Distance Measures for Effective Clustering of ARIMA Time-Series[1]

Konstantinos Kalpakis, Dhiral Gada, and Vasundhara Puttagunta
CSEE Department, UMBC, 1000 Hilltop Circle, Baltimore, MD 21250
{kalpakis, dgada1, vputta1}@csee.umbc.edu

Abstract

Many environmental and socioeconomic time–series data can be adequately modeled using Auto-Regressive Integrated Moving Average (ARIMA) models. We call such time–series ARIMA time–series. We consider the problem of clustering ARIMA time–series. We propose the use of the Linear Predictive Coding (LPC) cepstrum of time–series for clustering ARIMA time–series, by using the Euclidean distance between the LPC cepstra of two time–series as their dissimilarity measure. We demonstrate that LPC cepstral coefficients have the desired features for accurate clustering and efficient indexing of ARIMA time–series. For example, few LPC cepstral coefficients are sufficient in order to discriminate between time–series that are modeled by different ARIMA models. In fact this approach requires fewer coefficients than traditional approaches, such as DFT and DWT. The proposed distance measure can be used for measuring the similarity between different ARIMA models as well.

We cluster ARIMA time–series using the Partition Around Medoids method with various similarity measures. We present experimental results demonstrating that using the proposed measure we achieve significantly better clusterings of ARIMA time–series data as compared to clusterings obtained by using other traditional similarity measures, such as DFT, DWT, PCA, etc. Experiments were performed both on simulated as well as real data.

Keywords: time–series, similarity measures, clustering, ARIMA models, cepstral coefficients.

1 Introduction

Data–retrieval and data–mining applications in time–series databases have been gaining growing interest lately. Time–series form an important class of data objects that arise from various sources such as environmental and socioeconomic systems [4]. Typical applications on time–series deal with tasks like classification, clustering, similarity search, prediction, forecasting, outlier detection and noise removal. These applications rely heavily on the ability to measure the similarity or dissimilarity between time–series [1, 12]. The notion of similarity of complex objects such as time–series is specific to the application domain and also to the nature of the tasks [12]. Defining similarity is non–trivial. Simple equality or inequality is of little use. Also, time–series data tend to be very long because of which they suffer from the curse of dimensionality.

Previous Work. The problem of similarity search in time-series databases is extensively studied. Approaches differ mainly in their notion of similarity. Several different measures of pairwise similarity and dissimilarity have been proposed in the classification literature [11].

Agrawal et al [1] use the Euclidean distance between time–series of equal length as the measure of their similarity. They reduce sequences into points in low-dimensional space by using Discrete Fourier Transform (DFT). Parseval's theorem ensures that there are no false-dismissals in doing so. In addition, this approach improves upon the measurement of similarity between time-series since the effects of high frequency components in the DFT, which usually correspond to noise, are discarded. The idea has been generalized in [8] for subsequence matching. In a similar manner Struzit et al [17] use Discrete Wavelet Transform (DWT) and Gavrilov et al [10] use Principal Component Analysis (PCA) for measuring time–series similarity.

However there are many similarity queries where Euclidean distances between raw data elements fail to capture the notion of similarity (see [15] for examples). Agrawal et al [2] present a more intuitive idea that two series should be considered similar if they have enough non-overlapping time–ordered pairs of subsequences that are similar. The model allows translation and amplitude scaling. It also allows non-matching gaps in the matching subsequences. Rafiei et al [16] use moving window average for smoothening time–series and time-scaling (global stretching or shrinking of time axis). Yi et al [21] use time–warping distance as the similarity measure and look at the problem of indexing time–series when local time–warping transformations are allowed. Das et al [7] present a similarity model as follows: for a fixed set $\mathcal{F}$ of transformations (eg. the set of all linear transformations), two time–series X and Y are $\mathcal{F}$–similar if there exists a transformation f in $\mathcal{F}$ such that a long subsequence X' of X can be approximately mapped to a long subsequence Y' of Y using f.

Gavrilov et al [10] raise the question "Which (similarity)

[1]Supported in part by NASA under Contract NAS5–32337 and Cooperative Agreement NCC5–315. Please send all correspondence to Dr. Kalpakis.

measure is the best?" for mining stock market time–series. They observed that normalizing the time–series in any form always improved the quality of clustering. They conclude that piece-wise normalization and normalized derivatives results in highest quality clustering for stock–market data. But when they get the best clustering results, the time–series do not seem to be prone to dimensionality reduction.

Most existing approaches for mining time–series data do not appropriately take into account the stochastic properties of the time–series. A (1–dimensional) time–series is a sequence of observations of a particular variable. It consists of four components: a trend, a cycle, a stochastic persistence component, and a random element [4]. The stochastic component is present in almost all environmental and socioeconomic time–series. Therefore, to accurately represent time–series and to define similarity between them, it is important to consider all four components of the time–series. These four components can be captured by modeling it by a Box-Jenkins seasonal model [20] (see Section 2). This model is also called an Auto-Regressive Integrated Moving Average (ARIMA) model. This model has been found very useful for describing a variety of seasonal environmental and socioeconomic time–series. It is further described in section 3. We call time–series that can be described or generated by ARIMA models *ARIMA time–series*.

Suppose we have a collection of time–series generated by different ARIMA models. In this paper, we attempt to answer the question "Which (similarity) measure is the best?" for such time–series. We propose to use the Euclidean distance between the LPC cepstra of time–series as a measure of their distance (dissimilarity). The most striking characteristics of this measure we propose are that it gives more accurate clusterings and achieves large dimensionality reduction. Since we use the estimated model of the time–series for clustering, we overcome problems that arise due to time–series of different lengths and time–series that are growing (as long as the model does not change).

