Nearest Neighbour Algorithms for Forecasting Call Arrivals in Call Centers

Sandjai Bhulai, Wing Hong Kan, and Elena Marchiori
Vrije Universiteit Amsterdam
Faculty of Sciences
De Boelelaan 1081a
1081 HV Amsterdam
The Netherlands
E-mail: {sbhulai, wing, elena}@few.vu.nl

Abstract

In this paper we study a nearest neighbour algorithm for forecasting call arrivals to call centers. The algorithm does not require an underlying model for the arrival rates and it can be applied to historical data without pre-processing it. We show that this class of algorithms provides a more accurate forecast when compared to the conventional method that simply takes averages. The nearest neighbour algorithm with the Pearson correlation distance function is also able to take correlation structures, that are usually found in call center data, into account. Numerical experiments show that this algorithm provides smaller errors in the forecast and better staffing levels in call centers. The results can be used for a more flexible workforce management in call centers.

1 Introduction

A call center is a collection of resources providing a telephony interface between a service provider and its customers. The resources consist, among others, of customer service representatives, who talk to customers over the telephone. The service representatives are usually supported by elaborate information and communication (ICT) equipment. Due to the rapid growth of e-commerce, the contact between the service provider and its customers is additionally subject to other channels, such as e-mail and the Internet. This general interface between a service provider and its customers is now often referred to as a contact center.

Most organizations with customer contact have a call center nowadays, or hire specialized firms to handle their communications with customers through call centers. Current trends are towards an increase in economic scope and workforce (see Gans et al. [8]). Hence, there is an enormous financial interest in call centers that leads to the importance of efficient management of call centers. The efficiency relates to the efficient use of the workforce, since the costs in a call center are dominated by personnel costs.

The basis of efficient workforce management in call centers is the well-known Erlang-C model (see Erlang [7]). This queueing model, also referred to as an M/M/c model,

assumes that the process of incoming calls forms a Poisson process with parameter λ, that the service duration is exponentially distributed with parameter μ, and that there are c service representatives. Based on these assumptions, the model allows one to calculate the waiting time distribution under a given load $R = \lambda/\mu$ of the call center. The result is used to compute the minimum number of service representatives needed to meet a target service level β, i.e., the waiting time objective. Halfin and Whitt [13] suggest the square-root safety-staffing principle, recommending the number of representatives to be $c \approx R + \beta\sqrt{R}$, where R is the offered load and $\beta\sqrt{R}$ is safety staffing to compensate for stochastic variability.

The Erlang-C model is perhaps the simplest model in call center circles. For many applications, however, the model is an over-simplification due to the assumption of a constant arrival rate over the whole day. Common call center practice is to use the stationary independent period-by-period (SIPP) approximation (see Green et al. [12]). The SIPP approximation uses the average arrival rate over a period of 15 or 30 minutes, based on historical data, and the number of service representatives in that period as input to the stationary Erlang-C model to approximate performance in that period. The pointwise stationary approximation (PSA) (see Green and Kolesar [11]) is the limiting version of the SIPP approximation when the period length approaches zero.

The stationary models implicitly assume that the time required for the system to relax is small when compared to the length of the period. However, abrupt changes in the arrival rate, or overload situations during one or more periods lead to non-stationary behaviour that must be accounted for. Yoo [19] and Ingolfsson et al. [15] present methods to calculate the transient behaviour by numerically solving the Chapman-Kolmogorov forward equations for the time-varying $\mathrm{M}_t/\mathrm{M}/c_t$ queueing system. Yoo [19] and Ingolfsson et al. [14] approximate the continuously varying parameters with small, discrete intervals and use the uniformization method (see Grassmann [10]) to calculate the change in system occupancy from one period to the next. Experiments show that the method of randomization yields results that are close to the exact solutions in general.

While the non-stationary models perform well and can be used for workforce management, they assume that the overall arrival rate is known. In practice, the arrival rate is predicted from historical data and is not known with certainty in advance. The risk involved in ignoring this uncertainty can be substantial. A call center that is planning to operate at 95% utilization, can experience an actual utilization of 99.75% with exploding waiting times when the arrival rate turns out to be 5% higher than planned (see Gans et al. [8]). A natural way to deal with this uncertainty is to increase the safety staffing $\beta\sqrt{R}$, but it remains to be studied how much the increase exactly should be.

