ARTICLE IN PRESS

International Journal of Forecasting ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

International Journal of Forecasting

journal homepage: www.elsevier.com/locate/ijforecast

Identifying business cycle turning points in real time with vector quantization

Andrea Giusto [a], Jeremy Piger [b,*]

[a] Department of Economics, Dalhousie University, Canada

[b] Department of Economics, University of Oregon, United States

ARTICLE INFO

Keywords:
Classification
Reference cycle
Expansion
Recession

ABSTRACT

We propose a simple machine-learning algorithm known as Learning Vector Quantization (LVQ) for the purpose of identifying new U.S. business cycle turning points quickly in real time. LVQ is used widely for real-time statistical classification in many other fields, but has not previously been applied to the classification of economic variables, to the best of our knowledge. The algorithm is intuitive and simple to implement, and easily incorporates salient features of the real-time nowcasting environment, such as differences in data reporting lags across series. We evaluate the algorithm's real-time ability to establish new business cycle turning points in the United States quickly and accurately over the past five NBER recessions. Despite its relative simplicity, the algorithm's performance appears to be very competitive with those of commonly used alternatives.

1. Introduction

A traditional view of the US business cycle is that of alternating phases of expansion and recession, where an expansion corresponds to widespread, persistent growth in economic activity, and a recession consists of a widespread, relatively rapid, decline in economic activity. The timely identification of the turning points between these phases, or peaks and troughs, is of considerable importance to policymakers, financial markets, firms, and individuals. A substantial body of literature has focused on the prediction of future turning points using a variety of leading economic and financial time series, with some limited success.[1] A smaller body of literature has focused on the identification of turning points that have already occurred, using economic variables that are coincident with the business cycle.

The problem of the ex-post identification of turning points is of particular interest, because there are many examples of turning points that have not been predicted ex-ante. This spotty forecasting record means that economic agents are left trying to determine whether a new business cycle phase has already begun. Even this is a difficult task, with new turning points usually not being identified until many months after they occur. The official chronology of business cycle turning points in the United States is provided by the National Bureau of Economic Research's (NBER) Business Cycle Dating Committee, which has historically announced new turning points with a lag of between four and 21 months. Statistical models improve on the NBER's timeliness considerably, with little difference in the timing of the turning point dates established.[2]

* Corresponding author.
E-mail addresses: andrea.giusto@dal.ca (A. Giusto), jpiger@uoregon.edu (J. Piger).

[1] For recent contributions to this literature, see Berge (2015), Chauvet and Potter (2005), Kauppi and Saikkonen (2008), Ng (2014) and Rudebusch and Williams (2009).

[2] See for example Chauvet and Hamilton (2006) and Chauvet and Piger (2008).

http://dx.doi.org/10.1016/j.ijforecast.2016.04.006
0169-2070/

Please cite this article in press as: Giusto, A., & Piger, J., Identifying business cycle turning points in real time with vector quantization. International Journal of Forecasting (2016), http://dx.doi.org/10.1016/j.ijforecast.2016.04.006

However, in general these models don't identify turning points until several months after they occurred. As a recent example of this, Hamilton (2011) surveys a wide range of statistical models that were in place to identify business cycle turning points in real time, and finds that such models did not send definitive signals regarding the December 2007 NBER peak until late 2008.[3] One important reason for these identification lags is data reporting lags, as many key coincident indicators are released only with a lag of one to two months. Another factor is the need for several months of negative or positive data to accumulate before a definitive turning point signal can be uncovered.

The essence of real-time turning point identification is a problem of statistical classification. Given a set of observations on economic indicators, we wish to determine which of two "classes" these observations belong to, where the classes are expansion and recession. Much of the literature on the identification of business cycle turning points has focused on the use of parametric statistical models to link the observed data to the classes. For example, Chauvet (1998) proposed a dynamic factor model with Markov-switching (DFMS) for identifying expansion and recession phases from a group of coincident indicators, and Chauvet and Hamilton (2006) and Chauvet and Piger (2008) evaluated the performance of variants of this DFMS model for identifying NBER turning points in real time. Fossati (in press) alternatively evaluated the real-time performance of a dynamic probit specification linking NBER expansion and recession phase indicators to observed data. If the true data generating process (DGP) linking the observed data to the classes is known, then such parametric models allow for the construction of an optimal Bayesian classifier of a period's data as belonging to either the expansion or recession class.

However, in the absence of a known true DGP, nonparametric methods may be more robust, as they do not rely on the specification of the DGP. Of particular interest are non-parametric classification techniques based on machine learning algorithms, which have been utilized successfully for real-time classification in a large number of existing studies outside economics. Somewhat surprisingly, applications of these algorithms to the problem of the real-time classification of macroeconomic data to expansion or recession phases, and the resulting identification of business cycle turning points, are rare. Qi (2001) considered the ability of an artificial neural network to produce out-of-sample forecasts of US business cycle turning points between 1972 and 1995. In recent work, Berge (2015) evaluated the ability of a nonparametric forecasting model with predictors selected via a boosting algorithm to forecast and nowcast business cycle turning points since 1985. Importantly, this latter paper shows that the performance of the boosting algorithm improves on a Bayesian averaging of forecasts produced by parametric models.[4]

In this paper, we use a non-parametric classification algorithm known as Learning Vector Quantization (LVQ) to classify economic indicators as arising from either expansion or recession regimes in real time.[5] LVQ is used widely in real-time classification problems in a number of fields and applications, including production quality monitoring, power load forecasting, speech recognition, odor classification, intrusion detection, beer grading, and email spam detection. LVQ takes both historical data and its classification as inputs to train the algorithm. For this historical classification, we use the available NBER chronology. Based on this training, new data points that have not yet been classified by the NBER are then labeled as arising from the expansion or recession regimes, which provides a real-time tracking of new business cycle turning points. Our focus is on the algorithm's ability to provide a timely and accurate identification of new NBER business cycle turning points.

In addition to the potential advantages of a nonparametric approach when the DGP is unknown, LVQ also has computational advantages over the parametric methods that have been frequently used to identify business cycles turning points based on a group of coincident indicators. The algorithm is very simple to implement, generally taking only seconds of computer time. In contrast, the parameters of the DFMS model or a dynamic probit specification are generally estimated via computationally intensive Bayesian techniques. Frequentist approaches to estimation can also be used, but these require approximations, as the exact likelihood function for these models is generally not available. Also, the LVQ algorithm incorporates the possibility of data series arriving with different reporting lags, which is a salient real-world feature of the nowcasting environment that is not easy to handle with standard parametric models.

In our empirical analysis, we evaluate the LVQ algorithm's ability to classify US macroeconomic data into expansion and recession phases in real time, with particular emphasis on the timely identification of new business cycle turning points. The data to be classified are four monthly coincident series that have been highlighted by the NBER as providing information regarding business cycle phases. We consider an analyst applying the LVQ algorithm to this data each month between December 1976 and August 2013 in order to identify new business cycle turning points. Importantly, we use a vintage data set that is an accurate reflection of the information that would have been available to the analyst in real time. This is an important difference from the small number of existing studies that use non-parametric approaches to identify business cycle turning points, which have conducted their forecast evaluations using fully revised data. Also, as the economic indicators that we consider are released with different reporting lags, we allow the analyst in our forecast evaluation to update inferences about new business cycle turning points at different points during the month in real time as new data are released. The LVQ algorithm's performance over this period is impressive relative to the results

[3] The NBER announced the December 2007 peak on December 1, 2008.

