Chapter 6
Analytical Support

> It is useful to think of the human and the computer together as a single cognitive entity, with the computer functioning as a kind of *cognitive coprocessor* to the human brain. [...] Each part of the system is doing what it does best. The computer can pre-process vast amounts of information. The human can do rapid pattern analysis and flexible decision making.
>
> Ware (2008, p. 175)

Visualization and interaction as described in the previous chapters help users to visually analyze time-oriented data. Analysts can look at the data, explore them, and in this way understand them. This is possible thanks to human visual perception and the fact that humans are quite good at recognizing patterns, finding interesting and unexpected solutions, combining knowledge from different sources, and being creative in general[1]. This holds true unless the problem to be solved exceeds a certain size. Very large time-series or data that consist of many thousands of time-dependent variables can usually not be grasped by human observers. In such cases, we need the proficiency of computing systems to assist the knowledge crystallization from time-oriented data. Apparently, if the problem size is sufficiently large, computers are better (i.e., faster and more accurate) than humans at numeric and symbolic calculations, logical reasoning, and searching.

In general, *data mining* and *knowledge discovery* are commonly defined as the application of algorithms to extract useful structures from large volumes of data, where knowledge discovery explicitly demands that knowledge be the end product of the analytical calculations (see Fayyad et al., 1996, 2001; Han and Kamber, 2005). A variety of concepts and methods are involved in achieving this goal, including databases, statistics, artificial intelligence, neural networks, machine learning, information retrieval, pattern recognition, data visualization, and high-performance computing.

This chapter will illustrate how automatic analytical calculations can be utilized to facilitate the exploration and analysis of larger and more complex time-oriented

[1] Wegner (1997) makes some interesting statements about why interaction is better than algorithms.

W. Aigner et al., *Visualization of Time-Oriented Data*,
Human-Computer Interaction Series, DOI 10.1007/978-0-85729-079-3_6,

data. To this end, we will give a brief overview of typical temporal analysis tasks. For selected tasks, we will present examples that demonstrate how visualization can benefit from considering analytical support. Our descriptions will intentionally be kept at a basic level. For details on the sometimes quite complex matter of temporal data analysis, we refer interested readers to the relevant literature.

6.1 Temporal Analysis Tasks

Temporal analysis and temporal data mining are especially concerned with extracting useful information from time-oriented data. More specifically, analytical methods for time-oriented data address the following categories of tasks (see Antunes and Oliveira, 2001; Laxman and Sastry, 2006; Hsu et al., 2008; Brockwell and Davis, 2009; Mitsa, 2010):

Classification Given a predefined set of classes, the goal of classification is to determine which class a dataset, sequence, or subsequence belongs to. Applications such as speech recognition and gesture recognition apply classification to identify specific words spoken or interactions performed. The analysis of sensor data or spatio-temporal movement data often requires classification to make the enormous volumes of data to be handled manageable.

Clustering Clustering is concerned with grouping data into clusters based on similarity, where the similarity measure used is a key aspect of the clustering process. In the context of time-oriented data, it makes sense to cluster similar time-series or subsequences of them. For example, in the analysis of financial data, one may be interested in stocks that exhibit similar behavior over time. In contrast to classification, where the classes are known a priori, clusters are not defined upfront.

Search & retrieval This task encompasses searching for a priori specified queries in possibly large volumes of data. This is often referred to as *query-by-example*. Search & retrieval can be applied to locate exact matches for an example query or approximate matches. In the latter case, similarity measures are needed that define the degree of exactness or fuzziness of the search (e.g., to find customers whose spending patterns over time are similar but not necessarily equal to a given spending profile).

Pattern discovery While search & retrieval requires a predefined query, pattern discovery is concerned with *automatically* discovering interesting patterns in the data (without any a priori assumptions). The term *pattern* usually covers a variety of meanings, including sequential pattern, periodic pattern, but also temporal association rules. In a sense, a pattern can be understood as a local structure in the data or combinations thereof. Often, frequently occurring patterns are of interest, for example when analyzing whether a TV commercial actually leads to an increase in sales. But patterns that occur very rarely can also be interesting because they might indicate malicious behavior or failures.

Prediction An important task in analyzing time-oriented data is the prediction of likely future behavior. The goal is to infer from data collected in the past and present how the data will evolve in the future. To achieve this goal one first has to build a predictive model for the data. Examples of such models are autoregressive models, non-stationary and stationary models, or rule-based models.

In the context of visualization, these tasks share a common goal: *data abstraction* in order to reduce the workload when computing visual representations and to keep the perceptual efforts required to interpret them low. For classification and clustering, we abstract from the raw data and work with classes and clusters. For search & retrieval and pattern discovery we are foremost interested in relevant patterns and de-emphasize irrelevant data. For prediction, we focus on the future.

A variety of methods have to play in concert in order to accomplish temporal analysis tasks. Statistical aggregation operators (e.g., sum, average, minimum, maximum, etc.), methods from time-series analysis, as well as dedicated temporal data mining techniques are needed.

In what follows, we demonstrate the applicability of analytical methods for the analysis of time-oriented data using the three examples: clustering, temporal data abstraction, and principal component analysis. Clustering decreases the number of data items to be represented, and allows the discernment of similarities and unexpected behavior. Temporal data abstraction reduces data complexity by deriving qualitative statements, which are much easier to understand. Principal component analysis decreases the number of time-dependent variables by switching the focus to major trends in the data.

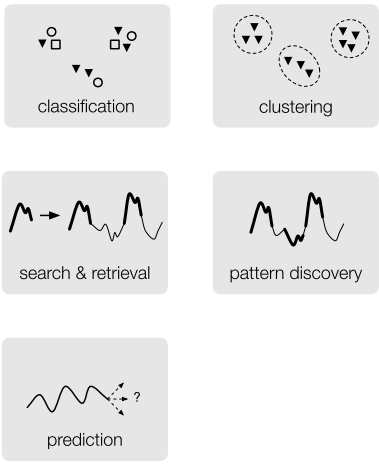

Fig. 6.1 Overview of temporal analysis tasks.