Call centers generate a vast amount of data from which the history of each call that enters the system can be reconstructed. Statistical data analysis on the records of individual calls reveals structural properties that can be used in determining the call arrival function. Brown et al. [4] have found that there is a significant correlation in call arrivals across different time periods in the same planning horizon. Moreover, the first few hours of a day often provide significant information about the call volumes for the remainder of the day. These findings motivate general arrival rate models such as those suggested by Avradimis et al. [2], Jongbloed and Koole [16], and Steckley et al. [18].

The rise in computational power and large call center databases bring forth new forecasting techniques that increase the accuracy of the forecasts. In this paper we present a K-nearest neighbour algorithm for forecasting the arrival rate function dynamically. The method is based on comparing arrival rate functions on different days with the observed pattern so far. The K arrival rate functions that are closest, with respect to some distance function, to the observed pattern are used to forecast the call arrival rate function for the rest of the planning horizon. The algorithm is able to take into account the correlations in call center data found by Brown et al. [4]. Moreover, the algorithm does not require a model and can be applied without pre-processing the historical data.

The resulting forecast of the K-nearest neighbour algorithm can be used to create more accurate calculations of the number of representatives that is needed to meet the service level (e.g., by using the non-stationary models). This leads to greater flexibility in the management of the workforce. Next to their job of handling calls, customer representatives also have administrative tasks, meetings, and trainings. These tasks could be re-scheduled or cancelled if the forecast shows a surge in calls that was unprecedented for. Representatives could also attend to other channels in periods that are predicted to be less busy. Similarly, inbound and outbound calls could be mixed in such periods (call blending), so that the workforce is used efficiently (see, e.g., Bhulai and Koole [3] and Gans and Zhou [9]).

The outline of the paper is as follows. In Section 2 we give the exact problem formulation. We then continue to present the K-nearest neighbour algorithm, that will be used to solve the problem, in Section 3. A case study with the results of applying this algorithm are presented in Section 4. Finally, Section 5 concludes the paper by summarizing the main results.

2 The Forecasting Problem

Consider a call center whose statistics are stored in a large database. Typically, the statistics contain a lot of information on each individual call, such as the starting time, the end time, the waiting time, the handling agent, and much more. In practice, many call centers store summarized historical data only, due to the historically high cost of maintaining and storing large databases even if these reasons are no longer prohibitive. Let us therefore assume that the data is aggregated over a period of length Δt minutes. Note that when Δt is sufficiently small, we get the information on individual calls back. For our purpose of forecasting the call arrival rate function, we concentrate on the number of calls in a period of length Δt.

The length of a period Δt together with the opening hours of the call center define n periods over the day, which we denote by $t_1, \ldots, t_n$. Assume that the database contains m records, i.e., data on aggregated call arrivals of m days. Then we can represent the database by an $m \times n$ matrix H, where entry $h_{d,i}$ represents the number of calls that occurred on day d in period t_i of length Δt for $d = 1, \ldots, m$ and $i = 1, \ldots, n$.

Data analysis (see, e.g., Brown et al. [4] and Steckley et al. [18]) shows that the shape of the call arrival rate function on a particular day of the week is usually the same over different weeks. In case of an unusual event, e.g., a holiday, the shape differs significantly.

Therefore, in practice, the data is cleaned first by removing records for weeks containing unusual events, and the matrix H is divided into submatrices $H^{(1)}, \ldots, H^{(7)}$ per day of the week. Conventional forecasting algorithms base their forecast on this data by taking the mean number of call arrivals for that period based on the historical values of that day. Thus, the forecast F_i^{CV} for the number of calls in period t_i on day j of the week is given by

$$F_i^{\text{CV}} = \frac{1}{k} \sum_{d=1}^{k} h_{d,i}^{(j)},$$

for $i = 1, \ldots, n$, where k is the number of records in the submatrix $H^{(j)}$.

The conventional algorithm ignores additional structure that is present in the call arrival pattern. From data analysis it is known that there is a significant correlation in call arrivals across different time periods in the same planning horizon. Typically, the first few hours of a day often provide significant information about the call volumes for the remainder of the day. The models suggested in Avradimis et al. [2], Jongbloed and Koole [16], and Steckley et al. [18] use this structure to determine the total number of calls over the day and distribute the calls according to the correlated structure over the different periods. These models, however, mostly focus on estimation rather than on prediction and also rely on the time-consuming data analysis and preparation to determine parameters of the models.

