Chapter 2
Analytics Fundamentals

In this chapter, we are introducing the key methods of data analytics and describe the underlying principles, assumptions, and the thought process behind applying these methods in LTE performance and fault analysis. We start with the most basic form of time-series analysis (statistical process control) and progress through the mathematical concepts of outliers, queueing theory, system performance laws, forecasting, regression, and clustering.

2.1 Statistical Process Control

Statistical process control (SPC) today is the hallmark of a mature technology. When properly established, not only does it allow NOC operators to promptly detect, quantify, scope, and analyze root causes of problems in the network, leading to quickly solving the problems; it also makes forecasting easier and more intuitive.

A number of SPC techniques have been developed since the approach was first introduced by Walter Shewhart of Bell Laboratories in the early 1920s. Shewhart was one of the "early rejecters" of the approach to production process analysis from the standpoint of "normal," or Gaussian, distribution. We have to keep in mind that in the 1920s, the Central Limit Theorem (CLT) was not yet as firmly established in the world of probability and statistics as it is today.

2.1.1 Central Limit Theorem

Central Limit Theorem today serves as a convenient justification for treating any random process as normally distributed, which ensures simplicity of mathematical

D. Kakadia et al., *Network Performance and Fault Analytics for LTE Wireless Service Providers*, DOI 10.1007/978-81-322-3721-1_2

analysis of data leading to actionable results. Indeed, if we assume normal distribution of data, we can get away with using well-established simple methods developed by William Gossett, Karl Pearson, and Ronald Fisher for statistical hypothesis testing. CLT gives us a way to justify such assumption: If we have "enough" independent samples that are "big enough," then the means (arithmetic averages) of these samples converge to a normal distribution with the mean equal to the "mean of the means" and the variance computed as the sum of variances of the samples.

$$\lim_{N\to\infty} \frac{\sum_{i=1}^{N} \mu_i}{N} = \mathbb{N}(\mu, \sigma) \tag{2.1.1}$$

Here

$\mathbb{N}(\mu, \sigma) =$ normal distribution with

$\text{mean} = \frac{\sum_{i=1}^{N} \mu_i}{N}$ and standard deviation $= \sqrt{\sum_{i=1}^{N} \sigma_i^2}$

In Eq. (2.1.1),

$$\text{mean} = \frac{\sum_{i=1}^{N} \mu_i}{N} \tag{2.1.2.a}$$

$$\text{standard deviation} = \sqrt{\sum_{i=1}^{N} \sigma_i^2} \tag{2.1.2.b}$$

Lyapunov's equation for the variance of the aggregate of independent samples (sum of squares of variances) marked an important milestone in probability theory and statistics: It allowed treating samples as dimensions of the process being analyzed, as long as they were independent.

This simple construct enabled a number of breakthroughs in data science; most importantly, it allowed engineers to quickly develop a number of statistically sound robust "rule of thumb," such as Western Electric Rules and Nelson Rules, for statistical process control. It also allowed a delineation of long-term and short-term process capability that has become one of the cornerstones of the Six Sigma Methodology, which became the next breakthrough in statistical quality control developed in the late 1980s and early 1990s and dovetailed into the concept of Design for Six Sigma (DFSS), which is currently used by manufacturers of hi-tech equipment all over the world.

2.1.2 Applications of Central Limit Theorem: Bernoulli Trials

Disclaimer: All charts and data in this chapter are produced by simulation and do not reflect actual vendor data.

"Bernoulli trials" is a term in probability theory representing a scenario where a definition of insanity falsely attributed to Einstein ("doing the same thing over and over and expecting different results") is challenged: Given a binary random variable X (or a process that can be in one of two states, typically "success" and "failure," or 0 and 1), we repeat the same action that produces and tallies the number of trials and the number of times when $X = 2.1$. Then,

$$\Pr\{X = 1\} = \lim_{N \to \infty} \left[\frac{N_{X=1}}{N} \right]$$

As the number of Bernoulli trials increases, the proportion of successful outcomes in the total number of trials tends to the actual probability of success. But this actually is the de Moivre's form of Central Limit Theorem that he published in 1733!

2.1.3 Examples of SPC for Bernoulli Trials

Attempts to establish a connection can be modeled as Bernoulli trials: They can either succeed or fail, and we are interested in the frequency (probability estimation) of successful attempts.

If we assign $p = \text{Probability of success}; N = \text{number of trials}$, then, given sufficient number of samples, CLT enables us to consider accessibility as a normally distributed metric with

$$\text{mean}_{\text{successful trials}}(n, p) = N * p \tag{2.1.3}$$

and

$$\text{variance}_{\text{successful trials}}(n, p) = N * p * (1 - p). \tag{2.1.4}$$

This will enable us to apply the SPC principles directly to attempts to establish connection. Two types of SPC charts, the p-chart and the np-chart (Fig. 2.1), which were originally proposed by Walter Shewhart and was named after him, are specifically designed for this kind of metric.

Collecting data for daily attempts to establish a connection will yield an np-chart like in Fig. 2.1.

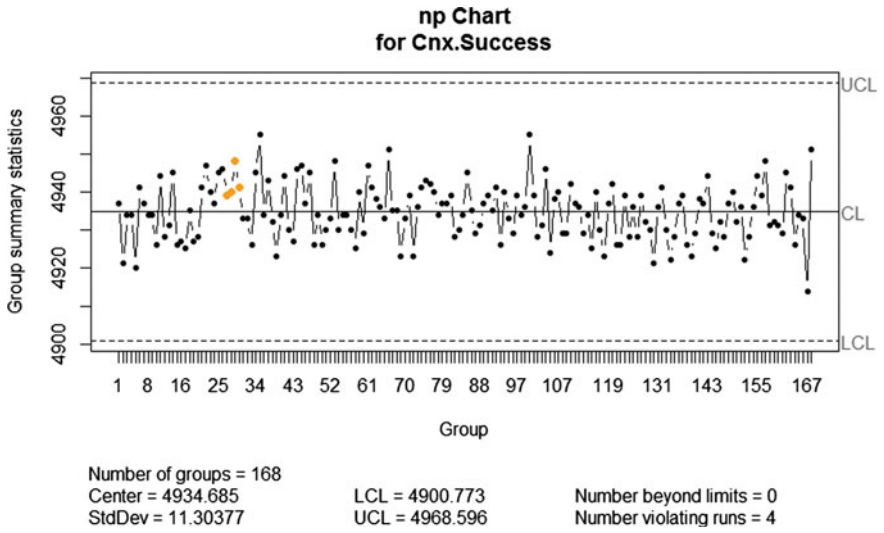

Fig. 2.1 Bernoulli trials: np-chart (simulated data)

Figure 2.1 shows the number of successfully established connections during each hour of one week. The control level (CL) is calculated using Eq. (2.1.3), while the Upper Control Limit and Lower Control Limit (UCL and LCL, respectively) are computed using the "3-sigma" rule:

$$\begin{aligned} \text{UCL} &= \mu + 3 * \sigma \\ \text{LCL} &= \mu - 3 * \sigma \end{aligned},$$

Here, $\mu = N * p = \text{mean}; \sigma = \sqrt{N * p * (1 - p)} = \text{standard deviation}$.

Figure 2.2 shows the number of successful ERABs established during each hour of one week. The control level (CL) is calculated using a form of Eq. (2.1.3) where both sides of the equation are divided by the number of samples (N), while the Upper Control Limit and Lower Control Limit (UCL and LCL, respectively) are computed using the "3-sigma" rule:

$$\begin{aligned} \text{UCL} &= \mu + 3 * \sigma, \\ \text{LCL} &= \mu - 3 * \sigma \end{aligned}$$

Here, $\mu = p = \text{mean}; \sigma = \sqrt{\frac{p*(1-p)}{N}} = \text{standard deviation}$.

The np- and p-charts enable the user to easily detect events when the data was out of control, e.g., the red data point at Group 87 (87th hour).

They also allow the analyst to find violations of the SPC rules of thumb, in particular Western Electric Rules (http://en.wikipedia.org/wiki/Western_Electric_rules) or Nelson Rules (http://en.wikipedia.org/wiki/Nelson_rules).

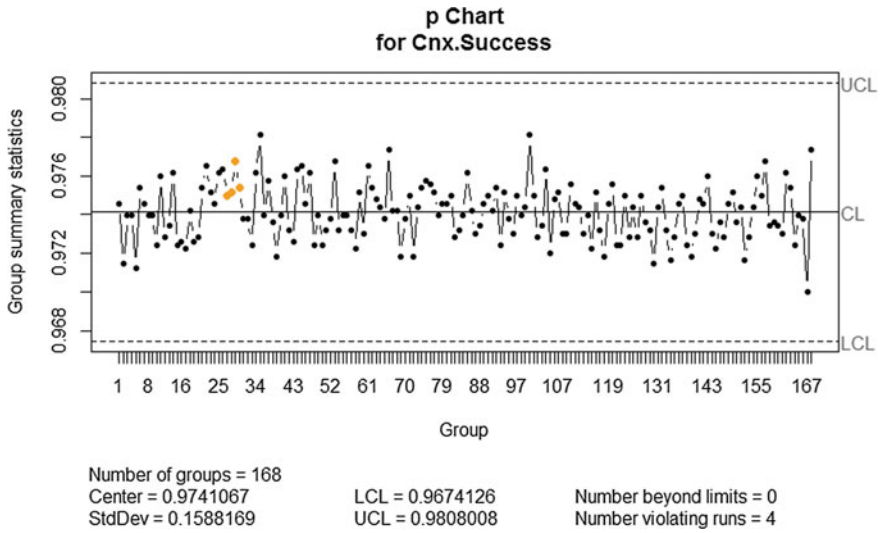

Fig. 2.2 Accessibility p-chart (simulated data)

Thus, the four yellow points in Figs. 2.1 and 2.2 correspond to violation of Nelson Rule 6 (or its equivalent Western Electric Rule 3): 4 out of 5 points fall outside the 1-standard-deviation band on the same side of the mean.

When the number of successful connections and the number of attempts is known for every hour, the R code to produce this graph is very simple:

```
################################################################
## Plot the np-chart
################################################################
if (! require ("qcc")){install.packages ("qcc")}
library (qcc)
npChart <- qcc (Cnx.Success, sizes = Attempts, type = "np")
pChart <- qcc (Cnx.Success, sizes = Attempts, type = "p")
```

2.2 Outliers

The "magic of LTE" involves a lot: A variety of users, sometimes with a number of different user elements per user, are trying to get access to the data. Uploads, downloads, VoIP communications, all requiring different connection times, different QoS, and different packet sizes, happen at the same time, ensuring that there is no limit to confusion and outliers.

This section is not a diversion into sociology or social psychology; however, it is important to note that despite the apparent chaos, there is a high degree of order and predictability in LTE. Outliers have been usually given the role of a nuisance whose only role is to annoy the engineers and throw off the beautiful developments that would have been so much easier if only there were no outliers. In this section, we will discuss outliers, their detection, and analysis in stationary (in the strict sense) and non-stationary systems. We will also try to show that not all outliers are bad and that they always carry useful actionable information, if only we can extract it.

2.2.1 QoS Outliers

The most common way to bring order out of chaos is to organize data in a manner that makes sense to the analyst. Traditionally, all IT services have been organized from the point of view of the quality of service, or QoS.

In LTE networks, there are nine QoS class indicators (QCIs), each with a specification on a packet delay budget and error loss rate (see Chap. 1 and Fig. 2.3).

The predefined packet delay budgets for each QoS class mean that an outlier may very well put a bearer into a lower-priority class, at least for the outlier. For example, a Priority 1 (QCI = 5, delay = 100 ms) connection may too easily fall into Priority 6 (QCI = 6, delay budget = 300 ms).

While packet error loss rate may be approximated as a normal distribution (see discussion in Sect. 2.1), packet delays are strongly right-skewed: Their distribution

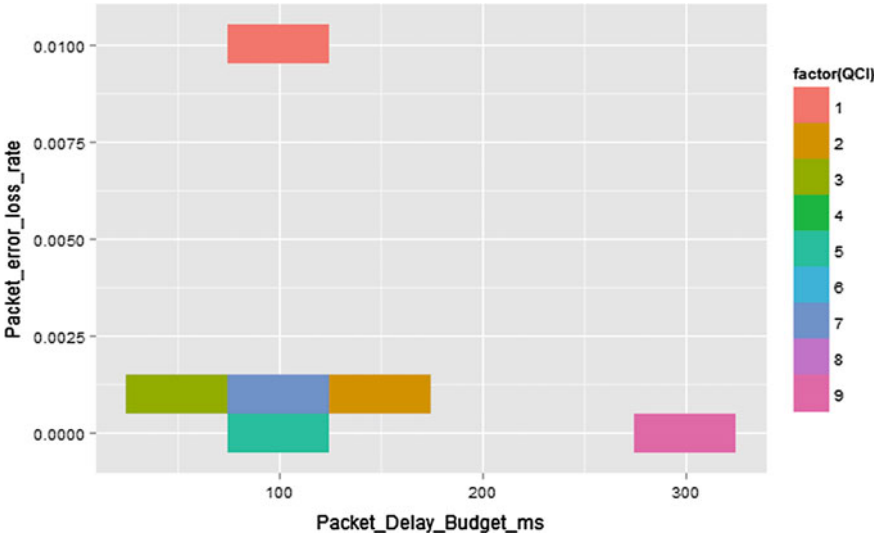

Fig. 2.3 LTE QoS class indicators (QCIs) defined by loss and delay specifications

has a long tail to the right. This phenomenon is often approximated by an exponential distribution. This makes the packet delay extremely sensitive to outliers.

