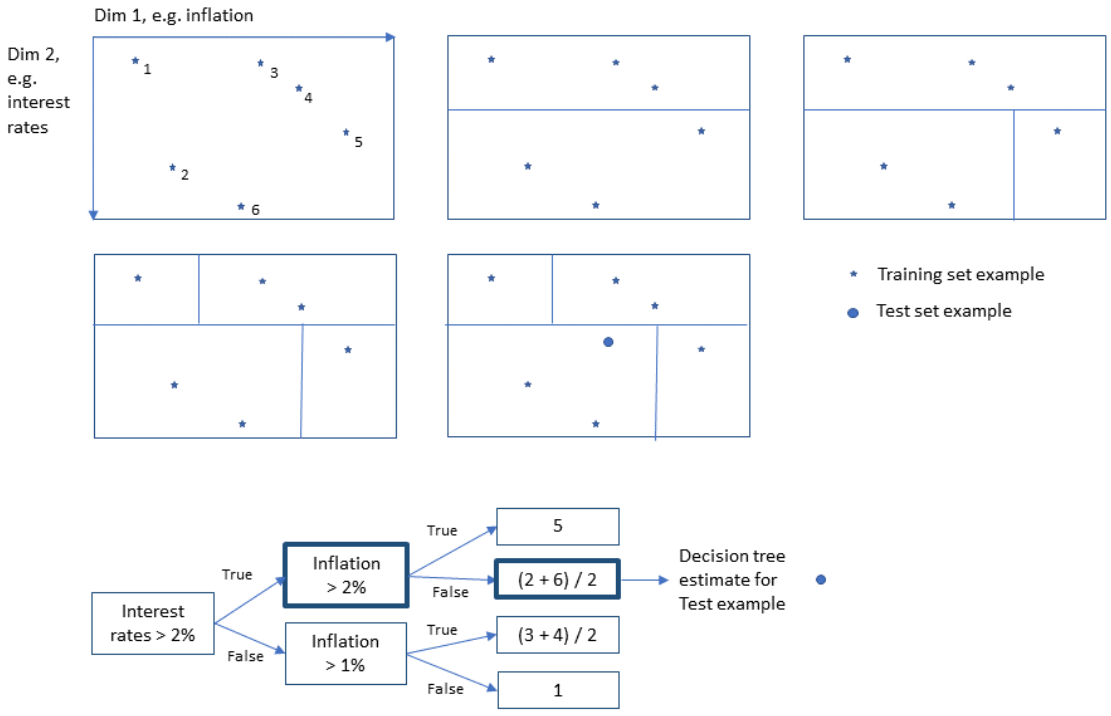


Figure 3: Example of single decision tree with 2 dimensions

3.2.3 Combining the Two

The two approaches are not entirely mutually exclusive. Some of the most powerful techniques are network models, each node of which effectively compares the input with a decision boundary, which is learnt like a curve fitting model, but then can be put through a non-linear function and squashed toward 0 or 1 so that it effectively forms a decision model. It seems intuitive that models that in some way combine these two approaches might be capable of performing the best. Although a multi-tiered solution would imply using more of the information in the data in model selection and less in noise reduction. The more complicated the model the more information is needed to accurately specify it, and so if there isn't enough information in the data we run a greater risk of overfitting.

3.3 Overfitting and Regularisers

Overfitting is one of the key concepts in machine learning, and handling this problem is vital to giving the model the best chance of making robust predictions on unseen data. A good way of thinking about the problem is that the observed y value is a combination of the true value and additive noise (equation:3.8). A model that overfits is one that has learned the noise as well as the signal in the data. The best that any predictor can possibly do, assuming that the model is absolutely correct, is to have an error of ϵ.

$$y_i = y_i^* + \epsilon \tag{3.8}$$

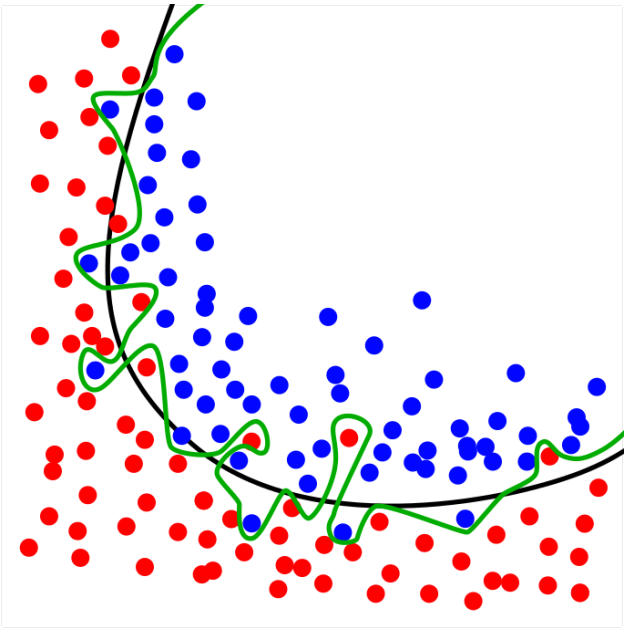


Figure 4: Overfitting - image of a regularised boundary and an overfitted boundary [2]

The black line here shows a regularised fit in this classification task. The green line shows a much more complicated decision boundary, which provides a perfect fit for the training data but is more likely to suffer from overfitting and generalise poorly to unseen test data. Fundamentally, the structure in the classifier (or regressor) should not be more complicated than the underlying true signal in the data.

Regularisation is the idea that you reward simplicity in the model so that only structure for which there is sufficient evidence will appear in the final result. In a curve fitting model this is normally imposed by adding a term to the cost function which represents the magnitude of the weight vector. This makes sense if you consider that each term in the input vector is likely to have some noise associated with it. When these terms are multiplied by the weight vector then the noise is also going to be multiplied and so large weight vectors will lead to large amounts of noise in the final prediction.

The most commonly used regularisers are the L1 and L2 norms of the weight vector, with the L2 norm perhaps the best in environments where we would expect the noise and signal to be evenly distributed among all dimensions. The L1 norm can be preferable when sparsity in the weight vector should be encouraged, e.g. when some dimensions are likely to have only a very weak influence on the final result and more likely to incorporate noise. In this experiment there are potentially many variables, which have little or no useful value; it makes sense, therefore to try and use L1 regularisation.

After incorporating a regulariser to our cost function for linear regression it becomes:

$$c = \sum_{i}^{m} (y_i^* - w^T x_i + b)^2 + \lambda w^T w \tag{3.9}$$

$$w = \arg\min_{w} \sum_{i}^{m} (y_i^* - w^T x_i + b)^2 + \lambda w^T w \tag{3.10}$$

[2]Image sourced from Wikipedia commons https://upload.wikimedia.org/wikipedia/commons/1/19/Overfitting.svg

Where λ is a hyper-parameter, which balances the preference for simplicity with the fit of the model to the training set. This will typically be set during training by using cross-validation. We would expect to see higher values of λ for noisy data and increasing lambda will typically increase the error in the training set, but (up to a point) will improve the generalisation error of the model.

3.4 Short Data Series

Machine learning is often thought of as being intrinsically a science of big datasets and one of the most well known articles in the field is 'The Unreasonable Effectiveness of Data' (Halevy et al. [2009]). This contribution to the conversation, by three Google researchers, argues that for natural language processing tasks, the advantages of having trillions of training examples (so called web-scale learning) outweighs the improved quality of the much smaller, annotated sets, which had previously be curated by academics. Support vector machines in particular are associated with large datasets, given that some of their main advantages are computational, particularly allowing the user to reduce the dimensionality of inference.

One of the defining difficulties of the economic forecasting problem on the other hand is the small datasets available for macro-economic time series (in section 4.2.1 of the literature review we reference Elliott and Timmermann's discussion of the problem). The training set runs from 1982 to 2010 and has 141 individual data points in it. Although there is an apparent contradiction between small datasets and machine learning techniques this not a hard constraint. There are a few guidelines that we can consider to maximize the value in the data (El Deeb), (Forman and Cohen [2004]).

- Limit the dimensionality of the problem - using feature selection outside of the model training procedure can help improve the signal to noise ratio in the data, and therefore make learning the signal easier in a small dataset.
- Use simple models - a commonly referred to 'rule of thumb' is that the model should have no more than$\sqrt{M}$ parameters, where M is the number of examples in the training set. The simplistic nature of this kind of rule is problematic, but if followed it would imply only 12 parameters can be supported by the 140 item training set.
- Take advantage of the ability to retrain the model many times - because the training set is small we can afford to use an expanding window and retrain from scratch for every test point.
- Use ensembles to train multiple times over the same dataset with different sub-sampling or different models.
- Use regularisation to help prevent over-fitting.

3.5 Solving Ridge Regression

By 'solve' we mean finding the w, which minimizes the cost function. It is very simple to solve this minimization for the squared loss by differentiating with respect to w and taking the minimum. In order to simplify the notation we represent the equation in matrix form.

$$c = (Y - Xw)^2 + \lambda w^T w \tag{3.11}$$

$$\frac{\delta c}{\delta w} = -2X^T(Y - Xw) + 2\lambda w = 0 \tag{3.12}$$

$$X^T X w + \lambda w = X^T Y \tag{3.13}$$

$$(X^T X + \lambda I)w = X^T Y \tag{3.14}$$

$$w = (X^T X + \lambda I)^{-1} X^T Y \tag{3.15}$$

Where 3.15 is the primal form of the solution for ridge regression.

3.6 Support Vector Machines

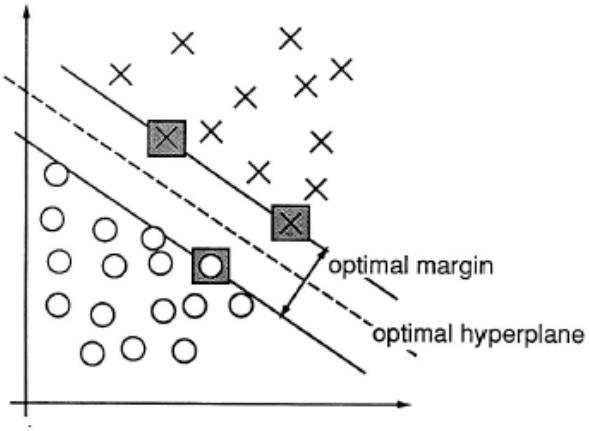


Figure 5: Optimal margin classifier on linearly separable data[3]

A support vector machine (SVM) is a classifier, which uses the optimal separating hyperplane, to assign data points into categories (Cristianini and Shawe-Taylor [2000]). SVMs were developed out of work on statistical learning theory by Vladamir Vapnik, introducing the idea of a margin, Epsilon (ϵ), defined as the minimum distance between the separating hyperplane and a data point. In the case of linearly separable data, that is data for which their exists a line (or a hyperplane in higher dimensions) which accurately classifies all of the training points, the optimal hyperplane is defined by the maximal separation margin (figure:5).
When the data is not linearly separable, the classifier is defined by the line which minimizes the hinge loss of the misclassified examples. The hinge loss is defined as:

$$cost = Max(0, abs(y_i^* - y_i) - \epsilon) \tag{3.16}$$

Where ϵ is the width of the margin from its centre. A graph of the loss function can be seen in figure 1. The margin width is a hyper-parameter that is set when defining the model and tells the regression to ignore noise of less than magnitude ϵ.

[3]Adapted from (Cortes and Vapnik [1995])

3.7 Support Vector Regression

Support Vector Regression is just a curve fitting regression model, using the ϵ-insensitive loss function. There are several nice side-effects of choosing this loss-function, however, because it is not differentiable everywhere (there are 'kinks' at +/- ϵ), the analytic solution cannot be found using a simple differentiation as in section 3.5. Instead a numeric solution can be found using convex optimisation solvers. A useful precursor to that can be to transform the equation into the Lagrangian dual form. A thorough examination of the steps required is available in Smola and Scholkopf's 'A tutorial on support vector regression' published in the journal Statistics and Computing (Smola and Vapnik [1997]). In the next sections we will run through the derivation of the dual form solution of SVR as an optimisation problem. In the subsection following the equations we will briefly explain what intuition we can gain from the final expressions.

3.7.1 Lagrangian form for minimizing SVR cost function

It starts by rewriting the cost function using slack variables ξ_i and ξ_i^*, which are defined as the scalar amount the error in each prediction is below or above (respectively) the true value plus the margin. The problem then becomes.

$$w = \arg\min_{w} \sum_{i}^{m} (\xi_i + \xi_i^*)^2 + \lambda w^2 \tag{3.17}$$

subject to

$$y_i - w^T x_i - b \leq \epsilon + \xi_i$$

$$w^T x_i + b - y_i \leq \epsilon + \xi_i^*$$

$$\xi_i, \xi_i^* \geq 0$$

Where the 'subject to' conditions can be understood as the definitions of ξ_i and ξ_i^*. Instead of using λ as the constant to balance the cost term and the regularisation term, the SVR traditionally uses C on the cost term. This is just a hyper-parameter which sets the relative weights of the two parts of the solution, so it is unimportant which is used, but for consistency with the material we will continue with C, which is proportional to $\frac{1}{\lambda}$

The essential key to the Lagrangian method is to incorporate the constraints and the objective function into a single function, called the Lagrangian, where the constraints are parameterised by constants called Lagrange Multipliers. It is guaranteed to be a problem, which is a lower bound of the original problem and is concave, so can be solved rapidly by convex optimisation methods.

$$L = \frac{1}{2} w^T w + C \sum_{i=1}^{m} (\xi_i + \xi_i^*) - \sum_{i=1}^{m} (\eta_i \xi_i + \eta_i^* \xi_i^*) - \sum_{i=1}^{m} \alpha_i (\epsilon + \xi_i - y_i + w^T x_i + b) - \sum_{i=1}^{m} \alpha_i^* (\epsilon + \xi_i^* + y_i - w^T x_i - b) \tag{3.18}$$

$$L = \frac{1}{2} w^T w + \sum_{i=1}^{m} ((C - \eta_i - \alpha_i)\xi_i + (C - \eta_i^* - \alpha_i^*)\xi_i^*) - \sum_{i=1}^{m} \alpha_i (\epsilon - y_i + w^T x_i + b) - \sum_{i=1}^{m} \alpha_i^* (\epsilon + y_i - w^T x_i - b) \tag{3.19}$$

Where $\alpha_i^*, \alpha_i, \eta_i^*, \eta_i$ are the Lagrange multipliers. Taking partial derivatives we get.

$$\frac{\delta L}{\delta b} = \sum_{i=1}^{m}(\alpha_i^* - \alpha_i) = 0 \tag{3.20}$$

$$\frac{\delta L}{\delta w} = w - \sum_{i=1}^{m}(\alpha_i - \alpha_i^*)x_i = 0 \tag{3.21}$$

$$\frac{\delta L}{\delta \xi_i^{(*)}} = C - \eta_i^{(*)} - \alpha_i^{(*)} = 0 \tag{3.22}$$

from 2

$$w = \sum_{i=1}^{m}(\alpha_i - \alpha_i^*)x_i \tag{3.23}$$

Substituting these back into the equation 3.19 one by one for clarity, starting with equation 3.22 to get.

$$\frac{1}{2}w^T w - \sum_{i=1}^{m}\alpha_i(\epsilon - y_i + w^T x_i + b) - \sum_{i=1}^{m}\alpha_i^*(\epsilon + y_i - w^T x_i - b) \tag{3.24}$$

Then substitute equation 3.23 for the first term

$$\frac{1}{2}\sum_{i=1}^{m}\sum_{=1}^{m}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)x_i^T x_j - \sum_{i=1}^{m}\alpha_i(\epsilon - y_i + w^T x_i + b) - \sum_{i=1}^{m}\alpha_i^*(\epsilon + y_i - w^T x_i - b) \tag{3.25}$$

Group the last two terms inside a single summation and then switch the sign inside the brackets and infront of the α_i^*:

$$\frac{1}{2}\sum_{i=1}^{m}\sum_{=1}^{m}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)x_i^T x_j - \sum_{i=1}^{m}(\alpha_i(\epsilon - y_i + w^T x_i + b) - \alpha_i^*(-\epsilon - y_i + w^T x_i + b)) \tag{3.26}$$

Now it is clear we can use 3.20 to get:

$$\frac{1}{2}\sum_{i=1}^{m}\sum_{=1}^{m}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)x_i^T x_j - \epsilon\sum_{i=1}^{m}(\alpha_i^* + \alpha_i) \tag{3.27}$$

which is the simplified expression for the dual form, i.e. it is expressed entirely in terms of the Lagrangian multipliers and can be maximized to provide an optimal (minimal) solution to the original problem in terms of α_i and α_i^*.