The discussion above shows that there is still a need for efficient computational algorithms that do not require the tedious data analysis and use the correlation structures in the call arrivals to yield accurate forecasts. Consequently, the algorithm should also be able to update the forecast as soon as new information on call arrivals is available. A possible way to do this is to compare arrival rate functions of different days with the observed pattern so far. The arrival rate functions that are closest, with respect to some distance function, to the observed pattern can then be used to forecast the call arrival rate function for the rest of the planning horizon. This leads to the class of nearest neighbour algorithms. The distance function can be used to capture the correlation structures in call arrivals. Moreover, the algorithm does not require a model and can be applied without pre-processing or analyzing the historical data. In the next section we explain a version of the K-nearest neighbour algorithm, that we will use to generate forecasts, in greater detail.

3 The K-Nearest Neighbour Algorithm

The K-nearest neighbour algorithm (see, e.g., Cover and Hart [5]) is a machine learning technique that belongs to the class of instance based learners. Given a training set, a similarity measure over patterns, and a number K, the algorithm predicts the output of a new instance pattern by combining (e.g., by means of weighted average) the known outputs of its K most similar patterns in the training set. The training data is stored in memory and only used at run time for predicting the output of new instances. Advantages of this technique are the (implicit) construction of a local model for each new instance pattern, and its robustness to the presence of noisy training patterns (cf., e.g., Mitchell [17]).

Nearest neighbour algorithms have been successfully applied to many pattern recognition problems, such as scene analysis (Duda and Hart [6]) and robot control (Atkeson et al. [1]).

Application of K-nearest neighbour algorithms to our forecasting problem in call centers translates into the prediction of the arrival rate function until the end of the planning horizon, based on matching K arrival rate functions to the pattern observed so far.

More formally, suppose that we have observed the call arrival rates $r_1, \ldots, r_x$ on a specific day in periods $t_1, \ldots, t_x$ where $x < n$. Let us call this information the reference trace $\vec{r}_x = (r_1, \ldots, r_x)$. The nearest neighbour algorithm compares the reference trace to historical data $\vec{h}_{d,x} = (h_{d,1}, \ldots, h_{d,x})$ from the matrix H for $d = 1, \ldots, m$. The comparison is based on a distance function $\mathrm{D}(\vec{r}_x, \vec{h}_{d,x})$, which is defined by a norm on the space of the traces. The K nearest traces with respect to the distance function D are used to generate a forecast for the arrival rates $\hat{r}_{x+1}, \ldots, \hat{r}_n$ in periods $t_{x+1}, \ldots, t_n$. The result depends on the value of K as well as on the choice of the distance measure D.

In the next subsections we will describe two distance functions, the Euclidean distance and the Pearson correlation distance, that we will use in the numerical experiments. The former distance function can be seen as the conventional forecasting algorithm when restricted to the submatrices $H^{(1)}, \ldots, H^{(7)}$. The latter distance function is our novel approach to forecasting the call arrival rate function. Each subsection will motivate the choice of the distance function, and describe how to generate forecasts.

Euclidean Distance (ED)

The Euclidean distance is the most widely used and the most natural distance function to use. It is defined by the Euclidean norm

$$\mathrm{ED}(\vec{r}_x, \vec{h}_{d,x}) = \Big[\sum_{i=1}^{x} (r_i - h_{d,i})^2 \Big]^{1/2}.$$

The effect of using this distance function in the nearest neighbour algorithm is that traces are considered to be near when the arrival rates in the historical data almost exactly match the observed arrival rates in the different periods. By selecting the K nearest traces, say on days $d_1, \ldots, d_K$, the forecast for period t_i is given by

$$F_i^{\mathrm{ED}} = \frac{1}{K} \big[h_{d_1,i} + \cdots + h_{d_K,i} \big],$$

for $i = x + 1, \ldots, n$.

Note that by restricting to a specific day j of the week, i.e., to the submatrix $H^{(j)}$, we get the conventional forecast F_i^{CV}. Also observe that the distance function is sensitive to trends and days with special events. Hence, this distance function actually requires cleaned historical data without trends and days with special events. Therefore, the forecast needs to be adjusted for these situations.