Our Approach. Our notion of similarity is that two time–series are similar if the underlying physical models that generate them are the same or close. The intuition behind this notion of similarity is that, if the parameters of models fitted to the time–series are close, then the time–series behave in a similar manner (probabilistically). Two models are similar when one model can be fitted to a sequence generated by another model. Such a similarity measure can be used to more accurately cluster time–series. It enables us to draw inferences (extract knowledge) about a time–series from others belonging to the same cluster and improves our knowledge about the dynamics of the system being studied. For eg.(pg. 15 in [4]), the Keynesian macro-economic model based on Keynes's general theory of employment, explains the changes in the national income Y_t as $Y_t = a_1 Y_{t-1} - a_2 Y_{t-2} + b_0 U_t$, where Y_t and U_t are the national income and government spending at quarter t, and a_1, a_2, and b_0 are the parameters of the model. (Intuitively, these parameters depend on the rate of consumption, savings, and investment in an economy.) Considering a collection of national–income time–series, from various economies, how could one cluster those time–series so that series that are clustered together have similar spending, savings, and investment characteristics? Consider a case where two countries have the same values of parameters - a_1, a_2 and b_0 for the model fitted to their national income time series(Y_t). In such a case, we can make meaningful inferences about the economy of one country from the spending, savings, and investment characteristics of another country.

In using fitted ARIMA models for clustering time–series data, it is necessary to define an appropriate distance measure between ARIMA models and hence time–series. We define such a distance measure by using the Euclidean distance between the Linear Predictive Coding (LPC) cepstrum of two time–series (models). Our results show that the LPC cepstrum provides higher discriminatory power between time–series and superior clusterings than other widely used methods such as Euclidean distance between the DFT, DWT (with the S8 wavelet [14]), PCA, and DFT of the auto-correlation function (ACF) of two time–series. The LPC cepstrum retains high amount of information about the underlying time–series in very few coefficients. This property makes the LPC cepstrum a very effective similarity measure for clustering time–series data and models.

Organization. The organization of the rest of the paper is as follows. Preliminaries and definitions are given in Section 2. Properties of the LPC cepstrum of the time–series are given in Section 3. In Section 4 we present our results from clustering synthetic ARIMA time–series and real datasets. We conclude the paper in Section 5.

2 Preliminaries

We start with a brief description of various concepts and definitions used in this paper. Interested reader is referred to [4, 20, 22] for more details. A stationary time–series is one whose probability distribution is time-invariant. Note that a non-stationary time–series may have its mean μ_t or variance σ_t varying with time or have a trend. As mentioned earlier, a time–series has four components: a trend, a cycle, a stochastic persistence component, and a random element. These can be captured by modeling it by a Box-Jenkins seasonal model: $\Phi_P(B^s)\phi_p(B)(1-B)^d(1-B^s)^D X_t = \theta_q(B)\Theta_Q(B^s)\epsilon_t$. This model is also called ARIMA $(p,d,q) \times (P,D,Q)$ model. Here B is an operator such that $B^p X_t = X_{t-p}$, s is the seasonality (periodicity) of the time series, p and q are the orders of the autoregressive and moving average components respectively, P and Q are the orders of the seasonal autoregressive and moving average components respectively, d is the order of differencing and D is the

order of seasonal differencing,(d=0 for a stationary time series, $d \geq 1$ for non-stationary time series), $\Phi_p(B^s) = (1 - \Phi_1 B^s - \ldots - \Phi_P B^{Ps})$ represents the correlation between the seasonal elements of the time–series, $\phi_p(B) = (1 - \phi_1 B - \ldots - \phi_p B^p)$ represents the correlation of X_t on its preceding values, $\Theta_q(B^s) = (1 - \Theta_1 B - \ldots - \Theta_q B^{Qs})$ represents the seasonal moving average component, $\theta_q(B) = (1 - \theta_1 B - \ldots - \theta_q B^q)$ represents the moving average component, ϵ_t is a sequence of uncorrelated random variables with constant mean and variance. Some special cases of the ARIMA $(p, d, q) \times (P, D, Q)$ model are as follows: AR(p) model is given as ARIMA $(1, 0, 0) \times (0, 0, 0)$; MA($q$) model is given as ARIMA $(0, 0, 1) \times (0, 0, 0)$; ARMA($p, q$) model is a combination of the stationary AR(p) and MA(q) models. For brevity, we refer to ARIMA $(p, d, q) \times (0, 0, 0)$ model as ARIMA (p, d, q) model.

We measure the dissimilarity between time–series using the Euclidean distance between their LPC cepstral coefficients. Cepstral analysis is a non–linear signal processing technique with a variety of applications in areas such as speech and image processing [9]. The cepstrum is defined as the inverse Fourier transform of the short-time logarithmic amplitude spectrum. One characteristic feature of the cepstrum is that it allows for the separate representation of the spectral envelope and fine structure. The real cepstrum is defined as the inverse Fourier transform of the real logarithm of the Fourier transform of the time–series. The complex cepstrum is defined as the inverse Fourier transform of the complex logarithm of the Fourier transform of the time–series. The cepstrum of an ARIMA time–series can be estimated using the parameters of an ARIMA model for that time–series. The cepstrum defined using the auto-regression coefficients is referred to as the LPC cepstrum, since it is derived through Linear Predictive Coding models (e.g. ARIMA) for the time–series. Hereafter, unless otherwise specified, we refer to the LPC cepstrum of a time–series simply as its cepstrum.