3.7.2 Karush-Kuhn-Tucker

The Karush Kuhn Tucker conditions (KKT) are an additional set of identities, which allow us to determine whether the point of zero gradient we have found is a genuine optimal solution to the constrained optimisation. They state that at the point of the solution, the product between dual variables and constraints has to vanish.

$$\begin{aligned}(C - \alpha_i)\xi_i = 0 \\ (C - \alpha_i^*)\xi_i^* = 0\end{aligned} \tag{3.28}$$

3.7.3 Qualitative Interpretations of the Derivation

There are two major insights that we can glean from the final form of this optimisation.

1. From equation 3.27 it can be seen that the solution includes the term $x_i^T x_j$, that is the dot product between two of the training examples, but it does not include any training data as a standalone term. This observation is crucial when it comes to kernel transformations of the input data, because the kernel function can be moved through the dot product, our solution only needs to expressed in terms of the original input space, rather than the potentially much higher dimension transformed space.

2. α_i and α_i* are the weights we are trying to learn in the dual form. From equation 3.28 we can see that α_i can only be non-zero when the expression inside the bracket is equal to zero. Due to the slack variable, this will always be true when the error between the true y and the prediction is less than ϵ. Furthermore as $\alpha_i \alpha_i* = 0$ only one of these can be non-zero for any particular training example. Put together this shows that the maths is accurately reflecting our construction of the problem, that the solution is uninfluenced by examples, which have error less than ϵ. This shows that the sparseness of the solution, that has been noted as a big benefit of support vector methods, is still valid for support vector regression.

3.8 Kernel Methods

It is trivially true that classifiers defined by lines can only separate linearly separable regions in feature space. Kernel methods allow us to surpass this limitation by transforming the data from the original feature space to a new feature space, typically with more dimensions, in which the data becomes separable (Müller et al. [2001]). A Kernel function is one which takes data items and transforms them into a new reproducing Hilbert space. Critically this transformation involves only the dot products of the features of the original data, the dot product of course can be thought of as a measure of similarity. When a new sample is tested then, the transformation into feature space considers, in some way, the similarity with every point in the data.

Examples of kernel functions which are in common use include the Gaussian (also known as radial basis function or RBF) kernel and polynomial kernel. The 'similarity score' given by the Gaussian kernel is equivalent to the value probability density function centered at the support vector training example and evaluated at the position of the test point. It would make sense to use this kernel in situations where there are not necessarily prescribed or predictable relationships between the input dimensions and the output variable, but where similar world states are likely to lead to similar outcomes.

The beauty of this approach is that it means that once our regressor is specified, it is expressible in terms of the input data, which limits the dimensionality of inference.

3.9 Random Forest Regression

The basic scheme for Random Forest regression is to have a series of decisions based on individual dimensions being greater than or less than a threshold value. Once a test sample is classified down to a certain level, defined in the implementation, an estimate of the target variable is produced by somehow averaging the training examples classified into the same set. The depth of the trees acts like a regulariser, where deeper trees correspond to more structure and therefore more likelihood of overtraining.

This simple tree structure is grown into a random forest by taking a large number of trees each of which only consider a subset of either the input data points or of the world state variables and averaging the values produced by each of those trees. This is an example of the ensemble method 'bagging'.

3.10 Ensemble Methods

3.10.1 Bagging

An example of bagging is described above with reference to random forest regression, however, it is also applicable to more complicated regressors, including SVR. The main advantage of bagging is to take a model which can vary greatly between trials and reduce the variance by averaging over a number of trials. Given the large variation we observed when making small changes to the parameterisation of our models, we decided that bagging was an appropriate technique to try and have described our implementation in section 6.9. It is also important to note that bagging as a concept is not restricted to combining equally weighted models, is is also possible to have an adaptive weighting scheme to generate the final output, as we have done in the final part of our experiment.

In an extremely approachable and yet wide-ranging paper (Domingos [2012]), Pedros Domingos refers to the power of ensembles in his tenth point, in particular he talks about the power of combining different models into a single ensemble, rather than just the same model trained multiple times on different subspaces of data or world state. For more in-depth information on bagging a good reference is available online from the University of Washington (Buja and Stuetzle).

Using ensembles of SVMs is not a new idea, although typically the ensembles are constructed of sub-spaces of the input dimensions (Kim et al. [2003]). Using ensembles of SVRs with random paramterisations is less common in the literature, although we believe it may hold value in cases where cross-validation will not necessarily produce reliable results (section: 6.6).

3.10.2 Boosting

Boosting is another related set of methods for combining multiple models into a larger model. In bagging each model is fitted to the original underlying training data, whereas in boosting the models are fitting sequentially either to the residuals of the previous models or to training examples somehow weighted by the fit of those models. The goal is to increase the power of relatively undescriptive models by building a more complicated aggregate model. Given that we are more worried about overfitting and a lack of information in the data than models without enough descriptive power, boosting did not immediately appeal to us as a technique to improve our experiment.

Chapter 4: Literature Review

We have divided this literature review into two main subsections, one, the development and theory of support vector machines and two, econometric approaches. Having already presented an introduction and derivation of some of the important results in regression and specifically SVR we believe it is instructive to review the original papers, which introduced some of these techniques, to enable us to pay specific attention to those aspects of the tool, which the authors identify as distinguishing them from previous results. Having highlighted these points and tried to suggest why the characteristics of SVR might equip us well to tackle economic problems, we go on to specifically review the state of the art in econometric forecasting. It is an area of huge significance, both practical and academic, and so it is of no surprise that econometrics has been an area of active development for the last 50 years. We examine some review articles and specifically try and draw comparisons, both similarities and differences between how the two separate fields of econometrics and machine learning have tried to approach fundamentally very similar problems. The last paper we review is perhaps of the most direct relevance to this piece of work as it is an application of SVR to financial data, in this case I seek to draw parallels with my own application and conclude.

4.1 The Development and Theory of Support Vector Machines

4.1.1 A Training Algorithm For Optimal Margin Classifiers, Boser, Guyon, Vapnik, 1992

Support Vector Machines were formulated primarily out of the work on statistical learning theory of Vladamir Vapnik and were presented as we know them now in this paper (Boser et al. [1992]). Along with Bernhard Boser of UC Berkley and Isabelle Guyon of Bell Laboratories, Vapnik presented SVMs as a classifier for data, which was perfectly linearly separable (after a kernel transformation) with several advantageous properties, all of which derive from two properties of the SVM.

1. The classifier can be fully represented by just those points closest to the decision boundary.
2. The optimisation problem, which identifies those points, is globally convex and as a consequence can be determined quickly and reliably.

The benefits derived from this are multiple. They identify that one aspect of the problem of model selection in machine learning is balancing a trade-off between choosing a model that captures every detail of the training set and one that generalises well to unseen data. They point out that SVMs take this decision out of the hands of the person using them, because of (1) they will dynamically increase or decrease the number of parameters in the model based on the number of points required to fully specify the decision boundary, in this respect they have some inbuilt regularisation.
The authors frequently compare SVM classification against building a decision boundary using a sum of square errors (SSE) cost function. They tell us that because of (2), the SVM will always find an errorless classification on the training set where one is available. This is similar to the perceptron but not the SSE classifier. They are also able to use simple logic to provide an estimate for the generalisation error of the method where

$$E_{gen} \leq \frac{num\ support\ vectors}{num\ training\ vectors}$$

This is intuitive because any point that is wrongly classified during training will become a support vector and then be correctly classified.

Another point, which they present as a benefit but may be positive or negative depending on the use case is that, because the classifier is specified by only a few data items, it will either completely ignore, or be strongly effected by, outlying bad data. They point out that when building a decision boundary using a typical square loss error function, every point in the training set will have only a small effect on the position of the boundary, bad data is more likely to remain hidden. **This argument is interesting but it implies a certain amount of post processing of the data is required if the SVM classification turns out to not generalise well.**

A great emphasis of the paper is the efficiency of the method computationally and this is a result of both points (1) and (2). The classifier can be trained sequentially on batches of the training set and we know that we will get the same decision boundary at the end as if we had trained on the whole set. Furthermore the paper goes through the derivation of solving the kernalised problem in dual form, because only a few of the points are significant to the boundary this greatly improves the efficiency of evaluating test data in the dual form.

4.1.2 Support-Vector Networks Cortes, Vapnik, Machine Learning 20(3), 1995

Three years after the original paper Cortes and Vapnik published a follow-up, presented in the journal Machine Learning in 1995 (Cortes and Vapnik [1995]). This paper elucidated two important extensions to the method:

1. The idea of the SVM as a component in a network. As explained in section 3.2, some of the most powerful machine learning models combine linear decision surfaces in layers, to build a piece-wise linear model. SVMs in layers can still build non-linear decision surfaces (in the input space) by using kernel methods, and then by chaining SVMs the power of this approach can be enhanced further. This is compared favorably with chained perceptrons and linear neural networks, the former which would not necessarily find an optimal decision surface, and the latter, which does not enjoy the power of kernel transformation.

2. The 'soft-boundary' extension to allow the method to function in the case of non-separable data. Cortes and Vapnik specify a new form for the cost function, which can be minimized to find the optimal hyperplane.

$$\frac{1}{2}w^2 + CF(\sum_{i-1}^{l} \zeta^{\sigma})$$

   Where w is the n dimensional vector of weights specifying the optimal hyperplane. C is a positive constant hyper-parameter set to balance the weight of the solution between minimizing the errors in the training set and minimizing the complexity of the solution. F is a monotonic convex function worked through with the example $F(u) = u^2$, ζ are the error terms, equal to the ϵ-insensitive loss (figure 1)

It was this idea of using slack variable and presenting the SVM as a Lagrangian optimisation problem, which enabled practitioners to use SVMs for regression problems, rather than just as classifiers. Indeed the format

of the loss function looks exactly like that of a more typical regularised kernel ridge regression, where the instead of a mean square error loss a ϵ-insensitive loss (sometimes 'Vapnik' loss) is used. Over the next few years researchers explored this new perspective, clearly the ϵ-insensitive loss was a powerful device, but why, and when in particular is it appropriate?

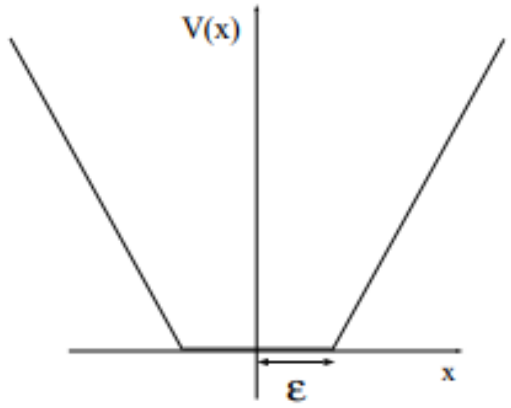


Figure 6: Vapnik's Insensitive Loss Function [4]

4.1.3 On the Noise Model of Support Vector Machine Regression, Pontil, Mukherjee, Girosim, 1998

Three years later, Massimiliano Pontil, Sayan Mukherjee and Federico Girosi, presented this informative paper, sharing their insights to these questions (Massimiliano Pontil Federico Girosi [1998]). They perform a Bayesian analysis to generate a general form for the loss function in terms of the probability distributions of the mean and the variance, which they then solve for the ϵ-ILF. They demonstrate that regular regression with the MSE loss can be generated by assuming that each data point suffers additive Gaussian noise with zero mean, but that the ϵ-ILF is a more nuanced picture of the noise, with every data point target value having a Gaussian addition, but with means and variances for this Gaussian drawn from the probability distributions demonstrated in (fig:2).

Roughly speaking, this corresponds to a mean uniformly distributed between +/-ϵ and the variance is a unimodal distribution which is independent of ϵ. This distribution of the mean is intuitively appealing, if the targets are equally likely to have a biased error anywhere within this region, then it makes sense to ignore errors of that magnitude.

[4] Adapted from (Massimiliano Pontil Federico Girosi [1998])

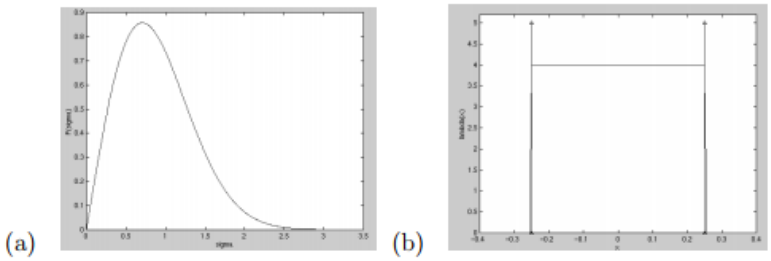


Figure 7: [5] **a) The probability distribution of the variance of the noise where $\sigma^2 = \frac{1}{2\beta}$ and $P(\beta) = \frac{C}{B^2} e^{\frac{1}{4\beta}}$**
b) The probability distribution of the mean of the noise for $\epsilon = 0.25$.