Pearson Correlation Distance (PD)
The Pearson Correlation distance is based on the correlation coefficient between two vectors. A correlation value close to zero denotes little similarity, whereas a value close to one signifies a lot of similarity. Hence, the definition of the distance function is given by

$$\begin{aligned}\mathrm{PD}(\vec{r}_x, \vec{h}_{d,x}) &= 1 - \left|\mathrm{correlation}(\vec{r}_x, \vec{h}_{d,x})\right| \\ &= 1 - \left| \frac{(x-1)\sum_{i=1}^{x}\left(r_i - \mathrm{mean}(\vec{r}_x)\right)\left(h_{d,i} - \mathrm{mean}(\vec{h}_{d,x})\right)}{\sum_{i=1}^{x}\left(r_i - \mathrm{mean}(\vec{r}_x)\right)^2 \sum_{i=1}^{x}\left(h_{d,i} - \mathrm{mean}(\vec{h}_{d,x})\right)^2} \right|,\end{aligned}$$

where $\mathrm{mean}(\vec{r}_x) = \frac{1}{x}\sum_{i=1}^{x} r_i$ and $\mathrm{mean}(\vec{h}_{d,x}) = \frac{1}{x}\sum_{i=1}^{x} h_{d,i}$. Note that the function looks at similarities in the shape of two traces rather than the exact values of the data. Consequently, this function answers the need to capture correlation structures in the call rates. Moreover, it does not require cleaned data, because it is less sensitive to trends and special events.

When the K nearest traces have been selected, say on days $d_1, \ldots, d_K$, the forecast for period t_i cannot be generated as in the case of the Euclidean distance. Since the distance function selects traces based on the shape, the offset of the trace for day d_j needs to be adjusted by

$$c_{d_j} = \frac{1}{x}\Big[\sum_{i=1}^{x}(r_i - h_{d_j,i})\Big].$$

Therefore, the forecast for period t_i is given by

$$F_i^{\mathrm{PD}} = \frac{1}{K}\left[(h_{d_1,i} + c_{d_1}) + \cdots + (h_{d_K,i} + c_{d_k})\right],$$

for $i = x+1, \ldots, n$.

In the next section we will use the K-nearest neighbour algorithm to forecast call arrival functions using these two distance functions based on real historical data.

4 Numerical Experiments

In this section we illustrate the nearest neighbour algorithm using real call center data of an Israeli bank. The call center data with documentation are freely available from `http://iew3.technion.ac.il/serveng/callcenterdata/`. The data contains records during a period of a year. The call center is staffed from 7 am to midnight from Sunday to Thursday, it closes at 2 pm on Friday, and reopens at 8 pm on Sunday.

In our experiments we take for practical staffing purposes Δt to be equal to 15 minutes. Thus, taking the opening hours on Sunday until Thursday into account, we have 68 periods for which the number of arriving calls are aggregated per 15 minutes. Given these 5 days, we therefore have 5×52 records, which are used to construct the matrix H and the submatrices $H^{(1)}, \ldots, H^{(5)}$.

In order to evaluate the quality of the forecasts produced by the nearest neighbour algorithm, we pick a reference day and assume that the call arrival function $\vec{r}_x$ up to period