6.2 Clustering

In general, grouping data into clusters and concentrating on the clusters rather than on individual data values allows the analysis of much larger datasets. Appropriate *distance* or *similarity measures* lay the ground work for clustering. Distance and similarity measures are profoundly application dependent and range from average geometric distance, to measures based on longest common subsequences, to measures based on probabilistic models. Based on computed distances, clustering methods create groups of data, where the number of available techniques is large, including hierarchical clustering, partitional clustering, and sequential clustering. Due to the diversity of methods, selecting appropriate algorithms is typically difficult. Careful adjustment of parameters and regular validation of the results are therefore essential tasks in the process of clustering. More details on clustering methods and distance measures can be found in the work by Jain et al. (1999), Gan et al. (2007), and Xu and Wunsch II (2009).

A prominent example of how analytical methods can assist the visualization of time-oriented data is the work by Van Wijk and Van Selow (1999). The goal is to identify common and uncommon subsequences in large time-series data and to understand their distribution over time. The problem is that simply drawing line plots for all subsequences is not a satisfactory solution due to the overwhelmingly large number of time points and line plots. In order to tackle this problem, clustering methods and a calendar-based visualization are used.

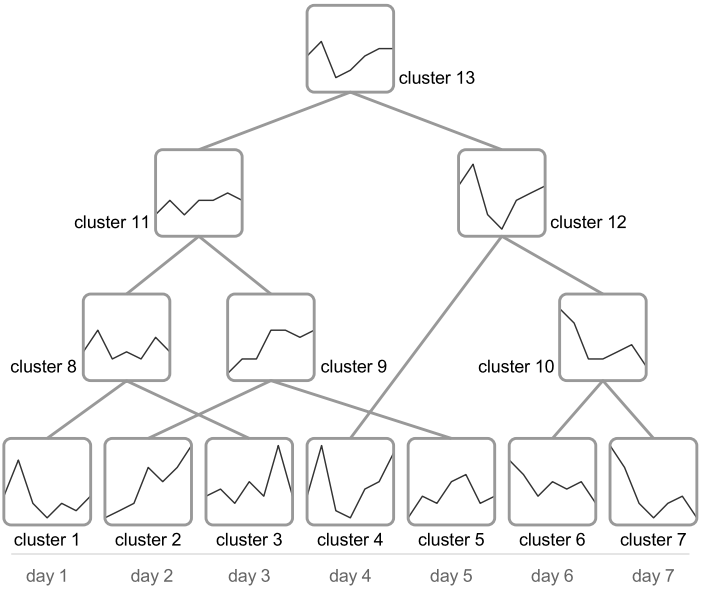

Fig. 6.2: By repeatedly merging the two most similar sequences into new clusters, a clustering hierarchy is generated. The root cluster is an aggregated representative of the entire dataset.

In particular, the approach works as follows. As Van Wijk and Van Selow (1999) are interested in patterns on the granularity of days, the first step is to split a large time-series into k day patterns, each of which stores the subsequence for one day. The clustering process starts with the k day patterns as initial clusters. Then the differences of all possible combinations of two clusters are computed and the two most similar clusters are merged into a new cluster (i.e., an aggregated representative of the two clusters). This process runs repeatedly and results in a *clustering hierarchy* with $2k - 1$ clusters, where the root of the hierarchy represents the entire dataset in an aggregated fashion. Figure 6.2 illustrates the clustering process with data for seven days.

The visualization of the clustered day patterns uses two different views for the two analysis tasks: (1) assess similarity among day patterns and (2) locate common and uncommon patterns over time. The first task is facilitated by a basic line plot (↪ p. 153) that shows a selected number of clusters, where each plot uses a unique color. To accomplish the second task, a calendar display is used where individual days are color-coded according to cluster affiliation. This way, analysts can see the day pattern and at the same time understand when during a year this pattern occurs. Various interaction methods allow adjustments of the visual representation and data exploration. In terms of assessing similarities, the user can select a day from the calendar and with the help of the clustering hierarchy, similar days (and clusters) can be retrieved automatically.

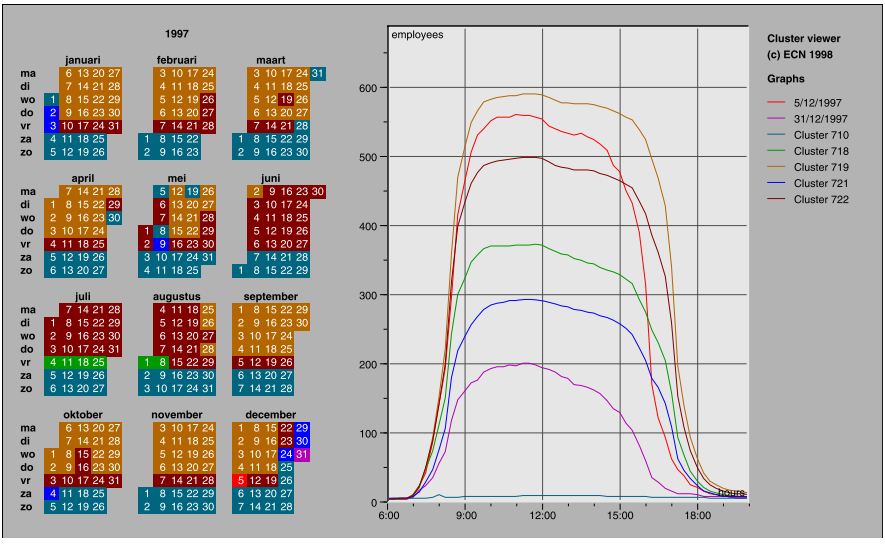

Fig. 6.3: Visual analysis of the number of employees at work. Day patterns for selected days and clusters are visualized as line plots (right). Individual days in a calendar display (left) are colored according to cluster affiliation.
Source: Van Wijk and Van Selow (1999), © 1999 IEEE. Used with permission.