Consider a time–series X_t defined by an AR(p) model $X_t + \alpha_1 X_{t-1} + \ldots + \alpha_p X_{t-p} = \epsilon_t$where $\alpha_1, \ldots, \alpha_p$ are the auto-regression coefficients and ϵ_t is white noise with mean 0 and certain non-zero variance. Note that for every ARIMA model there exists an equivalent AR model, that can be obtained from the ARIMA model by polynomial division. Hence, without loss of generality, for the remainder of this paper we focus on AR time–series.

The cepstral coefficients for an AR(p) time–series can be derived from the auto-regression coefficients [9]:

$$c_n = \begin{cases} -\alpha_1, & \text{if } n = 1 \\ -\alpha_n - \sum_{m=1}^{n-1}(1 - \frac{m}{n})\alpha_m c_{n-m}, & \text{if } 1 < n \leq p \\ -\sum_{m=1}^{p}(1 - \frac{m}{n})\alpha_m c_{n-m}, & \text{if } p < n \end{cases} \quad (1)$$

We use cepstral coefficients to extract the significant features of time–series and define distance between two time series as the Euclidean distance between their cepstra.

Given two clusterings $G = G_1, \ldots, G_k$(say the "ground truth") and $A = A_1, \ldots, A_k$ (obtained using any feature extraction method, distance measure, or clustering method), the cluster similarity metric is defined as $Sim(G, A) = (\sum_i \max_j Sim(G_i, A_j))/k$, where, $Sim(G_i, A_j) = 2|G_i \bigcap A_j|/(|G_i| + |A_j|)$. Sim(G,A) can be used to evaluate the clustering results [10].

Silhouette width [14] is another measure for the quality of clustering. It lies between -1 and 1. For each object that is clustered, the silhouette width for that object can be interpreted as follows: if it is close to 1 then the object is well clustered (classified); if it is close to 0 then the object lies between two clusters; and if it is close to -1 then the object is badly clustered. The average silhouette width of all the clustered objects is another measure of quality of the clustering achieved. To evaluate our clustering experiments, we use both Sim(G,A) and silhouette width.

3 Properties of Cepstral coefficients

In this section we present some of our findings on the properties of cepstral coefficients of stationary time–series generated by different models. These observations justify our use of cepstral distance as an effective similarity measure and also as a good dimensionality reduction method for clustering purposes. A detailed discussion and experimental results can be found in [13].

3.1 Effective at distinguishing models.

There can be various ways of distinguishing different AR models. Suppose we have 3 models (M_1–M_3): $x_t = 0.3x_{t-1} + 0.2x_{t-2} + \epsilon_t$ (M_1), $x_t = 0.3x_{t-1} + 0.2x_{t-2} + 0.1x_{t-5} + \epsilon_t$ (M_2), and $x_t = 0.4x_{t-1} + 0.2x_{t-2} + \epsilon_t$ (M_3). The coefficients of M_1 and M_2 differ by 0.1. The coefficients of M_1 and M_3 also differ by 0.1. However, M_2 differs from M_1 in it's correlation with the 5^{th} coefficient which is relatively less significant as compared to M_1 differing from M_3 in it's correlation with the 1^{st} coefficient. According to our notion of similarity, M_1 is more similar to M_2 than it is to M_3. In essence, it means that for an AR(p) model, lower the order of the AR coefficient lower its importance.

In the above example it can be seen that simple Euclidean or Manhattan distances between the model parameters will not be of use. One might think of using some other distance measures such as the maximum distance between the model parameters or the distance between the principal components of the model parameters. Our experiments reveal that these too do not serve our purpose (see [13]). One could use weighted Euclidean distance, $\sum_{i=1}^{p} w_i(\alpha_i - \alpha_i')^2$. Here, α_i and α_i' are the i^{th} AR coefficients of models 1 and 2 respectively, p is the order of the AR model. w_i is the weight for the i^{th} AR coefficient set to $w_i = c^i/K$, where $K = \sum_{i=1}^{p} c^i$, for some constant $0 < c < 1$. The results of the weighted Euclidean method depend on the combination of the weights and the model parameters. We observe that when cepstral distance is used as the similarity measure,

the distance between M_1 and M_2 is always lower than the distance between M_1 and M_3. The cepstral distance can be considered as a special case of the weighted Euclidean distance. Cepstral coefficients are effective as a similarity measure between models because they are sensitive to the position of the AR coefficients.

3.2 Decay rapidly to zero.

The cepstral coefficients decay rapidly to zero and thus, we need to retain only the first few coefficients to capture most of the information in the time–series. This property is important to overcome the curse of dimensionality. For an AR(1) model given as $X_t = \alpha_1 X_{t-1} + \epsilon_t$ it is easy prove this property from equation 1. Since we are considering stationary time–series, we have $|\alpha_1| < 1$, and thus $\lim_{n\to\infty} c_n = \lim_{n\to\infty} -\alpha_1^{n-1} c_1/n = 0$, where c_n is the n^{th} cepstral coefficient. We can prove this property for an AR(p) model using the Z-transform approach described in [19, pg. 212–213]. Also, the smaller the value of the AR parameters, the faster the cepstral coefficients will decay.

We performed experiments to find the effect of retaining only first few coefficients on the distances between two time–series. We generated AR(1) time–series of length 256. By fitting a model to the generated time–series the AR coefficients were estimated. Table 1 summerizes the results.