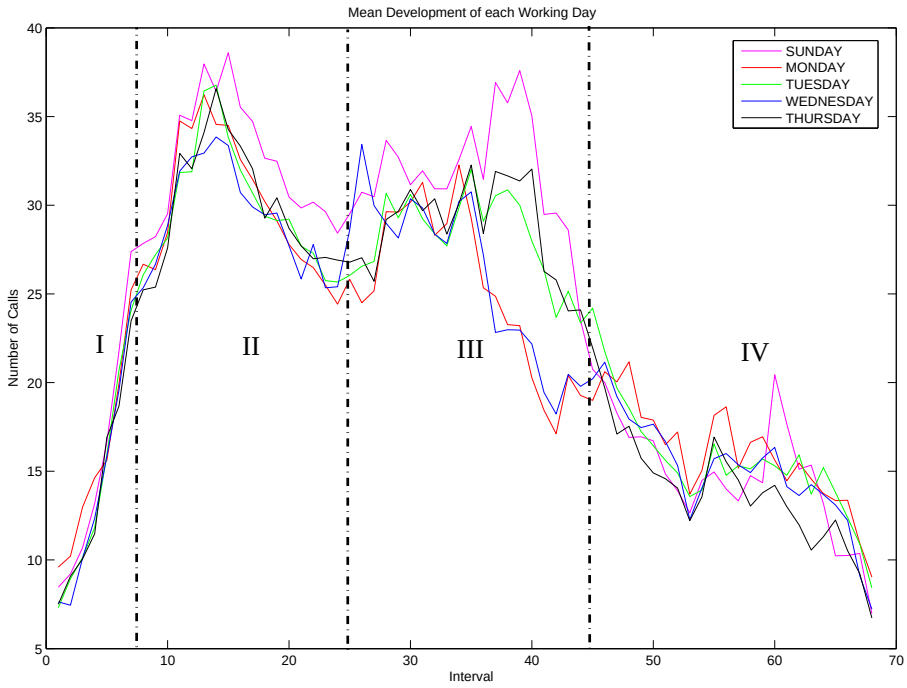

Figure 1: Forecast update moments over the day.

$x \in \{1, \ldots, 68\}$ is known. The nearest neighbour algorithm then selects K traces that are nearest to $\vec{r}_x$ and generates forecasts as explained in the previous section. Typically, historical data will be used, i.e., data only prior to the reference day selected. This is not prohibitive in practice, since call centers usually have enough data to work with. In our case, however, this could result in too few traces when a day early in the year is picked. Since the call center data has a negligible trend in the number of calls, we use all traces available for the matching procedure. Thus, we treat traces that are not prior to the reference day as historical data gathered from the previous year. This also enables us to make better comparisons, since every reference day can be compared with 51 records when using $H^{(j)}$ and 259 records when using H.

In principle, the nearest neighbour algorithm can be run every time new data of the reference trace is available. In practice, reacting to every new forecast might lead to many different staffing configurations which might not be manageable. Therefore, we choose to update the forecast only at specific moments of the day. The call arrival function of call centers typically has two peaks, with relatively more calls arriving as compared to other periods, during a day. Depending on the services and opening hours of the call center, the first peak usually occurs in the morning, and the second in the afternoon. It is important to have staffed enough service representatives to meet the service level and not to build up a backlog of waiting calls, which might affect the service level in the periods to come. Therefore, having a better forecast before and after such peaks can result in flexible solutions, as described in Section 1, to prevent violation of the service level. Based on

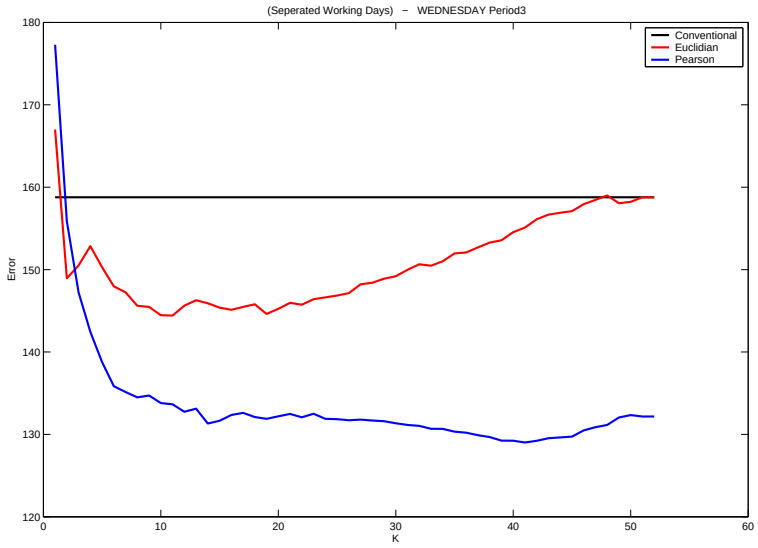

Figure 2: Forecast errors for CV, ED, and PD under SWD as a function of K.

the data of the call center, see Figure 1, we identify three moments over the day at which the forecast is updated. Figure 1 shows the mean arrival rates for the different days of the week, and identifies 9 am (start of period 9), 1 pm (start of period 25), and 6 pm (start of period 45) as update moments.

Having described the setup of the experiments, we continue to describe the forecasting algorithms that we will compare in the experiments. We shall compare the conventional way of forecasting (CV), described in Section 2, and the K nearest neighbour algorithm with the Euclidean distance (ED) and the Pearson correlation distance (PD), described in Section 3. Note that the conventional algorithm CV is equivalent to ED when K is set equal to the total number of records available. These algorithms will also be compared by forecasting based on separated days of the week $H^{(1)}, \ldots, H^{(5)}$ (SWD) and combined days of the week H (CWD). Note that at the start of period 1, the CV is the only applicable algorithm, since there are no traces available for the nearest neighbour algorithm.