Before we take a deeper dive into the topic, it is important to note that the concepts of outliers and of SLA (QoS) violations are, generally speaking, independent: Not all outliers are in violation of SLA and not all events of SLA violation are outliers.

2.2.2 Outliers: What Are They?

According to www.dictionary.com, an outlier in its statistical sense is

> … an observation that is well outside of the expected range of values in a study or experiment, and which is often discarded from the data set…

And here lies one of the most common temptations that we face when dealing with real-world data: The dictionary merely states it, but all too often we tend to discard outliers, even though more often than not, they are "in the eyes of the beholder."

Sometimes outliers do represent bad data collections, mangled data points, and simply unreliable data. However, when we know that the data is good, outliers become are important observations that can give us an advanced warning that the system's performance is beginning to degrade.

When the number of outliers for a KPI reaches an alarming level (again, "in the eyes of the beholder," but at least it can be quantified and parameterized), we want to continue monitoring the KPI specifically for the relative and absolute growth in outlier count and magnitude, to be able to predict when the performance will degrade if the system is left alone and to take appropriate corrective action.

Usually, the beholder—a subject-matter expert (SME)—knows what percentage of outliers in a sample the system can handle. Such knowledge is often guarded by the SME and eventually becomes "tribal knowledge" in the organization.

When dealing with thousands of objects providing data, the beholder's eyes become too strained and the beholder too bored analyzing each and every data set for outliers. In other words, an automatic outlier detection tool has to be implemented. Such tools are usually based on proven quantitative analysis (QA) techniques, which we shall consider here.

2.2.3 Outlier Detection: The Basic Approach

The SPC (see Sect. 2.1) represents outlier detection by applying the density-function method. As we discussed in Sect. 2.1, the mathematically simplest way to detect outliers is by using the mean (average) and standard deviation of the data. We then apply a multiplier to the standard deviation and get the confidence

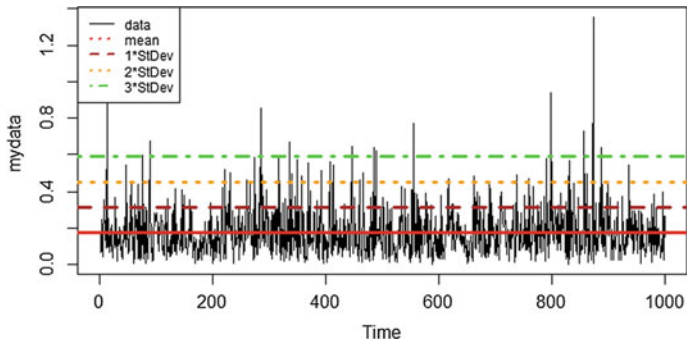

Fig. 2.4 A simulated right-skewed data set—representative of a typical latency (packet delay) observed on an LTE network—and its 0, 1, 2, and 3 standard deviations

interval and state that points outside such confidence interval (usually 3 standard deviations) are outliers. This is known as the "3-sigma rule."

Figure 2.4 shows a set of simulation data that we built to illustrate the concept. This view of the data allows us to identify outliers at different confidence levels:

$\pm 1 * \sigma$	68.2%
$\pm 2 * \sigma$	95%
$\pm 3 * \sigma$	99.7%

In other words, if a data point falls outside the $1 * \sigma$ range, we say with 68.2% confidence that this data point is an outlier. If another data point falls outside the $3 * \sigma$ range, that means that we are 99.7% certain that this data point is an outlier.

The code implementing these principles is here:

```
#####################################################################
## Returns outlier indices in vector X corresponding to Z*StDev
#####################################################################
GetZRangeOutlierIndices = function(X, Z = 3, Upper = TRUE, Lower = TRUE){
 # Upper = TRUE; Lower = TRUE ### for debugging
 Mean <- mean (mydata); SD <- sd (mydata)
 upper <- Mean + Z * SD
 lower <- Mean - Z * SD
 UpperOLindices <- which (Upper & X > upper)
 LowerOLindices <- which (Lower & X < lower)
 Outliers <- c(LowerOLindices, UpperOLindices)
 Outliers
}
#####################################################################
## Generate 1000 samples of gamma-distributed "packet delay" data:
mydata <- rgamma (1000, shape = 1.7, rate = 10)
```

```
#####################################################################
## Get outlier indices in mydata corresponding to 1, 2, and 3 StDev
#####################################################################
zOLs.1 <- GetZRangeOutlierIndices(mydata, Z = 1)
zOLs.2 <- GetZRangeOutlierIndices(mydata, Z = 2)
zOLs.3 <- GetZRangeOutlierIndices(mydata, Z = 3)
```

We intentionally did not show the left side of the standard-deviation bands: The data is right-skewed, and for this particular analysis, we are not interested in low outliers.

Advantages of this approach are obvious:

1. It is mathematically simple.
2. It is fast.
3. Measures (mean and standard deviation) can be computed recursively, hence
   (a) Simple data storage, regardless of size.
   (b) Simple computations, regardless of size.
4. It makes sense statistically, giving answers in terms of confidence intervals.
5. It makes sense philosophically: Everybody knows what confidence is.

Disadvantages are fewer, but they can overwhelm the advantages:

1. Assumptions have to be made about the distribution.
2. The method will not work when these assumptions are wrong.

2.2.4 Advanced Methods of Outlier Detection

Section 2.6.2 discusses advanced methods of outlier detection and demonstrates an example of using information from in network performance analysis.

2.3 A Few Words About Queueing Systems

An illustration of a basic queueing system is shown in Fig. 2.5. The number of queueing circuits available is defined by the spectrum range and number of channels occupied at any given time (Bandwidth Efficiency).

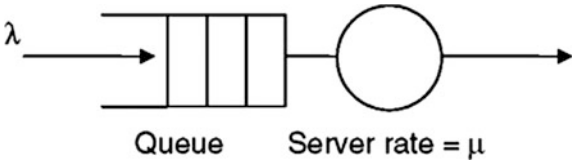

Fig. 2.5 A simple queueing system representing a EUTRAN channel

In this section, we will demonstrate why Central Limit Theorem is a hindrance, rather than help, in network performance analysis. When we are dealing with samples and sample averages, we will see them in data as normally distributed and apply statistical methods that will potentially lead us to wrong conclusions.

Indeed, for a queueing system with a single server (channel) and a buffering queue, the reason why we can describe the entire process easily via two variables, $\lambda = \text{rate}_{\text{arrivals}}$ and $\mu = \text{rate}_{\text{server}} = \frac{1}{T}$, is the assumption that they are not normally distributed ($T =$ average service time).

Moreover, assumption of normal distribution would lead to unintended consequences. For example, a normally distributed service time (latency for a network) would have allowed negative latencies (Fig. 2.6).

In addition, even if the likelihood of negative latencies allowed by a specific normal distribution was negligibly small, the workload on the network element [measured as number of packets in flight and expressed as $W = \lambda * T = \lambda/\mu$] would come out skewed (Fig. 2.7).

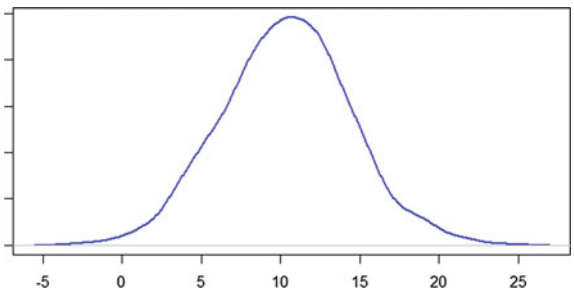

Fig. 2.6 A normal distribution allows negative latencies

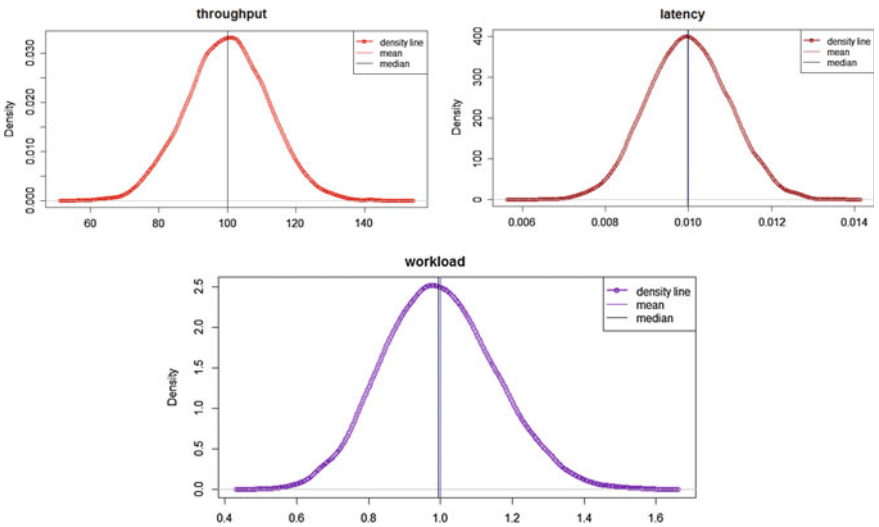

Fig. 2.7 An illustration of the skew in the product of two normal distributions: The gap between the *purple* (mean) and the *black* (median) *vertical lines* indicates that the distribution is skewed. For more information on skewness, see other literature

2.3.1 *"True" Process Distributions for LTE Network Components*

As Agner K. Erlang demonstrated in the 1920s, memoryless processes (where time intervals between adjacent events are independent of previous events) provide a very adequate representation of a queueing system. The only memoryless distribution known today is the exponential, and if we model network latencies using exponential distribution, we will have an adequate representation of the behavior of a network, including LTE network.

In order for event inter arrival times to be exponentially distributed, arrival process has to be Poisson.

Figure 2.8 illustrates these concepts in light of Little's Law.

This leads to workload distributions being non-normal as well.

This observation has two very important consequences: (1) if processes in an LTE link are memoryless, then Erlang models can be used to predict packet loss due to link overloads; (2) we cannot use any statistical methods created with the assumption of Gaussian (normal) distribution of underlying data.

Long story short, traffic metrics (throughput, latency, and workload) of a queueing system are not normally distributed.

2.3.2 *Little's Law: The "Big Three" of Queueing Dynamics*

A queueing-system dynamics can be fully characterized by three variables: (i) arrival rates, (ii) service rates, and (iii) workload. These are the "Big Three" of queueing dynamics.

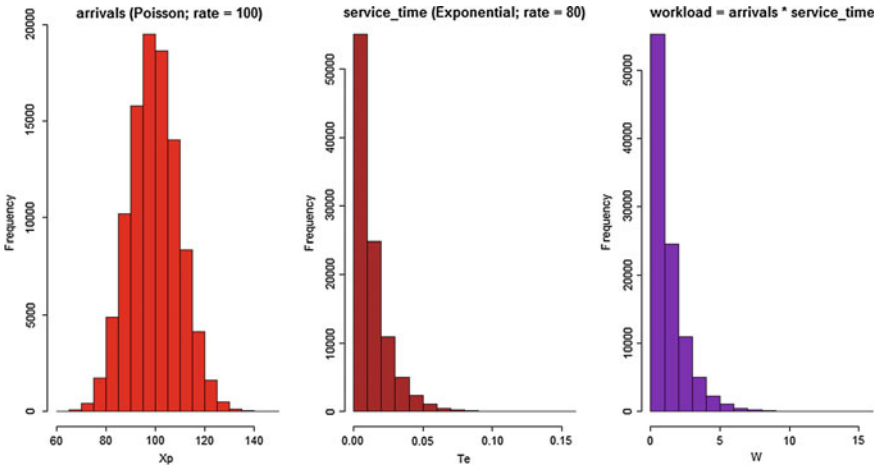

Fig. 2.8 The (throughput, latency, workload) tuple describing a queueing system

Packet Arrivals

Arrival rate is the characteristic of the demand on the queueing system.

Most commonly used models for sizing queuing systems today (Erlang models) rely on the assumption of certain prescribed distributions of arrival and service rates; in particular, that arrivals should be a Markov process, i.e., that job (packet) arrival rates should follow Poisson distribution.

For a long time, network engineering world has been rejecting the models derived for telephony on the grounds that Poisson distribution does not apply to network traffic, which is bursty and unpredictable. However, as our understanding of network, and in general IT traffic grew, so did our understanding that bursts could be modeled as Poisson arrivals of "trains" of packets, whose size (number of packets per "train") could be modeled as a Markov process. This opened an understanding of nested Markov arrival processes [IPPM2014], as well as made it possible to apply a quasi-Erlang model, approximating the nested Poisson distribution with a geometric distribution [FERR2014].

Modern methods of operations research (e.g., Monte Carlo analysis) open new opportunities for creating network analytical systems for LTE where traffic mixes with a variety of arrival processes can be analyzed from the perspective of the probability of delay and/or probability of blocking.