Table 1. Distances between two time–series by retaining k coefficients (in 10^{-3})

Measure	k = 256	25	20	15	10
dist. between $AR(1)$ timeseries with $\hat{\alpha_1} = 0.3$ and 0.34					
d_{CEP}	0.97	0.97	0.97	0.97	0.97
d_{DFT}	22.55	3.26	2.75	2.41	1.95
$d_{\mathrm{DFT(ACF)}}$	4.34	1.24	1.01	0.81	0.43
d_{DWT}	22.55	4.20	3.67	2.79	1.15
d_{PCA}&d_{MSE}	22.55	–	–	–	–
dist. between $AR(1)$ timeseries with $\hat{\alpha_1} = 0.6$ and 0.64					
d_{CEP}	0.60	0.60	0.60	0.60	0.60
d_{DFT}	33.82	9.29	8.40	7.63	6.49
$d_{\mathrm{DFT(ACF)}}$	5.39	2.16	1.98	1.67	1.31
d_{DWT}	33.82	13.79	12.26	9.40	4.93
d_{PCA}&d_{MSE}	33.82	–	–	–	–

In this table, d_{CEP}, d_{DFT}, $d_{\mathrm{DFT(ACF)}}$, d_{DWT}, and d_{PCA} are the Euclidean distances between their Cepstral coefficients, their DFT coefficients, the DFT coefficients of their ACFs, their DWT coefficients, and their PCA coefficients respectively; d_{MSE} is the mean squared error between the time–series; k is the number of coefficients retained, and $\hat{\alpha_1}$ is the estimated auto-regression coefficient. In calculating d_{PCA}, we used only the first two principal components of the PCA since they are found to contain over 98% of the information in the time–series. cepstral coefficients are always computed using the estimated AR parameters.

From Table 1, we observe that when methods such as DFT or DWT are used, the distance between the two time–series reduces significantly as the number of coefficients retained is reduced. This could result in a lot of false positives when only a few coefficients are retained. Hence, a large overhead is required to remove the false positives. On the other hand, retaining as few as only 10 cepstral coefficients are sufficient for distinguishing two time–series results with virtually no loss of information. Hence, only a few cepstral coefficients should be sufficient. This property of cepstral coefficients results in effectively reducing the dimensionality and is useful in indexing time–series data efficiently. An added benefit is that the number of cepstral coefficients to be retained does not depend on the length of the time–series.

3.3 High discriminatory power.

Discriminatory power of a feature is its ability to separate time–series generated by different models. Cepstral coefficients provide more discriminatory power than the other feature extraction methods.

We performed experiments to study the change in the distance obtained when the auto-regression parameter changes. We generate AR(1) time–series TS_1 and TS_2 with α_1=0.3 and 0.31 respectively. We then keep changing TS_2 time–series by generating AR(1) time–series with increasing values of α_1 and report the percentage increase in the distance between TS_1 and TS_2 w.r.t the first pair of time–series in table 2. We use the first 10 coefficients of the feature extraction methods to compute the distance. For MSE, all coefficients were used. From Table 2, we observe that

Table 2. Percentage increase in distances w.r.t. first row. TS_1 has $\hat{\alpha}_1 = 0.3$

$\hat{\alpha}_1$ of TS_2	% d_{CEP}	% d_{DFT}	% $d_{\mathrm{DFT(ACF)}}$	% d_{DWT}	% d_{PCA}	% d_{MSE}
0.31	–	–	–	–	–	–
0.33	23.272	0.44	0.87	0.38	0.38	0.38
0.34	106.64	3.28	7.94	2.68	2.41	2.41
0.39	286.51	4.76	12.89	3.69	3.58	3.58
0.43	551.07	1.92	28.45	-0.13	0.25	0.25
0.47	999.77	5.18	55.05	0.96	1.52	1.52
0.50	1345.7	10.93	70.56	4.67	5.08	5.08

the cepstral coefficients have higher percentage increase in the distance for small increase in the AR parameter of TS_2 than any other method. Therefore, the cepstral coefficients can distinguish between the two time-series much more accurately than the other feature extraction methods.

3.4 Invariant under basic transformations.

Amplitude Translation. By normalizing a time-series to have mean zero, and then computing the cepstrum we can achieve invariance of the cepstral distance to amplitude-translation. This property is useful to identify series that have similar patterns but fluctuate around different means.

Amplitude Scaling. Multiplying a time-series by a constant does not affect its cepstral coefficients. Consider two stocks which have the same price fluctuations, however one stock sells at twice the price of the other. This property is useful for identifying such patterns.

Time-Shifting. Translating (shifting) a time-series in time does not affect its cepstrum. Consider a time series monitoring the growth pattern of different bacteria. A particular bacteria could be triggered out of dormancy at a later time compared to another bacteria, however its growth pattern might follow that of the bacteria that got stimulated earlier.

3.5 Seasonal time-series.

For seasonal time–series, the cepstral coefficients have high values at the period of seasonality. The seasonal peaks show a decreasing trend and rapidly drop to zero in a manner similar to that of the cepstral coefficients in the case of AR time–series. We can think of the cepstrum of a seasonal time–series to be a superposition of two independent cepstra: the seasonal part which is formed using the coefficients at the peaks, and the non–seasonal part formed from the remaining coefficients. When comparing two seasonal models we find the distance between their seasonal parts and their non-seasonal parts.