[4] A substantial number of studies have used non-parametric algorithms to establish historical, in-sample chronologies of business cycle turning points. Examples include Berge and Jordá (2011), Fushing, Chen, Berge, and Jordá (2010), Harding and Pagan (2006), Stock and Watson (2010, 2014), and Vishwakarma (1994). The objective of our study differs from those of these studies in that we take the historical NBER chronology as given, and focus on the identification of turning points in end-of-sample data that have not yet been classified by the NBER.

[5] LVQ algorithms and related extensions are described in detail by Kohonen (2001).

Please cite this article in press as: Giusto, A., & Piger, J., Identifying business cycle turning points in real time with vector quantization. International Journal of Forecasting (2016), http://dx.doi.org/10.1016/j.ijforecast.2016.04.006

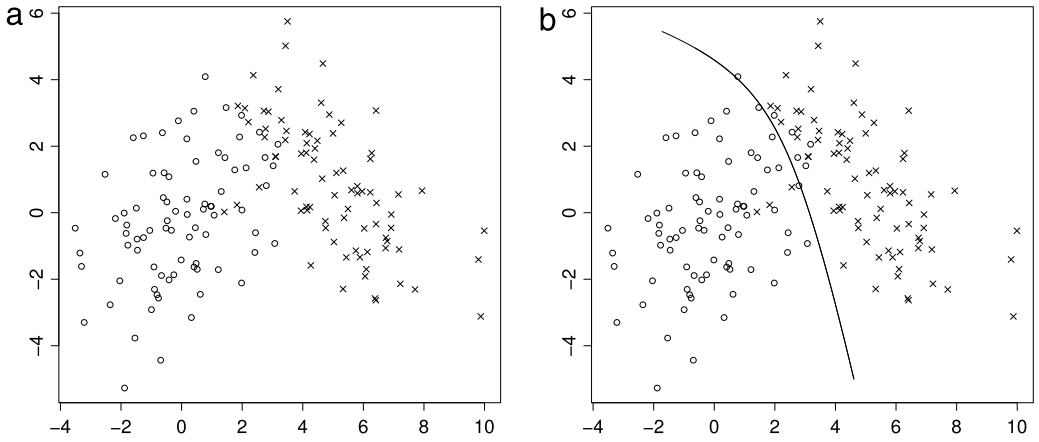

Fig. 1. Bivariate random samples generated from two independent bivariate normal distributions. The dataset represented with circles has mean (0, 0), variance 3 (along both dimensions), and covariance 1. The dataset represented with crosses has mean (5, 1), variance 3, and covariance −2. Panel B contains the separation manifold defined by the discriminant function in Eq. (1).

reported in the existing literature, particularly with regard to business cycle peaks. As one example of this, the LVQ algorithm would have called the peak of the Great Recession on June 6, 2008, which is ahead of any of the procedures reviewed by Hamilton (2011). The established date of this peak would have been January 2008, which is very close to the NBER peak date of December 2007.

The rest of the paper proceeds as follows. Section 2 provides a general description of the LVQ algorithm. Section 3 describes the performance of the LVQ algorithm for identifying new business cycle turning points in the United States. Section 4 concludes.

2. Classification using learning vector quantization

2.1. Data classification

The issue of automatic data classification has long been studied in statistics, with the traditional approach based on the calculation of conditional probabilities. Consider the problem of assigning an m-dimensional real vector $x \in \mathbb{R}^m$ to a class belonging to the set $\{C_k\}_{k=1}^K$. Let $p(C_k)$ denote the *a priori* probability that a vector belongs to class k, and $p(x|x \in C_k)$ be the probability that sample x is observed if it belongs to class C_k. The statistical approach to classification uses the discriminant functions:

$$\delta_k = p(x|x \in C_k)p(C_k),$$

where pattern x is assigned to class C_k, with $C_k = \max_k\{\delta_k\}$. This approach has several theoretical advantages, including the property that the probability of misclassification is minimized. From a practical point of view, though, the desirable theoretical properties of this classification rule may not be exploitable because of a lack of knowledge of the probabilities that govern the DGP. The usual approach in practical statistical classification consists of deriving a good estimate of the discriminant functions over the entire sample space of x.

To make things more concrete, consider panel (a) of Fig. 1, which shows two random samples of equal size, drawn from two independent bivariate normal distributions. In terms of the notation used above, we have that $x \in \mathbb{R}^2$, the set of classes is $\{C_1, C_2\}$, the *a priori* probability of each class is $\frac{1}{2}$, and the conditional probabilities are calculated as:

$$p(x|x \in C_k) = \frac{1}{2\pi|\Sigma_k|^{1/2}} \exp\Big(-\tfrac{1}{2}(x-\mu_k)' \times \Sigma_k^{-1}(x-\mu_k)\Big), \quad k = 1, 2, \tag{1}$$

where μ_k and Σ_k denote the mean and variance–covariance matrix of the two bivariate normal distributions, respectively. As is clear from Fig. 1, this classification problem is non-trivial, since the distributions describing the two classes have overlapping support. We classify the samples by defining a separation manifold between the two classes determined by the discriminant functions. The separating manifold partitions $\mathbb{R}^2$ into subsets over which each discriminant function is maximal. Panel (b) of Fig. 1 shows the separating manifold for these data. Once the boundary between the two regions has been established, it is easy to build a Bayesian classification system that assigns class 1 to new data points that lie below the curve and class 2 to those above it. Note that the optimality of this classifier system does not imply the perfection of its classification ability, but only the minimization of the probability of an erroneous classification.

Vector quantization (VQ) is an alternative strategy that can be used to obtain a separating manifold in the space of interest, but it is based on completely different principles to the Bayesian classifier. A VQ classifier relies on the definition of certain key points, called *codebook vectors*, in the data space. Each codebook vector is used to represent one class, and there can be more than one codebook vector per class. Once these codebook vectors have been singled out, data are classified as belonging to the same class as the nearest codebook vector in the Euclidean metric.[6]

Consider, for example, the situation depicted in Panel (a) of Fig. 2: two codebook vectors representing two

[6] The Euclidean metric is only one of many possible metrics that could be used, but it is the dominant choice in applied work.