Service Process

Little's Law (Eq. 2.3.1) is a universal law of nature. It ties together the concurrency (number of units of work, e.g. jobs, queries, and packets) that are in the system at any given moment in time) with arrival rate and waiting time:

$$W = \frac{\lambda}{\mu} = \lambda * T \tag{2.3.1}$$

In turn, waiting time can be represented as the sum of the time in and the time in service:

$$T = T_{\mathrm{W}} + T_{\mathrm{S}}, \tag{2.3.2}$$

From Eqs. (2.3.1) and (2.3.2), we get:

$$W = \lambda * T_{\mathrm{W}} + \lambda * T_{\mathrm{S}} \tag{2.3.3}$$

In turn, T_{W} is a function of the size of the queue (measured in units of work) and the arrival rate:

$$T_{\mathrm{W}} = \frac{Q}{\lambda} \tag{2.3.4}$$

T_S is the inverse of service rate:

$$T_S = \frac{1}{\mu} \tag{2.3.5}$$

One very important property of service time for a network is that it depends on the packet size. In other words, while the bit processing rate is a constant for any network element, the number of packets that it can turn out varies:

$$\mu_P \text{ [packets per second]} = 8 * \frac{\text{BW [bits per second]}}{S \text{ [bytes per packet]}} \tag{2.3.6}$$

The bandwidth of a network element is a critical parameter defining the packet service time!

Putting it All Together

Putting (2.3.3)–(2.3.6) together, we have an expression for the number of concurrent packets in the network element:

$$W \text{ [packets]} = \lambda \text{ [packets per second]} * \left\{ T_W + \frac{S \text{ [bytes per packet]}}{8 * \text{BW [bits per second]}} \right\} \tag{2.3.7}$$

This can be translated into:

$$W[\text{bits}] = \frac{8 * \lambda \text{ [bits per second]}}{S \text{ [bytes per packet]}} * \left\{ T_W + \frac{S \text{ [bytes per packet]}}{8 * \text{BW [bits per second]}} \right\} \tag{2.3.8}$$

Generally speaking, the packet size, S [bytes per packet], is not the same for arriving packets as it is for the packets already in flight (in service). Simple algebra leads to:

$$\forall i \in \{\text{incoming packet types}\}:$$

$$W_i^S \text{ [bits]} = \frac{8 * \lambda_i \text{ [packets per second]}}{S_i^{\text{in}} \text{ [bytes per packet]}} * \left\{ T_W + \sum_{f \in \{\text{packet types in flight}\}} \frac{S_f^{\text{out}} \text{ [bytes per packet]}}{8 * \text{BW [bits per second]}} \right\} \tag{2.3.9}$$

In Eq. (2.3.9),

S_i^{in} size of packet type i

W_i^S number of packets in flight of type i

λ_i [packets per second]	arrival rate of packets of type i
S_f^{out}	size of packets in flight of type f
BW	bandwidth of the network element

And total concurrency (number of bits in flight) will be:

$$W\ [\text{bits}] = \sum_{i \in \{\text{incoming packet types}\}} W_i^S\ [\text{bits}] \tag{2.3.10}$$

Equations (2.3.9) and (2.3.10) are very important: They allow us to compute the packet delay. Indeed, simple algebra yields:

$$\begin{aligned} T_{\text{W}} = & \sum_{i \in \{\text{incoming packet types}\}} \frac{W_i^S\ [\text{bits}] * S_i^{\text{in}}\ [\text{bytes per packet}]}{8 * \lambda_i\ [\text{packets per second}]} \\ & - \sum_{f \in \{\text{packet types in flight}\}} \frac{S_f^{\text{out}}\ [\text{bytes per packet}]}{8 * \text{BW}\ [\text{bits per second}]} \end{aligned} \tag{2.3.11}$$

Equation (2.3.11) is the general form of the waiting (queueing) time, which is the main component of packet delay. If we tie it back with Little's Law, it becomes mathematically obvious how to compute the buffer for packet delay.

2.3.3 System Performance Laws

Some LTE behavior patterns are unique to LTE. Others follow well-established laws of IT systems, many of them derived from queueing-system considerations.

2.3.3.1 Amdahl's Law

In 1967, Gene Amdahl formulated the argument against putting too much trust into parallel computing. According to this argument, which later became known as Amdahl's Law, there is a limit to the speedup that a computing system can obtain as a result of massive parallelization. This limit is controlled by the relationship between the number of parallel processors and the amount of serial processing in the program as percentage of total processing:

$$\text{Speedup} = \frac{1}{S + \frac{1}{N} * (1 - S)} \tag{2.3.12}$$

In Eq. (2.3.12), The S and the N are the fraction of processing that is strictly serial and the number of parallel servers, respectively.

2.3.3.2 Universal Scalability Law

In 1993, Neil Gunther formulated the Universal Scalability Law (aka USL), which took Amdahl's Law one step further and generalized it to the case of processors communicating within each other. For more details, we highly recommend [GUNT2007]. Gunther formulated the relationship between the system throughput and the number of concurrent users in the system:

$$C = \frac{N}{1 + \sigma * (N - 1) + \kappa * N * (N - 1)} \tag{2.3.13}$$

In (2.3.13), the C is the system throughput (capacity); σ and the κ are parameters describing contention for resource and incoherency (lack of communication) during the job execution, respectively.

Figure 2.9 shows how the USL curve changes with different USL parameters.

Depending on the system's performance parameters (contention and incoherency), the USL curve will grow more or less steep, level off at some number of concurrent users, or peak and start going down having reached the maximum number of users it can handle.

In the recent years, there have been a number of improvements to USL and its "cousin," the Superserial Scalability Law (SSL); in particular, [CHDY2014] suggested a third parameter, β, to account for queueing:

$$C = \frac{N}{1 + \sigma * (N - 1) + \kappa * N^{\beta} * (N - 1)} \tag{2.3.13a}$$

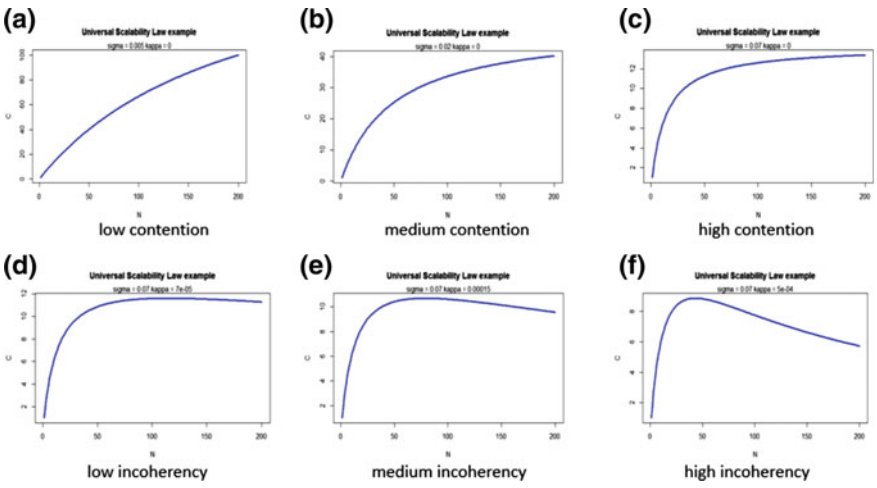

Fig. 2.9 Universal scalability law examples

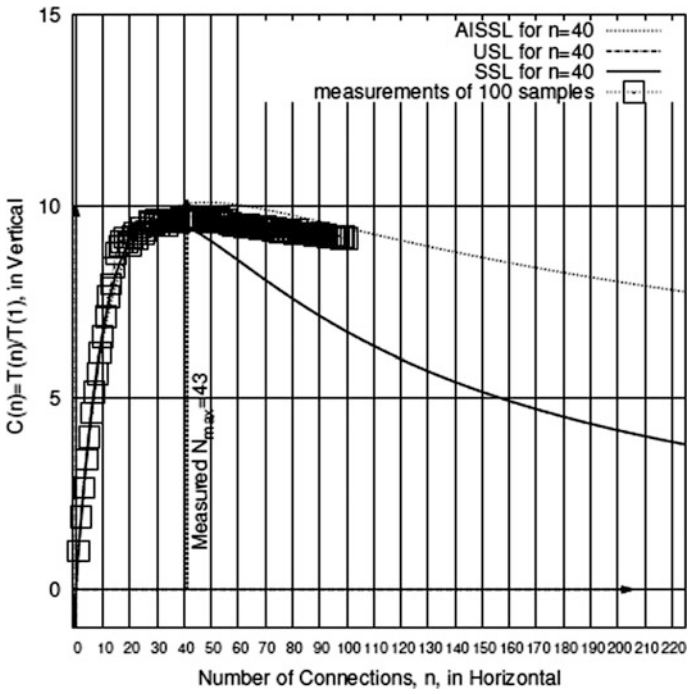

Fig. 2.10 Different versions of USL and SSL and real-world data (reproduced from [CHDY2014] with author's permission)

The Choudhury's AISSL (Asymptotically Improved SSL) form better fits real-world data by smoothing the sharp drop predicted by USL in Fig. 2.9f (Fig. 2.10).

2.3.4 Conclusion

In this section, we have covered queueing process distributions, Little's Law, and scalability laws (Amdahl's and the USL family) and described the packet delay due to retransmissions.

In the following three sections, we will cover forecasting, regression, and clustering, which are the three data-analytical techniques that must be in the toolbox of the LTE data analyst.

2.4 Forecasting

Forecasting is an important analytical technique that is especially useful in capacity planning and business analytics. There are a number of forecasting methods, and each of them deserves a separate book in its own right. In this book, we will only

cover forecasting techniques formulated for time-series analysis (TSA). However, in Sect. 2.5, we will discuss regression and its use in forecasting.

2.4.1 Time Series: Definition and Assumptions

A time series is a set of system metric measurements sampled in constant time intervals and arranged in the order of the time stamps. In TSA, assumptions are made that there are no missed data and that each metric's time series can be viewed as independent of other metrics.

In TSA forecasting, the additional assumption is made that the system's behavior in the past will continue into the future. While the first to assumptions can be justified or enforced, this last one is not as trivial.

Consider the system in Fig. 2.11. It can be any MISO system, e.g., two big groups of users as the input and SINR as the output.

Note that while Fig. 2.11 shows a dependency between the inputs and the output of the system, and the two inputs most likely are interacting (e.g., multiplexing), as long as the processes described by the system are stationary, assumption of independence will hold true for all 3 signals. This can be confirmed by running a number of statistical tests, from correlation tests to ANOVA. For more details, see earlier sections of this chapter, as well as other literature. As long as it is the case, we can apply the forecasting techniques that we are introducing in this section to predict future behaviors of $X1$, $X2$, and Y.

However, if a user control is introduced (Fig. 2.12), the situation will change.

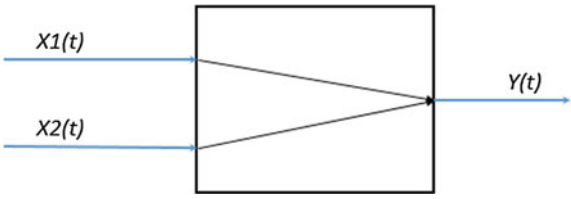

Fig. 2.11 A system with 2 incoming and one outgoing signals

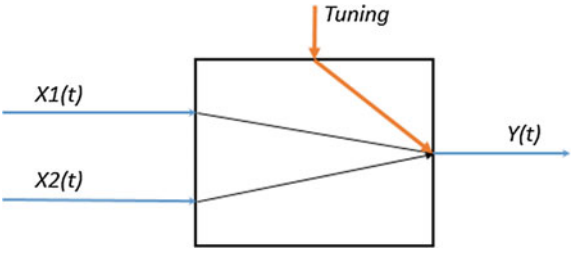

Fig. 2.12 The same system with 2 incoming and one outgoing signals

Now, we have introduced a tuning parameter, e.g., the azimuth of the tower's antenna. Now, the SINR has changed, and the predictions made for $Y(t)$ based on previous behavior will not be adequate anymore.

2.4.2 *Filling in the Gaps in Data*

2.4.2.1 Why?

All TSA techniques are based on having constant time intervals between sequential measurements and the data being ordered by time stamps. On the most basic level, a gap in data means that we have no knowledge about what was happening to the system during this time interval.

2.4.2.2 Methods: Extension

The most primitive method for filling in the gaps in data is extension of the last point's value that preceded the gap, aka LOCF—Last Observation Carried Forward. It is very effective and "unassuming" when it comes to small gaps that need to be filled. The downside is that the value changes in a step function (see the step at X = 30 in Fig. 2.13), which sometimes may mean an actual level shift and sometimes a gap filled by extension.

2.4.2.3 Methods: Interpolation

The most basic method of filling in the gaps is interpolation. The technique has been known since the times of Pythagoras, but we will start with it.