The practical conclusion of the paper is that if you are expecting your data to have Gaussian noise, which may not be centered around zero or be of fixed mean and variance between observations, then Support Vector Regression is an appropriate tool to pick. Beyond that we can inform our choice of ϵ as we now understand that it represents the range of possible biases in the mean of the Gaussian noise effecting the result.

4.2 Econometric approaches

While reviewing the economics literature for instances of using machine learning for forecasting we found that the two most common areas of application were:

1. Financial markets - large amounts of data in time series make forecasting financial asset prices a naturally tempting target for researchers. Neural networks and most recently recurrent neural networks are the tool of choice in most cases, but in a few cases we found people have experimented with support vector machines. Notable examples include *Forecasting stock market movement direction with support vector machines* (Huang et al. [2005]), *Financial distress prediction using support vector machines: Ensemble vs. individual* (Sun and Li [2012]) and *Estimating GARCH models using support vector machines* (Pérez-Cruz et al. [2003]), which we have analysed below.

2. 'Now-casting' - that is using alternative data sources to try and estimate data for the current period, but before data (or revisions) are released by the relevant agencies. The papers we looked at included:

   - (Fantazzini and Toktamysova [2015]), which looks at the ability to forecast German car sales using Google search data.
   - (Duarte et al. [2017]), which looks at forecasting retail sales using data from ATMs and point of sale software.
   - (Golinelli and Parigi [2004]), which looks at the predictive information in consumer sentiment surveys across countries.

   Although we are primarily focused on more forward looking forecasts, these are interesting and useful contributions and at the margin can help provide better longer term forecasts as they can work to improve the accuracy of the input parameters to the regression in real time. Furthermore, these now-casts can help resolve the 'jagged-edge' problem, which describes having some of the variables on

[5] Adapted from (Massimiliano Pontil Federico Girosi [1998])

interest available before others, and therefore only a partial world state for the current time period. It seems unlikely, however, that these kind of massive 'live' datasets are going to help predict consumer spending 4 quarters ahead, and even less likely they will contain information beyond that contained in the aggregate economic numbers.

The kind of longer term economic forecasting that is the subject of this investigation is rarely tackled using a machine learning approach, which is perceived to be better suited to the larger datasets available in these areas. A few attempts to apply neural networks to macroeconomic data can be seen in *Applications of Artificial Intelligence in Finance and Economics* (200 [2004]) and *Neural network versus econometric models in forecasting inflation* (Moshiri and Cameron [2000]). In these papers NNs are viewed purely as a way to introduce non-linearity into the forecasts, with little in the way of fundamental analysis as to why (and what type) of neural network would form an appropriate model for economics.

Having highlighted some of the relatively limited attempts to apply machine learning to economic forecasting we will now provide a more general overview of that field. In particular we will consider some of the similarities and differences to machine learning, because the more similar the fundamental content of the field the less sense it makes to say 'applying machine learning'. For example, if we apply a regularised least squares regression model, we could equally be considered to be 'applying econometrics'.

4.2.1 Economic Forecasting, Elliott, Timmermann, Journal of Economic Literature 2008

Graham Elliott and Allan Timmermann have produced two extensive review articles on the state of forecasting within the economics profession (Elliott and Timmermann [2008]), (Elliott and Timmermann [2016]). Some of the topics they touch on and the way in which they frame the problems are very familiar including:

4.2.2 The Similarities

- Vector Autoregression Models (VAR). This is basically the name for a linear regression model in econometrics. Autoregression refers to including the previous value of y (in time series models) in the prediction of the next value. The 'vector' part of the name indicates that the world state (x) is considered to be multi-dimensional, and not *just* consist of this previous y value.
- Cost (or Loss) functions. The authors go to particular effort to describe how the loss functions are considered to be related to the objective of the forecaster, so if the (economic) cost of overestimating the result is less than that of underestimating it, the cost function should be similarly asymmetric.
- Bayesian interpretations. Elliott and Timmermann present a thorough construction of regression as a Bayesian problem, although they correctly point out that the posterior distribution generated by the method contains little information beyond the point forecast, unless you believe that the form of the prior and likelihood terms have a fundamental explanation.
- They discuss possible methods for model selection and particularly parameter selection including principal component analysis, regression with a regulariser that encourages sparseness (the L1 norm) and ensemble methods, rerunning the regression multiple times with different subsets of variables.
- They present two approaches to dealing with non-linearity

1. A combination of linear models. They state that this is typically achieved by a combination of Gaussians representing the probability of y for each underlying state, weighted by an estimate of each state's probability. Structurally this sounds very similar to a neural network model. Another example of regime changing models are autoregressive conditional heteroskedasticity models (ARCH). These are very popular in asset price prediction as they build in 'volatility clustering', that is sustained periods of low variance followed by periods of high variance. As in machine learning, Hidden Markov Models are frequently used as the underpinning model for these regime shifts.
2. More complicated functional forms. They talk about various families of model that are in use in the economic literature including polynomials, splines, sigmoid functions and Fourier Series.

In both cases they describe the difficulties presented by higher dimensional models having a greater number of parameters to set and, therefore, overfitting on relatively short datasets. This is particularly a problem as non-linear models can change more rapidly and unpredictably for out-of-sample test examples causing ridiculous predictions. For this reason many practitioners build in a 'sanity test' to their models, which limit extreme values. Furthermore, as I suggested in the introduction, they highlight that there is even less data available for regimes in which the economy is acting in a 'non-typical' fashion, which makes it very hard to specify these parts of the model.

"*Nonlinearites do seem to be present in many macroeconomic series, but the data samples for these variables tend to be relatively short, thus hampering the precise estimation of nonlinear forecasting models.*"

4.2.3 The Differences

- There is no mention of the use of the kernel trick in the treatment of non-linear models. This was a big advance from the field of machine learning, which allowed users to limit the dimensionality of the model.
- There is no mention of using the ϵ-insensitive loss function and no suggestion of support vector machines as a tool for regularisation.
- Elliott and Timmermann only briefly touch on Dynamic Stochastic General Equilibrium models (DSGE) but they are an important separate approach used by central banks and other serious forecasting operations. They build up large scale models of the economy from underlying theories concerning the component parts and then stochastically sample from these models. This is a much more deterministic way of creating a forecast, which puts more emphasis on domain specific theory than on pure data analysis.

In the updated paper the two economists published in 2016, they repeated some of the same background information as in the earlier article, but they also include new information, including detail on specifically how large datasets are transforming economic forecasting. Some of the interesting points they raise are:

- There are three ways to increase the amount of data available to the model.
    1. Increase the length of the time series, although this makes you more vulnerable to model instability as regime shifts take place in the real economy.

2. Increase the frequency of the time series. This is a very popular solution in financial markets, where the data is often tick by tick, but less so in macroeconomics, where more frequent data might just introduce noise, which is averaged out over longer periods. This, along with the potential for more immediate financial rewards, is why a lot of the attempts to apply machine learning in this field hitherto have been in financial markets.
3. Increase the array of variables which constitute the world state. Again they reference the pros and cons of this decision, balancing the curse of dimensionality against including every weak predictor that can add extra information. The solutions they propose again are PCA and L1 regularisation.

- There is no one model, which always outperforms the others. Dynamic model selection is possible but raises more issues around data mining. A mixture of models, specifically an equally weighted mixture is likely to be a close to optimum solution.

4.2.4 Prediction Using Several Macroeconomic Models, Gianni Amisano, John Geweke, Working Paper for the ECB (2013)

This paper (Amisano and Geweke [2013]) is of particular interest as it uses some of the same data series as are used by this thesis, including US personal consumption, and compares the actual performance of the different models reviewed by Elliott and Timmermann on real economic forecasting. It is of further interest because one of the key conclusions of the paper is that the best results were derived by 'pooling' probability estimates from the different models. This strongly correlates with our findings that ensembles over different model parameters outperformed any single model.

4.2.5 Estimating GARCH models using support vector machines, Fernando Perez-Cruz, Julio A Afonso-Rodriguez and Javier Giner, Quantatitive Finance Volume 3 (2003)

This paper (Pérez-Cruz et al. [2003]) details a relatively early attempt to apply support vector regression to the problem of predicting financial time series data and can be read as an introduction to SVMs for the econometric community. Sequential prices are assumed to be generated by a stochastic process consisting of the mean plus an additive noise term. The target of the forecast is not the change in the price itself but rather the variance of a price and the prices from the four previous periods. The variance is modeled according to:

$$\sigma_t^2 = \omega + \alpha y_{t-1}^2 + \beta \sigma_{t-1}^2 \tag{4.1}$$

where ω is the bias term and is to be learned in training along with α and β.

The authors initially trialled the model on data generated by simulation and found that, when using a purely Gaussian noise model, the least squares maximum likelihood better explains the out of sample data. When they tried additive noise generated by a Student t distribution, however, which has fatter tails (higher kurtosis) the SVR started to outperform. This ties in with the authors expectations and our previous insights into the noise model of SVR versus LS regression.

Once they try it on real financial data (stock indices and individual stocks prices) they find that SVR outperforms LS in 5 out of 6 cases. They note that the kurtosis of daily log difference stock returns is relatively high, which gives them some justification to expect that SVR would outperform.

One further issue they note is that many of the evaluation metrics which are typically used in econometrics are based off the expectation that errors are additive Gaussian. This biases the metrics towards favoring a model which assumes the same and minimizes the error accordingly. We need to be careful then, when evaluating the success or failure of the predictions, that we choose metrics which accurately reflect the desired properties of the fit, and don't necessarily have built in assumptions about the noise.

4.3 Conclusions from the Literature Review

To recap the most important points:

- The optimal separating hyperplane derived from support vector regression can be determined quickly and reliably, is memory and processor efficient, exhibits some robustness to bad data and has inbuilt regularisation.
- SVR assumes a noise model of additive Gaussian noise, where the Gaussian mean has equal probability of being between $+/-\epsilon$ and the variance is drawn from a uni-modal distribution.
- Economic data series often exhibit kurtosis and therefore there is reason to assume that SVR might outperform LS regression.
- Econometricians have investigated various ways to introduce non-linearity into their predictions, some of which look like machine learning approaches, kernel SVR does not appear to be well established as one of these methods.
- All of this gives us reasons to be optimistic that applying kernel SVR to the forecasting problem might yield useful results.

Chapter 5: Data

5.1 FRED

The Federal Reserve Economic Data service, or FRED, is a set of nearly 0.5 million economic data series made publicly available by the Reserve Bank of St. Louis economic research team (St. Louis Fed). As well as being accessible through the website, FRED provide a simple RESTful webservice, which allows users to search for series and return data. We were able to use one of the recommended python clients, which is freely available at Github (mortada).

It is important to note that the data available in FRED is the best available number for each date and series. This means that any revisions to the data, which are made after the event are represented in the data whereas they would not be available to contemporary forecasters. This issue is well known in the econometric community (Koenig et al. [2003]) and suggestions for approaching it are considered in the section on ideas for further work (9).

5.2 The Target

Real Personal Consumption Expenditures (FRED KEY - PCECC96): The target variable is quarter on quarter percentage change in personal consumption, seasonally adjusted and inflation adjusted. Figure 8 shows a histogram and compares it to a Gaussian distribution with the same mean and standard deviation. You can see that the true distribution of personal consumption is somewhat narrower than the Gaussian. The visual check is confirmed by the calculated kurtosis, which is 5.75, higher than 3, showing that the distribution is Leptokurtic. As discussed in the prior sections, support vector regression is thought to work well in domains where the target variable has a fat tailed noise model.

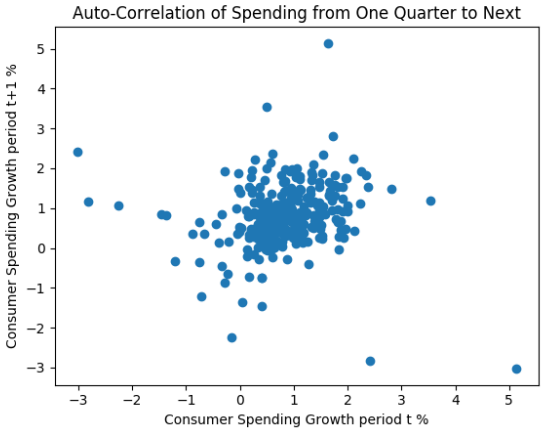
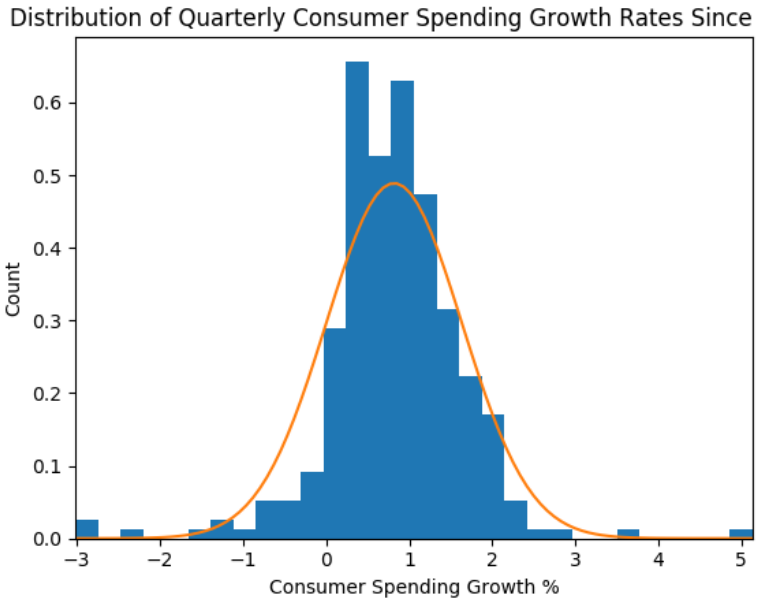


Figure 9: Plotting growth rate for one period against the next shows a weak positive correlation of 0.096

Figure 9 shows that there is very little information, to predict what the value of the next period's growth number will be, in this period's growth number. There is a slight positive correlation, which implies that the tendency for the numbers to be similar from period to period is slightly greater than the tendency to



mean	0.82
std	0.82
kurtosis	5.75

Figure 8: Histogram of consumer spending growth rates fitted with a normal distribution

revert to the mean, but we can see that we will need to find other variables to include in our regression and we certainly should not expect a plain autocorrelation to produce useful results.