Figure 6.3 shows an example of the visualization design. The data displayed in the figure contain the number of employees at work. The line plot currently shows the day patterns of two days (5/12/1997 and 31/12/1997) and five clusters (710, 718, 719, 721, and 722). Van Wijk and Van Selow (1999) demonstrate that several conclusions can be drawn from the visual representation. To give only a few examples:

- Employees follow office hours quite strictly and work between 8:30 am and 5:00 pm in most cases.
- Fewer people work on Fridays during summer (cluster 718).
- During weekends and holidays only very few people are at work (cluster 710).
- It is common practice to take a day off after a holiday (cluster 721).

These and similar statements were more difficult or even impossible to derive without the integration of clustering. Van Wijk and Van Selow (1999) most convincingly demonstrate the advantages of analytical support for the visual analysis of time-oriented data. While here the benefit lies in the abstraction from raw data to aggregated clusters, we will see in the next section that other kinds of abstraction are useful as well.

6.3 Temporal Data Abstraction

In practice, time-oriented datasets are often large and complex and originate from heterogeneous sources. The challenging question is how huge volumes of possibly continuously measured data can be analyzed to support decision making. On the one hand, the data are too large to be interpreted all at once. On the other hand, the data are more erroneous than usually expected and some data are missing as well. What is needed is a way to abstract the data in order to make them eligible for subsequent visualization.

The term *data abstraction* was originally introduced by Clancey (1985) in his classic proposal on heuristic classification. In general, the objective of data abstraction is:

> ... to create an abstraction that conveys key ideas while suppressing irrelevant details.
> Thomas and Cook (2005, p. 86)

The basic idea is to use qualitative values, classes, or concepts, rather than raw data, for further analysis or visualization processes (see Lin et al., 2007; Combi et al., 2010). This helps in coping with data size and data complexity. To arrive at suitable data abstractions, several tasks must be conducted, including selecting relevant information, filtering out unneeded information, performing calculations, sorting, and clustering.

Principles

Let us now illustrate the concept of *temporal data abstraction* in medical contexts with a simple example. Figure 6.4 shows time-oriented data as generated when monitoring newborn infants that have to be ventilated artificially. The figure visualizes three variables plotted as points against a horizontal time axis: S_aO_2 (arterial oxygen saturation), $P_{tc}CO_2$ (transcutaneous partial pressure of carbon dioxide), and P_aCO_2 (arterial partial pressure of carbon dioxide). S_aO_2 and $P_{tc}CO_2$ are measured continuously at a regular rate, but with different frequency. New values for P_aCO_2 arrive irregularly and some values for $P_{tc}CO_2$ are missing.

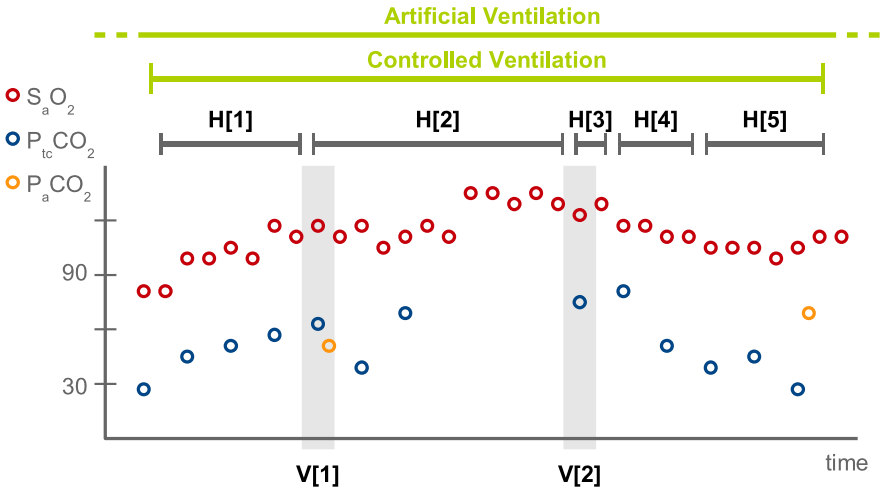

Fig. 6.4: Temporal data abstraction in the context of artificial ventilation. Vertical temporal abstractions are illustrated as V[1] and V[2] and horizontal temporal abstraction are illustrated as H[1] – H[5]. The context is given as "artificial ventilation" and its sub-context "controlled ventilation".

The aim of temporal data abstraction is to arrive at qualitative values or patterns over time intervals. *Vertical* temporal abstraction (illustrated in V[1] and V[2]) considers multiple variables over a particular time point and combines them into a qualitative value or pattern. *Horizontal* temporal abstraction (illustrated as H[1] – H[5]) infers a qualitative value or pattern from one or more variables and a corresponding time interval. Usually the abstraction process is context-dependent. In Figure 6.4, the abstraction is done in the context of artificial ventilation and in the sub-context of controlled ventilation.

In medical applications, there are different types of abstraction methods, ranging from rather simple to quite complicated ones. However, as pointed out by Combi et al. (2010), no exhaustive schema exists to categorize the available methods. Nevertheless, the common understanding is that even in very simple cases the process is knowledge-driven. The use of knowledge is the main characteristic that

distinguishes data abstraction from statistical data analysis (e.g., trend detection using time-series analysis).

Simple methods involve single data values and usually do not need to consider time specifically. They generate vertical abstractions. The knowledge used are concept associations or concept taxonomies. Combi et al. (2010) distinguish three types of simple methods:

- *Qualitative abstraction* means converting numeric expressions to qualitative expressions. For example, the numeric value of 34.8°C of body temperature can be abstracted to the qualitative value "hypothermia".
- *Generalization abstraction* involves a mapping of instances into classes. For example, "hand-bagging is administered" is abstracted to "manual intervention is administered", where "hand-bagging" is an instance of the concept class "manual intervention".
- *Definitional abstraction* is a mapping across different concept categories. The movement here is not within the same concept taxonomy, as for the generalization abstraction, but across two different concept taxonomies.