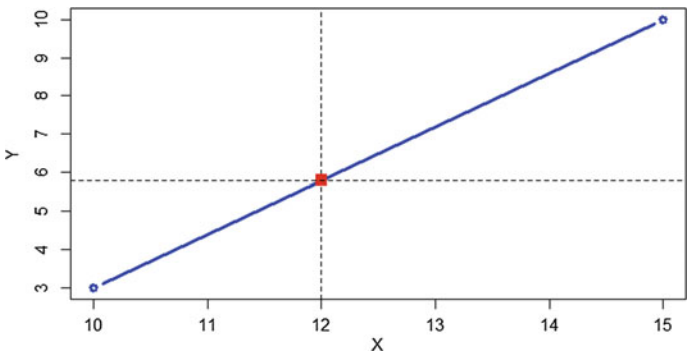

Fig. 2.13 Interpolation

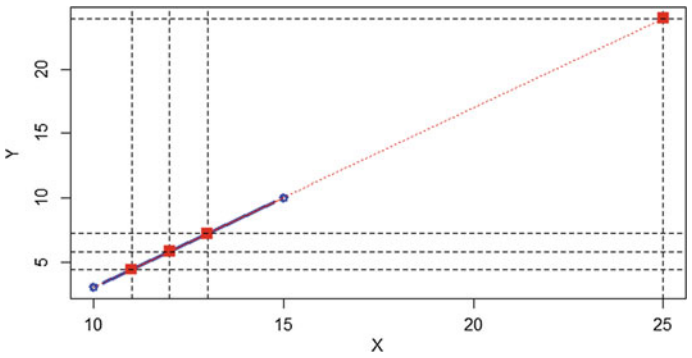

Fig. 2.14 Multi-point inter- and extrapolation

If Y is the value of the dependent variable that we need to find given X—the value of the independent variable, and we are given $(X_1, Y_1), (X_2, Y_2)$ pairs, then:

$$Y = Y_1 + (X - X_1) * \frac{Y_2 - Y_1}{X_2 - X_1} \tag{2.4.1}$$

Naturally, the formula (2.4.1) works just as well for multiple values of X and even for points outside the initial $[X_1, X_2]$ range in which case it is known as extrapolation (Fig. 2.14).

If we are dealing with a time series, interpolation gives us a simple tool to deal with missing data: If X is the point ID (or time stamp), and Y is the value we have measured on the system, and some X values do not have a corresponding Y, we can use interpolation to estimate the missing values (Fig. 2.15).

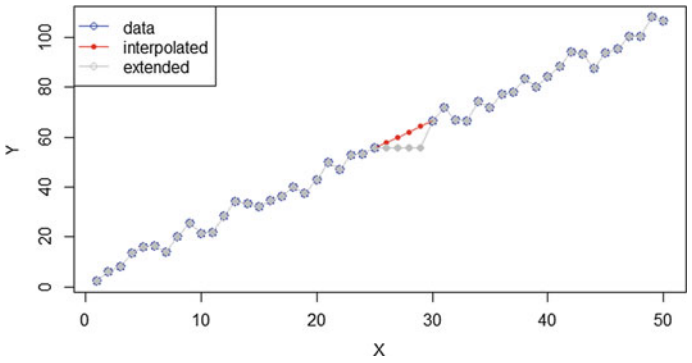

Fig. 2.15 Time series with extension and interpolation of missing data

2.4.3 Moving Averages

The moving average is one of the most important techniques used in the analysis of time series. An illustration of what moving average does is provided in Fig. 2.16.

One common thing among all MA methods is that they work as filters, greatly reducing the variability of the data. This makes them invaluable in outlier detection: look at the residuals between the filter and the data and identify outliers using any outlier detection method that we discussed in the earlier sections of this chapter, and we are done.

The main principle in applying moving-average filter techniques is in computing the average of N (usually called the filter's period) points near the current point. It may be N points behind, N points centered on the current point, or N points in front of it. In TSA forecasting, it is usually N points behind.

To improve the performance of the MA filter, each point is endowed with a weight, with the idea that the points far in the history should have relatively little say in the behavior of the filtered data, whereas more recent data points should carry a lot more weight.

The techniques produce different curves. Closest to the data line (Y in Fig. 2.16) are typically SMA (simple moving average) and WMA (weighted moving average). The SMA in time-series analysis is also known as "random walk," because of its sensitivity to the stochastic component of the data. The most sensitive to trending and periodic patterns in data is the EWMA, aka EMA (exponentially weighted moving average), aka exponential smoothing. The LWMA (linearly weighted moving average) is in between these two.

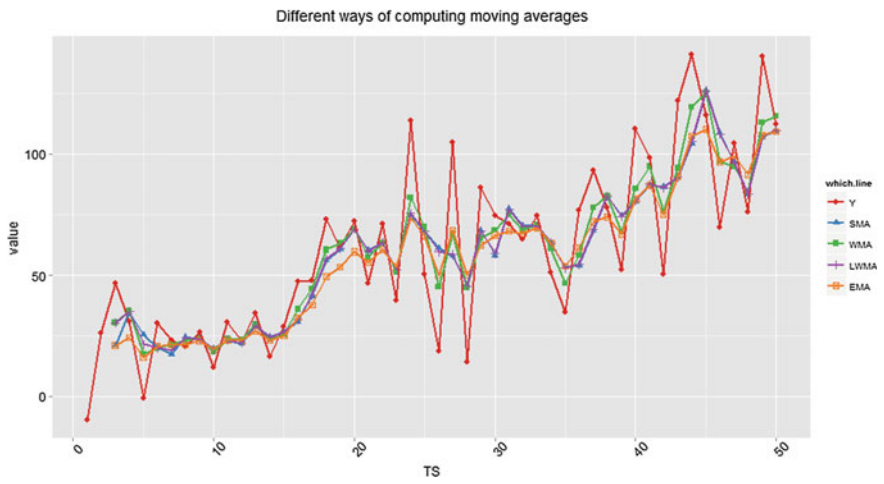

Fig. 2.16 An illustration of different ways of computing moving average

2.4.4 EWMA

The word "exponential" in "exponential smoothing" causes a lot of confusion outside the data-analytical community. It has nothing to do with the trend being exponential, linear, or even flat. The EWMA model computes the next value based on weighted previous data points in a manner that can be recursively written as shown in Eq. (2.4.2):

$$\begin{aligned} y_0 &= x_0 \\ y_t &= \alpha * x_t + (1-\alpha) * y_{t-1}, \quad \forall t > 1 \end{aligned} \tag{2.4.2}$$

In Eq. (2.4.2),

x the time series;
y the fitted model;
$0 < \alpha < 1$.

Equation (2.4.2) can be rewritten as:

$$\begin{aligned} y_t = \alpha * \Big[x_t + (1-\alpha) * x_{t-1} + (1-\alpha)^2 * x_{t-2} + \cdots + (1-\alpha)^{t-1} * x_1 \Big] \\ + (1-\alpha)^t * x_0, \quad \forall t > 1 \end{aligned}$$

Or in a more compact form:

$$y_t = \alpha * \sum_{i=0}^{t} (1-\alpha)^{t-i} * x_i + (1-\alpha)^t * x_0, \quad \forall t > 1 \tag{2.4.3}$$

It is because of the exponential form of the $(1-\alpha)$ weights that the method is called exponentially weighted moving average.

2.4.5 EWMA Forecasting

The only limitation on the t in Eq. (2.4.3) is that it should be greater than 1. This means that we can extend the series using EWMA forecasting to any point in the future.

An illustration is shown in Fig. 2.17.

Prediction Interval

The two nested gray envelopes around the blue line in Figs. 2.17 and 2.18 are measures of forecast precision. They outline the zones where 60% (dark gray) and 90% (light gray) of the data will likely fall for each time stamp in the forecast horizon.

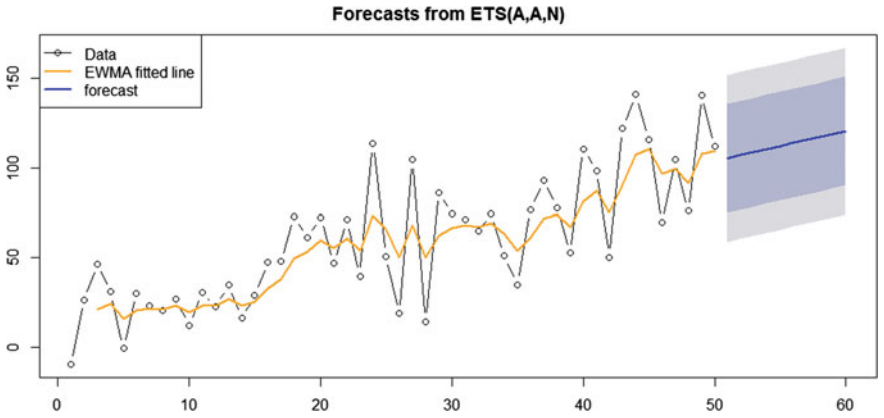

Fig. 2.17 EWMA forecast of the time series shown in Fig. 2.16

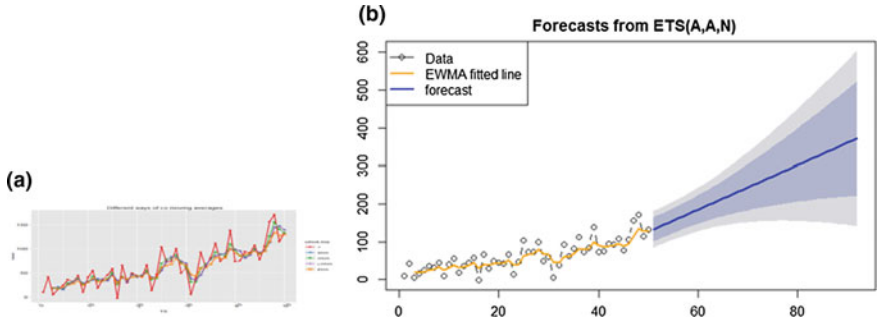

Fig. 2.18 A time series with a runaway prediction interval: **a** the four moving-average filters applied to the time series, **b** the EWMA forecast

In Sect. 2.1, when we discussed SPC, we talked about confidence intervals. For forecasts, the term "confidence interval" is replaced by "prediction interval." It is computed based on the model residuals (for a more in-depth discussion of model residuals, reader is referred to Sect. 2.5).

The "any point in the future" should be taken with a grain of salt: Forecasting a time series into a horizon bigger than the size of the historical data set is ill-advised. Typical forecast horizons are usually between half and 90% of the length of the historical data.

Consider the time series in Fig. 2.19.

Here, the forecast prediction interval is growing as we are moving farther away from the last historical data point, and by the time the forecast horizon (42 points) is reached, the upper 95th percentile of the prediction interval is already in the low 600 s, while the lower 95th percentile is still in the low 140 and even below the value of the time series at the last historical data point.

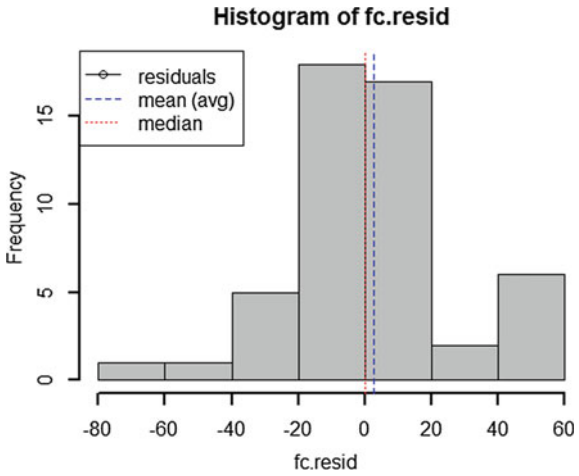

Fig. 2.19 Histogram of the EWMA forecast residuals

About Point Forecasts and Model Residuals

A point forecast is a prediction of the value being forecasted that does not have a prediction interval associated with it: Each time stamp of the forecast horizon is a single point. In Figs. 2.17 and 2.18, it is represented by the blue line.

It is important to remember that regardless of what shape the forecasted line takes, it is not meaningful to talk about a point forecast: Data volatility necessitates ranges of uncertainty around the centerline of the forecast.

Forecast residuals are computed as

$$\text{resid} = \text{data} - \text{forecast}$$

Their values are an important measure of how well the model fit the data. For the forecast shown in Fig. 2.18, the histogram of the residuals is shown in Fig. 2.19.

We see from the histogram that (i) the distribution of the residuals is not normal (it is compressed on both sides); (ii) there is a big bulge in the distribution where the tail usually is; and (iii) the forecasting model was anchored to the median (the red vertical line is zero) and not to the mean of the data.

Boxplot (see details in Sect. 2.2) reveals that we have outliers, and plotting the residuals as a time series and drawing the outlier boundary lines ("whiskers") will help us identify them (see details in Sect. 2.1). These concepts are illustrated in Fig. 2.20.

Measures of Forecast Model Quality

A number of measures of model quality have been developed over the years that EWMA forecasting has been in operation. Any forecasting tool can be tuned to optimize any of these statistics, reducing the error margins. The forecast we analyzed has been anchored to the median, but it could have been anchored to the mean of the historical data.