Evaluation from a statistical perspective

In this subsection we will compare the algorithms CV, ED, and PD based on SWD and CWD, resulting in 6 algorithms. Since the presentation of these algorithms for five days of the week with three update moments in a day (giving 15 combinations) is too extensive, we only present the Wednesday with 1 pm as the update moment as a representative case. We evaluate the forecast at the start of period 25 and we define the *error of the forecast* as the average over the absolute differences between the forecasted number of calls and the actual realization in periods 25, ..., 44. It is not necessary to take the other periods into account, since at the start of period 45 the algorithm will update the forecast again.

We start by comparing the different algorithms by studying the error as a function of the size of the neighbourhood K. Figures 2 and 3 show the error for the CV, ED, and PD

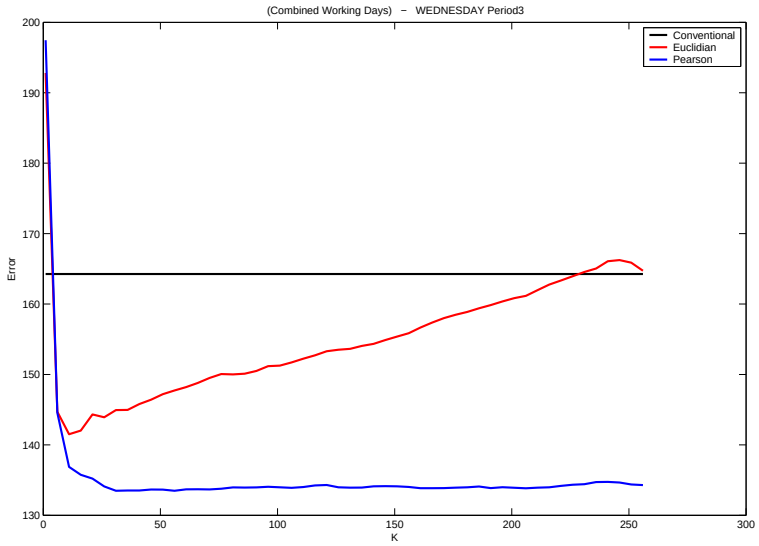

Figure 3: Forecast errors for CV, ED, and PD under CWD as a function of K.

algorithms under both SWD and CWD as a function of K. Note that the CV algorithm does not depend on the value of K, and hence results in a horizontal line in the figures. Also note that when K increases the ED algorithm resembles the CV algorithm more, and consequently has the same performance as the CV algorithm for large K.

Both figures consistently show that the PD algorithm performs better than the CV and the ED algorithms. Also, the PD algorithm does not need data preparation and actually profits from working with the full dataset that is not split up for every day of the week. The results under CWD seem to be more stable for different values of K. Moreover, under CWD already moderate values for K suffice to run the PD algorithm.

The quality measure used to compare the performance of the considered algorithms is an average of the errors obtained by taking every Wednesday out of the 52 Wednesdays as a reference day. Therefore, it is interesting to analyze the variability in the errors of these forecasts. Figures 4 and 5 show the spread of the error under SWD for the ED and PD algorithms illustrated by boxplots. The variance of the CV algorithm is obtained by looking at the graph for the ED algorithm with $K = 51$.

The figures show that the errors for the nearest neighbour algorithms are denser around the lower error values. This strengthens the conclusion that the nearest neighbour algorithm outperforms the conventional forecasts. More formally, one can statistically test which of the algorithms performs better by using the Wilcoxon rank test. This test determines if for each pair of methods, with each using its best value of K, the median of the first method is significantly lower than that of the second one. By carrying out the test for every pair of methods (CV, ED, and PD), for every dataset (SWD $H^{(1)}, \ldots, H^{(5)}$, and CWD H), and for every update moment we obtained the significance values for all the data. For illustration we give the significance values in Table 1 for the comparison of ED and CV under SWD.

	Sunday	Monday	Tuesday	Wednesday	Thursday
II	0.0052	0.0373	0.1931	0.0512	0.0009
III	0.0215	0.0210	0.0008	0.0167	0.0000
IV	0.1988	0.0036	0.0640	0.0098	0.0934

Table 1: Significance values for ED and CV under SWD.