5.3 The Regression Variables

We initially selected 17 variables that might have some predictive power on personal consumption by browsing through the available data series on the FRED website. These variables are listed in table 1 with some information about the series itself and some calculated correlations. The 'Series Name' is just a textual description of the data, the 'Series Type' is either 'lin', which is short for linear and just corresponds to the absolute value of the field at each timestep, or 'pch', which is short for percentage change and corresponds to the change of the variable from the preceding period. As the target is a percentage change it makes sense for almost all of the descriptor variables to be percentage changes as well. We want the regression to learn the immediate information, rather than the long term trend in the data. For example if we include a stock index price, which on average increases 10% a year, the fact that it is has doubled over 10 years will be less significant in predicting the next quarter's personal consumption growth, than if it has dropped by 20% in the previous quarter.

Four correlation numbers are presented for each series:

- 1 fwd - correlation between the variable and the next period consumer spending (CS) growth.
- 4 fwd - correlation between the variable and the CS growth in 4 periods time.
- Delta 1 fwd - correlation between the difference between the variable's value for the current period and the previous period, and the next period CS growth.

- Delta 4 fwd - correlation between the difference between the variable's value for the current period and the previous period, and the variable and the CS growth in 4 periods time.

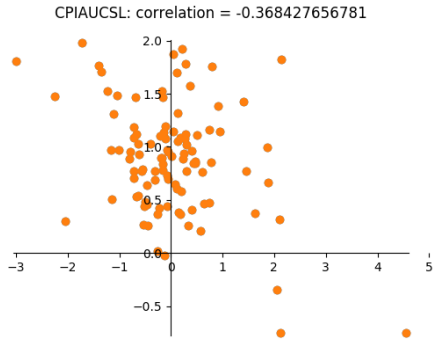
SERIES INFO			CORRELATIONS			
FRED Key	Series Name	Series Type	1 fwd	Delta 1 fwd	4 fwd	Delta 4 fwd
PCECC96	personal consumption	pch	0.191	-0.066	0.001	-0.04
TWEXM	dollar index	pch	-0.061	-0.138	0.057	-0.072
HOUST	housing starts	pch	0.284	-0.053	0.044	-0.04
TOTALSL	total consumer credit	pch	0.084	-0.02	0.004	-0.064
CIVPART	labour participation rate	pch	0.117	0.096	-0.035	0.026
POP	total population	pch	-0.171	-0.092	0.023	0.098
UNRATE	unemployment rate	pch	-0.182	0.015	-0.07	-0.033
PSAVERT	personal savings rate	lin	0.087	0.081	0.223	-0.027
W875RX1	real personal income	pch	0.2	0.001	0.041	-0.03
TOTALSA	total vehicle sales	pch	-0.114	-0.16	0.01	-0.054
M2	m2 money supply	pch	0.16	-0.014	0.069	-0.063
T10Y3M	10Y 3M T spread	pch	-0.078	-0.129	0.047	0.022
UMCSENT	consumer sentiment	pch	0.22	0.021	0.223	0.113
USTRADE	retail employees	pch	0.33	0.228	0.073	-0.001
CPIAUCSL	inflation index	pch	-0.368	-0.116	-0.115	-0.132
WTISPLC	oil spot price	pch	-0.219	-0.145	-0.074	0.019
WILL5000INDFC	broad equity index	pch	0.24	-0.029	0.082	0.059

Table 1: All input variables with correlation to target

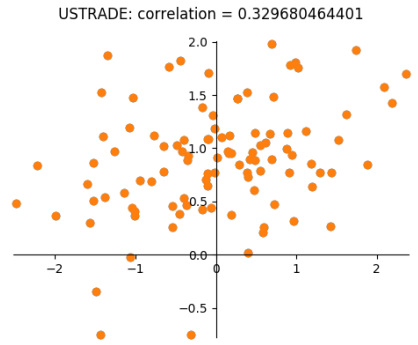
All of the correlations in the table were calculated over the training period 1982-2007. Many of the variables have longer time series, however, in order to have complete data for every period we decided to begin the training period in 1982. It is important not to include the test period in this section because if we decide to do dimension reduction based on this analysis, including the test set would constitute looking ahead.

5.3.1 A Few Example Correlations:

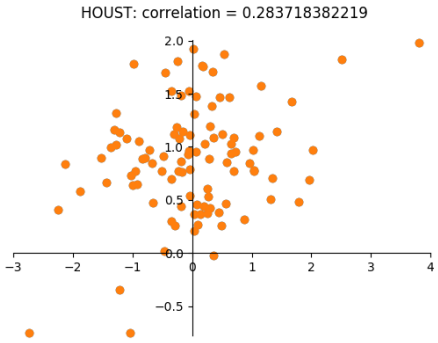
From this quick look at the correlation between the input variables and the target we notice that there are a few variables which seem to have a relationship. If the variables are assumed to be normally distributed, we can calculate that a 95% significance level over this sample size corresponds to a correlation of greater than around 0.2. It is important to bear in mind, therefore, that some of these relationships might be pure chance. Not withstanding this, the most significant relationships appear to be:



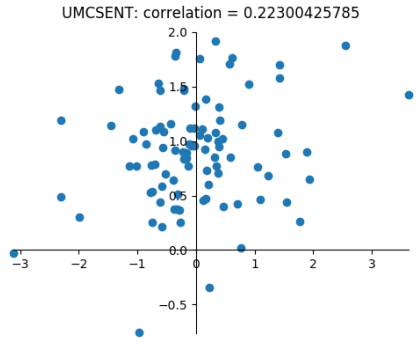
CPIAUCSL: The Consumer Price Index measure of inflation has a negative correlation with consumer spending even including the 4 period ahead forecast and looking at the second derivative in the change of the price index. This makes sense, because the consumer spending numbers we are targeting are inflation adjusted, a high inflation number increases the denominator and so will mechanically cause consumer spending to come down.



USTRADE: The change in the number of employees working in the retail sector seems to have a reasonably significant positive correlation with an increase in sales the following quarter. This suggests that companies are generally good at predicting increased demand in the short term. The correlation also exists, although not as strongly, with broader measures of employment, the participation rate and unemployment rate. It is likely there is some duplication of information between these variables (collinearity).



HOUST: An increase in the number of housing starts seems to be a relatively good predictor of future increases in consumer spending. Again this may be due to direct factors, housing costs are a part of consumer spending, or it may be due to housing starts increasing as part of a sign of a wider uptrend in the economy, which coincidentally occurs with people having increased income and spending.



UMCSENT: Consumer sentiment, unsurprisingly, had a positive correlation with increased consumer spending in the periods ahead. Consumer sentiment surveys can be performed continually and so are very often used to try and solve the now-casting problem; this result shows that it also appears to have longer term forecasting power.

5.3.2 Intuitive Relationships:

Most of the other relationships are either weak or in the direction that would make sense from an economic point of view. Money supply and personal income are both positively correlated with the 1 quarter ahead. It is intuitive that consumption increases when the either the income of consumers is increasing or the aggregate debt levels are (which M2 represents). Interestingly the personal savings rate is also positively correlated, although we might expect people to save less if they are spending more, apparently if they are spending more they are also saving more. This indicates that overall growth in the economy is a more important factor than the relative shift between saving and spending.

5.3.3 Four Quarters Ahead:

In general very few of the variables have a strong relationship with the change in consumer spending four periods ahead. In a way this is not surprising, we are not looking at the aggregate change between now and 12 months time but specifically the change between 9 months and 12 months. Many aspects of the future are unknown so any ability to specify this number beyond random chance should be well received.

5.3.4 Non-Linear relationships:

This analysis of the correlations does not consider all of the possible relationships between the data and the target variable. Non-linear or higher order relationships in one variable would still be distinguishable on a scatter plot but might not show up in the correlation numbers. Multi-variate relationships may also exist but would not show up in either the scatter plot or the correlation number. For example, an increased saving rate when unemployment is going down might indicate a strongly growing economy with consumers storing up purchasing power for the future, whereas an increased savings rate with unemployment going up might signify a shrinking economy with consumers saving their dollars out of fear for the future. Linear regression is unable to express relationships like this whereas kernel methods are. This is because the kernel function will generally including 'cross-terms' which are the product of the test samples dimensions with the other dimensions of the training set samples.

5.3.5 Discarding Weak Predictors:

As a result of regularisation, support vector regression should be able to discard input variables, which don't add significantly to the fit over the training set. This does not mean it is a good idea to throw in lots of irrelevant variables, especially when dealing with relatively short time series as we are here. As discussed above, the absence of a correlation between a dimension and the target variable is not necessarily proof that it will not add useful structure to the solution when combined with other dimensions. We experimented running the regression with and without the weaker predictors. As discussed in section 6.9, deltas and seasonal data were not considered when running ensemble methods.

Chapter 6: Implementation

Python's Numpy and Scikit Learn libraries make it very easy to train and run the models without needing to re-implement them. In general we found these were sufficiently powerful and flexible, with our EnsembleSVR.py script the only occasion in which we chose to implement our own functionality. This made it easier for us to combine many models with different hyper-parameters, instead of just multiple copies of the same model, trained over different datasets or input data.

6.1 Python Scripts:

1. **Main.py:** Entry point for experiment, configure experiment parameters and test.
2. **Valid.py:** Methods to handle a grid search over the model parameters using a validation set.
3. **EnsembleSVR.py:** Wrapper for handling ensemble methods.
4. **DataHelper.py:** Load data from the FRED webservice and save to local file for faster retrieval.
5. **PlotHelper.py:** Draw charts.
6. **ModelHelper.py:** Construct regression model and save model configuration.
7. **AutoRegression.py:** Preliminary investigation into the target variable.
8. **DisplayFactors.py:** Preliminary investigation into the input data variables.

6.2 Kernel functions

6.2.1 Linear

A linear kernel is equivalent to no kernel, each dimension of the input is treated independently and of first order, not capable of expressing non-linear relationships.

6.2.2 Polynomial

For polynomial you also specify the degree, for example degree 2 will include the square of each dimension as well as each dimension crossed with every other. Degree 3 would have each dimension cubed, plus each dimension squared with every other, plus every combination of 3 first order terms.

6.2.3 Gaussian/RBF

This kernel function is sometimes called the radial basis function kernel because the similarity term is based on the Euclidian distance in n dimensional space from the centre of the training example. It is also known as the Gaussian kernel because that similarity score decays like a Gaussian function as you move further away from the centre point. The parameter γ specifies the width of the Gaussian and thus the rate of decay of influence of training points on the solution. As gamma increases the influence of far off points drops more rapidly, the effect is similar to the k nearest neighbour algorithm. As gamma increases only the closest points in world state space effect the test prediction, equivalent to decreasing k. The extra hyper-parameter increases the search space for parameterisation of the model, although heuristics can be found, which typical enable users to outperform linear kernels (Keerthi and Lin [2003]).

6.3 Scaling Parameters

The support vector machine typically does not perform well in situations where dimensions of the input data are on completely different scales. If we are looking at the Euclidean distance between two points and one of the dimensions contains numbers several orders of magnitude larger than the others, then even small variations in this dimension will swamp large changes in the others. In order to nullify this problem we scaled the parameters using the Scikit Learn standard scaler. This maintains the distribution of values in the training set for each variable but scales them so that they have zero mean and unit variance.

It is important to note that the exact same transformation must happen to the test set points as the regression weights that have been learned during training depend on the scaling of the input data.

6.4 Hyper-Parameters

6.4.1 Epsilon

ϵ - the width of the insensitive region around the correct predictions, any guess within this much of the true value is treated as correct when fitting the model. Increasing the value regularises the solution as we have fewer points effecting the outcome and only pay attention to large deviations from the fit.

6.4.2 C

C - equivalent to $\frac{1}{2\lambda}$ in ridge regression, C is the parameter specifying the relative weights of the fit term and the norm of the weight vector. A decrease in C increases the regularisation of the solution.

6.4.3 Gamma

γ is a hyper-parameter, which is only relevant to the RBF-kernel and determines the rate of decay of the impact of training points on the value assigned to a test point. See section 6.2.3 for a complete description.

6.4.4 Ensemble Methods Parameters

Once we start considering ensemble models, put together as and aggregate of individual regressors, we introduce a new set of hyper-parameters that can effect performance. These include:

- The number of regressors in the ensemble
- The number of dimensions to consider for each regressor (in sub-space sampling)
- The number of training items to consider for each regressor (in bootstrap aggregation)
- How to combine the predictions of each regressor (equally weighted or some other weighting scheme)

6.4.5 Random Forest Parameters

All of the hyper-parameters we specified for ensemble methods naturally apply to Random Forest. In addition there are some specific params, including the maximum tree depth, and the loss function to use to best split each feature. Because this investigation is primarily into SVR and we are only using random forest as a comparison we have almost used the SKLearn random forest regressor straight out of the box, which gives

reasonable default parameters without the need optimise every facet. The only hyper-parameter we changed was the number of decision trees in the ensemble, which we increased to 50 to be in line with the other ensembles, which we tried.

6.5 Rolling Window Versus Expanding Window

We experimented with two modes of defining the training set:

Expanding window: Retrain the model at every step t including all of the examples up to t-1. We would expect this model to outperform if the relationship between these variables and the growth in personal consumption is effectively fixed. It generally makes sense to use as much training data as possible, particularly in situations like ours because with a time series of only 160 periods and a large amount of internal variation, it may be that it is hard to find training examples which accurately represent the current state. **All of the results stated in section 7 are generated using an expanding window.**

Rolling window: Retrain the model at every step with a window of say the last 20 years of data. We would expect this to outperform if the economy effectively transitions through different underlying true states, which are not fully represented by the data in the model. This means that examples of the relationships between macro-variables and personal consumption in the recent past are far more relevant predictors of the current relationship than examples from other periods.

6.6 Cross-Validation

It is important to have a validation set as well as training and test sets, the idea being that you can optimise the parameters of the model without data mining by peeking into the test set at training time. In order to do this most effectively we have used a two stage grid search over cross-validation.

A grid search means choosing an a priori feasible range for your parameter, in the case of SVR this would be ϵ and C, and then breaking those ranges into intervals and testing at every combination of the parameters. Cross-validation means breaking the training set into a number of parts, say 5, and then training over 4 of those parts and testing on the last. You repeat this 5 times, each time testing on one of the parts and training over the rest. This can be a fairly laborious way to optimise, however, with small datasets it is practical and has the advantage of being thorough, a smaller dedicated validation set would be more likely to result in parameters, which were optimised only for that set.

We performed the grid search in two steps, dropping the step size down by a factor of 10 after the first stage. This means we can drill down on a more precise optimum while testing fewer parameter combinations.

Scikit learn provides a cross-validation library, which makes breaking the set into equal pieces and keeping tracking of the optimum simple. Instead of using a grid-search it is sometimes possible to use a gradient based method such as gradient descent, this can be much faster, but relies on you having some idea of the shape of the cost function is response to changing the hyper parameters. If this problem is non-convex, or worse non-linear, then the optimisation would be more difficult.