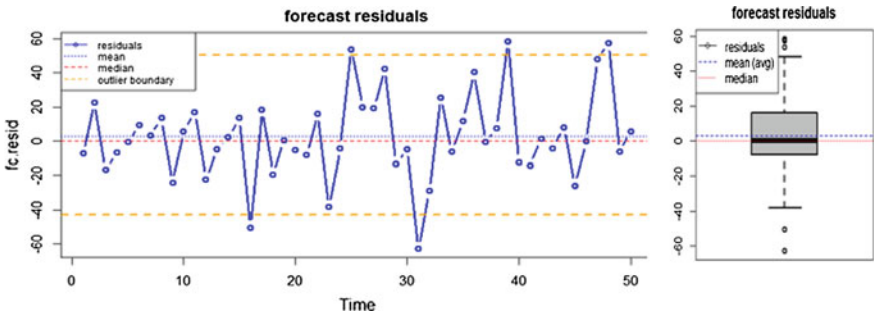

Fig. 2.20 Forecast residuals time series and boxplot

Seasonality and Trend

The same EWMA forecasting Eq. (2.4.3) can be, and have been, generalized to forecast trend and seasonality. The idea is the same: adding exponentially weighted values of the historical time series to calculate the trend and the seasonality at each point of the horizon (the so-called additive trend and additive seasonality). Moreover, they can be, and have been, modified by replacing "+" with a "*" to implement multiplicative seasonality and trend in EWMA forecasting.

Seasonality in time-series analysis is not necessarily tied into the four seasons. Any periodic pattern in a time series is called seasonality. It can be annual, semi-annual, quarterly, monthly, weekly, diurnal, and even hourly, but these are absolutely not the only periodic patterns that can be observed in the data. Seasonality in time-series data reveals a great deal of critical information about the dynamics of the processes that the data was collected from and can help control these processes.

If the magnitude of the seasonal changes does not increase or decrease in time, such seasonality is called additive, and Eq. (2.4.3) can be used with it "as is." When the magnitude of seasonal changes increases or decreases consistently, then it is called multiplicative seasonality, and the modified form of Eq. (2.4.3) needs to be applied to such data.

Seasonality and Trend Detection

It is not the purpose of this book to focus on seasonality and trend detection methods, but cluster-based seasonality analysis [GILG2010(1)], Fourier analysis, autocorrelation, and other techniques have proven themselves to be very useful in the field, leading to great improvements in the quality of the forecasts. This area, as well as its cousin change-point detection, is very rapidly growing areas in statistics, and one should always keep an eye on new developments there.

2.4.6 ARIMA Forecasting

"ARIMA" stands for "AutoRegressive Integrated Moving Average." It too is a moving-average technique, where the seasonality and the trend are detected automatically via time-series analysis. It is also known in time-series analysis as the Box-Jenkins forecasting method.

"AR"
Instead of applying an exponentially changing weight to the data points, ARIMA seeks to build a linear regression of each point's value to all of its predecessors and uses the coefficient of such model as the weights applied to each data point:

$$Y_t = c + \sum_{i=1}^{p} \varphi_i * X_{t-i}$$

The p is the number of degrees of freedom for the "AR" element of the ARIMA.

"I"
One problem with autoregressive models is they require the time series to be stationary (mean and variance to be "close enough" to constant). In the real world, it is not the case. One way to deal with it is to "difference" the time series:

$$X'_{\mathrm{t}} = X_{i+1} - X_i$$

The number of times it takes to make a time series stationary is usually designated as d and is the parameter of the "I" part of ARIMA.

"MA"
The "MA" (moving average) part works the same way as the already described SMA. Its parameter is usually designated as q and implies the width of the moving-average window.

Putting It All Together for ARIMA
When we apply ARIMA to the time series we have been examining in this section (Fig. 2.16), we have to be careful as to what values for (p, d, q) parameters we use. The parameters for most forecasting problems stay within the range of [0…2] (Fig. 2.21).

Let us explore the variety of ARIMA forecasts with different parameters.

2.4.7 Selection of Forecasting Model

The best overview of methods used in selecting a forecasting model is in [MAKR1998].

The model is usually selected based on how well it fits the data. The most common general strategy is similar to the approach used in the well-known, yet

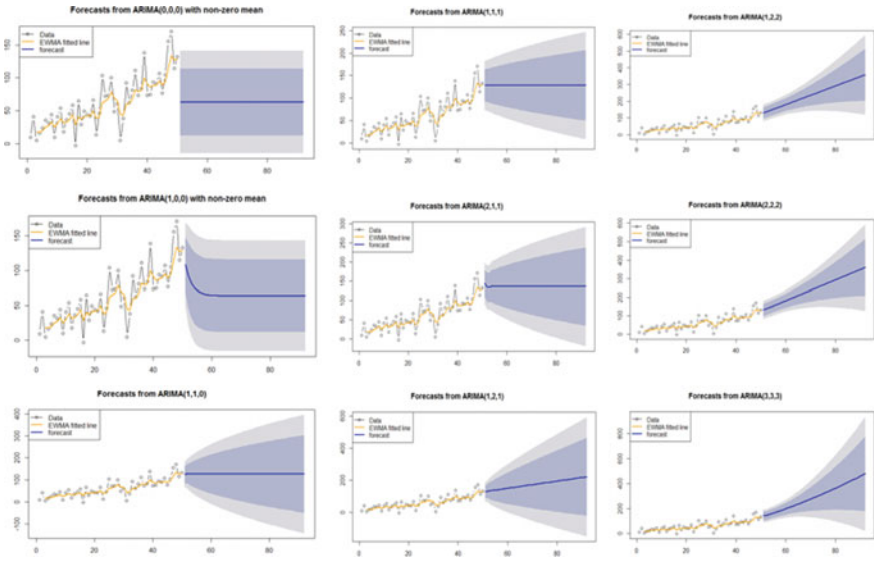

Fig. 2.21 ARIMA forecasts with different parameters

little understood, R^2 metric: compare model prediction with the data in every point of historical data and formulate the metric based on this comparison. Other approaches, such as out-of-sample testing, are used as well.

One of the problems is the variety of model quality evaluation methods, and finding ways to deal with it is one of the frontiers in TSA.

A weighted-sum synthetic approach to formulating the TSA model score has been successfully implemented, e.g., by [GILG2010(2)].

Regardless of what is chosen to be the model quality metric, two basic approaches are now in use in the predictive-analytics world: (i) select the model that fits best and (ii) use a so-called ensemble method. The latter relies on a Delphi approach, where several models are applied, analyzed, and scored, and then, their predictions are averaged with weights assigned based on their scores.

2.5 Regression

> … all models are wrong, but some are useful.
>
> —G.E.P. Box

Regression is another important tool in the network analyst's toolbox. It is the link between exploratory and predictive data analytics. As part of the exploratory data analysis (EDA), we always want to see whether we can find any correlations among the variables.

For example, if we see diurnal variations in throughput, we have an understanding that it is most likely driven by the number of user elements and their demand on the network.

This leads to several interesting observations: (i) We can forecast throughput using TSA techniques; (ii) we can also forecast throughput using business metrics (UE count and per-user throughput); (iii) if indeed there is a correlation between these two business metrics, it becomes a function of two variables: user count and per-user throughput.

In turn, total throughput is a nonlinear function of the number of users (see Sect. 2.3, Queueing Theory), which makes a seemingly simple task of predicting total throughput on a network a daunting one, due to all the interdependencies.

2.5.1 A Few Words on Terminology

The term "linear" is overloaded when it comes to regression. Indeed, there is an "official" mathematical definition of the term: http://en.wikipedia.org/wiki/Linearity, but this definition implies that an equation of the form

$$y = a * x + b, \tag{2.5.1}$$

(a, b = parameters; x, y = variables) is already nonlinear: the Homogeneity of Degree 1 is not true, unless $b = 0$:

$$m * (\alpha * x) + b = \alpha * m * x + b \neq \alpha * (m * x + b) = \alpha * m * x + \alpha * b$$

Nevertheless, when describing a regression line of the form (2.5.1), we call it linear, the reason being that its right-hand side is a linear polynome *in parameters*. By this definition, a quadratic, cubic, and any higher-degree polynomial function of x of the form $y = a_0 + a_1 * x + a_2 * x^2 + \cdots + a_n * x^n$ will be linear for all intents and purposes of regression: Each parameter $a_0, a_1, \ldots, a_n$ is entered in this polynomial as linear elements.

2.5.2 Linearizable Relationships

The "Big 6" of most commonly used linearizable relationships are as follows:

1. Straight-line relationship: $y = a_0 + a_1 * x$
2. Quadratic relationship: $y = a_0 + a_1 * x + a_2 * x^2$
3. Exponential relationship: $y = a_0 * e^{a_1 * x}$
4. Logarithmic relationship: $y = a_0 + a_1 * \ln(x)$

5. Power relationship: $y = a_0 * x^{a_1}$
6. Hyperbolic relationship: $y = \frac{a_0}{a_1 + a_2 * x}$

(Polynomials of higher degree than quadratic are very rarely used in regression analysis: Occam's razor demands that models should be as simple as possible, and cubic parabola and above rarely meet that requirement.)

We call these 6 linearizable because by simple substitution, we can convert them to linear:

1. Straight-line relationship: $y = a_0 + a_1 * x$:
   (a) no need to change anything
2. Quadratic relationship: $y = a_0 + a_1 * x + a_2 * x^2$:
   (a) Substitute $x_1 = x; x_2 = x^2$
   (b) $y = a_0 + a_1 * x_1 + a_2 * x_2$
3. Exponential relationship: $y = a_0 * \mathrm{e}^{a_1 * x}$:
   (a) Take In of both sides
   (b) Substitute $y_1 = \ln(y)$
   (c) $y_1 = \ln(a_0 * \mathrm{e}^{a_1 * x}) = \ln(a_0) + \ln(\mathrm{e}^{a_1 * x}) = b_0 + a_1 * x$
      i. $b_0 = \ln(a_0)$
4. Logarithmic relationship: $y = a_0 + a_1 * \ln(x)$:
   (a) Take exp of both sides
   (b) Substitute $y_1 = \mathrm{e}^y$
   (c) $y_1 = e^{a_0 + a_1 * \ln(x)} = a_0 + a_1 * x$
5. Power relationship: $y = a_0 * x^{a_1}$:
   (a) Take In of both sides
   (b) Substitute $y_1 = \ln(y); x_1 = \ln(x)$
   (c) $\ln(y) = \ln(a_0) + a1 * \ln(x) \Leftrightarrow y_1 = b_0 + a_1 * x_1$
      i. $b_0 = \ln(a_0)$
6. Hyperbolic relationship: $y = \frac{a_0}{a_1 + a_2 * x}$
   (a) Divide numerator and denominator by a_0: $y = \frac{1}{b_1 + b_2 * x}$:
   (b) Take the inverse of both sides: $\frac{1}{y} = b_1 + b_2 * x$
   (c) Substitute $y_1 = \frac{1}{y}$
   (d) Finally, $y_1 = b_1 + b_2 * x$

2.5.3 The Main Idea Behind Regression

In Fig. 2.22, we see a scatter plot around the line $y = 5 + 1.5 * x$.

The scatter (jitter/noise) around the solid black line is the stochastic (i.e., random) component of the data. In the real world, we usually know the nature of the relationship between the X and the Y, but do not know the equation of the solid black line; so we try to approximate (infer) it from the data. We want to draw a line that will minimize the error of such approximation.

Several measures of error have been proposed, the most prominent one, and the oldest one of all, being the sum of squared errors. We draw a line

$$y' = a_0 + a_1 * x: \quad (a_0, a_1) = \arg\min\left[\sum (y - y')^2\right] \tag{2.5.2}$$

In Eq. (2.5.2), $y' =$ fitted data; $y =$ measured data.

This form of regression has got the name of *Least-Squares regression*; sometimes, it is called *Ordinary Least-Squares (OLS) Regression* and is what is commonly referred to as *linear regression.*

By applying the transformations outlined in Sect. 2.5.2, and expanding it to a multivariate form (hint: the quadratic relationship has been converted to a bivariate form), we can model a wide class of relationships between the X and the Y.

The X and the Y are usually referred to as independent and dependent variables, respectively. Often in multivariate problems, the *Xs* are called "covariates," and in machine-learning applications, the *Xs* are usually referred to as "features."

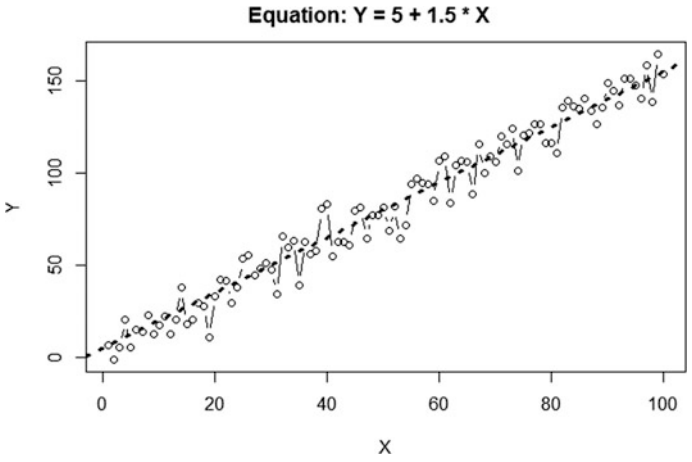

Fig. 2.22 Main idea behind regression

2.5.4 Solving Eq. (2.5.2)

To solve Eq. (2.5.2) is to fit a regression line into the data. Mathematically, it is done by solving a linear matrix equation which is outside the scope of this book. Interested readers are referred to any text in basic statistics. We are focused on the applied aspects of regression.