Consider a seasonal ARIMA $(1,0,0) \times (1,0,0)$ model $(1-\alpha_1 B)(1-\Phi_1 B^s)X_t = \epsilon_t$, where α_1 is the auto-regression parameter, Φ_1 is the seasonal auto-regression parameter, and s is the period of seasonality. We conducted experiments by varying each one of these parameters while keeping the other two constant. We made the following observations: (a) when s is varied, the distance between the cepstra of the two time–series does not change. (b) when α_1 is varied, the distance between the non-seasonal cepstra is the most significant component of the distance between the cepstra of the two time-series. (c) when Φ_1 is varied, the distance between the seasonal cepstra is the most significant component. Due to observation (a), we can group time–series which differ only in the seasonality together. Observations (b) and (c) enable us to identify time–series which differ only in one of the two parameters – auto-regression parameter or seasonal auto-regression parameter.

3.6 Relationships between time–series.

Consider a filter with impulse response h_t applied to a time–series X_t to give a new time–series Y_t: $Y_t = h_t \star X_t$. The relationship between the cepstra of X_t and Y_t can be shown to be $\mathrm{Cepstrum}(Y_t) = \mathrm{Cepstrum}(X_t) + \mathrm{Cepstrum}(h_t)$, (see also [9, eqs. 4.18–4.20]). Therefore, given two time–series Y_t and X_t which are related to each other, the difference in the cepstra of the two time–series is equal to the cepstrum of h_t. Suppose we want to find all the series X_t which are related to Y_t through the filter function h_t. Considering the cepstra of time–series as a multi-dimensional point, this problem reduces to that of finding those "points" (cepstra of time series) that are within a small distance from the "point" $\mathrm{Cepstrum}(Y_t) - \mathrm{Cepstrum}(h_t)$.

4 Clustering time–series.

We performed experiments to analyze the ability of cepstral coefficients to distinguish between ARIMA time series. We compared the clustering results obtained using cepstral coefficients with those obtained using other similarity measures such as DFT, DWT, PCA, DFT(ACF) and MSE. Experiments were conducted both on simulated as well as real datasets(collections). We used the Partitioning Around Medoids (PAM) clustering method [14] to cluster the time–series in each collection. To measure the accuracy and quality of clustering we use the similarity metric and silhouette width that are described in Section 2.

4.1 Clustering simulated datasets.

We perform clustering on a database of AR(1) time series and analyzed the results. We generate four groups (E, F, H, and I) each with 75 AR(1) time–series, with the α_1 parameter for the time–series in each group uniformly distributed in the ranges (0.3 ± 0.01), (0.34 ± 0.01), (0.6 ± 0.01), and (0.64 ± 0.01) respectively. The white noise ϵ_t used, had mean 0 and variance 0.01. We formed 10 collections from these time–series and ran clustering on each of the groups. Collections 1–5 were built by selecting 15 time–series each from groups E and F. Similarly, collections 6–10 were built from groups H and I.

Table 3. Clustering results of simulated datasets

Collection	Distance measure used					
	CEP	DFT	DFT(ACF)	DWT	PCA	MSE
Cluster results: Cluster Similarity Metric						
1	1	0.623	0.559	0.600	0.600	0.600
2	1	0.665	0.766	0.633	0.600	0.566
3	1	0.665	0.733	0.633	0.633	0.633
4	1	0.600	0.531	0.545	0.566	0.566
5	1	0.593	0.593	0.562	0.593	0.592
6	1	0.531	0.531	0.531	0.531	0.531
7	1	0.571	0.571	0.571	0.571	0.571
8	1	0.559	0.562	0.559	0.559	0.559
9	1	0.595	0.594	0.583	0.583	0.583
10	1	0.605	0.700	0.605	0.605	0.605
Cluster results: Average Silhouette Widths						
1	0.812	0.497	0.426	0.529	0.521	0.521
2	0.814	0.465	0.424	0.518	0.528	0.527
3	0.833	0.427	0.333	0.452	0.449	0.449
4	0.805	0.455	0.396	0.523	0.516	0.515
5	0.826	0.532	0.467	0.554	0.554	0.553
6	0.825	0.513	0.511	0.549	0.578	0.548
7	0.801	0.523	0.541	0.561	0.588	0.563
8	0.826	0.529	0.487	0.595	0.621	0.595
9	0.850	0.487	0.497	0.533	0.533	0.532
10	0.817	0.540	0.433	0.574	0.574	0.573

Table 3 shows the cluster similarity metric and silhouette width obtained when each of the collections was clustered using the different similarity measures. We get a perfect 1.0 for the cluster similarity metric when the cepstrum was used. Thus cepstral coefficients provide an accurate clustering for each of the collections. Also, the silhouette width obtained using the cepstrum is the highest and is always above 0.80. This indicates that the objects are clustered with a high confidence level.

The cluster plot [2] obtained using cepstral coefficients has a very low average dissimilarity within a cluster and a high dissimilarity between clusters. Hence the two clusters formed are well separated and it is a good clustering. The cluster plots obtained using the DFT and DWT coefficients have a high average dissimilarity within a cluster. Their dissimilarity within a cluster is higher than the dissimilarity between clusters and the two clusters overlap. Also, for some time–series, the individual silhouette width is close to zero and even negative. This indicates that the time–series were not clustered properly. The above results affirm that

[2] Omitted due to space limitation. See fig. 1 for an eg. cluster plot.

Euclidean distance between the cepstra is better than the other distance measures.