6.7 Stochastic Parameterisation

Instead of using cross-validation to find a single optimum parameterisation, another option, when using ensemble methods, is to select random parameters (from a feasible range) for each regressor and trust the induced randomness to average out over the ensemble. As well as saving the time associated with cross-validation, this can potentially reduce the bias, which might result from a bad parameterisation, which would result from differences between the train and test sets.

6.8 Performance Metrics

The way we evaluate our forecast is of course intrinsically linked to the cost function we started off with in (section: 3.2). Although it might seem obvious to use the hinge loss given that this is what we are optimising for in SVR, part of the reason the hinge loss is attractive is because of the sparsity and inbuilt regularisation it lends to the dual form. It is, therefore, reasonable to consider a range of error metrics alongside the hinge loss error, we have included.

6.8.1 Mean Square Error

Symmetric, penalises very bad predictions more heavily.

$$E = \frac{1}{M}\sum_{i=1}^{i=M}(y_i - y_i^*)^2 \tag{6.1}$$

6.8.2 Mean Absolute Error

Symmetric, penalises all discrepancies equally.

$$E = \frac{1}{M}\sum_{i=1}^{i=M}|y_i - y_i^*| \tag{6.2}$$

6.8.3 Epsilon Insensitive Error

$$E = \frac{1}{M}\sum_{i=1}^{i=M} Max(0, abs(y_i^* - y_i) - \epsilon) \tag{6.3}$$

6.8.4 Correlation

The Pearson correlation is defined as:

$$corr(X,Y) = \frac{\sum_{i=1}^{i=M}(X-\mu_X)(Y-\mu_Y)}{(m)\sigma_X\sigma_Y} \tag{6.4}$$

Where μ_X is the mean of X, and σ_X is the standard deviation of X. This correlation coefficient is a measure of whether two distributions are above or below their mean values at the same time and is independent of the scale of the two distributions. It varies from 1, which means perfectly positively correlated to -1 or perfectly negatively correlated. Perfectly positively correlated means that whenever series X is above its mean value, series Y is also above its mean value by the same amount, relative to their respective distributions.

6.8.5 R^2 - Coefficient of Determination

The coefficient of determination or R^2 Dougherty et al. [2000] is a measure of the explained variance a model has for data, defined as:

$$R^2(X,Y) = 1 - \frac{\sum_{i=1}^{i=M}(Y_i - X_i)^2}{\sum_{i=1}^{i=M}(Y_i - \mu_Y)^2} \tag{6.5}$$

It is similar to the correlation coefficient in that it is a measure of fit of one series to another and it has a maximum value of 1, which means perfectly fitted. The difference is that the R^2 does not tolerate one series to have a different scale to the other, which correlation does, so that if you doubled the value of every X, the correlation would be unchanged, but the R^2 would get worse (or possibly better if the scale of the variance was too small originally). R^2 is then in some sense a more challenging metric, and a value of zero means that each prediction is on average the same distance from the true value as the mean and prediction of the mean value at every timestep will therefore, have a R^2 of zero. If the fit is worse than that the R^2 can be negative and arbitrarily large. Note that a R^2 of zero is not indicative of there being zero information in the predictions, as there is no reason that the mean of the test set would be known a priori.

6.9 Ensemble SVR Implementation

During the course of the experiment we noticed that small changes in the model hyper-parameters could have large and unpredictable effects on the goodness of the fit of the model to the test data. We decided, therefore, to try combining multiple models with different parameters, in order to reduce the variance of the final solution and hopefully average out any artifacts.

6.9.1 Random Models

We combined 50 separate regressors in the ensemble model and each one of these had its own randomly assigned selection of the input dimensions and hyper-parameters. The input dimensions for each model were chosen at random without replacement as it would make no sense to replicate input dimensions. We generated random values of C and epsilon that were in the typical range of values, which were found to be useful in the previous cross validation. This was done using the following code:

```
self.ensemble_indices = []
for i in range(n_ensemble):
    self.ensemble_indices.append(np.random.choice(n_total, size=n_each, replace=False))

 random_c_exp = np.random.rand(n_ensemble) * 3
 random_epsilon = np.random.rand(n_ensemble) * 0.4
 for i in range(n_ensemble):
     c = 0.1 * 10 ** random_c_exp[i]
     epsilon = random_epsilon[i]
     model = mh.get_model(model_type, c, epsilon)
     self.models.append(model)
```

6.9.2 Sub-Space Sampling

Using different random sub-spaces of the world state for each regressor is a good way to try and get different regressors to focus on different dimensions of the input data and therefore allow the ensemble to tease out information from those dimensions with weaker predictive power, rather than just being completely dominated by the same one or two dimensions. We investigated using a range of different numbers of dimensions in the optimisation stage of the experiment.

For the ensemble section the 'delta' dimensions and the dimensions corresponding to the seasons were never used. The information in those dimensions is less than in the primary values and so when sub-sampling it was decided to exclude them.

6.9.3 Weighting Regressors Based on Cross-Validation

The same code that was to used to handle the cross-validation for the non-ensemble methods was used to help us weight the individual regressors within the ensemble. Each one generated a score based on the goodness of the fit during cross-validation. They were then ordered based on this score from worst to best and assigned weights in blocks based on their position in the list. The first block were assigned a weight of zero, so they were effectively pruned entirely from the ensemble, the next groups were assigned linearly increasing weights based on their block and then all of the weights were scaled so that they sum to one. The code to do this is included below:

```
for i in range(self.n_ensemble):
    model = self.models[i]
    pipeline = make_pipeline(preprocessing.StandardScaler(), model)
    score = sum(cross_val_score(pipeline, x_train.iloc[:,self.ensemble_indices[i]], y_train, cv=
    all_scores.append(score)

sorted_indices = np.argsort(all_scores)
weight_index = 1
num_per_bracket = self.n_ensemble / NUM_WEIGHT_BRACKETS
for i in range(self.n_ensemble):
    if weight_index * num_per_bracket < i:
        weight_index += 1

    self.model_weights[sorted_indices[i]] = weight_index - 1

#make them sum to 1
total_weight = np.sum(self.model_weights)
self.model_weights = self.model_weights / total_weight
```

Chapter 7: Tests and Results

7.1 Cross-Validation Results

Periods Forward	Kernel	Cost Function	C	Epsilon	Gamma	Score
1	linear	MAE	0.006	0.07	-	-1.482
1	linear	MSE	0.009	0.4	-	-1.224
1	linear	R^2	0.009	0.4	-	-1.099
4	linear	MAE	0.001	0.0	-	-1.557
4	linear	MSE	0.005	0.18	-	-1.355
4	linear	R^2	0.005	0.2	-	-1.1
1	rbf	MAE	1.0	0.09	0.0011	-1.468
1	rbf	MSE	90.0	0.44	0.0001	-1.223
1	rbf	R^2	9.0	0.0	0.0001	-1.104
4	rbf	MAE	10.0	0.0	0.0001	-1.518
4	rbf	MSE	10.0	0.2	0.0001	-1.375
4	rbf	R^2	10.0	0.2	0.0001	-1.118
1	poly (2nd order)	MAE	0.1	0.0	-	-1.624
1	poly (2nd order)	MSE	1.0	0.9	-	-1.531
1	poly (2nd order)	R^2	1.0	1.1	-	-1.518
4	poly (2nd order)	MAE	0.08	1.1	-	-1.511
4	poly (2nd order)	MSE	0.001	0.3	-	-1.481
4	poly (2nd order)	R^2	0.1	1.1	-	-1.608
1	poly (3rd order)	MAE	1.2	0.32	-	-1.604
1	poly (3rd order)	MSE	0.12	0.33	-	-1.318
1	poly (3rd order)	R^2	0.12	0.33	-	-1.011
4	poly (3rd order)	MAE	0.07	1.1	-	-1.534
4	poly (3rd order)	MSE	0.001	0.3	-	-1.478
4	poly (3rd order)	R^2	0.001	0.3	-	-1.645

Table 2: Cross-validation results

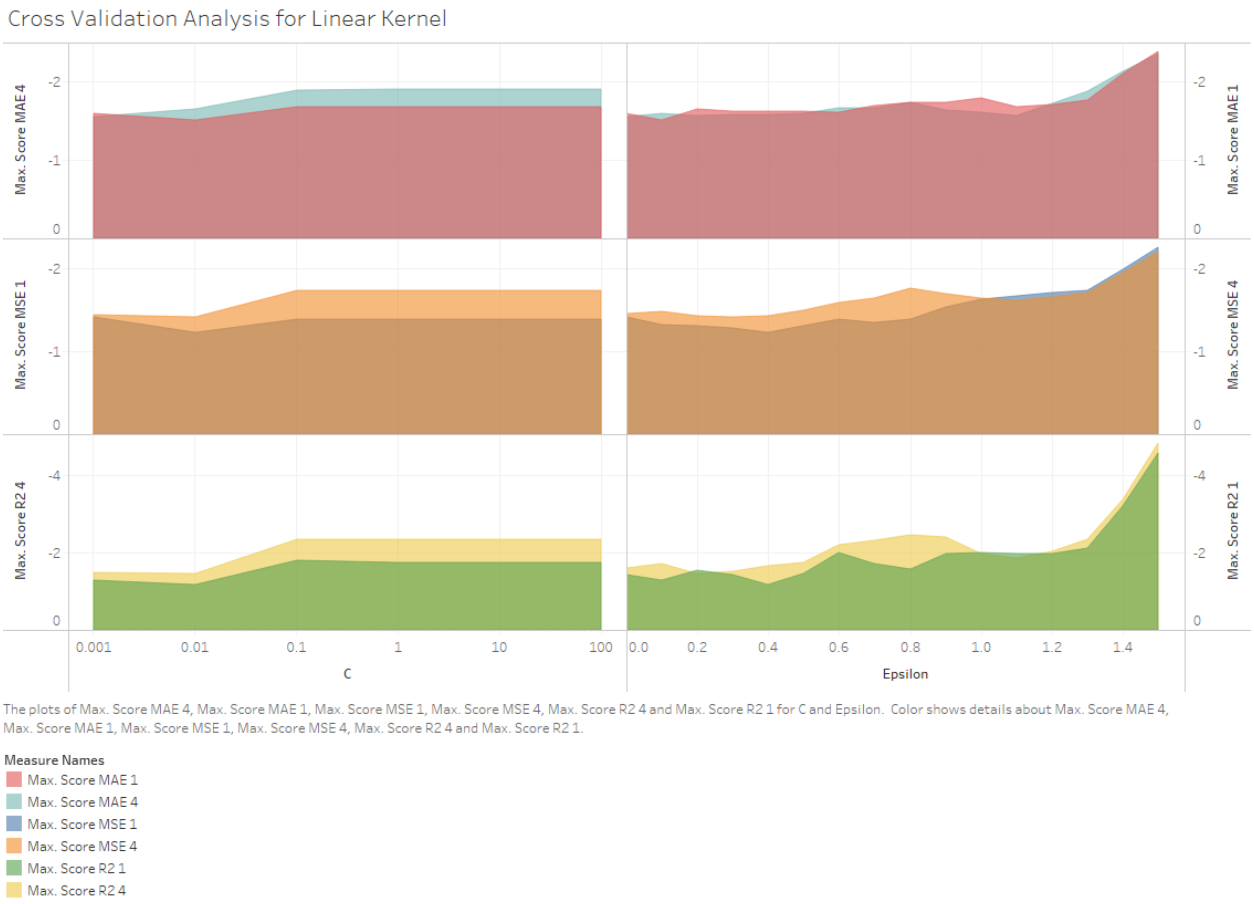


Figure 10: Min cost of 1 and 4 quarter forecasts for cross-validation with different cost functions

Discussion:

- All of the loss metrics are worse for the 4 quarter ahead forecast than the 1 period ahead, this backs up the idea that there is more information about the immediate future in the data than further ahead.
- The 4 quarter ahead forecasts do not vary much in accuracy as you change the hyper-parameters, they should not, therefore, be used for cross-validation.
- For the RBF kernel, two of the cost functions recommended an optimum epsilon of 0. With epsilon $= 0$, SVR just becomes kernel ridge regression with an absolute error loss function. This is a configuration that should be investigated but is of less interest because it loses the advantageous properties of SVR.
- The loss function that seems to be most sensitive to a change in hyper-parameters is the R^2 for that reason we used this loss function in the cross-validation for production test runs.

7.2 Single Regressor - Forecast Results

7.2.1 1 Period Forward

Kernel	C	Epsilon	Gamma	MSE	MAE	Mean ϵ-IE [6]	Correlation	R^2
linear SVR	0.009	0.4	-	0.2534	0.3832	0.0868	0.5695	-0.1269
rbf SVR	12.0	0.0	8e-05	0.2983	0.3966	0.0981	0.3615	-0.3269
poly2 SVR	1.0	1.1	-	0.3048	0.3868	0.1006	0.1225	-0.3556
poly3 SVR	0.12	0.33	-	0.2758	0.3913	0.0927	0.5383	-0.2268
poly4 SVR	0.07	1.2	-	0.2982	0.3792	0.1025	0.0103	-0.3263
Random Forest	-	-	-	0.3125	0.4395	0.1078	0.392	-0.3901
linear SVR	0.0012	0.0	-	0.2838	0.4601	0.0915	0.5429	-0.2622
Mean Experts	-	-	-	0.1634	0.2938	0.0467	0.6166	0.2731
Test Set Mean	-	-	-	0.225	0.346	0.073	0	0

Table 3: Forecast accuracy 1 period ahead

Discussion: For the single regressors (non-ensemble) the contemporary predictions of the experts of the Survey of Professional Forecasters (SPF) were better than the models predictions under every metric. Furthermore the model failed to predict the recession that came with the financial crisis, with only the RBF kernel SVR predicting a negative number at any point in the test (this prediction was accurate, but at the end rather than at the start of the recession, so of less importance).

The linear kernel and the third degree polynomial showed the highest correlation and R^2 values out of the models tested. The second degree and fourth degree polynomial chose very high values of Epsilon in cross-validation, and so in some sense 'gave-up' on making a particularly accurate prediction, preferring to over-regularise and just present a nearly flat line around the predicted average. It is worth investigating lower values of Epsilon because, although they might not present better average errors, a flat line prediction does not contain any actionable information.

All of the methods predicted greater than realised consumer spending growth, both during the crisis and in the recovery period from 2011-2013. It is perhaps unsurprising given the depth of the recession that predictions based on previous data overshoot the true value. Because there is only one other occasion (in 1990 for 2 quarters) since the start of the training set during which negative growth numbers were printed, it is a very rare occurrence in the data, and therefore, not likely to be well explained by the model.

It is reassuring that towards the end of the test period as the economy arguably approached something closer to normal conditions, the predictions get much more accurate. A model, which is only effective in good times is not helpful to policy makers, however, and it will only be possible to tell if the model has learned valuable information from the 2008 recession the next time things turn negative.