Please cite this article in press as: Giusto, A., & Piger, J., Identifying business cycle turning points in real time with vector quantization. International Journal of Forecasting (2016), http://dx.doi.org/10.1016/j.ijforecast.2016.04.006

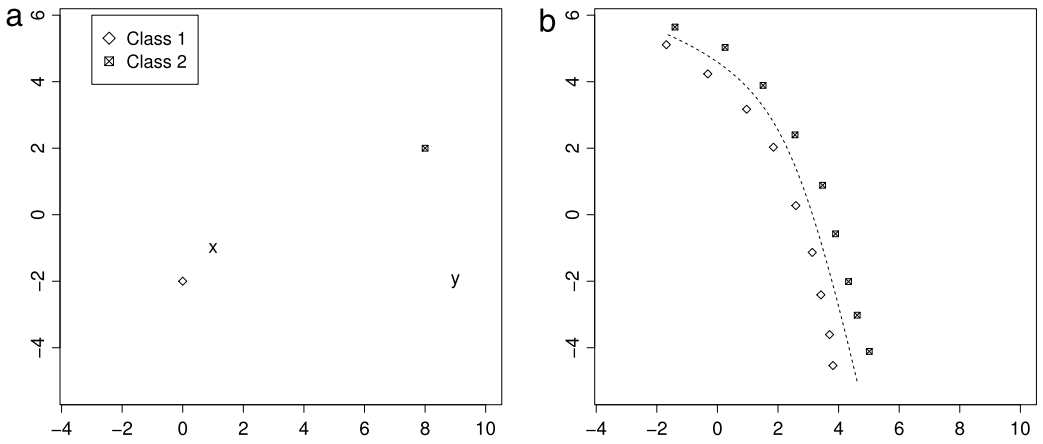

Fig. 2. Panel (a) illustrates the nearest neighbor classification rule. Vector x belongs to class 1, while vector y belongs to class 2. Panel (b) shows a hypothetical placement of the codebook vectors for the data shown in Fig. 1.

different classes are plotted with a diamond and a crossed square, respectively. New data points of unknown classification are classified by means of a "nearest neighbor" strategy. For example, vector x in Fig. 2 would be classified as belonging to class 1, while vector y would be assigned to class 2. In short, a VQ classifier is based on *quantizing* (i.e., approximating) all of the salient features of the data into the codebook vectors, which can be thought of as the most representative points of each class. Just like in the Bayesian classifier, the codebook vectors define a separating manifold in the space $\mathbb{R}^2$, through the midplanes between neighboring pairs of vectors. For the data represented in Fig. 1, the codebook vectors may be placed as in panel (b) of Fig. 2, where the midplanes between neighboring pairs of vectors would approximate the optimal Bayesian separation curve. A key difference between Bayesian and VQ classifiers is that while the Bayesian strategy seeks to approximate the discriminant functions over the whole sample space, the VQ classifier may focus on a smaller region of this space, if this is where most of the relevant information regarding classification is.

The identification of good codebook vectors may seem difficult without prior knowledge of (or assumptions about) the statistical distributions involved. However, the algorithms described in the next section solve this problem in a surprisingly simple and computationally light manner.

2.2. LVQ algorithms

Learning vector quantization is an adaptive learning algorithm in which the locations of the codebook vectors are established through adjustments of decreasing magnitudes. Let X be a collection of N observations $x_n \in \mathbb{R}^m$, $n = 1, \ldots, N$, for which the classification in the set $\{C_k\}_{k=1}^K$ is known. Let there be $\bar{N} \in [K, N]$ codebook vectors $m_i \in \mathbb{R}^m$, $i = 1, \ldots, \bar{N}$, with given initial locations. Finally, let $g = 1, 2, \ldots, G$ denote iterations of the algorithm, and let α^g be a decreasing sequence of real numbers bounded between zero and one. Given the initial locations of the $\bar{N}$ codebook vectors, the LVQ algorithm makes adjustments to their locations through the following steps.

Algorithm 1 (*LVQ*)**.**

1. Let $g = 1$ and $n = 1$.
2. Identify the codebook vector m_c^g that is closest to the data point x_n in the Euclidean metric:
$$c = \underset{i \in \{1,\ldots,\bar{N}\}}{\operatorname{argmin}} \left\{ \| x_n - m_i^g \| \right\}.$$
3. Adjust the location of the codebook vector with index c according to the following rule:
$$\begin{cases} m_c^{g+1} = m_c^g + \alpha^g (x_n - m_c^g) \\ \quad \text{if } x_n \text{ and } m_c^g \text{ belong to the same class} \\ m_c^{g+1} = m_c^g - \alpha^g (x_n - m_c^g) \quad \text{otherwise.} \end{cases}$$
4. If $n + 1 \leq N$, let $n = n + 1$, and repeat from step 2. Otherwise, let $n = 1$ and $g = g + 1$ and repeat from step 2 if $g \leq G$; otherwise stop.

The LVQ algorithm is very simple. At each iteration, a new data vector is considered, and its nearest codebook vector is identified. If this codebook vector agrees with the actual classification of the data vector, its location is moved closer to the data vector. If the selected codebook vector does not classify the data vector correctly, then it is moved farther away from the data vector. These adjustments are made in a simple linear fashion. Fig. 3 shows a hypothetical example of the two cases. These calculations are repeated for each data vector in the data set. When they have all been used, a new iteration is started with a decrease in the weight α^g, which controls the size of the adjustment to the codebook vectors. This continues for G iterations.[7]

[7] The algorithm that we have laid out is the basic LVQ algorithm, which has been shown to work well in many practical applications. Various modifications of this algorithm that may improve its classification ability in some contexts have been proposed. These include LVQ with nonlinear updating rules, as in the generalized LVQ algorithm of Sato and Yamada (1995), as well as LVQ employed with alternatives to the Euclidean measure of distance, such as the generalized relevance LVQ of Hammer and Villmann (2002). The latter allows for an adaptive weighting of the data series in the dimensions that are most helpful for classification, and may be particularly useful when applying LVQ to large datasets.

Please cite this article in press as: Giusto, A., & Piger, J., Identifying business cycle turning points in real time with vector quantization. International Journal of Forecasting (2016), http://dx.doi.org/10.1016/j.ijforecast.2016.04.006

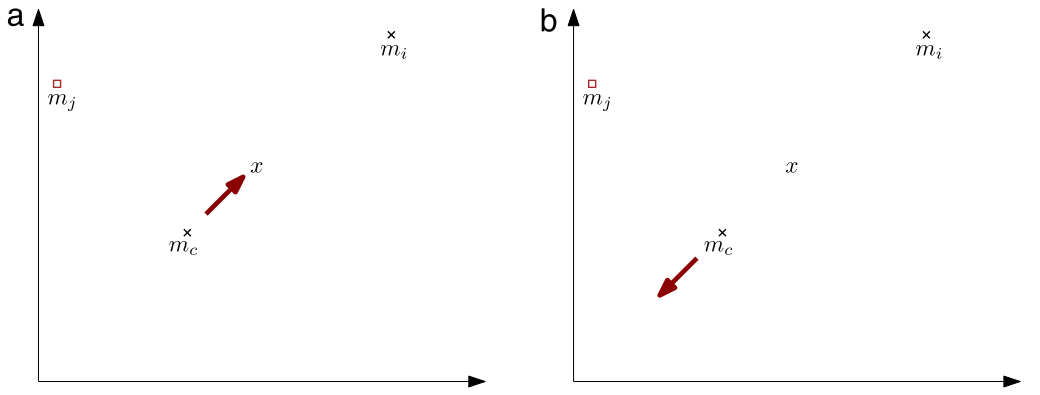

Fig. 3. Adjustments to the codebook vector m_c in the two cases in which m_c either (a) classifies x correctly or (b) does not.