More complex methods consider one or more variables jointly and specifically integrate the dimension of time in a kind of temporal reasoning. These methods generate horizontal temporal abstractions. According to Combi et al. (2010), four types of complex methods exist:

- *Merge (or state) abstraction* is the process of deriving maximal time intervals for which some constraints of interest hold. For example, several consecutive days with high fever and increased blood values can be mapped to "bed-ridden".
- *Persistence abstraction* means applying persistence rules to project maximal intervals for some property, both backwards and forwards in time. For example, "headache in the morning", can be abstracted to "headache in the evening before" or "headache in the afternoon afterwards".
- *Trend (or gradient or rate) abstraction* is concerned with deriving significant changes and rates of change in the progression of some variable. For example, $P_{tc}CO_2$ has decreased from 130 to 90 in the last 20 minutes would result in "$P_{tc}CO_2$ is decreasing too fast".
- *Periodic abstraction* aims to derive repetitive occurrence, with some regularity in the pattern of repetition. For example, "headache every morning, but not during the day", would result in "repetitive headache in the morning".

Application examples

The principles described in the previous paragraphs can be applied in various ways. In the following, we will give a few examples of systems that utilize temporal data abstraction. For more examples, we refer to the survey of temporal data abstraction in clinical data analysis by Stacey and McGregor (2007).

Monitoring artificially ventilated infants VIE-VENT is an open-loop knowledge-based monitoring and therapy planning system for artificially ventilated infants (see Miksch et al., 1996). In order to derive qualitative descriptions for different kinds of temporal trends (i.e., very-short, short, medium, and long-term trends) from continuously arriving quantitative data, the system utilizes context-sensitive and expectation-guided methods and incorporates background knowledge about data points, data intervals, and expected qualitative trend patterns. Smoothing and adjustment mechanisms help to keep qualitative descriptions stable in case of shifting contexts or data oscillating near thresholds. Context-aware schemata for data point transformation and curve fitting are used to express the dynamics of and the reaction to different data abnormalities. For example, during intermittent positive pressure ventilation (ippv), the transformation of the quantitative value $P_{tc}CO_2 = 56mmHg$ results in the qualitative abstraction "$P_{tc}CO_2$ substantially above target range". During intermittent mandatory ventilation (imv) however, $56mmHg$ represents the "target value". Qualitative abstractions and schemata of curve fitting are subsequently used to decide if the value progression happens too fast, at normal rate, or too slow.

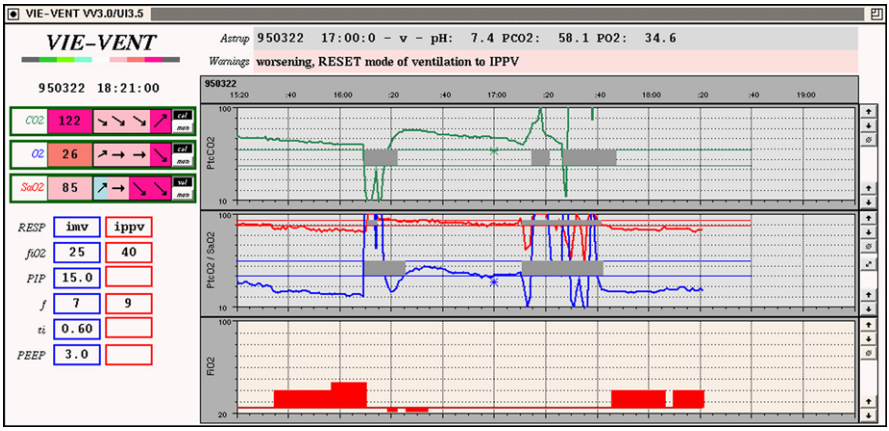

Fig. 6.5: VIE-VENT displays measured quantitative values as line plots. Qualitative abstractions and trends are represented by different colors and arrows in the top three boxes on the left.
Source: Miksch et al. (1996), © 1996 Elsevier. Used with permission.

Figure 6.5 shows the user interface of VIE-VENT. In the top-left corner, the system displays exact values of the quantitative blood gas measurements CO2, O2, SaO2. Arrows depict trends and qualitative abstractions are indicated by different colors (e.g., deep pink represents "extremely above target range"). The left panel further shows current and recommended ventilator settings in blue and red boxes, respectively. The right-hand side shows line plots of the most important variables for the last four hours.

Dealing with oscillating data Strongly oscillating data pose a formidable challenge for methods that aim to extract qualitative abstractions and patterns from the data. The problem is that derived abstractions could change too quickly as to be interpretable by the observer. Therefore, Miksch et al. (1999) developed the Spread, a time-oriented data abstraction method that is capable of deriving steady qualitative abstractions from oscillating high-frequency data. The tool performs the following steps of processing and data abstraction:

1. *Eliminate data errors*: Sometimes up to 40% of the input data are obviously erroneous, i.e., exceed the limits of plausible values.
2. *Clarify the curve*: Transform the still noisy data into the *spread*, which is a steady curve with some additional information about the distribution of the data along that curve.
3. *Qualify the curve.* Abstract from quantitative values to qualitative values like "normal" or "high" and concatenate intervals with equal qualitative values.

Figure 6.6 illustrates how the analytical abstractions can enhance the visualization. The Spread smooths out the strongly oscillating raw data. Even the increased oscillation in the center of the display is dealt with gracefully: it leads to increased width of the spread, but not to a change of the qualitative value. With these abilities, the Spread can support physicians in making better qualitative assessments of otherwise difficult-to-interpret data.

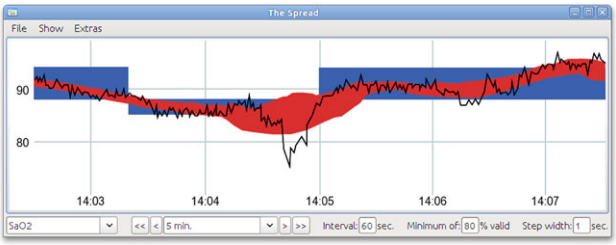

Fig. 6.6: The thin line shows the raw data. The red area depicts the *spread* and the blue rectangles represent the derived temporal intervals of steady qualitative values. The lower part of the figure shows the parameter settings.
Source: Adapted from Miksch et al. (1999), © 1999 Springer. Used with permission.