4.2 Clustering Real Data

We further performed experiments with four different real datasets: per capita personal income dataset; ECG recordings dataset; temperature dataset; and population dataset. The general methodology followed with each dataset consisted of identifying the different groups of time–series in the dataset, identifying the ARIMA model to be fitted, computing coefficients for each one of the methods, performing clustering using these coefficients and finally analyzing the clustering results obtained. The datasets were found to have non–stationarities in mean and/or variance, so there was a need for some preprocessing. Each step involved in the preprocessing and model identification for each of the datasets is explained in detail in [13]. Each of the datasets used had two groups of time–series. We performed the experiments on the normalized as well as the un-normalized time–series. Normalization is necessary to allow for differences in level and scale. In the experiments with DFT, DFT(ACF), PCA, DWT and CEP, we used the first 10 coefficients for clustering the time–series. For MSE, all the raw data values of the time–series were used. Due to space limitations we present only a few of the data and result plots. Interested readers are encouraged to see [13].

4.2.1 Personal Income Dataset

The personal income dataset [18] is a collection of time–series representing the per capita personal income from 1929-1999 in 25 states of the USA[3]. We define group 1 as the group of the east coast states, CA, and IL in which the personal income grows at a high rate. The mid-west states form a group in which the personal income grows at a low rate is called group 2.

The per capita income time–series are non-stationary in mean as well variance. To remove this non–stationarity, we do the following: (a) smoothen the original series by taking a window average over a window of size 2. This reduces the frequent variations and enables us to fit a lower–order model. (b) The non-stationarity in variance is dealt with by taking a logarithmic transform over the smoothened series. (c) after studying the ACF and PACF[4] of the resulting series, we decide the order of the ARIMA model to be fitted. We fitted ARIMA$(1,1,0)$ models to each of the series in the dataset after the preprocessing. We compute the cepstral coefficients using these ARIMA models (by converting the ARIMA model to an AR model through polynomial division). We performed the clustering experiments on the normalized as well as the un-normalized per capita personal income time–series. The results are summarized in Table 4. CEP gives the highest similarity metric and is also the only method in which the separation between clusters is higher than the average dissimilarity within a cluster. The silhouette width is the highest in case of CEP. Thus, CEP produces the most accurate clustering of the per capita personal income dataset for both the normalized and un–normalized time–series. The MSE, DFT, PCA, and DWT perform worse in the case of the un–normalized time–series than they do in the case of the normalized time–series.

Table 4. Clustering the personal income dataset.

Method	Normalized		Unnormalized	
	Sim(G,A)	Sil. Width	Sim(G,A)	Sil. Width
CEP	0.844	0.752	0.844	0.752
DFT	0.750	0.391	0.679	0.596
DFT(ACF)	0.762	0.522	0.762	0.522
DWT	0.740	0.475	0.602	0.508
PCA	0.788	0.335	0.679	0.572
MSE	0.788	0.325	0.679	0.555

The high/low income growth rate is accurately captured in the AR models fitted to the personal income time–series. The cepstral coefficients are effective in distinguishing between these models. The other methods do not seem to be capable of that. For example, upon examining the cluster plots when clustering the normalized time–series using the cepstral coefficients and the PCA metric in two classes, we observe the following. When the cepstral metric is used, the low personal income growth states of ID, IA, NE, SD, and ND have been put in one cluster, while the other cluster consists mainly of the east coast states, CA and IL (high personal income growth states). On the other hand, when the PCA metric is used, the resulting clusters contain both low and high personal income growth states.

4.2.2 ECG Dataset

The ECG dataset was obtained from the ECG database at PhysioNet [3]. We use 3 groups of those ECG time–series in our experiments: Group 1 included 22 time–series representing the 2 sec ECG recordings of people having malignant ventricular arrhythmia; Group 2 included 13 time–series that are 2 sec ECG recordings of healthy people representing the normal sinus rhythm of the heart; Group 3 included 35 time–series representing the 2 sec ECG recordings of people having supraventricular arrhythmia.

The time–series in this dataset exhibit considerable periodicity. We fit an ARIMA$(2,3,0)$ model to each time–series in the dataset, after smoothening it by taking a window average over a window size of 3. We performed clustering on ECG time–series collection 1 (comprising time–series from groups 1 and 2) and collection 2 (comprising time–series from groups 2 and 3) Results of clustering normalized time–series from collection 1 can be seen in Table 5 and Fig. 1 shows the cluster and silhouette plots using DFT(ACF) and CEP. From table 5 we observe that DFT(ACF) gives the most accurate clustering, but gives the lowest silhouette width among all the methods. This is because many of the time–series have been clustered into a particular group with very low confidence level with the sil-

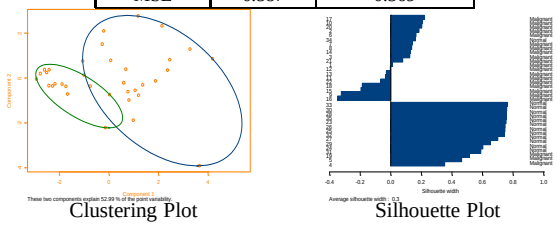
[3]The 25 states included were: CT, DC, DE, FL, MA, ME, MD, NC, NJ, NY, PA, RI, VA, VT, WV, CA, IL, ID, IA, IN, KS, ND, NE, OK, SD.

[4]Partial Auto–Correlation function. See [20] for definitions

houette width sometimes even negative (see Fig. 1(a)). This could be because the time–series lie between two clusters.

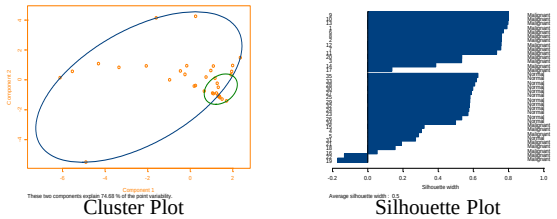
Table 5. Clustering normalized ECG collection 1.