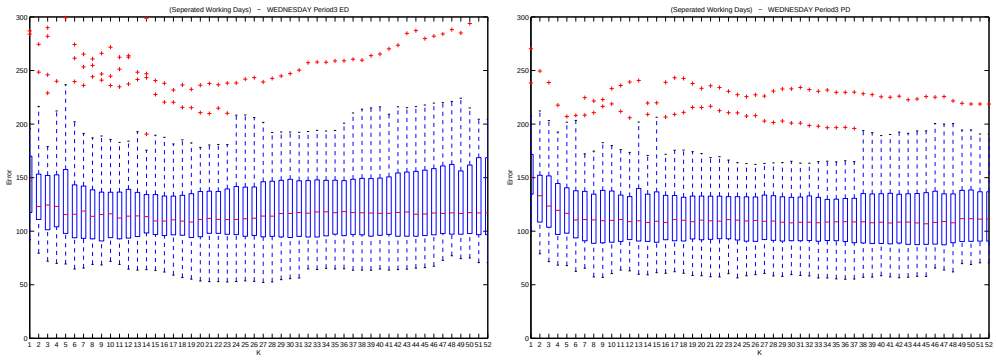

Figure 4: Spread of the errors for CV and PD under SWD as a function of K.

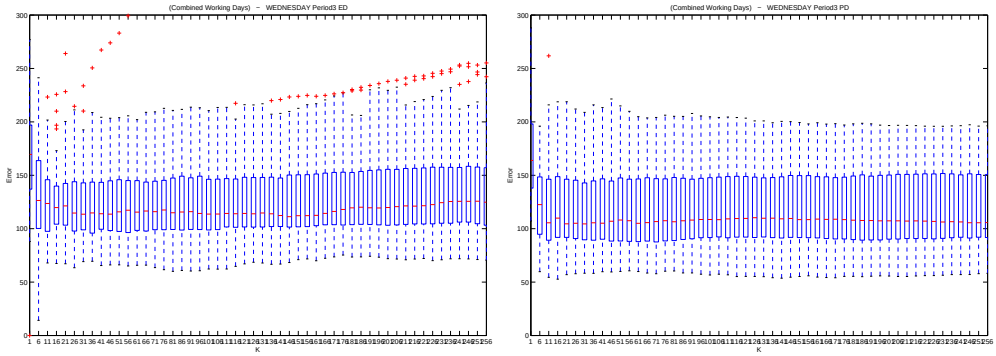

Figure 5: Spread of the errors for CV and PD under CWD as a function of K.

The table shows, for each interval (see Figure 1) and day combination, if the ED algorithm performs better than the CV algorithm. With a significance level of $\alpha = 5\%$, the table shows that for interval III the ED is significantly better than CV. With the exception of Tuesday and Wednesday, it is also better in interval II. Interval IV is somewhat unclear; Monday and Wednesday are predicted more accurately.

In general we can conclude that intervals II and III are forecasted more accurately when the nearest neighbour algorithm is used. Preference is given to the Pearson correlation distance, due to lower forecasting errors and less variability in the spread of the errors. For interval IV there was no clear indication that any method was significantly better that the others. However, the PD algorithm performed best in most cases.

Evaluation from a staffing perspective

In the previous subsection we studied the forecasting algorithms from a statistical perspective. However, it is not apparent how the conclusions of the previous subsection translate to improvements in staffing. Therefore, in this subsection we evaluate the results of the algorithms in the context of staffing in call centers. We first illustrate the effect of the nearest neighbour forecasting algorithm on staffing when the SIPP approach is used (see Green et al. [12]). Next, we study the effect on staffing based on transient models (see Ingolfsson et al. [14]).

To illustrate the SIPP approach, we model the call center as an M/M/c queue in every period. Furthermore, we assume that the call center has a service level target specifying that 80% of the customers should have a service representative on the line within 20 seconds. >From the call center data, we can derive that the average service time of the representatives is 2.4 minutes per call. The staffing can now be based on the Erlang-C formula (see Erlang [7]), using the forecasts of the call arrivals, such that the service level is met in every period.