Linking temporal and visual abstraction In interactive environments, the visualization of time-oriented data and abstractions thereof can change dynamically due to user interaction, where resizing and zooming are among the most commonly applied operations. In such scenarios, the visualization must be able to capture as much temporal information as possible without losing overview and details, even if the available display space is very limited. Bade et al. (2004) demonstrate that this is possible by means of *semantic zooming* (see p. 112 and ↪ p. 230). The semantic zoom functionality relies on an appropriate set of temporal data abstractions

and associated visual representations for different levels of detail as illustrated in Figure 6.7. Depending on the available display space (or the current zoom level), a suitable temporal abstraction is selected automatically and its corresponding visual abstraction is displayed. The advantage of this procedure is that it relieves the user of managing the levels of abstraction by hand. Moreover, the semantic zoom corresponds much better with the interactive nature of flexible and dynamic visual analysis scenarios.

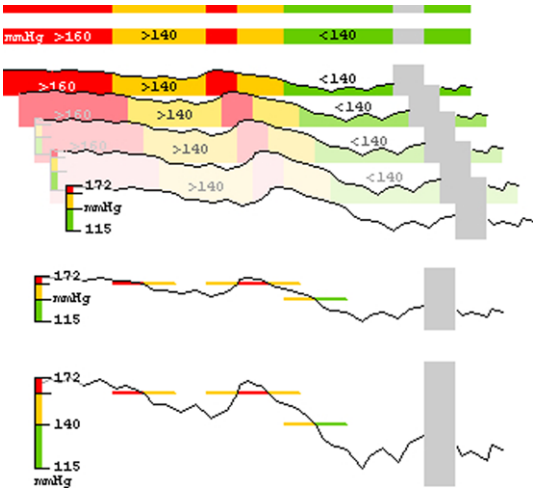

Fig. 6.7 Different steps of semantic zooming of a time-series visualization from a broad overview with qualitative values (top) to a detailed view with fine structures and quantitative details (bottom). Gray areas indicate missing data.

The examples described can only indicate the possible benefits that basic and complex temporal abstraction methods and their integration with the visualization can have for dealing with time-oriented data in medical applications. We know of quite positive feedback from medical experts who found it easy to capture the health conditions of their patients. Moreover, these qualitative abstractions can be used for further reasoning or in guideline-based care for a simplified representation of treatment plans.

What our previous examples have in common, however, is that they consider only a relatively small number of time-dependent variables. As we will see in the next section, if the number of variables gets larger, we need further analytical methods.

6.4 Principal Component Analysis

Time-oriented data are often of multivariate nature, but too large a number of variables poses considerable difficulties for the visualization. These difficulties can be overcome by applying principal component analysis (PCA), which offers an excellent basis for data abstraction (see Jolliffe, 2002; Jackson, 2003; Jeong et al., 2009).

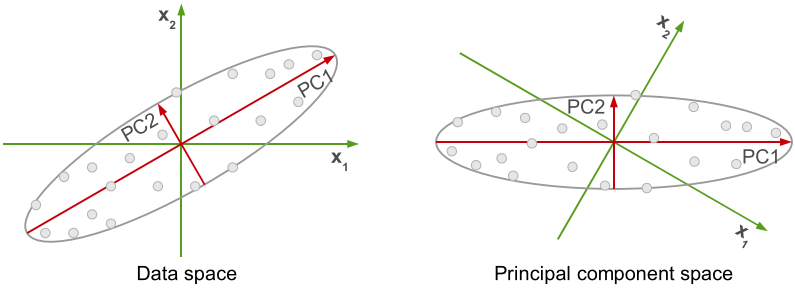

Fig. 6.8: Principal component analysis transforms multivariate data (with variables $\mathbf{x_1}$ and $\mathbf{x_2}$ in this case) into a new space, the so-called principal component space, which is spanned by the principal components (here PC1 and PC2).

The key principle of PCA is a transformation of the original data space into the principal component space (see Figure 6.8). In the principal component space, the first coordinate, that is, the first principal component represents most of the original dataset's variance, the second principal component, which is orthogonal to the first one, represents most of the remaining variance, and so on. Visualizing the data in the new principal component space shows us how closely individual data records are related to the major trends, and thus PCA helps us to reveal the internal structure of the data. Moreover, since principal components are ordered by their significance, we can focus on fewer principal components than we have variables in our data.

In the following we will take a brief look at the basics of principal component analysis and illustrate by means of examples the benefit that this analytical concept has for the visual analysis of time-oriented data.

Basic method

Assume that we have modeled our multivariate dataset as a matrix:

$$\mathbf{X} = \left(\mathbf{x_1}\mathbf{x_2}\cdots\mathbf{x_m}\right) = \begin{pmatrix} x_{1,1} & \cdots & x_{1,m} \\ x_{2,1} & \cdots & x_{2,m} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ x_{n,1} & \cdots & x_{n,m} \end{pmatrix}$$

where the columns of $\mathbf{X}$ correspond to the m variables $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_m}$ of the dataset, and the rows represent n records of data (e.g., n repetitions of an experiment). For a time-oriented dataset, one of the $\mathbf{x_i}$ is usually the dimension of time.

Depending on the application it can make sense to prepare the data such that they are mean-centered and normalized (by subtracting off the mean of each variable and scaling each variable according to its variance). Now our goal is to trans-

form the data into the principal component space that is spanned by $r \leq m$ principal components.