Method	Sim(G,A)	Silhouette Width
CEP	0.771	0.502
DFT	0.629	0.539
DFT(ACF)	0.881	0.299
DWT	0.587	0.523
PCA	0.587	0.377
MSE	0.587	0.369



Clustering Plot Silhouette Plot

(a) Using DFT of auto-correlation coefficients

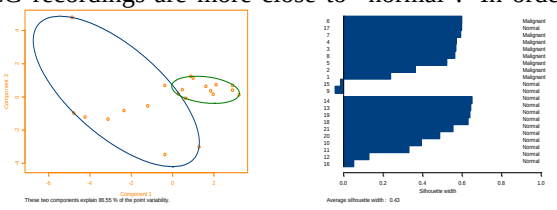


Cluster Plot Silhouette Plot

(b) Using cepstral coefficients

Figure 1. Clustering ECG collection 1 with DFT(ACF) and CEP

The cepstral coefficients give the second best clustering for collection 1. Fig. 1(b) shows the results of clustering the ECG time–series using the cepstral coefficients. We observe from the silhouette plot in Fig. 1(b) that some of the "malignant" time–series have been clustered in the same group as "normal" time–series. Upon further inspection we observed that these series indeed looked more similar to the normal time–series than to the malignant arrhythmia time–series. This could be because the corresponding subjects are in the initial stages of arrhythmia and hence their ECG recordings are more close to "normal". In order to



(a) Cluster Plot (b) Silhouette Plot

Figure 2. Clustering ECG dataset using CEP

verify that the wrongly clustered time–series were indeed separable, we formed a collection consisting of the wrongly clustered time–series along with the normal sinus rhythm time-series. and clustered it using CEP. The results of this clustering are shown in Fig. 2. Sim(G,A) value of 0.87 was obtained and we observe that the malignant arrhythmia time–series have been well separated from the normal time–series. This is very interesting because it shows that CEP can distinguish efficiently between series which could be difficult to separate. As can be seen in table 5, the remaining methods perform poorly.

Table 6 shows the results of clustering ECG collection 2 (both normalized and un-normalized). We observe that CEP gives the most accurate clustering for this collection.

Table 6. Clustering the ECG collection 2.

Method	Normalized		Unnormalized	
	Sim(G,A)	Sil. Width	Sim(G,A)	Sil. Width
CEP	0.779	0.519	0.779	0.519
DFT	0.579	0.649	0.639	0.972
DFT(ACF)	0.593	0.425	0.593	0.425
DWT	0.561	0.635	0.639	0.972
PCA	0.601	0.521	0.639	0.975
MSE	0.561	0.440	0.639	0.964

4.2.3 Temperature Dataset

This dataset, obtained from the National Climatic Data Center [6], is a collection of 30 time–series of the daily temperature in the year 2000 in various places in Florida, Tennessee and Cuba. It had temperature recordings from 10 places in Tennessee, 5 places in northern Florida, 9 places in southern Florida and 6 places in Cuba. Tennessee and northern Florida form group 1, because they are geographically close and their temperatures are known to be similar. Similarly, Cuba and southern Florida form group 2.

To fit a model, we first smoothen each of the time–series by taking a window average over a window size of 4, and then fit an ARIMA$(2,1,0)$ to the resulting time–series. We performed clustering experiments on both normalized and unnormalized time–series. Table 7 summarizes the results. MSE gives Sim(G,A) value of 0.933 with 366 data values. However, CEP gives the same accuracy with 10 coefficients. Thus, CEP gives the most accurate clustering on the normalized temperature time–series using very few coefficients. From Table 7, we observe that DFT, DWT, PCA and MSE

Table 7. Clustering the temperature dataset.

Method	Normalized		Unnormalized	
	Sim(G,A)	Sil. Width	Sim(G,A)	Sil. Width
CEP	0.933	0.531	0.933	0.531
DFT	0.828	0.533	1.000	0.700
DFT(ACF)	0.670	0.603	0.670	0.603
DWT	0.820	0.473	1.000	0.695
PCA	0.899	0.398	1.000	0.673
MSE	0.933	0.368	1.000	0.648

give accurate clustering for the unnormalized temperature data. This is not surprising because the temperature ranges of the two groups are distinctively different. Hence, MSE can clearly distinguish on the basis of these raw data values. This is also the reason for the success of DFT, DWT, and PCA in the un–normalized case. However, when we normalize the temperature time–series, the data values do not fall into a clearly defined separate range and therefore, DFT, DWT and PCA fail to give accurate clustering.

4.2.4 Population Dataset

The population dataset was a collection of time–series representing the population estimates from 1900-1999 in 20 states of the US [5]. Some of these time–series had an exponentially increasing trend while others had a stabilizing trend. The 20 states were partitioned into two groups based

on their trends: group 1 consisted of CA, CO, FL, GA, MD, NC, SC, TN, TX, VA, and WA had the exponentially increasing trend while group 2 consisted of IL, MA, MI, NJ, NY, OK, PA, ND, and SD had a stabilizing trend. We followed the same steps of preprocessing as we did for the Personal Income Dataset and fitted an ARIMA$(1, 1, 0)$ model to each time–series in the dataset.