For the scenario shown in Fig. 2.22, the best fit we could get with OLS regression is shown in Fig. 2.23.

We see that the equation is off.

The intercept for the best-fitted model is $a_0 = 2.69$, while the coefficient is $a_1 = 1.53$. To verify that this error is induced by the noise, we can fit a regression line into the data produced by simply generating the data using the equation $Y = 5 + 1.5 * X$, and such regression's parameters will be exactly $a_0 = 5.00$ and $a_1 = 1.50$.

This error is what forces the fitted line (the red dashed line in Fig. 2.23) to be off the black solid line there.

Caution

Do not try to predict the line in the reverse order: A regression built for $Y(X)$ will not necessarily work to predict $X(Y)$, even with the algebra done correctly. The reason for that is in the way OLS parameters are computed and is a consequence of the laws of matrix multiplication.

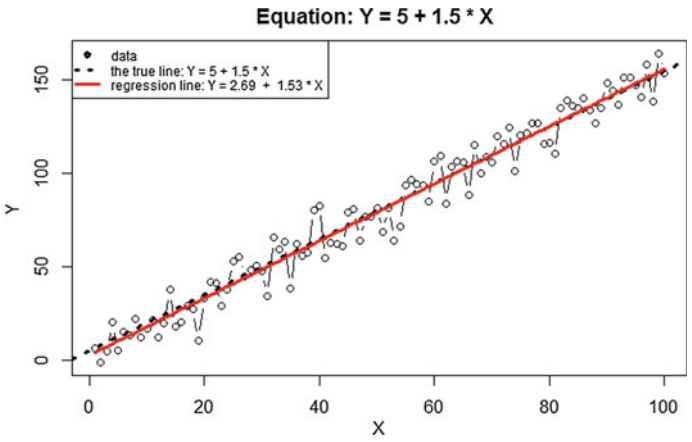

Fig. 2.23 Regression line (*red*) drawn through the data

2.5.5 Goodness of Fit

R-Squared

The most familiar to all goodness-of-fit estimator is the R^2. Created as a nonnegative version of ρ-Pearson's correlation factor, R^2 represents the proportion of the data variance that has been explained by the model. For more mathematical details, interested readers are referred to, e.g., [MILT2002]. Being a square of Pearson's correlation factor, it has some properties that make it preferred over the ρ. Namely, Pearson's ρ is formulated so as to indicate the directionality of the correlation ($\rho > 0$ if Y increases with X, and $\rho < 0$ if Y decreases as X grows). The R^2 eliminates this issue, making it better suited as a metric of model quality.

One of the less known properties of R^2 is that it is a central measure of a random variable that is not normally distributed. Ronald Fisher, however, developed a transformation, which he called the z-transform, which allowed comparing models by creating a normally distributed random variable with mean and variance defined by the R^2 and the number of data points that were used in the regression. Most of the standard (parametric) statistical tests are designed for normal variables, which allows one to compare models based on the z-transform R^2.

Information Criteria

In the 1950s, information theory was on the rise. The concept of entropy as the measure of chaos was adopted from physics and finally put to good use by Claude Shannon, John Von Neumann, and Norbert Wiener. The approach to measurement of goodness of fit from the information-theory side is beautiful: to view regression from the standpoint of its contribution to the reduction in the chaos of our perception of the world.

From this approach, the Akaike and the Bayesian Information Criteria (AIC and BIC, respectively) emerged. A model is an approximation, and therefore, it is guaranteed to lose some information (see Fig. 2.24) about the world it describes. If we have two models, A and B, then the AIC and the BIC measure how much more information is lost by applying model A than by applying model B. Consequently, we choose the model that loses less information. The difference is in the way they account for the number of points in the independent variable that are used in building the regression.

2.5.6 Model Competition

By using the goodness-of-fit criteria and/or their combination (e.g., [GILG2010]), we can identify the model that would be the best fit for the data.

For example, consider the data shown in Fig. 2.24.

We have a quadratic relationship between the two data sets (Y_q and X). We will compare two regression lines: a straight line and quadratic parabola.

First, let us look at the straight line (Fig. 2.25).

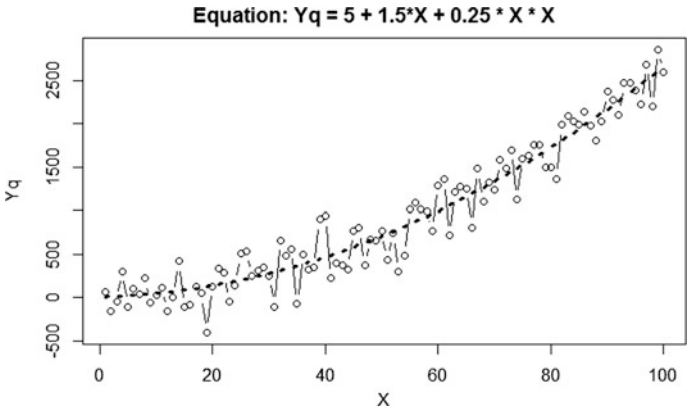

Fig. 2.24 An illustration

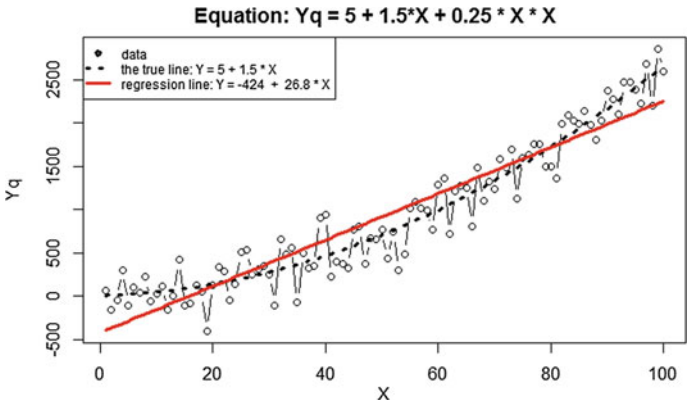

Fig. 2.25 Linear model

This line does show the direction of the correlation and fits the data fairly well; however, the $R^2 = 0.87$—not a very good fit (could be better); 13% of the variance has remained unexplained. The AIC for this model is 1420.

Next, we apply the quadratic model (Fig. 2.26).

The blue line seems to follow the data much closer. The $R^2 = 0.93$—a better fit is confirmed; only 7% of the variance has remained unexplained. The AIC for this model is 1367.

In this particular case, the R^2 and the *AIC* concur that the quadratic model does indeed provide a better fit to the data.

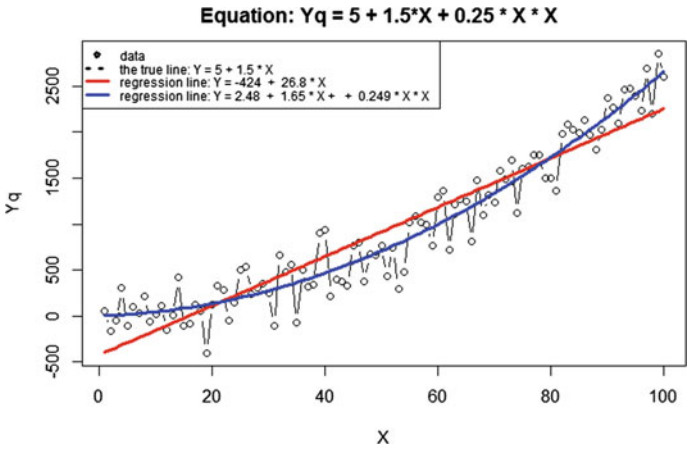

Fig. 2.26 Linear (*red*) and quadratic (*blue*) models

2.5.7 *Analysis of Residuals*

Visual

The goal of Least-Squares regression is to draw a line through the data, such that the residuals (observed−predicted) are random, with the distribution centered at zero.

In Fig. 2.26, we see that the quadratic line is a better fit than the straight line. Its residuals look as shown in Fig. 2.27a. Residuals from the straight-line regression are shown in Fig. 2.27b.

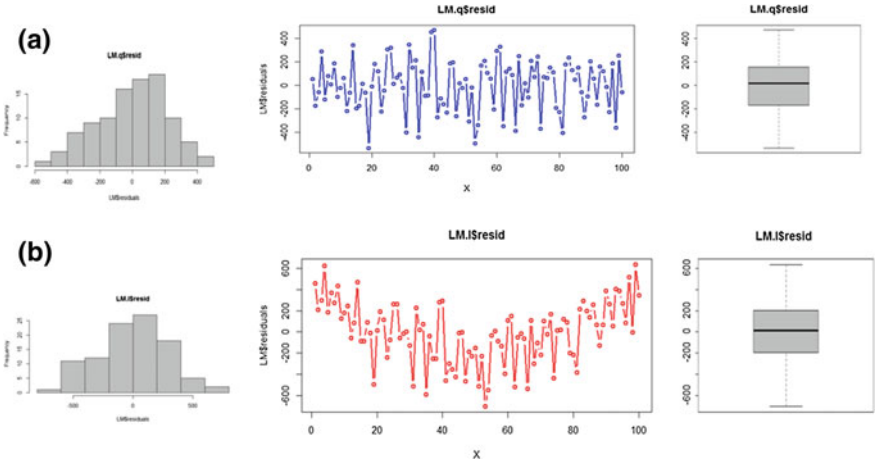

Fig. 2.27 Residuals from quadratic (**a**) and straight-line (**b**) regression

We see from Fig. 2.27 that the distribution of residuals, in and of itself, is not a sufficient signal on the quality of the regression. Other parameters need to be considered as well, such as latent correlation of residuals with independent variable(s).

Correlation Test

Correlation between residuals and the independent variable(s) can be performed via (i) the standard Pearson's correlation coefficient, which answers the question whether there is a *linear* relationship between the two data sets, (ii) the Spearman's correlation parameter, which answers the question whether there is any *monotonic* (going only up or going only down) relationship between the two data sets, or (iii) the Kendall's correlation parameter, which is not really applicable to analysis of residuals.

Sadly, neither one of the two will be adequate for answering the question whether Fig. 2.27b shows a residual correlation with the independent variable: This relationship is neither linear nor monotonic.

Yet we need to know whether the model is bad.

One way to do it is to check the correlation of the first derivative ($\frac{\Delta \text{Residuals}}{\Delta X}$) with the independent variable. If the line is quadratic, then the first derivative will be a straight line, and we will accurately capture the correlation if it is present.

This does look like a flat line; however, the formal correlation test tells us that we are still not out of the woods, the reason being the variance in the data: The signal-to-noise ratio is too low.

The other method involves splitting the independent variable into halves and running the Spearman's correlation test on both sides. If one is going up, while the other is going down, it means that residuals have a unimodal correlation.

In the scenario we are considering (Fig. 2.28), this method turned out to be beneficial. Correlation test for the first half returned.

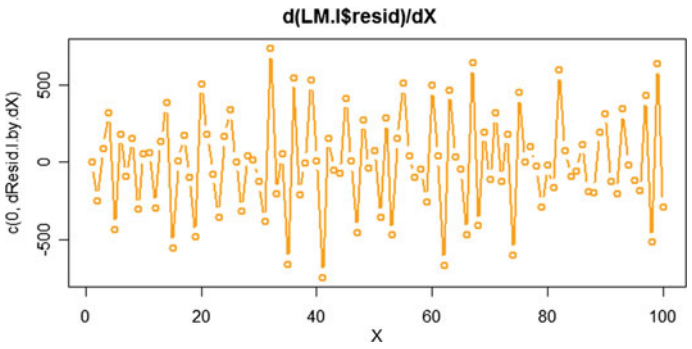

Fig. 2.28 First derivative of the residuals from straight-line regression

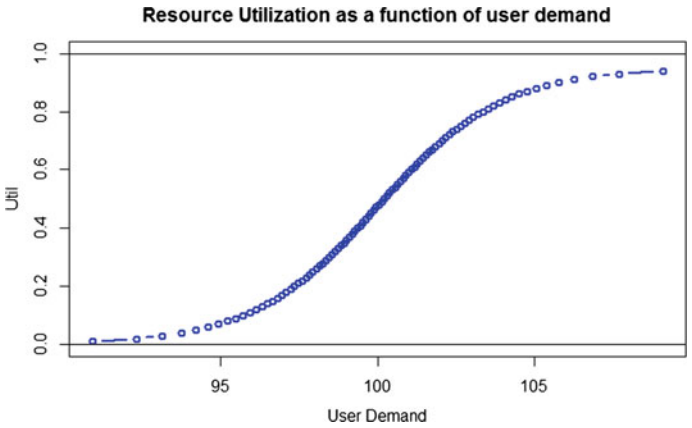

Fig. 2.29 Resource utilization is typically characterized by the sigmoid curve

```
> cor.test (first.half, X1, method = "spearman")
    Spearman's rank correlation rho
data: first.half and X1
S = 33668, p-value = 3.174e-06
alternative hypothesis: true rho is not equal to 0
sample estimates:
     rho
-0.6167107
```

For the second half:

```
> cor.test (second.half, X2, method = "spearman")
    Spearman's rank correlation rho
data: second.half and X2
S = 7212, p-value = 1.779e-07
alternative hypothesis: true rho is not equal to 0
sample estimates:
    rho
0.6736652
```

The product of the two correlation coefficients is $\rho_1 * \rho_2 = -0.6167 * 0.6737 = -0.4161 < 0$.