2.3. Class prediction

The LVQ algorithms described in the previous section require a sample of data $x_n \in \mathbb{R}^m$, $n = 1, \ldots, N$, for which the classification in the set $\{C_k\}_{k=1}^K$ is known. This sample of data is used to produce codebook vectors via the LVQ algorithm, a process that is typically referred to as "training" the classifier. Once this training is complete and the codebook vectors have been established, the classifier can be used to predict the classes of new observations for which the true class is not yet known. This process is analogous to the in-sample parameter estimation and out-of-sample prediction steps employed with parametric statistical models.

The specifics of class prediction is as follows. Suppose that we have a new data point, x_{N+1}, for which the classification is unknown. We can use the LVQ classifier to predict the class for this data point by first finding the codebook vector m_c that is closest to x_{N+1} in the Euclidean metric:

$$c = \underset{i \in \{1,\ldots,\bar{N}\}}{\operatorname{argmin}} \{\|x_{N+1} - m_i\|\}.$$

We then assign x_{N+1} to the same class as is assigned to the codebook vector m_c.

In some applications, the new data point to be classified may be observed only partially. This is particularly true for our application to the identification of business cycle turning points in real time, since relevant data are released at different times and with varying reporting lags. In such cases, class prediction is achieved simply by finding the codebook vector m_c that is closest to x_{N+1} in the Euclidean metric for the elements of x_{N+1} that are observed.

2.4. Implementing the LVQ algorithm

In order to implement the LVQ algorithm, we must define a modest number of parameters, namely the number of codebook vectors, $\bar{N}$, the decay rate, α, and the number of simulations G. For each of these, we follow the recommendations given by Kohonen (2001), which are based on a survey of a large number of empirical implementations of the LVQ algorithm. These recommendations are implemented in the R package "class", which we use to generate our empirical results below. The specifics are as follows. We assign the same number of codebook vectors to each class, with this number being equal to the number of data points in the training sample that fall into the less frequently occurring class. In the context of identifying expansions vs. recessions, the class of recessions occurs less frequently. Thus, the total number of codebook vectors is set equal to twice the number of recession periods in the training period, with half of these being assigned to the recession class and half to the expansion class. We set α, which controls the decay rate in the LVQ algorithm, equal to 0.3.[8] Finally, the number of algorithm iterations, G, is set so as to achieve the convergence of the codebook vectors. We define convergence as being when the individual elements of the codebook vectors move by no more than 10^{-20} through one complete iteration of the algorithm.

Finally, to start the LVQ algorithm, we must initialize the codebook vectors. This initialization can affect the resulting class prediction, as the final placement of the codebook vectors in an LVQ algorithm is not invariant to the initialization. Here, we follow a simple automatic procedure. First, the codebook vectors that are assigned to the recession class are initialized to the actual data vectors from the recession periods in our training sample. For expansions, where there are fewer codebook vectors than data vectors, we initialize the codebook vectors to a randomly-drawn set of data vectors from the expansion periods in our training sample. In our empirical analysis, we base our turning point identification procedure on 100 runs of the LVQ algorithm, where different random initializations for the expansion codebook vectors are used in each run. This is discussed in more detail below.[9]

[8] Kohonen (2001) argues that the classification results from LVQ should be largely invariant to the choice of alternative values of α, provided that $\alpha^g \to 0$ as $g \to \infty$, which ensures that the sizes of the codebook vector updates eventually converge to zero. We have verified this for our application by generating results for nine alternative values of α between 0.1 and 0.9, in increments of 0.1. The results across these alternative values of α were nearly identical.

[9] In the Appendix A, we present results for LVQ algorithms with alternative numbers of codebook vectors, as well as for an LVQ algorithm with an alternative, data-based, approach to determining the initial locations of the codebook vectors for the expansion class.

Please cite this article in press as: Giusto, A., & Piger, J., Identifying business cycle turning points in real time with vector quantization. International Journal of Forecasting (2016), http://dx.doi.org/10.1016/j.ijforecast.2016.04.006

3. Real-time identification of turning points

In this section, we evaluate the performance of the LVQ classifier for the real-time identification of new US business cycle turning points over the period from November 1976 to July 2013, over which there were five complete NBER recession episodes, meaning five peaks and troughs. We evaluate both the accuracy and the speed with which turning points are identified, with the eventual NBER chronology serving as the standard for accuracy.

3.1. Data to be classified

The data series that we use as inputs to the algorithm are the four monthly coincident series that are highlighted by the NBER in their decisions about the timing of US business cycle turning points, namely (1) growth in non-farm payroll employment (E), (2) growth in the industrial production index (I), (3) growth in real personal income excluding transfer receipts (P), and (4) growth in real manufacturing and trade sales (M). These series are the focus of the existing literature on the real-time identification of US business cycle turning points.[10] The LVQ algorithm could easily be extended to consider larger numbers of indicators, with little increase in computational complexity. However, we restrict our attention to these four series in our primary analysis in order to maintain comparability with studies that use alternative approaches to identify turning points. We will consider the results from including additional series in Section 3.4.

We collect monthly data from February 1967 to July 2013, and use the NBER's recession chronology over this sample period to define expansion and recession months. We consider an analyst who applies the LVQ algorithm in real time each month between December 1976 and August 2013. To replicate the data that would have been available to this analyst accurately, we use a dataset that contains each release, or vintage, of these four coincident variables over each month in our evaluation period. Specifically, we collect a time series for each variable as it would have appeared at its monthly release, beginning with data released in December 1976, and ending with data released in August 2013. For each vintage, the sample collected begins in February 1967 and ends with the most recent data available for that vintage. We collect this vintage dataset beginning with the dataset used by Chauvet and Piger (2008), which was collected partly from the Federal Reserve Bank of Philadelphia real-time data archive described by Croushore and Stark (2001), and partly from past releases by government statistical agencies. This dataset was then extended to include data for the recent "Great Recession" and beyond using data releases from the Bureau of Economic Analysis and the Federal Reserve Bank of St. Louis ALFRED database.

The NBER recession and expansion phases are persistent, with the estimated probability over the 1967–2013 sample of remaining in an expansion month being equal to 0.98, and the corresponding probability for recessions months being equal to 0.9. We incorporate the information provided by this persistence into the classification problem by classifying a vector of data that contains both current values and one lag of the monthly series mentioned above. Thus, the vector to be classified is

$$x_t = (E_t, I_t, P_t, M_t, E_{t-1}, I_{t-1}, P_{t-1}, M_{t-1})' .$$

3.2. Real-time classification environment

The real-time classification environment is as follows. An analyst is attempting to determine whether a new business cycle turning point has occurred in the recent past, and is doing so in month $T + 1$. We assume that the analyst updates her inferences twice during the month, once at the beginning of the month, immediately following the release of the employment series for month T, and once near the middle of the month, immediately following the release of the industrial production series for month T.[11] The analyst will be able to form an inference about the business cycle phase that was in operation between the beginning of the sample period and month T at each of these points; however, different amounts of information will be available for forming this inference at each point.