Table 8 summarizes the results of clustering the normalized and unnormalized time–series in this dataset. We observe that the MSE, DFT and PCA give the most accurate clustering for the normalized population dataset. MSE is based on the raw data values. The growth rate of the population time–series can be clearly distinguished from the normalized raw data values. This is because the group 1 time–series increase rapidly with time, hence at any instant the population estimate is only a small percentage of the total population in the year 1999. On the other hand, the group 2 time–series increase slowly with time, hence at any instant the population estimate is a large percentage of the total population in the year 1999. Thus, when we look at the normalized population time–series values, we observe that it is easy to separate the two groups based on the raw data values. This is the reason for the success of MSE, DFT and PCA on the normalized time–series.

We also observe that CEP gives the best results for the unnormalized population time–series. Comparing the results for the normalized and un–normalized datasets, we observe that normalization proves very beneficial for MSE, DFT, PCA and DWT on the population time–series. When the series are not normalized, the growth rates of the population time–series are difficult to separate by simply looking at the raw data values. Hence, MSE, DFT, PCA do not perform very well on the un-normalized time-series.

Table 8. Clustering the population dataset.

Method	Normalized		Unnormalized	
	Sim(G,A)	Sil. Width	Sim(G,A)	Sil. Width
CEP	0.744	0.687	0.744	0.687
DFT	1.000	0.651	0.596	0.652
DFT(ACF)	0.643	0.793	0.643	0.793
DWT	0.792	0.240	0.643	0.658
PCA	1.000	0.622	0.627	0.579
MSE	1.000	0.604	0.627	0.576

5 Conclusion

We consider the problem of defining an appropriate similarity measure for time–series that is crucial for data-mining applications in time–series databases (eg. similarity search and clustering).

We form a succinct representation of time–series using ARIMA models and then define a highly effective similarity measure for this representation. Representing a time–series as an ARIMA model captures the various important components of the time–series. We call two time–series similar when the same model fits the time–series. We propose a new distance measure using the LPC cepstral coefficients of the time–series for finding similarity between time–series models. We have demonstrated that the use of LPC cepstral coefficients for feature extraction helps in obtaining accurate clustering and also results in high dimensionality reduction both in the case of simulated as well as real data. Computing the feature vector from the model implies several desired features. The length of the time–series or differences in their lengths are no more a concern.

We compare the clustering results obtained using LPC cepstral coefficients with those obtained using other widely used methods such as DFT, DWT, PCA, and DFT of auto-correlation of the time–series. LPC cepstral coefficients clearly perform much better than the other methods for clustering synthetic ARIMA time–series. Our method is competitive with the other approaches for clustering various real datasets and in many cases significantly better than the other methods.

Our approach is limited to time–series that can be modeled by ARIMA models. Future work includes: (a) extending this approach to non–ARIMA time–series, such as chaotic and other non–linear time–series, and (b) extending this approach to multi–variate ARIMA time–series.

References

[1] R. Agrawal, C. Faloutsos, and A. Swami. Efficient similarity search in sequence databases. *Proc. of FODO*, pg. 69–84, 1993.

[2] R. Agrawal, K. Lin, H. Sawhney, and K. Shim. Fast similarity search in the presence of noise, scaling, and translation in time–series databases. In *21st VLDB*, pg. 490–501,1995.

[3] P. Archive. http://www.physionet.org/physiobank/database.

[4] R. Bennett. *Spatial Time Series*, Pion Limited, 1979.

[5] http://www.census.gov/population/www/estimates/st_stts.html.

[6] http://www.ncdc.noaa.gov/rcsg/datasets.html.

[7] G. Das, D. Gunopulos, and H. Mannila. Finding similar time series. In *Proc. of European Conf. on Principles of Data Mining and Knowledge Discovery*, pg. 88–100, 1997.

[8] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time–series databases. In *Proc. of SIGMOD*, pg. 419–429, 1994.

[9] S. Furui. *Digital Speech Processing, Synthesis, and Recognition*. Marcel Deckker, Inc., New York, 1989.

[10] M. Gavrilov, D. Anguelov, P. Indyk, and R. Motwani. Mining the stock market: Which measure is best? In *Proc. of the KDD*, pg. 487–496, 2000.

[11] A. Gordon. *Classification*. Chapman and Hall, CRC, 1999.

[12] H. V. Jagadish, A. O. Mendelzon, and T. Milo. Similarity–based queries. In *Proc. of the 14th SIGACT-SIGMOD-SIGART Symp. of Database Systems*, pg. 36–45, 1995.

[13] K. Kalpakis,D. Gada,V. Puttagunta. Distance measures for effective clustering of ARIMA time–series. Technical Report TR–CS–01–14, CSEE, UMBC, 2001.

[14] Mathsoft, Inc. *SPlus–2000 guide to Statistics*.

[15] D. Rafiei. On similarity–based queries for time series data. In *Proc. of the 15th ICDE*, pg. 410–417, 1999.

[16] D. Rafiei and A. Mendelzon. Similarity–based queries for time series data. In *Proc. of SIGMOD*, pg. 13–24, 1997.

[17] Z. Struzik and A. Sibes. Measuring time series similarity through large singular features revealed with wavelet transformation. In *Proc. of the 10th Intl. Workshop on Database and Expert Systems Appl.*, pg. 162–166, 1999.

[18] http://www.bea.gov/bea/regional/spi.

[19] R. Vich. *Z Transform Theory and Applications*. D. Reidel, Holland, 1987.

[20] W. Wei. *Time Series Analysis*. Addison–Wesley, 1994.

[21] B. Yi, H. Jagadish, and C. Faloutsos. Efficient retrieval of similar time sequences under time warping. In *Proc. of 14th ICDE*, pg. 201–208, 1998.

[22] X. Zhongjie. *Case Studies in Time Series Analysis*. World Scientific, Singapore, 1993.