For the purpose of explanation, we resort to *singular value decomposition (SVD)* according to which any matrix $\mathbf{X}$ can be decomposed as:

$$\mathbf{X} = \mathbf{W} \cdot \Sigma \cdot \mathbf{C}^T$$

where $\mathbf{W}$ is an $n \times r$ matrix, Σ is an $r \times r$ diagonal matrix, and $\mathbf{C}^T$ is an $r \times m$ matrix:

$$\mathbf{X} = \begin{pmatrix} w_{1,1} & \cdots & w_{1,r} \\ w_{2,1} & \cdots & w_{2,r} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ w_{n,1} & \cdots & w_{n,r} \end{pmatrix} \cdot \begin{pmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_r \end{pmatrix} \cdot \begin{pmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,m} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,m} \\ \vdots & \vdots & & \vdots \\ c_{r,1} & c_{2,2} & \cdots & c_{r,m} \end{pmatrix}$$

The matrix $\mathbf{C}^T$ has in its rows the transposed eigenvectors $\mathbf{c_1}^T, \ldots, \mathbf{c_r}^T$ of the matrix $\mathbf{X}^T\mathbf{X}$, which corresponds to the *covariance matrix* of the original dataset. The $\mathbf{c_i}$ form the orthonormal basis of the principal component space; they are the principal components. Each $\mathbf{c_i}$ is the result of a linear combination of the original variables where the factors (or *loadings*) of the linear combination determine how much the original variables contribute to a principal component. The first principal component $\mathbf{c_1}$ is chosen so as to be the one that captures most of the original data's variance, the second principal component most of the remaining variance, and so forth. The significance values $\sigma_1, \ldots, \sigma_r$ in Σ are determined by the likewise ranked square roots of the eigenvalues $\sqrt{\lambda}_1, \ldots, \sqrt{\lambda}_r$ of the eigenvectors (i.e., the principal components) $\mathbf{c_1}, \ldots, \mathbf{c_r}$. Finally, the i-th row of the matrix $\mathbf{W}$ contains the coordinates of the i-th data record in the new principal component space. The individual coordinates are often referred to as the *scores*.

This brief formal explanation provides a number of key take-aways. Let us summarize the ones that are most relevant for visualization:

- the significance values determine the ranking of principal components,
- the ranking is the basis for data abstraction, where principal components that bear little information can be omitted,
- the loadings describe the relationship of the original data variables and the principal components, and
- the scores describe the location of the original data records in the principal component space.

Application examples

We will now demonstrate how PCA can be applied to enhance the visual analysis of time-oriented data. Our general goal is to uncover structure in the data and to reduce the analysis complexity by focusing on significant trends. In a first example,

we will see that even a single principal component can bear sufficient information for discerning main trends in the data. Secondly, an example will illustrate how one can determine the principal components to be retained for the visualization as well as the ones that can be omitted due to their low significance.

Before we start with the examples, however, it is important to mention that PCA does not distinguish between independent and dependent variables. In particular, the dimension of time is processed indiscriminately, which sacrifices the temporal dependencies in the data. Therefore, it is often preferable to exclude time from the analysis, and to rejoin time and computed principal components afterwards to restore the temporal context. This is what we will do in the next example.

Revealing internal structures with PCA We consider the visual analysis of a meteorological dataset that contains daily observations of temperature (T_{min}, T_{avg}, and T_{max}) for a period of 105 years, which amounts to approximately 38,000 data records (see Nocke et al., 2004). As we are only interested in the summer seasons' weather conditions, the daily raw data are first aggregated into yearly data. To this end, five new variables are calculated for each year:

- *total heat (p1)* as the sum of the maximum temperatures for days with $T_{max} \geq 20°C$,
- *summer days (p2)* as the number of days with $T_{max} \geq 25°C$,
- *hot days (p3)* the number of days with $T_{max} \geq 30°C$,
- *mean of average (p4)* as the mean of the daily average temperatures T_{avg}, and
- *mean of extreme (p5)* as the mean of the daily maximum temperatures T_{max}.

These five quantitative variables are strongly correlated. The extracted dataset can be visualized as a centered layer area graph (⟶ p. 195), as illustrated in Figure 6.9. This visual representation is quite useful to get an overview of the data. We can clearly distinguish valleys and peaks in the graph, which indicate particularly cold and hot summers, respectively. The general trend in the data is communicated quite well.

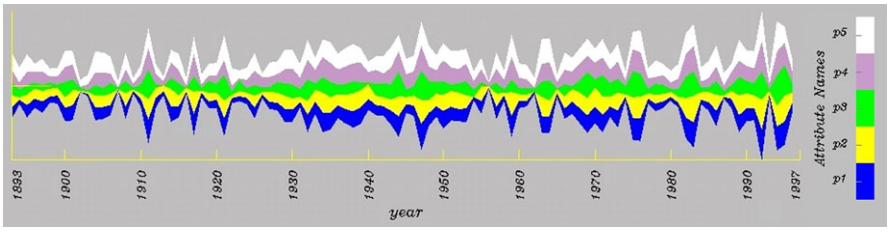

Fig. 6.9: Summer conditions (*p1–p5*) visualized as a centered layer area graph.
Source: Image courtesy of Thomas Nocke.

As we will see next, we can confirm our previous findings and gain further insight with the help of PCA and a simple bar graph (⟶ p. 154). But instead of visualizing all five parameters, our visual analysis will be based on just a single principal

component. So what we do is to apply PCA to the five variables extracted from the raw data. The dimension of time is excluded from the PCA. The computed PCA results are then fed to the visualization. In order to restore the temporal context, the bar graph in Figure 6.10 shows time along the horizontal axis, and the first principal component (PC1), to which all variables contribute because of their strong correlation, at the vertical axis. For each year, a bar is constructed that connects the baseline with the year's PC1 coordinate (i.e., the year's score in principal component space). This effectively means upward bars encode a positive deviation from the major trend, that is, they stand for warmer summers, where long bars indicate summers with extreme conditions. In contrast, downward bars represent colder-than-normal summers. As an additional visual cue, frequencies of score values are mapped onto color to further distinguish typical (saturated green) and outlier (bright yellowish green) years. This visual representation allows us to discern the following interesting facts:

- The first third of the time axis is dominated by moderately warm summers mixed with the coldest summers.
- The hot summers in the 1910s and 1920s are immediately followed by cold summers.
- There were relatively nice summer seasons between 1930 and 1950.
- In general, outlier summers, positive and negative ones, accumulate at the end of the time axis.