For each month $T + 1$, the LVQ classifier is trained on a sample that extends from the beginning of the sample to the end of month $T - j$, over which the NBER classification is assumed to be known. In our analysis, j varies across data vintages, and is set using the following assumptions. (1) The date of a new peak or trough is assumed to be known once it is announced by the NBER. (2) A new peak will be announced by the NBER within twelve months of it occurring, where twelve months is the longest historical lag of the NBER's announcement of a new business cycle peak.[12] This assumption allows us to update the knowledge of the NBER phase during long expansions, despite the lack of any official NBER announcements over these periods. (3) Once the date of a new turning point has been announced by the NBER, the new NBER business cycle phase (expansion or recession) is assumed to last at least six months.[13] After training, the analyst then attempts to assess whether a new turning point has occurred over the period between $T - j + 1$ and T. That is, the classification is assumed to be unknown over this period, and the analyst uses the LVQ classifier to predict the eventual NBER classification.

As an example, suppose that an analyst is forming an inference in December 1980, so that T is November 1980.

[10] See e.g. Camacho, Perez-Quiros, and Poncela (2012), Chauvet and Piger (2008), and Fossati (in press).

[11] We could consider an analyst who also updates her inferences at the end of month $T + 1$, following the release of the personal income and manufacturing and trade sales series. However, because these releases generally occur only a few days before the employment series release at the beginning of month $T + 2$, we simply consider updates made at the beginning of each month here.

[12] The NBER announcement dates for past turning points are available at http://nber.org/cycles/main.html.

[13] These assumptions regarding the lags of knowledge of the NBER classification are meant to be a reasonable description of reality. However, they are not essential for our results, which are similar for other, more conservative assumptions.

Please cite this article in press as: Giusto, A., & Piger, J., Identifying business cycle turning points in real time with vector quantization. International Journal of Forecasting (2016), http://dx.doi.org/10.1016/j.ijforecast.2016.04.006

At this point, the most recent NBER announcement was made on June 3, 1980, and was an announcement of a business cycle peak in January 1980. Since we assume that this new recession phase will last a minimum of six months, the NBER classification is then known through July 1980, with the period from February 1980 to July 1980 classified as recession months. Thus, the value of j in this case is four months, and the analyst would be searching for a new business cycle trough over the period between August 1980 and November 1980. As a second example, suppose that the analyst was doing her analysis in December 2005, so that T is November 2005. Here, the most recent NBER announcement was made on July 17, 2003, and announced the November 2001 trough in economic activity. In this case, the value of j would be set to 12 months, with the period from December 2001 to November 2004 classified as expansion months. The analyst would then be searching for a new business cycle peak over the period between December 2004 and November 2005.

Note that when the analyst uses the LVQ classifier to predict the class out of sample, the most recent data to be classified will be incomplete. For example, when the analyst applies the LVQ classifier at the beginning of month $T+1$, the end-of-sample data vector x_T is

$$x_T = (E_T, E_{T-1}, I_{T-1}, P_{T-1})'.$$

Here, this vector is missing the time T values of I, P and M, and the time $T-1$ values of M. As was discussed in Section 2.3, we classify this incomplete data vector according to the nearest codebook vector in the dimensions on which x_T is not missing.

To determine whether a new turning point has occurred over the prediction period, we must convert the class predictions for individual months into predictions about a new turning point. Here, we follow Chauvet and Piger (2008) and require three consecutive months to be classified as a new business cycle phase prior to calling a new turning point. This enforces the prior belief that business cycle phases are persistent, which is consistent with the NBER definition of a business cycle phase. Specifically, if the most recent known NBER classification was a business cycle trough (peak), we identify a new business cycle peak (trough) if the LVQ classifies each month between $T-2$ and T as a recession (expansion) month. As was discussed in Section 2.4, we base our class prediction on 100 runs of the LVQ algorithm, where each run is based on a different random initialization of the codebook vectors. We consider a month between $T-2$ and T to be a recession or expansion month if at least 80% of the runs return this classification. This 80% rule builds in a preference against the identification of false turning points. It also mirrors the rule used by Chauvet and Piger (2008) to convert the recession probabilities from a Markov-switching model into the identification of new turning points, which aids the comparability of our results with this earlier study. At the end of Section 3.3, we consider the robustness of our results to thresholds other than 80%.

Once a new turning point has been identified, the date of this new turning point is established to be the month prior to the string of consecutive months that is classified as belonging to the new business cycle phase. For example, suppose that the most recent business cycle turning point identified was a business cycle trough, and months $T-2$ to T are classified as recession months, but month $T-3$ is not. In this case, a new business cycle peak would be established as occurring in month $T-3$.[14]

Table 1
US business cycle peaks identified in real-time.

NBER peak date	LVQ peak date	NBER identification lag	LVQ identification lag
1/1980	1/1980	123	92
7/1981	8/1981	158	126
7/1990	6/1990	267	78
3/2001	3/2001	239	216
12/2007	2/2008	335	158
Average:		**224**	**134**

Notes: The table gives NBER-established peak dates for recessions that occur over the sample period from November 1976 to July 2013, along with the peak date established in real time by the LVQ algorithm over this period. Identification lags are measured as the number of days after the last day of the NBER peak month that the NBER or LVQ peak month was identified.

3.3. Results

Table 1 presents the results of the turning point classification experiment for NBER business cycle peaks, while Table 2 shows the results for troughs. The first column of each table shows the monthly dates of NBER turning points over the period November 1976–July 2013, while the third column shows the number of days between the end of the official NBER turning point month and the day of the corresponding turning point announcement made by the NBER's Business Cycle Dating Committee. The second and fourth columns show the corresponding results for the LVQ algorithm. The second column shows the months that the LVQ algorithm established as turning points at the time when it first detected a new turning point. In cases where the turning point detected by the LVQ algorithm matches that of an NBER turning point reasonably well, the fourth column shows the number of days after the end of the official NBER turning point month that the LVQ algorithm would have detected the turning point initially. Thus, comparing the NBER dates to those established by the LVQ algorithm (columns 1 and 2) quantifies the accuracy of the dates established by the LVQ algorithm, and comparing columns 3 and 4 measures the timeliness of the LVQ algorithm for detecting NBER turning points relative to the NBER business cycle dating committee.

Beginning with the accuracy of the LVQ algorithm, it is clear from the tables that the LVQ algorithm replicates

[14] This decision rule uses the convention that a business cycle trough represents the last month of a recession and a business cycle peak represents the last month of an expansion. On the other hand, the NBER represents these turning points as inflection points that do not necessarily belong to either expansion or recession phases. However, previous statistical classification exercises have confirmed that associating peak months with expansions and trough months with recessions provides the best fit (see e.g. Chauvet & Piger, 2008).

Please cite this article in press as: Giusto, A., & Piger, J., Identifying business cycle turning points in real time with vector quantization. International Journal of Forecasting (2016), http://dx.doi.org/10.1016/j.ijforecast.2016.04.006

Table 2
US business cycle troughs identified in real-time.