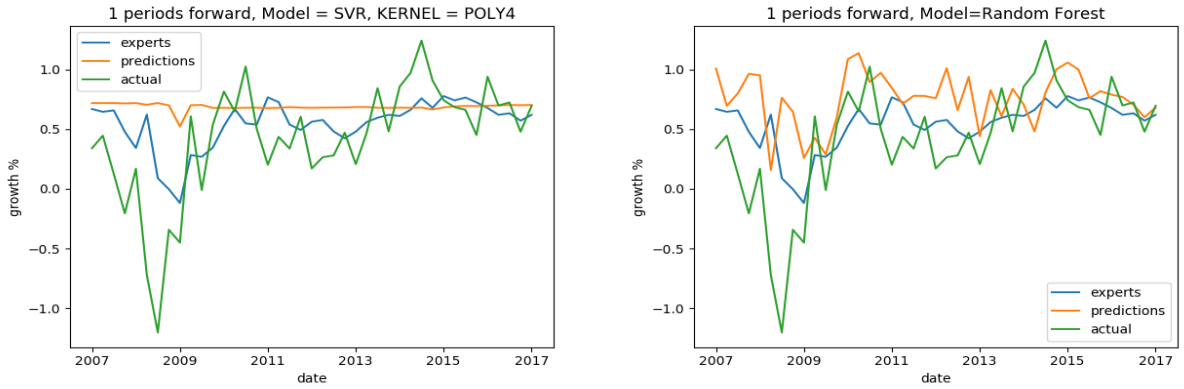
It is important to note that the Survey of Professional investors 'Experts' number is a mean value, typically averaged over 30 separate forecasts. This has the advantage that it pools the knowledge of different forecasters but also clearly suppresses the variance of the predictions. As can be seen in by looking at the experts line

[6] ϵ-IE calculated everywhere using $\epsilon = 0.5$

in the graphs in figure 11. This makes the expert predictions look something like a regularised model, where any outliers predictions are smoothed out by averaging. In section 7.4 we break out the SPF into individual experts for a more granular analysis.

The optimum fixed prediction would be at the test set mean of 0.416%, which would result in a MSE of 0.225 and MAE of 0.346. This is the optimum for an over-regularised solution, which just picks the same number every time. It is a challenging benchmark to beat in terms of squared errors, although the models we have identified as over-regularised do indeed approach this value for the absolute error. The linear SVR with no epsilon insensitive loss is an example of a model where the MAE significantly under-performs, despite showing a relatively high correlation between the predictions and the true values. A possible interpretation of this is that the variance of the model is higher than the true variance, so whereas the correlation (which takes no account of variance) is good, the errors are bad. This lends evidence to the claim that the insensitive region of the SVR is acting as a regulariser.

Figure 11: Graphs of model predictions, contemporary experts and true values



(a) **(b)**

Figure 12: Graphs of model predictions, contemporary experts and true values

7.2.2 2 Period Forward

Kernel	C	Epsilon	Gamma	MSE	MAE	Mean ϵ-IE	Correlation	R^2
linear SVR	0.009	0.4	-	0.3188	0.4278	0.1111	0.2366	-0.3844
rbf SVR	12.0	0.0	8e-05	0.2963	0.408	0.0791	0.3569	-0.2867
poly2 SVR	1.0	1.1	-	0.2355	0.3523	0.0749	0.3551	-0.0224
poly3 SVR	0.12	0.33	-	0.3212	0.4174	0.11	0.2408	-0.3949
poly4 SVR	0.07	1.2	-	0.2554	0.3549	0.0826	0.0808	-0.1091
Random Forest	-	-	-	0.3061	0.4415	0.0954	0.4082	-0.3293
linear SVR	0.0012	0.0	-	0.3094	0.4671	0.1086	0.3397	-0.3434
Mean Experts	-	-	-	0.1958	0.3256	0.0619	0.603	0.1496

Table 4: Forecast accuracy 2 periods ahead

Discussion: The model performance dropped off considerably after the 1 quarter ahead, with the best SVR model correlation dropping from 0.57 to 0.36, the expert baseline on the other hand continued to perform quite well, with a correlation of 0.6. The random forest was the best of the machine learning methods in terms of correlation, but not in terms of MAE where some of the more regularised models are the best performers. This shows that as the signal to noise ratio in the data decreases, more regularisation is beneficial.

Figure 13: Graphs of model predictions, contemporary experts and true values

7.2.3 3 Period Forward

Kernel	C	Epsilon	Gamma	MSE	MAE	Mean ϵ-IE	Correlation	R^2
linear SVR	0.009	0.4		0.2945	0.4218	0.0993	0.2687	-0.2471
rbf SVR	12.0	0.0	8e-05	0.3119	0.4018	0.0904	0.1838	-0.3204
poly2 SVR	1.0	1.1	-	0.3756	0.4271	0.1313	-0.1147	-0.5902
poly3 SVR	0.12	0.33	-	0.374	0.4417	0.1351	0.0291	-0.5834
poly4 SVR	0.07	1.2	-	0.297	0.3832	0.1021	-0.3604	-0.2574
Random Forest	-	-	-	0.2607	0.38	0.0811	0.3217	-0.1038
linear SVR	0.0012	0.0	-	0.3219	0.4856	0.1154	0.2707	-0.3628
Mean Experts	-	-	-	0.2686	0.3602	0.089	0.1608	-0.1371

Table 5: Forecast accuracy 3 periods ahead

Discussion: At 3 quarters ahead all of the models perform poorly including the expert predictions although some, including linear SVR and random forest, beat the experts in terms of correlation. It can be seen

from the charts that the variance of the expert predictions decreases as the forecasts become longer dated, reflecting the fact that a lack of detailed information encourages people to make predictions around the long term average. As we are using the same regularisation, for each prediction, we instead see increased errors in the 3 ahead forecasts.

We noticed from inspecting the graphs that the support vector regressions clearly display a tendency to follow the true values with a lag. This indicates that the current personal consumption growth number is having a significant effect on the prediction. The random forest does not seem to suffer from the same problem, which may reflect the advantage of the 'dropout' style regularisation, whereby different features are considered in every individual tree within the ensemble model.

7.2.4 Correcting For Seasonality

After observing that this pattern exists in the 2 and 3 ahead forecasts, but not noticeably in the 1 or 4 (see figure 14), we speculated that there is some residual seasonality, either in the input variables or the target (although the consumer spending number is seasonally adjusted). To attempt to counteract this we added the season as a four dimensional 1 hot encoded vector to every data point. By comparing charts c and d in figure 13, we see that it does indeed appear to help solve the problem of the predictions following the true values with a lag, and provides a better fit. This evidence supports the idea that there is still seasonality in the data, despite adjustments made to the target variable.

(a) **(b)** **(c)** **(d)**

Figure 14: Graphs of model predictions, contemporary experts and true values

7.2.5 4 Period Forward

Kernel	C	Epsilon	Gamma	MSE	MAE	Mean ϵ-IE	Correlation	R^2
linear SVR	0.009	0.4		0.2895	0.3962	0.092	0.3525	-0.2058
rbf SVR	12.0	0.0	8e-05	0.35	0.4089	0.0953	-0.0792	-0.4579
poly2 SVR	1.0	1.1	-	0.2861	0.3677	0.1016	-0.0609	-0.1916
poly3 SVR	0.12	0.33	-	0.346	0.4147	0.121	0.1126	-0.441
poly4 SVR	0.07	1.2	-	0.2839	0.3629	0.1009	-0.6247	-0.1824
Random Forest	-	-	-	0.2552	0.3763	0.08	0.374	-0.0631
linear SVR	0.0012	0.0	-	0.3282	0.5013	0.1142	0.3475	-0.3671
Mean Experts	-	-	-	0.2771	0.3537	0.096	0.0835	-0.1541

Table 6: Forecast accuracy 4 periods ahead

Discussion: Once again the linear SVR and random forest models have significantly beaten the mean expert predictions, in terms of correlation with the true values. It is interesting that the linear kernel provides the

best fit for the longer dated forecast as it is the simplest model, requiring the fewest number of parameters to specify. This reinforces the assertion that when there is less useful information in the data a simple model is preferable, as less information will be taken up reducing model error and can be used instead in reducing the generalization error.

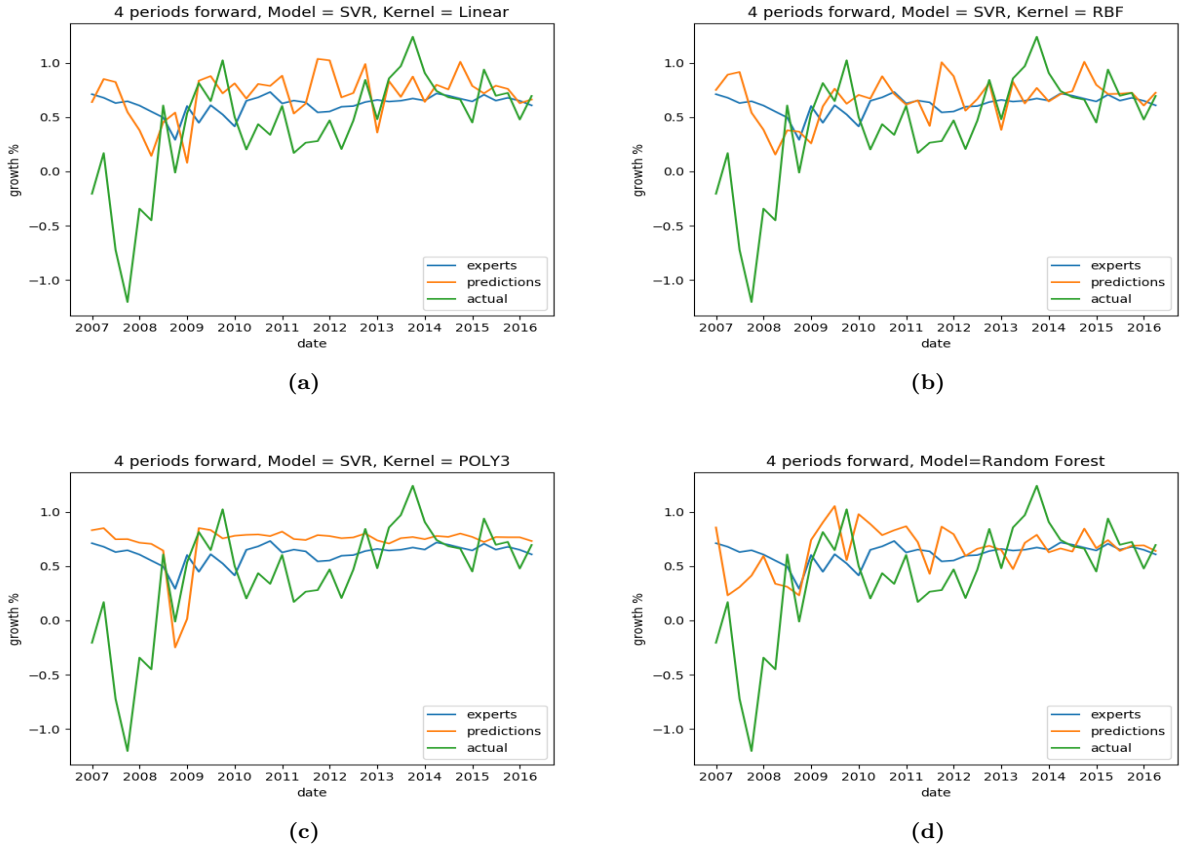


Figure 15: Graphs of model predictions, contemporary experts and true values

7.3 Residual Analysis

When performing regression it is common to look at the residuals (the difference between the true values and your predictions) and to analyse if there is any systematic error that you are making. If the mean of residuals is consistently higher than 0 for example, then you are consistently over-estimating when you make a forecast. Ideal residuals contain no information, so they are centered around zero and have 0 correlation with the predictions themselves.

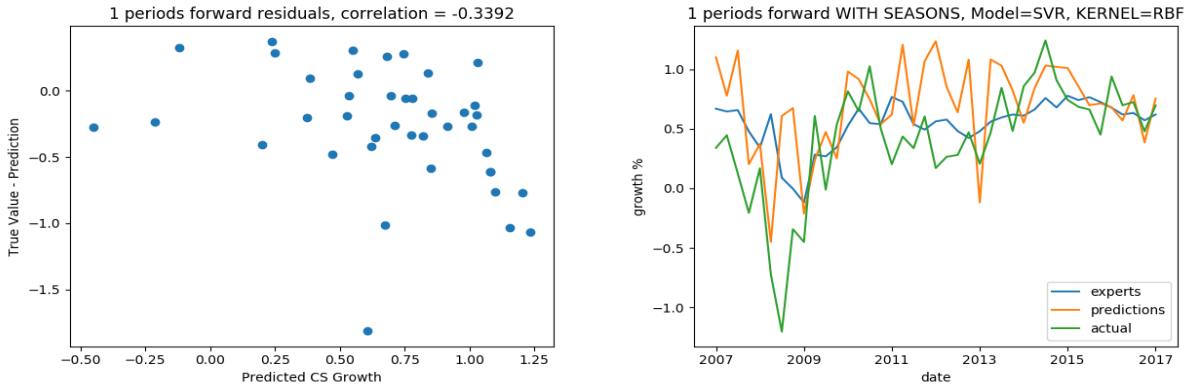
Plotting the residuals against the predictions themselves is also an important exercise. If they are positively correlated, meaning that as the predictions get larger in magnitude so do the residuals, it implies that we could add some extra regularisation, as our forecasts will be improved if we reduce the variance of the predictions. On the other hand, if there is a negative correlation (i.e. as our predictions increase we get closer to the true value) then we should reduce the regularisation in order to let the model try and fit the data more closely.

Below is an experiment demonstrating the effect of changing the hyper-parameters C and Epsilon on the

fit of the data. Adding regularisation by decreasing C or increasing Epsilon increases the correlation of the residuals to the predictions and the best fit is achieved when the residuals are completely uncorrelated to the predictions. It is not fair to use this analysis to provide hyper-parameter values for the production results because it involves data-mining the test set. That said, incorporating residual analysis can be useful for choosing hyper-parameters for future forecasts and for assessing whether the model is theoretically capable of performing better following a test.