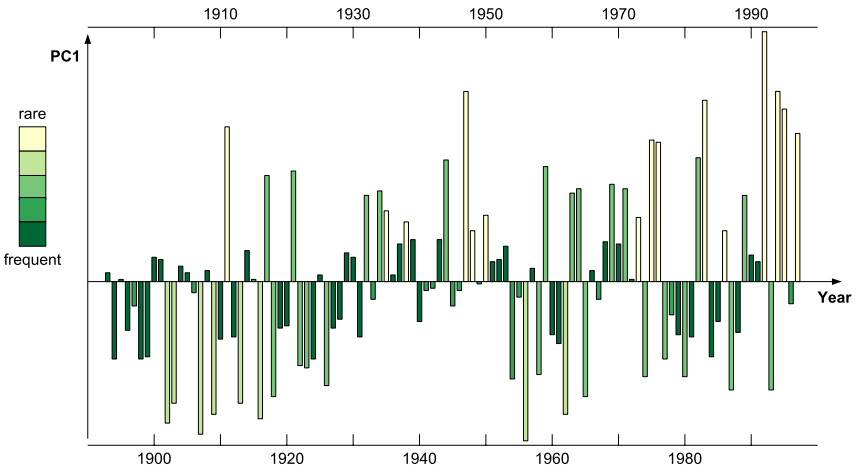

Fig. 6.10: The bar graph encodes years along the horizontal axis and the scores of the first principal component (PC1) along the vertical axis. Color indicates the frequency of score values.

Although the visualization in Figure 6.10 shows only the first principal component, rather than the five data variables, it depicts corresponding trends very well. Nonetheless, one should recall that our data represent a special case where all five

variables are strongly correlated. This correlation is the reason why PC1 separates warm and cold summers so well. When analyzing arbitrary time-oriented datasets, further principal components might be necessary to capture major structural relationships. The following example will illustrate how users can be assisted in making informed decisions about which principal component's scores to display.

Determining significant principal components We now deal with a census dataset with multiple variables, including population, gross domestic product, literacy, and life expectancy. As before, the independent dimensions (i.e., time and space) are excluded to maintain the data's frame of reference, leaving ten variables to be processed analytically by the PCA. Accordingly, the analysis yields ten principal components, which correspond to the major trends in the data. The principal components' significance-weighted loadings indicate how individual variables participate in these trends.

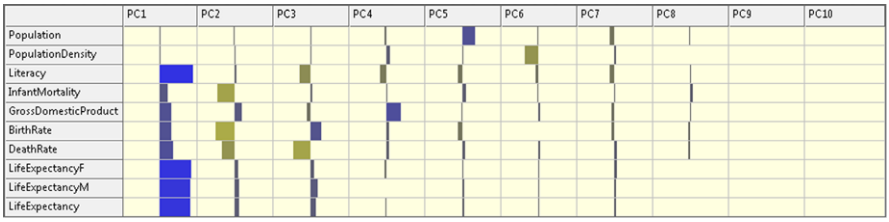

Fig. 6.11: The bars in the table cells visualize the loadings of principal components weighted by their significance. This clearly echoes the ranking of the principal components.

The significance-weighted loadings of our example are depicted in Figure 6.11, where longer bars stand for stronger participation, and blue and yellow color are used for positive and negative values, respectively. By definition, the principal components are ranked according to their significance from left to right. The figure indicates that the data's major trends (PC1-PC4) are largely influenced by the eight variables from literacy to life expectancy. But we can also see that if we consider only these first four principal components, we certainly lose the relation to the two variables population and population density, which do not contribute to the top four trends. Therefore, at least the principal components up to PC5, which is proportional to population, and PC6, which is indirectly proportional to population density, should be retained. In turn, if we are interested in the main trends only, we can safely omit the remaining principal components (PC7-PC10).

If we are interested in outlier trends as well, we should be less generous with dropping principal components. This can be illustrated by a visualization of the plain (i.e., unweighted) loadings of the principal components as shown in Figure 6.12. The figure clearly reveals contradictory contributions of the variables to the lower-ranked trends. In particular, we can see a contradiction between life expectancy of females and males in the ninth principal component (PC9).

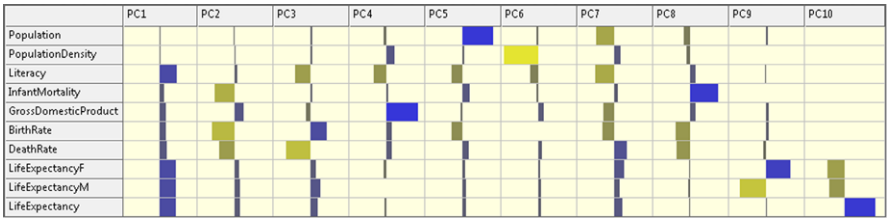

Fig. 6.12: The bars in the table cells visualize the unweighted loadings of principal components, that is, they indicate how much the individual variables contribute to any particular principal component.

The visualization of the loadings helped us in identifying the top-ranked principal components and those that might bear potentially interesting outlier information. The knowledge that we derived about the principal components can also be interpreted in terms of the variables of the original data space. A number of findings can be gained, including:

- All the positive loadings in the main trend (PC1) indicate a direct proportional relationship for the literacy, infant mortality, gross domestic product, birth rate, death rate, and life expectancy.
- The second trend (PC2) is constituted by the gross domestic product, life expectancy as well as infant mortality, death rate, and birth rate, where the latter three variables are indirectly proportional to this trend.
- The major trends in the data (PC1-PC3) are largely independent of population and population density.
- An outlier trend is present in PC9, where the contradictory loadings of life expectancy of females and males might hint at an interesting aspect.

In summary we have seen in this section that PCA is a useful tool for crystallizing major structural relationships in the data and for identifying possible candidates for data reduction.

6.5 Summary

The information seeking mantra proposed by Shneiderman (1996) should guide the users when exploring the data visually:

> Overview first,
> zoom and filter,
> then details-on-demand.
>
> Shneiderman (1996, p. 2)

However, with massive, heterogeneous, dynamic, and ambiguous datasets at hand, it is difficult to create overview visualizations without losing interesting patterns. Therefore, Keim et al. (2006) revised the information seeking mantra, in order

to indicate that it is not sufficient to just retrieve and display the data using a visual metaphor:

> Analyze First -
> Show the Important -
> Zoom, Filter and Analyse Further -
> Details on Demand.
>
> Keim et al. (2006, p. 6)

In fact, it is necessary to analyze the data according to aspects of interest, to show the most relevant features of the data, and at the same time to provide interaction methods that allow the user to get details of the data on demand (see Keim et al., 2010).