We have correctly captured the fact that the residuals hit a low point near the middle of the range of the independent variable. By contrast, for the quadratic line, not only are the correlations not very likely (the p-values are high), but also the correlation parameters for both halves of the range of the independent variable are on the same side of zero:

```
> cor.test (first.half.q, X1, method = "spearman")
    Spearman's rank correlation rho
data: first.half.q and X1
S = 20302, p-value = 0.8623
alternative hypothesis: true rho is not equal to 0
sample estimates:
     rho
0.02511405


> cor.test (second.half.q, X2, method = "spearman")
    Spearman's rank correlation rho
data: second.half.q and X2
S = 21068, p-value = 0.7442
alternative hypothesis: true rho is not equal to 0
sample estimates:
     rho
0.04669683


> cor(first.half.q, X1) * cor (second.half.q, X2)
[1] 0.001776423
```

Based on this analysis, despite the preference we should always give to the more simple models, given the choice between straight-line and quadratic relationships, we choose the quadratic model for the data in Fig. 2.27.

2.5.8 *Advanced Regression Methods*

Detailed coverage of advanced topics on regression is outside the scope of this book. Interested readers are referred to specialized texts covering in detail nonlinear regression; logistic regression; the general linear model (GLM), and other methods of regression. Majority of statistical and data mining software products have libraries that implement such methods. However, we will say a few words here about each of them.

Nonlinear Regression

Nonlinear regression is used when we know the form of the equation, and it is not one of the "Big 6." In such cases, iterative methods are used to compute the coefficients.

Generalized Linear Model (GLM)

The GLM is used for a wide variety of scenarios where the distribution of the residuals cannot be normal (Gaussian). In such cases, the GLM still tries to draw a line through the mean of the distribution; however, the mean will not necessarily correspond to the central point in the distribution.

Logistic Regression

A very special, and widely used, case of GLM is the logistic regression. It is so prominent in predictive analytics that most statistical software products have it implemented as a separate method. Logistic regression is used when we want to determine the probability [or any other variable bound between two hard limits (typically 0 and 1, but that can be scaled to any other pair of limits) as a function of some independent variable(s)]. In the IT world in general, and in LTE networks in particular, utilization (e.g., CPU, memory, bandwidth) falls into that category: it can be viewed as the probability that at any moment in time the system will be in the busy state, which makes it important to discuss logistic regression in this book.

We will also use it to illustrate the need to reverse all transformations and substitutions that were done when linearizing the regression equations.

Utilization of a live (operational) resource can never be greater than 1 and less than 0, but it can approach these limits asymptotically. This imposes a very distinct shape on the curve expressing utilization as the function of, e.g., user demand (Fig. 2.26).

The sigmoid (aka logistic) curve is the 7th of the "Big 6" linearizable curves. Indeed, if we consider that utilization is probability of being busy, we can compute the odds of being busy:

$$\text{Odds}_{\text{busy}} = \frac{\text{Util}}{1 - \text{Util}} \tag{2.5.3}$$

That, however, will result in an exponential-looking curve (Fig. 2.30).

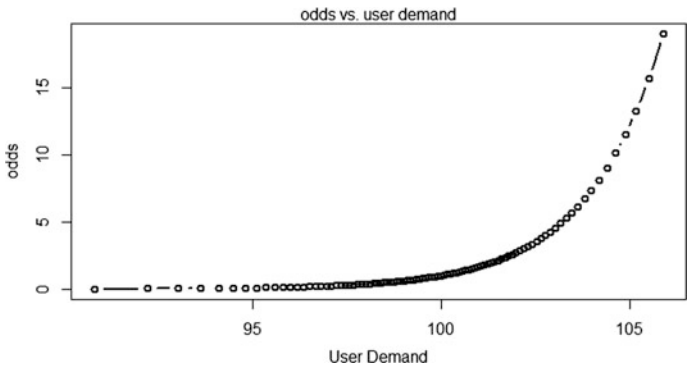

Fig. 2.30 Odds of seeing the resource in a busy state as a function of user demand

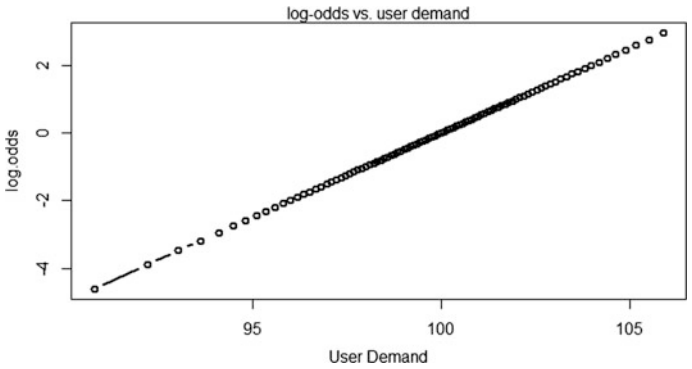

Fig. 2.31 Log-Odds of seeing the resource in a busy state as a function of user demand

One final operation—taking the natural logarithm of the odds—will result in a straight line (Fig. 2.31):

$$\text{log-Odds}_{\text{busy}} = \ln\left(\text{Odds}_{\text{busy}}\right) = \ln\frac{\text{Util}}{1-\text{Util}} \tag{2.5.4}$$

One final step is to reverse the calculations to go from linear model prediction for $\text{log-Odds}_{\text{busy}}$ to Util in Fig. 2.29.

$$\text{prob}_{\text{busy}} = \frac{e^{\text{log}-\text{Odds}_{\text{busy}}}}{1+e^{\text{log}-\text{Odds}_{\text{busy}}}} \tag{2.5.5}$$

Equation (2.5.5.) was derived from Eq. (2.5.4) by simple algebraic transformations (Fig. 2.32).

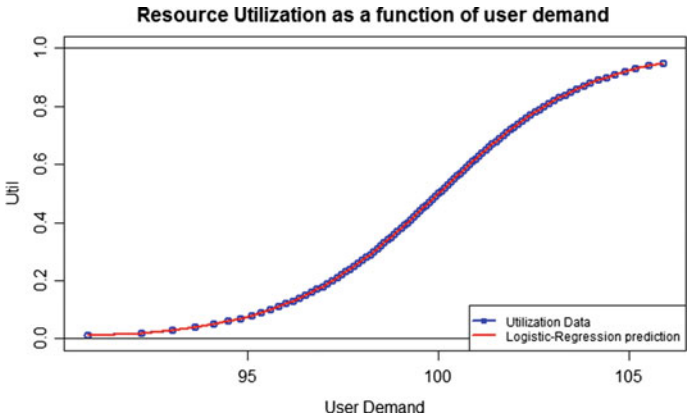

Fig. 2.32 The final regression fit

In performance and fault analysis, sometimes the prediction line ends up above the data. This happens as an artifact of applying Eq. (2.5.5): In effect, we are scaling the range of utilization to the [0, 1] range. If it happens, it is a strong indicator of hidden saturation (congestion) on a related resource.

Quantile Regression

The quantile regression [KNKR2001] approach is relatively new. It was first formulated in 1987; however, the computing power was not up to par, making the software implementing QR sluggish at best. Fifteen years later, the quantreg package was implemented for *R*, and another few years after that, quantile regression was implemented in SAS. Today, it is widely used everywhere, especially in social sciences, where families in the socioeconomic standings react differently to different stimuli. It is useful in building predictive models for LTE networks, e.g., where technically different user elements share a bandwidth. This will lead to a variety of usage patterns, some of which can even be used to identify the user's equipment.

Quantile regression is highly efficient when we do not know, and cannot guess, the type of distribution of the dependent variable.

It can also be used to dynamically detect outliers in correlated data: Indeed, if we can draw regression lines through the 25th and 75th percentiles (first and third quartiles) of the dependent variable, compute the IQR (see section on Outliers in this chapter) at every "vertical" slice and compute the trajectories of high and low whiskers (high and low outlier boundaries).

Other examples of using quantile regression include capacity planning, busy-hour-traffic analysis, and Predictive SPC [FERR2014].

2.5.9 Do We Have to Compete?

Sometimes, it is hard to choose the "right" model, even when formally the parameters are pointing at one. When formal competition yields a tie, it becomes even more complicated. When that happens, we need to either choose the model based on some heuristics or use the so-called *ensemble approach*.

With ensemble, or Delphi (in honor of the Delphi Oracle in Ancient Greece) approach, we combine predictions of two or more models into one. For some models, it can be easily done by simple formulae; for others, more advance techniques, such as bootstrapping and jackknifing, have to be used.

Ordinary Least-Squares (OLS) Regression

As we discussed, OLS regression's residuals are expected to be normally distributed and centered at zero. This provides an easy way to compute the distribution of the ensemble. Indeed, since all models in the ensemble are independent, and so are their predictions, we can compute the mean at all values of independent variable (s) as the sum of means, and the variances of the ensemble prediction as the sum of variances, of the models that were used in the ensemble:

$$\forall \boldsymbol{X}: \quad \mu_{\boldsymbol{X}} = \sum_{m=1}^{M} (\mu_{\boldsymbol{X}})_m \tag{2.5.6a}$$

And

$$\forall \boldsymbol{X}: \quad \mathrm{Var}_{\boldsymbol{X}} = \sum_{m=1}^{M} (\mathrm{Var}_{\boldsymbol{X}})_m \tag{2.5.6b}$$

In Eqs. (2.5.6a) and (2.5.6b),

$\boldsymbol{X}$	is the independent variable (or a combination of covariates)
M	is the number of models in the ensemble
$\mathrm{Var}_{\boldsymbol{X}} = \sigma^2$	is the variance of residuals predicted at $\boldsymbol{X}$

Because (see Section 2.5.7) we do not want to use a model that leaves residuals cor-related with X; Eq. (2.5.6b) can be simplified to:

$$\forall \boldsymbol{X}: \quad \mathrm{Var}_{\boldsymbol{X}} = \sum_{m=1}^{M} \mathrm{Var}_m \tag{2.5.6c}$$

Note: Model predictions will appear to be correlated. This is expected: We are modeling the same process using different models. However, the models are independent; therefore, Eqs. (2.5.3a–c) will work.

Non-OLS Regression

For non-OLS regression, when the distribution of residuals is generally non-normal (see Sects. 2.1–2.3), ensemble methods will involve more complicated statistical techniques, falling under the general classification as Monte Carlo analysis. They are outside the scope of this book. A simple, down-to-earth, technique that allows the analyst to avoid using Monte Carlo and yet get decent results involves resampling. The idea is to fulfill the requirements of the Central Limit Theorem: take multiple independent samples at random from the values predicted by each model. If enough such samples are taken, their means will converge to a normal distribution, and then, Eqs. (2.5.6a)–(2.5.6c) will be usable for computing the ensemble prediction.

2.6 Clustering

In LTE analytics, it is often important to break data into natural groups.

One of the most immediate use cases involves bundling curves into clusters. Another use case is dealing with geographic clustering (e.g., identifying natural groupings of UEs to determine RAN metro boundaries).

We shall start with the first use case, as it builds on the material we introduced in this chapter in Sects. 2.4 and 2.5.

2.6.1 Bundling the Curves

In Fig. 2.33, we show a group of data simulated to reflect 500 days of bandwidth utilization data for five network links.

We see interesting patterns in data: In the early days (May through August, 2010), the five links were ramping up together, reached what appears like saturation at ∼90%. This was followed by a drop in utilization of L4 and L2 (perhaps they were taken out of traffic for a few months). The other three links (L0, L1, and L3) saw a sharp drop in utilization around the same time. Such drops are usually associated with an increase in capacity.

L1 started ramping up gradually, while the other two (L1 and L3) stayed at the same level of utilization to which they came after the drop.

Finally, in February 2011, links L2 and L4 were put back into operation.

This is all a very basic analysis that can be performed visually, but when dealing with hundreds, even thousands, of links, we want an automatic system to perform it for us. We need a mathematically soun way to automatically group data into bundles.

K-means

The K-means method of clustering is the simplest yet surprisingly powerful. K "Centers" are put at random on the field (y-axis), and all data points are assigned to each of the K clusters so as to minimize their distance to the center. The new

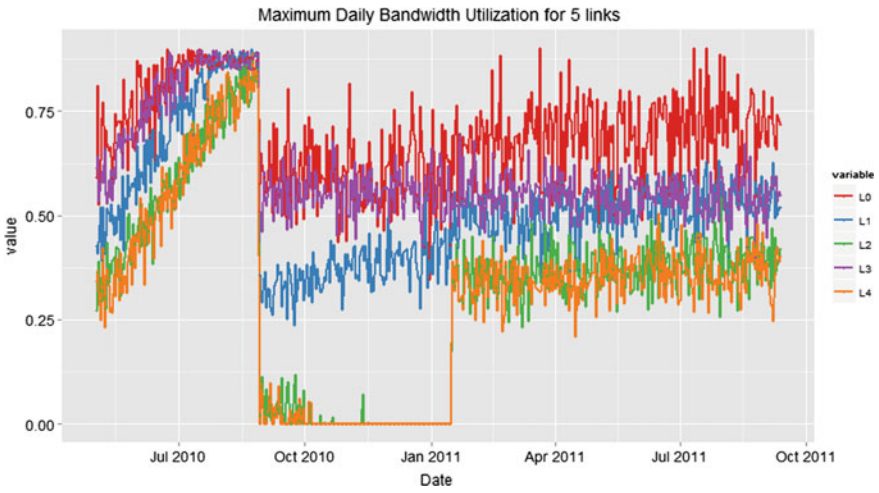

Fig. 2.33 Maximum bandwidth utilization for a group of 5 links in a WAN region

centroid (mean) is calculated for each cluster, and the data points are reassigned. The iterations continue until no more improvements to the distances to the centroids can be made. Due to the way distances are calculated (typically sum of squares of Euclidean distances), this method is equivalent to moving K variable-width Gaussian bell curves (surfaces if the clustering is multi-dimensional) until the narrowest possible bell curves are achieved.