NBER trough date	LVQ trough date	NBER identification lag	LVQ identification lag
7/1980	7/1980	341	127
11/1982	12/1982	219	136
3/1991	4/1991	631	443
11/2001	1/2002	593	308
6/2009	6/2009	446	157
Average:		**446**	**234**

Notes: The table gives NBER-established trough dates for recessions that occur over the sample period from November 1976 to July 2013, along with the trough date established in real time by the LVQ algorithm over this period. Identification lags are measured as the number of days after the last day of the NBER trough month that the NBER or LVQ trough month was identified.

the NBER peak and trough dates very accurately. There are no instances in which an NBER turning point date in column 1 is not matched by a similar date produced by the LVQ algorithm. Also, the LVQ algorithm never produces a turning point date that does not match an NBER date. In other words, the LVQ algorithm produces no false positives and no false negatives. Furthermore, the accuracy of the peak and trough dates is impressive. For both peaks and troughs, the average absolute difference between the turning point dates established by the LVQ algorithm and the corresponding NBER dates is only 0.8 months.

These results suggest that the LVQ algorithm is a very accurate technique for establishing the dates of NBER turning points in real time. Having determined this, the primary question of interest is the timeliness with which this accurate identification is achieved. Focusing on the fourth column of each table, the LVQ algorithm establishes business cycle peaks with an average delay of 134 days, and business cycle troughs with an average delay of 234 days. This is substantially faster than the NBER's business cycle dating committee. As the second column of each table shows, the committee has announced peak and trough dates with average lag times of 224 days for peaks and 446 days for troughs. Thus, for the last five recessions, the LVQ algorithm would have established essentially the same turning point dates as the NBER, but an average of three months faster for peaks and seven months faster for troughs.

Admittedly, the NBER dating committee is not overly concerned with timeliness, and instead has focused historically on the establishment of correct dates. Thus, it is more relevant to compare the algorithm with alternative statistical approaches that are used in the literature. To this end, we first compare the results from the LVQ algorithm to those of other statistical approaches that were in use for identifying the beginning and end of the 2007–2009 Great Recession, as summarized by Hamilton (2011). To keep the analysis comparable, we focus on alternative approaches that use the same four coincident series that we use here. The LVQ procedure would have established the business cycle peak for the most recent recession on June 6, 2008, which is 158 days after the end of the official NBER peak month, December 2007. In contrast, Hamilton (2011) reports that other real-time approaches that were in use at the time did not identify a business cycle peak until the Fall of 2008 or the Winter of 2009.

Table 3
Average real-time identification lag: robustness to threshold.

Threshold	Peaks (days)	Troughs (days)	False cycles
90%	136.4	319.6	0
80%	134	234.2	0
70%	97.8	205.8	1
60%	97.8	200	1
50%	97.8	200	3

Notes: The table gives the average number of days required for the LVQ algorithm to identify NBER-established peak and trough dates, where we vary the threshold required to signal a new turning point from the LVQ algorithm output. For each value of this threshold, the table also shows the number of non-NBER recession episodes (false cycles) identified by the LVQ algorithm.

For recessions prior to the Great Recession, we compare our results to those given by Chauvet and Piger (2008), who investigated the real-time performance of the dynamic factor Markov-switching (DFMS) model of Chauvet (1998) for identifying the four NBER peak and trough dates that occur over the period from November 1976 to June 2006. The comparison to this analysis is particularly applicable, as the variables to be classified and the requirements used to convert the monthly classifications into turning point calls are identical to those used here. Also, the DFMS model is a technique that is used commonly to identify turning points in real time.[15]

Chauvet and Piger (2008) find that, on average, the DFMS model does not improve on the timeliness of the NBER business cycle dating committee for identifying business cycle peaks, but does improve on the NBER by an average of 5.5 months for calling NBER business cycle troughs. Over the same four recessions that were covered by Chauvet and Piger (2008), the LVQ algorithm improves on the NBER by an average of 2.3 months for calling business cycle peaks and 6.5 months for business cycle troughs. Thus, the LVQ algorithm in our forecast experiment identifies both business cycle peaks and troughs faster than the DFMS model in the forecast experiment of Chauvet and Piger (2008). In comparing the two sets of results, it is important to note that Chauvet and Piger (2008) assume that the analyst only estimates the DFMS on data sets that end with the most recent data on manufacturing and trade sales, which puts the DFMS model at up to a two-month disadvantage over our implementation of the LVQ algorithm, which uses all data as they become available.[16] Even if we adjust a full two months to allow for the maximum potential advantage of the data analyst in our forecast experiment over that of Chauvet and Piger (2008), the LVQ algorithm is still competitive with DFMS, being slightly quicker at identifying peaks and slightly slower at identifying troughs. Thus, the LVQ algorithm appears to be promising relative to a commonly used alternative.

[15] See Camacho et al. (2012) and Chauvet and Hamilton (2006).

[16] The LVQ algorithm has the ability to identify a new business cycle turning point as of the month T employment release on the first Friday of month $T+1$, which could be nearly two months prior to the release of the month T manufacturing and trade sales release at the end of month $T+2$.

Please cite this article in press as: Giusto, A., & Piger, J., Identifying business cycle turning points in real time with vector quantization. International Journal of Forecasting (2016), http://dx.doi.org/10.1016/j.ijforecast.2016.04.006

As was described in Section 3.2, the results in Tables 1 and 2 are based on the use of a threshold of 80% for signaling new business cycle turning points. Table 3 assesses the robustness to this threshold by presenting the average numbers of days required for the LVQ algorithm to identify NBER peak and trough dates when we vary this threshold from 50% to 90%. As expected, lowering the threshold leads to a more timely identification of the turning point dates. However, this comes at the cost of some false positives, in the form of non-NBER recession episodes that are identified by the algorithm. For example, lowering the threshold to 70% identifies NBER peak and trough dates about one month faster than the 80% threshold on average, but the algorithm also identifies one false recession. Interestingly, lowering the threshold below 70% provides small to no gains in timeliness. The number of false positives remains modest, reaching a high of three when the threshold is set to 50%.

3.4. Results with additional data series

The analysis of the previous section applied the LVQ algorithm to four monthly series that have been highlighted consistently by the NBER in announcements regarding the dates of US business cycle turning points. However, it is possible that considering extra monthly series in addition to those mentioned explicitly by the NBER may help to provide a more timely identification of business cycle turning points. We explore this possibility here by repeating the out-of-sample experiment described in the previous section, but with additional data added to the analysis.

There are many additional monthly coincident indicators that we could include, and the recent literature has largely taken the path of extracting a small number of factors from a large set of indicators for use in forecasting. For example, Stock and Watson (1999) forecast inflation successfully using a factor constructed from 61 indicators of macroeconomic activity, while Ludvigson and Ng (2009) explained bond risk premia using factors constructed from 132 macroeconomic and financial time series that are thought to comove with the business cycle. Our approach here is similar, with our dataset being augmented with a factor that is obtained from monthly coincident indicators of macroeconomic activity. We measure this factor following Fossati (in press), and use the Chicago Fed National Activity Index (CFNAI). This index, which is released near the end of each month by the Chicago Federal Reserve Bank, is the first principal component of 85 series that have been identified as coincident with the US business cycle. We are able to obtain each release, or vintage, of the CFNAI stretching back to its inception in January 2001, which allows us to retain the realism of our real-time out-of-sample nowcasting experiment, albeit over a shorter out-of-sample period than our primary analysis. An alternative would be to construct a similar factor directly from the underlying data series, recursively over a longer out-of-sample period. We do not take this approach because it is not feasible to obtain the vintages that would have been available to an analyst who was applying LVQ in real time for such a large set of data series.