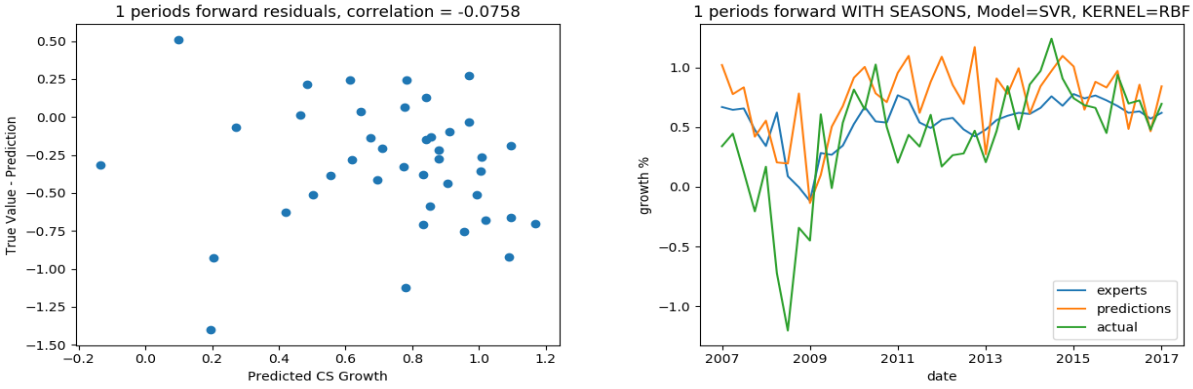
Kernel	C	Epsilon	Gamma	Residual Corr	MSE	MAE	Mean ϵ-IE	Results Corr	R^2
rbf SVR	10	0.0	0.001	-0.339	0.2705	0.3896	0.0917	0.4786	-0.2033
rbf SVR	10	0.4	0.001	-0.076	0.258	0.4026	0.0882	0.555	-0.1475
rbf SVR	1	0.4	0.001	0.24	0.3176	0.4121	0.1167	0.4562	-0.4126
rbf SVR	7	0.4	0.001	0	0.2536	0.3915	0.0855	0.5637	-0.1279

Table 7: Residual correlation with changing regularisation parameters



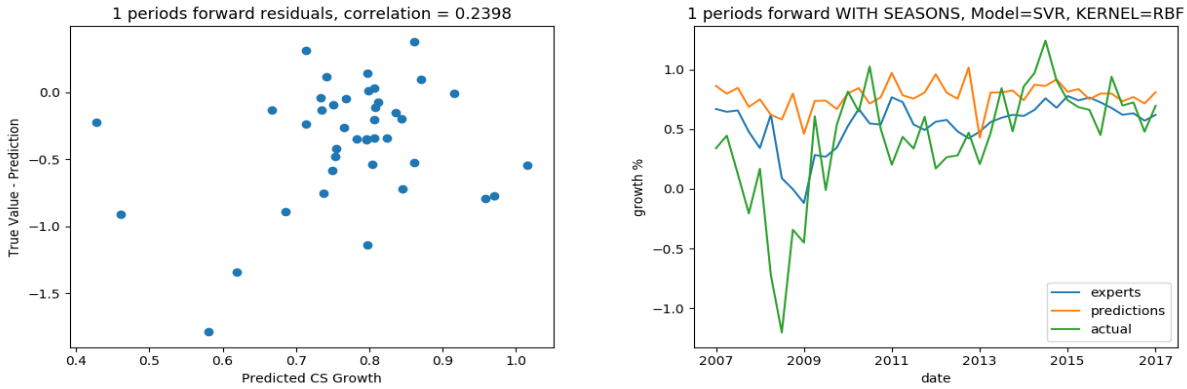
(a) **(b)**

Figure 16: under-regularised, negatively correlated residuals



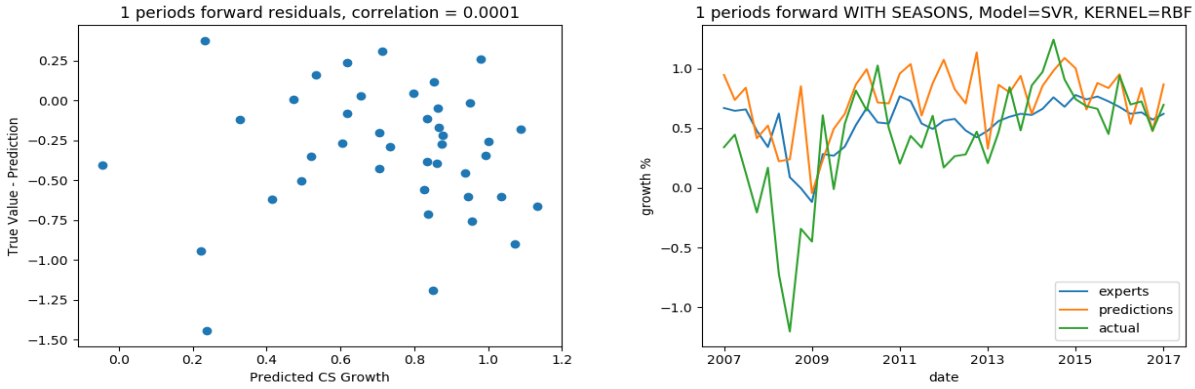
(a) **(b)**

Figure 17: slightly under-regularised



(a) **(b)**

Figure 18: over-regularised, positively correlated residuals



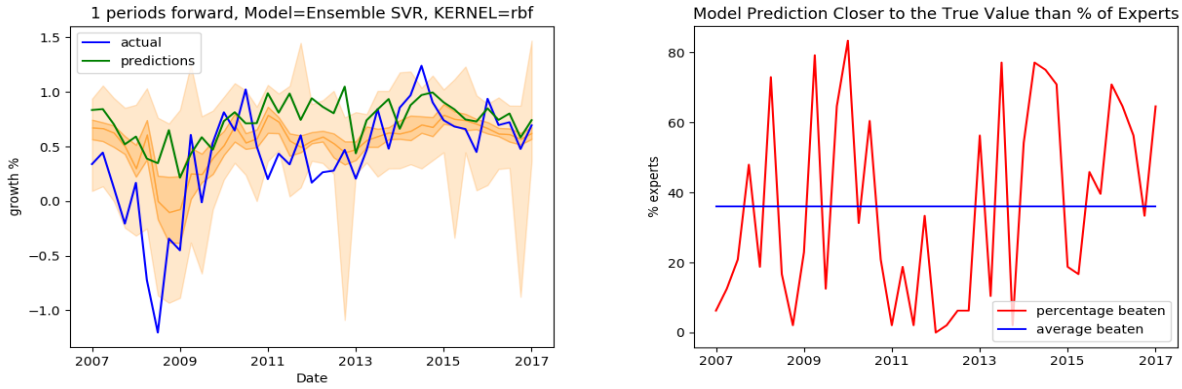
(a) **(b)**

Figure 19: best-fit, zero residuals

7.4 Individual Experts

For an extra insight into how well the model performs against the professionals surveyed we have plotted the 1 period ahead forecast using the RBF kernel SVR against the expert predictions broken into quartiles. In figure 20 the lightly shaded regions correspond to the range of predictions between the min and lower quartile / max and upper quartile; the more opaque shaded region corresponds to the region between the quartiles and the median and the median prediction is marked by a thin line.

We can see that the expert predictions cluster fairly tightly around the median value, particularly in 'normal' times, indeed the average difference (over the whole period) between the upper and lower quartile is only 0.21%. This may be because many forecasters are using similar assumptions and models, or it could be a result of a tendency of forecasters to 'herd', that is they aim for predictions that are close to other forecasts as the consequences of being an outlier and being wrong are more severe than being in the middle and being wrong. This means that figure 20, which shows the percentage of forecasters our prediction 'beats', looks digital.



(a) **(b)**

Figure 20: Breaking out expert predictions

7.5 Ensemble SVR

7.5.1 Optimisation

Num Dims	Re-weighted	MSE	MAE	Mean ϵ-IE	Correlation	R^2
3	False	0.304	0.403	0.1077	0.4453	-0.352
4	False	0.2678	0.3875	0.0889	0.541	-0.1911
6	False	0.2331	0.3689	0.0728	0.6192	-0.0366
8	False	0.2158	0.3627	0.0665	0.6559	0.0401
10	False	0.2231	0.3769	0.0666	0.6262	0.0076
12	False	0.2202	0.3733	0.0666	0.6184	0.0205
14	False	0.2205	0.3739	0.0672	0.622	0.0191
16	False	0.2325	0.3786	0.0734	0.5823	-0.0343
17	False	0.2429	0.3852	0.0794	0.5461	-0.0802
17	True	0.2504	0.388	0.0824	0.5302	-0.1139
16	True	0.2406	0.3863	0.0761	0.5752	-0.0702
14	True	0.2308	0.3785	0.0723	0.5888	-0.0265
12	True	0.2169	0.3711	0.0679	0.6409	0.0351
10	True	0.2005	0.3546	0.0562	0.6505	0.1081
8	True	0.2126	0.3567	0.0664	0.6622	0.0543
6	True	0.2137	0.3618	0.0616	0.6621	0.0493
4	True	0.2585	0.3779	0.0874	0.5806	-0.1496
3	True	0.26	0.3769	0.0877	0.5216	-0.1565

Table 8: Ensemble parameter optimisation - using test set 1 period ahead

Table 8 shows the forecast performance 1 quarter ahead for an ensemble of 40 support vector regressors each with a linear kernel and randomised model parameters as specified in section 6.9.1. The parameters of the ensemble that were varied between trials were: the number of dimensions of the input data considered by each regressor; and whether or not the forecast was generated using a weighted combination of the individual regressors based on their performance in cross-validation.

We can see a clear improvement on our previous results is gained by using the ensemble model, in the following experiment we took our best performing configuration and compared the performance directly with the expert predictions and the previous trials. We also want to note that there was a stable pattern of performance that resulted from varying the ensemble parameters. That pattern was characterised by:

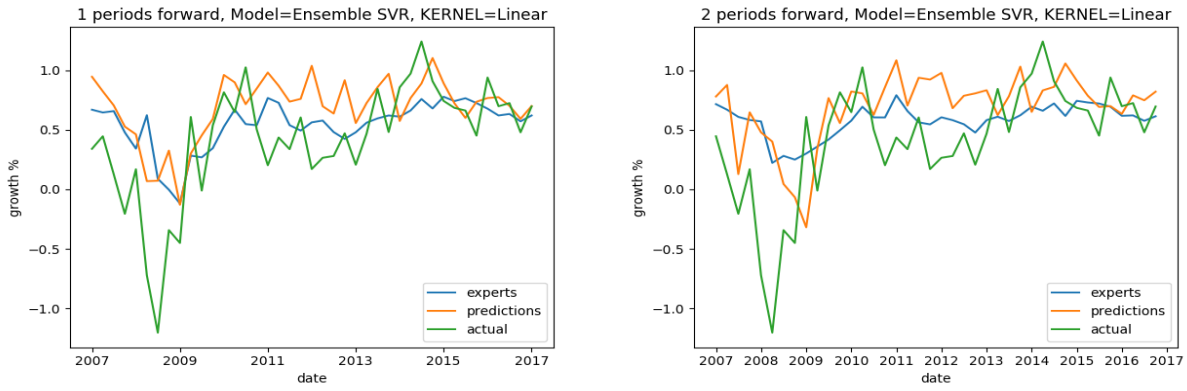
- Changing the number of input dimensions caused a broad peak in performance between 6 and 14 and dropped off notably above and below that. It is obvious why performance should drop when fewer input dimensions are considered by each regressor but it is perhaps surprising that performance also got worse when too many of the dimension were considered. A possible reason for this is that if all of the dimensions are considered by each model, then there is less variation in the regressors and the stochasticity is lower, resulting in less noise cancellation. It is also worth noting that the speed of training and testing was significantly increased for lower dimensions.

- Switching between equal weighting and performance weighting did not have as much of an effect as we were expecting and may have no significant effect at all. It seemed in fact that although the peak performance was not improved by re-weighting the performance for trials, where the dimension was away from optimum, performance fell off more rapidly for the equal weighted ensemble. This makes intuitive sense in the case of reduced dimensions, because if there are some regressors which perform exceptionally badly by chance, they can be completely ignored by the re-weighted ensemble.

7.5.2 Performance

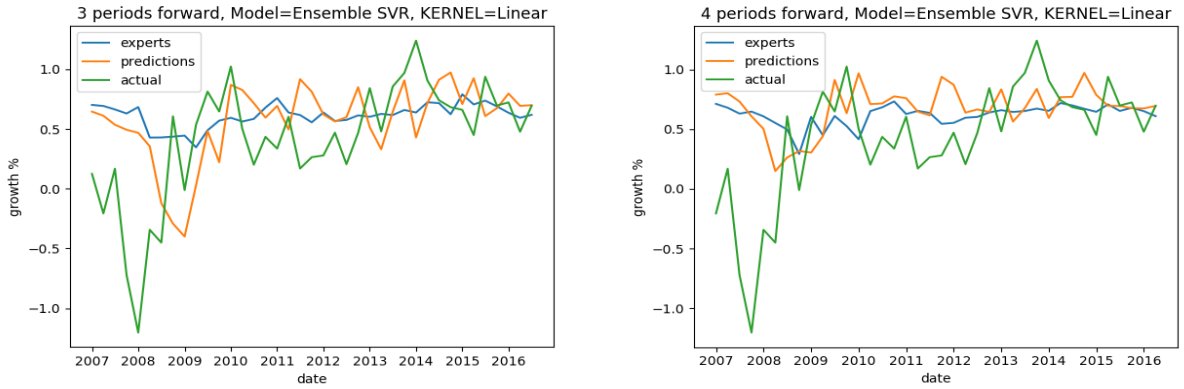
Model	Forecast Quarters	MSE	MAE	Mean ϵ-IE	Results Corr	R^2
Individual SVR	1	0.2534	0.3832	0.0868	0.5695	-0.1269
Ensemble SVR	1	0.2038	0.3585	0.0596	0.6693	0.0936
Mean Experts	1	0.1634	0.2938	0.0467	0.6166	0.2731
Individual SVR	2	0.3188	0.4278	0.1111	0.2366	-0.3844
Ensemble SVR	2	0.258	0.4016	0.0812	0.4563	-0.1203
Mean Experts	2	0.1958	0.3256	0.0619	0.603	0.1496
Individual SVR	3	0.2945	0.4218	0.0993	0.2687	-0.2471
Ensemble SVR	3	0.2829	0.4069	0.0903	0.216	-0.1977
Mean Experts	3	0.2686	0.3602	0.089	0.1608	-0.1371
Individual SVR	4	0.2895	0.3962	0.092	0.3525	-0.2058
Ensemble SVR	4	0.2927	0.4022	0.0915	0.1873	-0.2191
Mean Experts	4	0.2771	0.3537	0.096	0.0835	-0.1541

Table 9: Performance of ensemble SVR against individual SVR and experts



(a) **(b)**

Figure 21: Ensemble SVR against Expert Forecasts



(a) **(b)**

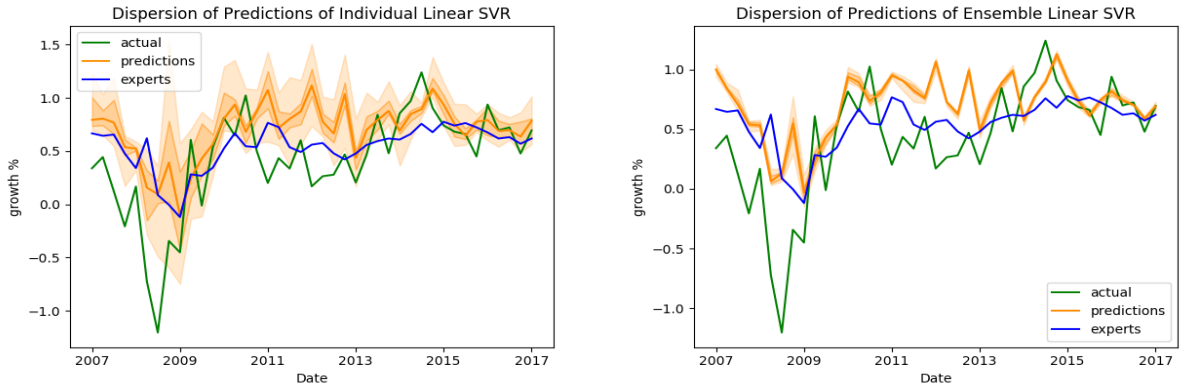
Figure 22: Ensemble SVR against Expert Forecasts

The first thing to note is that, as suspected, the ensemble beats the single support vector regressor under every metric on the 1 and 2 quarter ahead forecast. The 1 quarter ahead in particular finally produces an R^2 over 0, meaning that it is better than just predicting the mean result at every timestep (which is of course not a trivial task in itself as the mean of the test set is not equal to the mean of the training set). Although the single regressor shows a higher correlation over 3 and 4 quarters ahead, and is comparable on other metrics, it is likely there is much higher variation in the performance of the single regressor. This makes it possible that the better performance is purely a result of chance and also makes the forecast less valuable in itself as low variance in the prediction is a desirable quantity. This is a question we look into directly in section 7.6.