Applied to the data shown in Fig. 2.33, K-means clustering produces decent results, which, however, fail to detect the patterns in data trajectories.

Advantages of K-means Clustering
It is simple, powerful, and versatile.

Downsides of K-means
K-means method groups data based on their value, without consideration for trending and other patterns in data. Consequently, cluster overlaps are to be expected when this method is applied to time-series data. We see these overlaps in Fig. 2.34.

The clustering shown in Fig. 2.34 better reflects what we think we see from the data, but both groupings are wrong, and both groupings are right.

Model-Based Clustering
Model-based clustering (commonly abbreviated as MBC) is an extension of K-means for multivariate clustering, where the bell surfaces (Gaussian clouds) are extended along the primary axes of the data. For technical details, please see [FRAF2002] and [FRMS2012].

It is a highly sophisticated algorithm, fitting different shapes of Gaussian clouds into data and using the BIC (see Sect. 2.5) to decide when to stop adding clusters (detect the point of diminishing returns in information gain).

The particular data set that we have been looking at is too big for MBC on a regular laptop, even with 8 GB of memory, however. The algorithm is very memory-intensive. To alleviate the problem, we are going to aggregate the data for

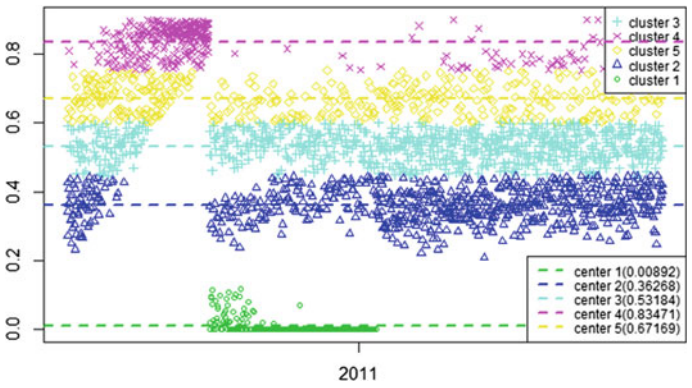

Fig. 2.34 K-means clustering at $K = 5$

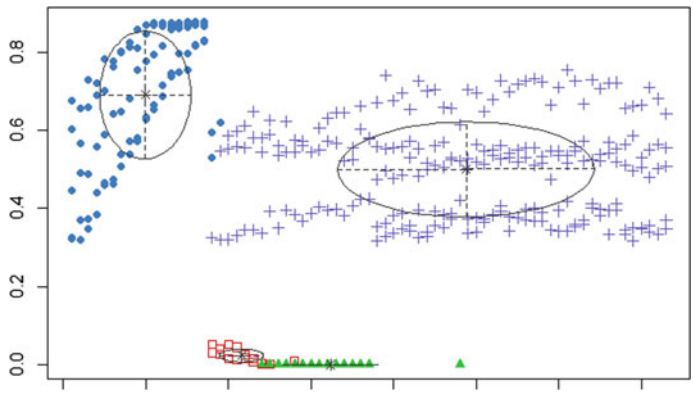

Fig. 2.35 Clustering by model-based method

each set into weekly averages. The reason we choose averages is because MBC relies on a Gaussian Mix Model; so using averages actually helps in this case by fulfilling the conditions of the Central Limit Theorem. Doing so also reduces the data by a factor of 7. It still takes a couple of minutes, but the result is very informative (Fig. 2.35).

The ellipses that MBC method has added show the Gaussian kernels that characterize the clusters. The colors correspond to the clusters. We see how different sets from Fig. 2.33 group naturally with other sets at different times, which allows us to see the dynamics of the data. This allows us to build an automatic performance monitoring system that will answer important performance-related questions.

The classification uncertainty plot (Fig. 2.36) allows the analyst to see regions where the model-based classifier had higher and lower confidence of the cluster assignment.

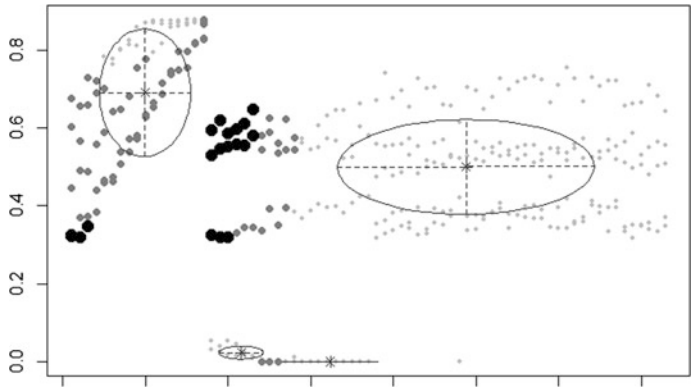

Fig. 2.36 Classification uncertainty chart

Here, point size and darkness indicate the degree of uncertainty in grouping the data the way MBC did.

2.6.2 Geographic Clustering

With geographic clustering, the same approaches apply. Figure 2.37 shows a distribution of population in 3 urban centers.

The model-based clustering will provide the following plots.

The plots shown in Fig. 2.38 show the natural patterns in data, the overlap between the Gaussian kernels (uncertainty), and a contour plot showing the data density distribution. The cluster assignment (count and shape) is optimized based on the Bayesian Information Criterion (BIC). The process is illustrated in Fig. 2.39.

Here, the 3-letter combinations correspond to the shapes of the cluster kernels, and the term "components" corresponds to the clusters into which the data is grouped. For more details, please refer to the literature provided below.

2.6.3 Geographic Clustering of Signal

The method we have illustrated here assigns clusters based on the density of the points on the map (see Fig. 2.39). When we are dealing with signal distribution within each cluster, a different method has to be used, the DBSCAN. Its biggest advantage over the K-means and the MBC methods is that it allows identifying groupings of data with nonlinear separation boundaries. Its discussion is outside the scope of this chapter. R, python, SAS, SPSS, and other statistical tools all have

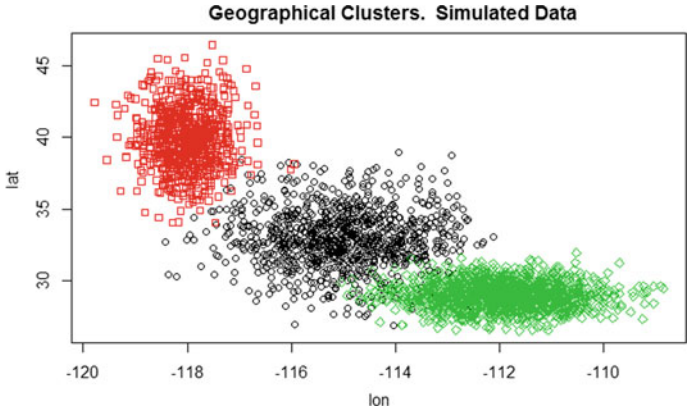

Fig. 2.37 The three urban centers

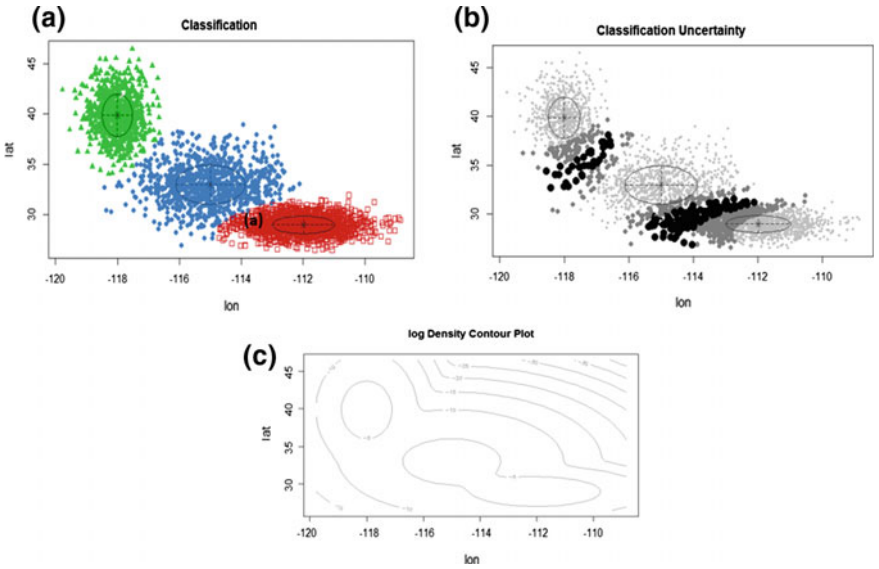

Fig. 2.38 The three plots provided by the model-based clustering

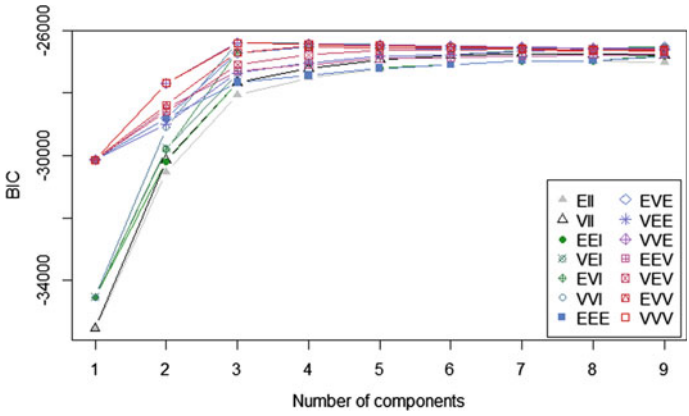

Fig. 2.39 An illustration of the cluster assignment process

implementations of K-means, MBC, DBSCAN, and other separation methods shipped either as part of the standard distributions or as additional libraries.

We have only touched upon the most basic use cases and tools for clustering. For more information, techniques, and tools, please see the literature referenced in this book.

2.7 Conclusion

We have covered the basics of machine learning and applied data analytics, with some examples from the LTE networking world. In Chap. 6, after the reader gets introduced to the key elements comprising the LTE network—EUTRAN, EPC, IP Backhaul, Metro, and Core, we revisit the topic, describing why advanced techniques are needed to adequately analyze LTE network performance as a whole and introducing such techniques for statistical process control, outlier detection, and queueing-system analysis.

References

[GUNT2007] Gunther NJ (2007) Guerrilla capacity planning: a tactical approach to planning for highly scalable applications and services. Springer, New York, Secaucus, NJ, USA © 2006. ISBN 3540261389

[CHDY2014] Choudhury J (2014) Parameter Estimation of asymptotically improved super-serial scalability law. In: Performance and capacity international conference by computer measurement group (CMG'14) (PDF)

[MAKR1998] Makridakis S, Wheelright SC, Hyndman RJ (1998) Forecasting: methods and applications, 3rd edn. Wiley. ISBN 978-0-471-53233-0

[MILT2002] Milton S, Arnold J (2002) Introduction to probability and statistics: principles and applications for engineering and the computing sciences, 4th edn. McGraw-Hill. ISBN-13 978-0072468366

[GILG2010(1)] Gilgur A, Perka M Computer Storage capacity forecasting system using cluster-based seasonality analysis. US Patent 7783510

[GILG2010(2)] Gilgur A, Perka M (2010) Forecast model quality index for computer storage capacity planning. US Patent 7788127. Filed 06/25/2007. Awarded 08/24/2010

[KNKR2001] Koenker R, Hallock KF (2001) Quantile regression. J Econ Perspect 15(4): 143–156

[FRAF2002] Fraley C, Raftery AE (2002) Model-based clustering, discriminant analysis and density estimation. J Am Stat Assoc 97:611–631

[FRMS2012] Fraley C, Raftery AE, Brendan Murphy T, Scrucca L (2012) mclust Version 4 for R: normal mixture modeling for model-based clustering, classification, and density estimation, technical report no. 597, Department of Statistics, University of Washington

[FERR2014] Ferrandiz J, Gilgur A (2014) Capacity planning for QoS - the journal of capacity management. A publication of the computer measurement group. Issue 135, Winter, 2014

http://www.springer.com/978-81-322-3719-8

Network Performance and Fault Analytics for LTE Wireless Service Providers

Kakadia, D.; Yang, J.; Gilgur, A.

2017, XVIII, 204 p. 146 illus., Hardcover
ISBN: 978-81-322-3719-8