Table 4
US business cycle peaks and troughs identified in real-time: additional monthly coincident indicators.

NBER peak date	LVQ peak date	NBER identification lag	LVQ identification lag
3/2001	3/2001	239	251
12/2007	2/2008	335	158
NBER trough date	**LVQ trough date**	**NBER identification lag**	**LVQ identification lag**
11/2001	1/2002	239	166
6/2009	7/2009	335	199

Notes: The table replicates the analysis in Tables 1–2 when the dataset is augmented with the Chicago Fed National Activity Index.

Table 4 replicates the analysis in Tables 1–2 over the out-of-sample period from January 2001 to July 2013, but with the dataset augmented with the CFNAI.[17] We see some mixed results from including additional monthly coincident indicators. First, including the CFNAI does not change the dates of turning points established in real time materially. For the four turning points in the out-of-sample period, only one date was altered, and this by only one month. Second, the timeliness of the algorithm is improved significantly for one turning point in the sample, namely the trough of the 2001 recession. The inclusion of CFNAI allowed this trough to be identified by the algorithm 142 days earlier than with the algorithm applied to data without CFNAI. Finally, the addition of CFNAI slows the identification of two other turning points, by 35 days for the 2001 business cycle peak, and by 42 days for the 2009 business cycle trough. Thus, overall, the addition of CFNAI provides an improvement in the average speed with which turning points are identified, although this is not true for all, or even the majority, of the turning points in the out-of-sample period.

4. Conclusion

Non-parametric machine learning algorithms are used commonly in many disciplines to classify data as belonging to one of a set of classes. We have proposed a particularly salient algorithm, known as learning vector quantization, for the purpose of classifying economic data into expansion and recession regimes. Of particular interest is the ability of the algorithm to identify US business cycle turning points accurately and quickly in real time.

We evaluate the real-time performance of the LVQ algorithm for identifying business cycle turning points in the United States over the past 35 years and five recessions. The LVQ algorithm identified the dates of all five recessions over this period accurately, with no false positives, and at an impressive speed. For example, the LVQ algorithm would have identified the December 2007 peak in economic activity by early June 2008, several months ahead of the statistical tracking procedures reviewed by Hamilton (2011) as being in use at the time. Looking across all recessions, the algorithm's speed of identifying peaks and troughs over our sample period is very competitive

[17] Technically, we use the "CFNAI3" series, which is the three-month moving average of the CFNAI.

Please cite this article in press as: Giusto, A., & Piger, J., Identifying business cycle turning points in real time with vector quantization. International Journal of Forecasting (2016), http://dx.doi.org/10.1016/j.ijforecast.2016.04.006

with that of the dynamic factor Markov-switching model, a technique that is used commonly for dating business cycles in real time.

Acknowledgments

We thank two anonymous referees, Travis Berge, and seminar participants at Dalhousie University and the Federal Reserve Bank of St. Louis Workshop on Applied Econometrics for helpful comments.

Appendix A

As was described in Section 2.4, the implementation of the LVQ algorithm requires both the specification of the number of codebook vectors and the development of a strategy for the initialization of these codebook vectors. In this appendix, we consider the performance of the LVQ algorithm when we set the number and initial location of codebook vectors using alternative approaches to that used in the baseline implementation that generated our primary results. In particular, we repeat the analysis presented in Tables 1–2 when using these different implementations of the LVQ algorithm.

Again, we set the number of codebook vectors for our baseline version of LVQ, denoted $\bar{N}$, by following the prescriptions of Kohonen (2001) and assigning the same number of codebook vectors to both the recession and expansion classes, where this number is equal to the number of data points in the training sample that fall into the recession class. We consider two alternative values for $\bar{N}$ here. In the first, $\bar{N} = 2$, so that each class is described by only a single codebook vector. The results for this case are given in Tables A.1–A.2. In the second, we set $\bar{N}$ equal to the number of recession months, so that each class is described by a number of codebook vectors that is equal to half of the number of recession months. The results for this case are given in Tables A.3–A.4.

Table A.1
US business cycle peaks identified in real-time: an alternative number of codebooks.

NBER peak date	LVQ peak date	NBER identification lag	LVQ identification lag
1/1980	2/1980	123	127
7/1981	9/1981	158	161
7/1990	7/1990	267	129
NA	10/1991	NA	NA
3/2001	8/2001	239	251
12/2007	7/2008	335	312
Average:		**224**	**196**

Notes: The table replicates the analysis in Table 1, but using an alternative number of codebook vectors in the LVQ algorithm. The total number of codebook vectors is set equal to two, so that the expansion and recession classes each have a single codebook vector.

As the tables demonstrate, lowering the number of codebook vectors substantially relative to our baseline implementation generally causes the performance of the LVQ algorithm to deteriorate. When there is only one codebook vector per class, the algorithm identifies each of the five NBER peaks more slowly than our baseline implementation of LVQ, being roughly two months slower at calling peaks on average. For troughs, though, LVQ with a single codebook per class identifies two troughs substantially more quickly than the baseline LVQ. However, the extent of this improvement is misleading, as one of these quickly identified troughs is followed by a false recession episode in late 1991. The turning point dates established by LVQ with a single codebook per class are generally less accurate as well. For example, the December 2007 NBER peak is dated to July 2008. Increasing the number of codebooks so that $\bar{N}$ equals the number of recession months (Tables Table A.3–A.4) improves the performance of the algorithm somewhat over the case of $\bar{N} = 2$, particularly in the speed

Table A.2
US business cycle troughs identified in real-time: an alternative number of codebooks.

NBER trough date	LVQ trough date	NBER identification lag	LVQ identification lag
7/1980	7/1980	341	127
11/1982	11/1982	219	136
3/1991	4/1991	631	124
NA	1/1992	NA	NA
11/2001	12/2001	593	126
6/2009	7/2009	446	192
Average:		**446**	**141**

Notes: The table replicates the analysis in Table 2, but using an alternative number of codebook vectors in the LVQ algorithm. The total number of codebook vectors is set equal to two, so that the expansion and recession classes each have a single codebook vector.

Table A.3
US business cycle peaks identified in real-time: an alternative number of codebooks.

NBER peak date	LVQ peak date	NBER identification lag	LVQ identification lag
1/1980	2/1980	123	127
7/1981	9/1981	158	161
7/1990	7/1990	267	94
NA	10/1991	NA	NA
3/2001	3/2001	239	108
12/2007	7/2008	335	312
Average:		**224**	**160**

Notes: The table replicates the analysis in Table 1, but using an alternative number of codebook vectors in the LVQ algorithm. The total number of codebook vectors is set equal to the number of recession months in the training sample, with the expansion and recession classes each being assigned half of these codebook vectors.

Table A.4
US business cycle troughs identified in real-time: an alternative number of codebooks.

NBER trough date	LVQ trough date	NBER identification lag	LVQ identification lag
7/1980	8/1980	341	127
11/1982	12/1982	219	164
3/1991	4/1991	631	215
NA	9/1992	NA	NA
11/2001	1/2002	593	166
6/2009	7/2009	446	192
Average:		**446**	**173**

Notes: The table replicates the analysis in Table 2, but using an alternative number of codebook vectors in the LVQ algorithm. The total number of codebook vectors is set equal to the number of recession months in the training sample, with the expansion and recession classes each being assigned half of these codebook vectors.