In this chapter, we provided a brief overview of how analytical methods can support the visual analysis of time-oriented data. We gave a list of typical temporal analysis tasks and illustrated the utility of analysis methods with the three examples: clustering, temporal data abstraction, and principal component analysis. All of these examples perform a particular kind of data abstraction. Admittedly, our examples are simple, but still we believe that they demonstrate the benefits of analytical methods quite well.

In fact, when confronted with really huge datasets, a single analytical method alone will most certainly not suffice. Instead, a number of analytical methods must play in concert to cope with the size and complexity of time-oriented data. Moreover, analytical methods are not solely a preprocessing step to support the visualization of data. The full potential of analytical methods unfolds only if they are considered at all stages of interactive exploration and visual analysis processes in an integrated fashion depending on the data, users, and tasks.

However, as we will see in the next chapter, more in-depth research and development is necessary to arrive at an intertwined integration of visual, interactive and analytical methods for the bigger goal of gaining insight into large and complex time-oriented data.

References

Antunes, C. M. and Oliveira, A. L. (2001). Temporal Data Mining: An Overview. Workshop on Temporal Data Mining at the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD).

Bade, R., Schlechtweg, S., and Miksch, S. (2004). Connecting Time-oriented Data and Information to a Coherent Interactive Visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 105–112, New York, NY, USA. ACM Press.

Brockwell, P. J. and Davis, R. A. (2009). *Time Series: Theory and Methods*. Springer, New York, USA, 2nd edition.

Clancey, W. J. (1985). Heuristic Classification. *Artificial Intelligence*, 27(3):289–350.

Combi, C., Keravnou-Papailiou, E., and Shahar, Y. (2010). *Temporal Information Systems in Medicine*. Springer, Berlin, Germany.

Fayyad, U., Grinstein, G. G., and Wierse, A. (2001). *Information Visualization in Data Mining and Knowledge Discovery*. Morgan Kaufmann, San Francisco, CA, USA.

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, 17(3):37–54.

Gan, G., Ma, C., and Wu, J. (2007). *Data Clustering: Theory, Algorithms, and Applications*, ASA-SIAM Series on Statistics and Applied Probability. SIAM, Philadelphia, PA, USA.

Han, J. and Kamber, M. (2005). *Data Mining: Concepts and Techniques*. Morgan Kaufmann, San Francisco, CA, USA.

Hsu, W., Lee, M. L., and Wang, J. (2008). *Temporal and Spatio-Temporal Data Mining*. IGI Global, Hershey, PA, USA.

Jackson, J. E. (2003). *A User's Guide to Principal Components*, Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., New York, NY, USA.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data Clustering: A Review. *ACM Computing Surveys*, 31:264–323.

Jeong, D. H., Ziemkiewicz, C., Fisher, B., Ribarsky, W., and Chang, R. (2009). iPCA: An Interactive System for PCA-based Visual Analytics. *Computer Graphics Forum*, 28(3):767–774.

Jolliffe, I. T. (2002). *Principal Component Analysis*, Springer Series in Statistics. Springer, New York, NY, USA, 2nd edition.

Keim, D., Kohlhammer, J., Ellis, G., and Mansmann, F., editors (2010). *Mastering the Information Age – Solving Problems with Visual Analytics*. Eurographics Association, Geneve, Switzerland.

Keim, D. A., Mansmann, F., Schneidewind, J., and Ziegler, H. (2006). Challenges in Visual Data Analysis. In *Proceedings of the International Conference Information Visualisation (IV)*, pages 9–16, Los Alamitos, CA, USA. IEEE Computer Society.

Laxman, S. and Sastry, P. (2006). A Survey of Temporal Data Mining. *Sādhanā*, 31:173–198.

Lin, J., Keogh, E. J., Wei, L., and Lonardi, S. (2007). Experiencing SAX: A Novel Symbolic Representation of Time Series. *Data Mining and Knowledge Discovery*, 15(2):107–144.

Miksch, S., Horn, W., Popow, C., and Paky, F. (1996). Utilizing Temporal Data Abstraction for Data Validation and Therapy Planning for Artificially Ventilated Newborn Infants. *Artificial Intelligence in Medicine*, 8(6):543–576.

Miksch, S., Seyfang, A., Horn, W., and Popow, C. (1999). Abstracting Steady Qualitative Descriptions over Time from Noisy, High-Frequency Data. In *Proceedings of the Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making (AIMDM)*, pages 281–290, Berlin, Germany. Springer.

Mitsa, T. (2010). *Temporal Data Mining*. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, Boca Raton, FL, USA.

Nocke, T., Schumann, H., and Böhm, U. (2004). Methods for the Visualization of Clustered Climate Data. *Computational Statistics*, 19(1):75–94.

Shneiderman, B. (1996). The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*, pages 336–343, Los Alamitos, CA, USA. IEEE Computer Society.

Stacey, M. and McGregor, C. (2007). Temporal Abstraction in Intelligent Clinical Data Analysis: A Survey. *Artificial Intelligence in Medicine*, 39(1):1–24.

Thomas, J. J. and Cook, K. A. (2005). *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Computer Society, Los Alamitos, CA, USA.

Van Wijk, J. J. and Van Selow, E. R. (1999). Cluster and Calendar Based Visualization of Time Series Data. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis)*, pages 4–9, Los Alamitos, CA, USA. IEEE Computer Society.

Ware, C. (2008). *Visual Thinking for Design*. Morgan Kaufmann, Burlington, MA, USA.

Wegner, P. (1997). Why Interaction Is More Powerful Than Algorithms. *Communications of the ACM*, 40(5):80–91.

Xu, R. and Wunsch II, D. C. (2009). *Clustering*. John Wiley & Sons, Inc., Hoboken, NJ, USA.