When comparing against the expert predictions we see that the ensemble method beats in terms of the 1 period ahead correlation but loses against the mean expert on most of the other metrics. It is important and interesting to note that while both the model forecasts and the expert predictions exhibit less variance for the longer dated forecasts, the ensemble SVR retains more variation than the mean expert prediction. As previously discussed a constant prediction around the sample mean is of little use or interest to us, so the correlation exhibited by the ensemble model over the longer time horizons might make this a more useful forecast, even if the average error in the predictions is greater.

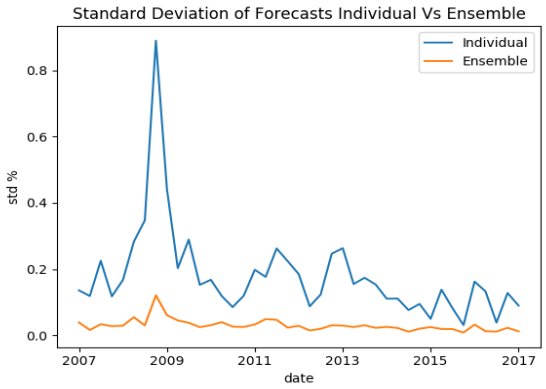
7.6 Variation

We have claimed without evidence at a few points earlier in the discussion that the variance between trials is lower for the ensemble SVR than the individual SVR. In order to give some empirical justification for this claim we have included some data generated by running multiple trials of each.



(a) **(b)**

Figure 23: 10 random parameterisations with shading indicating quartiles of predictions



(a)

Figure 24: Standard deviations of individual Vs ensemble SVR forecasts

Model	Forecast Quarters	Mean STD of Forecast	Mean Correlation	STD of Correlation
Linear SVR	1	0.1828	0.4968	0.1117
Ensemble Linear SVR	1	0.0304	0.6362	0.02066

Table 10: Average performance and std of 10 trials of individual and ensemble SVR

Table 10 contains the most important evidence of the superiority of the ensemble method over the individual SVR. We would expect the standard deviation of the forecast for each particular point to be reduced by a factor of $\sqrt{40} = 6.3$ (40 being the number of regressors in the ensemble) which is almost exactly what we find. The effect on the correlation with the underlying true values is less predictable, however, and what we find is that not only is the variance of that correlation reduced significantly but the mean correlation is far higher for the ensemble than the individual regressor. This presents strong evidence that the averaging taking place in the regressor is actually helping the model find a better fit to the underlying data.

Chapter 8: Conclusions

Simple models perform better when there is limited information.
One of the defining challenges of working with economic time series datasets is that they are short, with at most a few hundred useable data points. We found that the SVR with the linear kernel, the simplest model, was the best performing, particularly for the 3 and 4 quarters ahead forecast, where the noise to signal ratio is higher.

Ensemble methods not only reduce variance but dramatically improve average performance.
There are a number of hyper-parameters, including the variable selection and model specific parameters such as C and Epsilon, which introduce a large and unpredictable variance into the accuracy of the results. By averaging over random assignments of these parameters in an ensemble we were not only able to reduce the variance of our predictions, but reliably produced something approaching the best fit of the individual underlying model. The most important result from this thesis is, therefore, that the power of averaging over many parameterizations is significant and far outweighs a single model, even one chosen by careful cross-validation.

Support vector regression slightly outperformed kernel ridge regression.
The performance of the ensemble regression when Epsilon was set equal to zero was very similar to the performance when random Epsilons were generated between zero and 0.4. For the 4 quarters ahead prediction, however, SVR slightly outperformed. This may be because further ahead a slightly more regularised solution is desirable. Perhaps if the regularisation parameter of KRR was adjusted upwards to compensate for the lack of regularisation from an epsilon-insensitive loss, better performance could be retrieved. More investigation would be needed to confirm this but ensemble SVR is our regressor of choice based on this experiment.

All models tended to overestimate consumer spending growth.
This was notable both during the recession, which was very severe and not comparable to anything in the training set, and in the recovery since, which is widely recognised to have been unusually subdued by historical standards. Since 2014 the performance of our model has been very good, hopefully the model can continue to learn and provide useful forecasts both in normal times and now. With more data about the condition of the economy prior to and during the last recession it will have a better chance of predicting the next one.

Chapter 9: Suggestions for Further Work

1. This thesis has found that the performance of a model can be greatly improved by combing a variety of different parameterisations and variable subspaces into ensembles. It would be interesting to combine a greater variety of models including different kernels and the random forest into a single ensemble to see if the benefits of averaging can be extended further.

2. We have investigated using 17 economic variables but there are potentially some which do not help at all and still others which have not been considered. More investigations into the individual dimensions might reveal a better final model.

3. We have focused entirely on data from the United States. It would be interesting to apply the same techniques to other developed countries. Furthermore, if the countries are directly comparable, the training set could be expanded by considering multiple countries together, possibly with extra dimensions to represent the country.

4. The amount of data in the training set is limited by those datasets that begin the latest, for example the M2 money supply, which was not published before 1980. It would be interesting to investigate if the length of the training data could be extended by perpetuating average values for the shorter series back in time or by constructing realistic values for those series, using other available data from the time.

5. We loaded data from the FRED data service, which includes all subsequent revisions to data in the historic values. The ALFRED data service (Archival FRED, Alf) contains data that would have been available at the time to contemporary forecasters, absent of any revisions. It would be valuable to try using these pre-revision values. However, there is a danger which is introduced for any series which have had their methodology changed over the years and so the values prior to the methodology change would not be sensibly comparable with values post change.

Chapter 10: Forecasts for the Future

For the interest of the reader we include the predictions of the best ensemble SVR model for the next four quarters.

Date	Lin SVR	RBF SVR	Random Forest	Ensemble SVR
2017 Q3	0.3598	0.5618	0.6998	0.5369
2017 Q4	0.7575	0.6284	0.7031	0.6889
2018 Q1	0.4731	0.4631	0.6758	0.5605
2018 Q2	0.4043	0.4959	0.5551	0.5019

Table 11: Predictions of future quarterly personal consumption growth in %

The models provide a range of estimates for upcoming growth but all show a similar pattern of increasing personal consumption growth towards the ends of 2017 which then starts to slow in 2018. The ensemble SVR, which we have established is our best model, is also quite close to the average of the other models.

References

Alfred Data Service. URL https://alfred.stlouisfed.org.

Applications of artificial intelligence in finance and economics / edited by Jane M. Binner, Graham Kendall, and Shu-Heng Chen. Advances in econometrics ; v. 19. Emerald, Bingley, U.K., 2004. ISBN 9781849503037.

Gianni Amisano and John Geweke. Prediction Using Several Macroeconomic Models. 2013. URL https://www.ecb.europa.eu/pub/pdf/scpwps/ecbwp1537.pdf?665147f94986b2e4d7bfdf857b7c9f9a.

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory - COLT '92*, 1992. ISBN 089791497X. doi: 10.1145/130385.130401.

Andreas Buja and Werner Stuetzle. Observations on Bagging. URL https://www.stat.washington.edu/wxs/Learning-papers/sinica-bagging-buja-stuetzle.pdf.

Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Machine Learning*, 1995. ISSN 15730565. doi: 10.1023/A:1022627411411.

N Cristianini and J Shawe-Taylor. *An introduction to Support Vector Machines*. 2000. ISBN 0521780195. doi: 0521780195.

Pedro Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 2012. ISSN 00010782. doi: 10.1145/2347736.2347755.

Edward R. Dougherty, Seungchan Kim, and Yidong Chen. Coefficient of determination in nonlinear signal processing. *Signal Processing*, 2000. ISSN 01651684. doi: 10.1016/S0165-1684(00)00079-7.

Cl??udia Duarte, Paulo M M Rodrigues, and Ant??nio Rua. A mixed frequency approach to the forecasting of private consumption with ATM/POS data. *International Journal of Forecasting*, 2017. ISSN 01692070. doi: 10.1016/j.ijforecast.2016.08.003.

Ahmed El Deeb. What to do with small data. *2015*. URL https://medium.com/rants-on-machine-learning/what-to-do-with-small-data-d253254d1a89.

Graham Elliott and Allan Timmermann. Economic Forecasting. *Journal of Economic Literature*, 46(1): 3–56, 3 2008. ISSN 0022-0515.

Graham Elliott and Allan Timmermann. Forecasting in Economics and Finance. *Annual Review of Economics*, 8, 1 2016. ISSN 1941-1383. URL http://search.proquest.com/docview/1844211080/.

Dean Fantazzini and Zhamal Toktamysova. Forecasting German car sales using Google data and multivariate models. *International Journal of Production Economics*, 2015. ISSN 09255273. doi: 10.1016/j.ijpe.2015.09.010.

George Forman and Ira Cohen. Learning from Little: Comparison of Classifiers Given Little Training. *Knowledge Discovery in Databases: PKDD 2004*, 2004. ISSN 03029743. doi: 10.1007/978-3-540-30116-5_17.

Roberto Golinelli and Giuseppe Parigi. Consumer Sentiment and Economic Activity A Cross Country Comparison Consumer Sentiment and Economic Activity: A Cross Country Comparison. *Journal of Business Cycle Measurement and Analysis Journal of Business Cycle Measurement and Analysis* –, 2004. ISSN 1729-3618. doi: 10.1787/jbcma-v2004-art10-en.

Alon Halevy, Peter Norvig, and Fernando Pereira. The Unreasonable Effectiveness of Data. *IEEE Intelligent Systems*, 2009. ISSN 1541-1672. doi: 10.1109/MIS.2009.36.

David F. Hendry and Michael P. Clements. Economic forecasting: Some lessons from recent research, 2003. ISSN 02649993.

W Huang, Y Nakamori, and S Y Wang. Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*, 2005. ISSN 0305-0548. doi: 10.1016/j.cor.2004.03.016.

S Sathiya Keerthi and Chih-Jen Lin. Asymptotic behaviors of support vector machines with Gaussian kernel. *Neural computation*, 2003. ISSN 0899-7667. doi: 10.1162/089976603321891855.

Hyun Chul Kim, Shaoning Pang, Hong Mo Je, Daijin Kim, and Sung Yang Bang. Constructing support vector machine ensemble. *Pattern Recognition*, 2003. ISSN 00313203. doi: 10.1016/S0031-3203(03)00175-4.

Evan F. Koenig, Sheila Dolmas, and Jeremy Piger. The Use and Abuse of Real-Time Data in Economic Forecasting. *Review of Economics and Statistics*, 2003. ISSN 0034-6535. doi: 10.1162/003465303322369768.

Spyros Makridakis and Michèle Hibon. The M3-Competition: results, conclusions and implications. *International Journal of Forecasting*, 2000. ISSN 01692070. doi: 10.1016/S0169-2070(00)00057-1.

Sayan Mukherjee Massimiliano Pontil Federico Girosi. On the Noise Model of Support Vector Machine Regression. *Massachusetts Institute of Technology*, 1998.

mortada. mortada fred api. URL `https://github.com/mortada/fredapi`.

Saeed Moshiri and Norman Cameron. Neural network versus econometric models in forecasting inflation. *Journal of Forecasting*, 2000. ISSN 02776693. doi: 10.1002/(SICI)1099-131X(200004)19:3¡201::AID-FOR753¿3.0.CO;2-4.

Klaus Robert Müller, Sebastian Mika, Gunnar Rätsch, Koji Tsuda, and Bernhard Schölkopf. An introduction to kernel-based learning algorithms, 2001. ISSN 10459227.

Rickard Nyman and Paul Ormerod. Predicting Economic Recessions Using Machine Learning Algorithms. 2017. URL `https://arxiv.org/abs/1701.01428`.

Paul Ormerod and Craig Mounfield. Random matrix theory and the failure of macro-economic forecasts. *Physica A: Statistical Mechanics and its Applications*, 2000. ISSN 03784371. doi: 10.1016/S0378-4371(00)00075-3.

F Pérez-Cruz, Ja Afonso-Rodríguez, and Javier Giner. Estimating GARCH models using support vector machines. *Quantitative Finance*, 2003. ISSN 1469-7688. doi: 10.1088/1469-7688/3/3/302.

maxine singer. Thoughts of a nonmillenarian. *Bulletin of the American Academy of Arts and Sciences*, 1997. URL `http://www.jstor.org/stable/3824486`.

A Smola and V Vapnik. Support vector regression machines. *Advances in neural information processing . . .*, 1997.

St. Louis Fed. Federal Reserve Economic Data. URL `https://fred.stlouisfed.org/`.

Jie Sun and Hui Li. Financial distress prediction using support vector machines: Ensemble vs. individual. *Applied Soft Computing Journal*, 2012. ISSN 15684946. doi: 10.1016/j.asoc.2012.03.028.

Hal Varian. Big data: New tricks for econometrics. *The Journal of Economic Perspectives*, 2014. ISSN 0895-3309. doi: 10.1257/jep.28.2.3.