Please cite this article in press as: Giusto, A., & Piger, J., Identifying business cycle turning points in real time with vector quantization. International Journal of Forecasting (2016), http://dx.doi.org/10.1016/j.ijforecast.2016.04.006

with which it calls business cycle peaks. However, the algorithm still generates a false positive recession episode, and is less accurate than the baseline implementation at dating business cycle turning points.

We next investigate the effects on the algorithm's performance of an alternative approach to the initialization of the location of codebook vectors in the LVQ algorithm. In our baseline implementation, we initialize the recession codebooks, of which there are the same number as recession months, to equal the actual data vectors for the recession months. We initialize the expansion codebooks, of which there are fewer than the number of expansion months, to be drawn randomly from the data vectors for the expansion months. In contrast, we could use a data-based approach to identify clusters in the expansion data, which one might think would be good initial guesses for the codebook vectors. To implement this idea, we apply a k-means clustering algorithm to the expansion data vectors in the training sample in order to identify $\bar{N}/2$ clusters. These clusters are then used as initial values for expansion codebooks in the LVQ algorithm.

The results from using LVQ with this alternative initialization are shown in Tables A.5–A.6. The results are largely similar to those obtained from our baseline algorithm. The turning point dates established in real time are almost identical, never deviating from those established by the baseline specification by more than one month. The speed at which these dates are established is somewhat worse, 12 days slower on average for peaks and 24 days slower on average for troughs. Overall, the LVQ algorithm's performance for dating business cycle peaks is fairly robust to this alternative initialization procedure, although the simple initialization procedure used in our baseline version of LVQ performs best.

Table A.5
US business cycle peaks identified in real-time: an alternative initialization of the LVQ algorithm.

NBER peak date	LVQ peak date	NBER identification lag	LVQ identification lag
1/1980	1/1980	123	127
7/1981	7/1981	158	98
7/1990	7/1990	267	94
3/2001	3/2001	239	251
12/2007	2/2008	335	158
Average:		**224**	**146**

Notes: The table replicates the analysis in Table 1, but using a k-means clustering algorithm to initialize the codebook vectors for the expansion class in the LVQ algorithm.

Table A.6
US business cycle troughs identified in real-time: an alternative initialization of the LVQ algorithm.

NBER trough date	LVQ trough date	NBER identification lag	LVQ identification lag
7/1980	8/1980	341	127
11/1982	12/1982	219	220
3/1991	4/1991	631	443
11/2001	12/2001	593	308
6/2009	7/2009	446	192
Average:		**446**	**258**

Notes: The table replicates the analysis in Table 2, but using a k-means clustering algorithm to initialize the codebook vectors for the expansion class in the LVQ algorithm.

Appendix B. Supplementary material

Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.ijforecast.2016.04.006.

References

Berge, T. (2015). Predicting recessions with leading indicators: Model averaging and selection over the business cycle. *Journal of Forecasting*, *34*(6), 455–471.
Berge, T., & Jordá, O. (2011). The classification of economic activity into expansions and recessions. *American Economic Journal: Macroeconomics*, *3*(2), 246–277.
Camacho, M., Perez-Quiros, G., & Poncela, P. (2012). Markov-switching dynamic factor models in real time. CEPR Working Paper No. 8866.
Chauvet, M. (1998). An econometric characterization of business cycle dynamics with factor structure and regime switching. *International Economic Review*, *39*, 969–996.
Chauvet, M., & Hamilton, J. D. (2006). Dating business cycle turning points. In P. R. Costas Milas, & D. van Dijk (Eds.), *Nonlinear time series analysis of business cycles* (pp. 1–53). North Holland: Elsevier.
Chauvet, M., & Piger, J. (2008). A comparison of the real-time performance of business cycle dating methods. *Journal of Business and Economic Statistics*, *26*(1), 42–49.
Chauvet, M., & Potter, S. (2005). Forecasting recessions using the yield curve. *Journal of Forecasting*, *24*(2), 77–103.
Croushore, D., & Stark, T. (2001). A real-time data set for macroeconomists. *Journal of Econometrics*, *105*, 111–130.
Fossati, S. (2016). Dating US business cycle with macro factors. *Studies in Nonlinear Dynamics and Econometrics*, in press.
Fushing, H., Chen, S.-C., Berge, T., & Jordá, O. (2010). A chronology of international business cycles through non-parametric decoding. Working paper.
Hamilton, J. D. (2011). Calling recessions in real time. *International Journal of Forecasting*, *27*(4), 1006–1026.
Hammer, B., & Villmann, T. (2002). Generalized relevance learning vector quantization. *Neural Networks*, *15*(8–9), 1059–1068.
Harding, D., & Pagan, A. (2006). Synchronization of cycles. *Journal of Econometrics*, *132*, 59–79.
Kauppi, H., & Saikkonen, P. (2008). Predicting US recessions with dynamic binary response models. *The Review of Economics and Statistics*, *90*(4), 777–791.
Kohonen, T. (2001). *Self-organizing maps*. Berlin: Spring-Verlag.
Ludvigson, S. C., & Ng, S. (2009). Macro factors in bond risk premia. *Review of Financial Studies*, *22*(12), 5027–5067.
Ng, S. (2014). Boosting recessions. *Canadian Journal of Economics*, *47*(1), 1–34.
Qi, M. (2001). Predicting US recessions with leading indicators via neural network models. *International Journal of Forecasting*, *17*, 383–401.
Rudebusch, G., & Williams, J. (2009). Forecasting recessions: The puzzle of the enduring power of the yield curve. *Journal of Business and Economic Statistics*, *27*(4), 492–503.
Sato, A., & Yamada, K. (1995). Generalized learning vector quantization. In D. T. G. Tesauro, & T. Leen (Eds.), *Advances in neural information processing systems* (pp. 423–429). North Holland: Elsevier.
Stock, J. H., & Watson, M. W. (1999). Forecasting inflation. *Journal of Monetary Economics*, *44*(2), 293–335.
Stock, J. H., & Watson, M. W. (2010). Indicators for dating business cycles: Cross-history selection and comparisons. *American Economic Review Papers and Proceedings*, *100*(2), 16–19.
Stock, J. H., & Watson, M. W. (2014). Estimating turning points using large data sets. *Journal of Econometrics*, *178*(1), 368–381.
Vishwakarma, K. (1994). Recognizing business cycle turning points by means of a neural network. *Computational Economics*, *7*, 175–185.

Andrea Giusto is an Associate Professor of Economics at Dalhousie University. He received his Ph.D. from the University of Oregon in 2009.

Jeremy Piger is Professor of Economics at the University of Oregon. He received his Ph.D. from the University of Washington in 2000. His previous employment includes the Federal Reserve Board and the Federal Reserve Bank of St. Louis. He is an Associate Editor for the *Journal of Money, Credit and Banking and Studies in Nonlinear Dynamics and Econometrics*.

Please cite this article in press as: Giusto, A., & Piger, J., Identifying business cycle turning points in real time with vector quantization. International Journal of Forecasting (2016), http://dx.doi.org/10.1016/j.ijforecast.2016.04.006