We analyze three scenarios in Table 2 for staffing on a specific Wednesday in interval III (i.e., periods 25 to 44). The first scenario uses the real call arrival rates h_i to staff the minimum number of service representatives n_i such that the attained service level SL_i is above 80%. The second scenario uses the conventional forecasting method to derive estimates h_i^{CV} of the call arrival rate. Based on these forecasts, the number of representatives n_i^{CV} is determined such that the service level is met. However, in reality the service level experienced under the real call arrival rates is given by $\mathrm{SL}_i^{\mathrm{CV}}$. The last scenario is similar to the second with the exception that the forecast is based on the nearest neighbour algorithm with the Pearson correlation distance.

Table 2 shows that the conventional method systematically overestimates the call arrivals. This results in planning 137 representatives and an average weighted service level $\sum_i h_i \,\mathrm{SL}_i / \sum_i h_i$ of 0.960. The nearest neighbour algorithm does not meet the service level in every period. It staffs 116 representatives with an average weighted service level of 0.890. It is close to the optimal staffing level of 108 with an average weighted service level of 0.869. In general, numerical experiments show that the nearest neighbour algorithm produces results closer to the optimal staffing levels and the optimal average weighted service level. On special days, i.e., days with relatively high or low call volumes, the service level of the nearest neighbour algorithm is generally also better when viewed at each interval as compared to the service levels generated with the conventional method.

To illustrate the effect of the forecasting methods on staffing with transient models, we model the call center as an $\mathrm{M}_t/\mathrm{M}/c_t$ queueing system. The forecasts in scenarios 2 and 3 are updated at the beginning of interval II, III, and IV. As in the previous case, scenario 2 uses the conventional forecasting algorithm. In scenario 3, the best forecasting algorithm is used in each interval, i.e., the forecasting algorithm that gives the lowest errors in each interval for Wednesdays. Thus, the conventional forecast is used in interval I, the nearest neighbour forecasting algorithm PD is used in interval II and III, and for interval IV the nearest neighbour algorithm ED is used. Due to the computational complexity of this experiment, we take Δt to be one hour, with an additional half hour at the end of the day for handling all remaining calls in the system. For the same reason, the service level is set

h_i	n_i	SL_i	h_i^{CV}	n_i^{CV}	$\mathrm{SL}_i^{\mathrm{CV}}$	h_i^*	n_i^*	SL_i^*
23	6	0.846	28.706	7	0.940	20.821	6	0.846
23	6	0.846	33.627	8	0.979	22.089	6	0.846
20	6	0.914	30.176	8	0.991	22.309	6	0.914
21	6	0.894	29.137	7	0.962	25.016	7	0.962
19	5	0.812	28.333	7	0.977	23.431	6	0.931
18	5	0.845	30.608	8	0.995	26.041	7	0.983
25	7	0.911	29.941	8	0.966	25.382	7	0.911
21	6	0.894	28.510	7	0.962	24.187	6	0.894
24	6	0.817	27.922	7	0.927	23.455	6	0.817
24	6	0.817	30.333	8	0.973	26.065	7	0.927
20	6	0.914	30.961	8	0.991	27.480	7	0.970
27	7	0.873	27.196	7	0.873	22.577	6	0.708
15	5	0.921	22.980	6	0.976	18.675	5	0.921
20	6	0.914	23.039	6	0.914	18.138	5	0.775
11	4	0.912	23.196	6	0.994	18.528	5	0.976
14	4	0.813	22.333	6	0.982	18.211	5	0.939
14	4	0.813	19.549	6	0.982	14.797	5	0.939
8	3	0.871	18.431	5	0.994	13.333	4	0.969
16	5	0.899	20.549	6	0.968	15.870	5	0.899
17	5	0.874	19.843	6	0.957	15.163	5	0.874
	108	0.869		137	0.960		116	0.890

Table 2: Numerical experiments for SIPP staffing.

to have an average speed of answer (ASA) of at most 30 seconds.

Table 3 shows the results for the three scenarios for several days (hence the index d instead of i). When we compare scenario 2 and 3 with each other we cannot draw firm conclusions. Out of the 50 Wednesdays that we examined, the nearest neighbour algorithm performed better in 26 cases, worse in 17 cases, and had similar performance in 7 cases. The variability in performance is caused by the estimate in interval I. The staffing based on this estimate can result in an ASA far from 30. This result has a big influence on the staffing in the next intervals, since the staffing levels will be adjusted to correct the average speed of answer. Hence, dynamic forecasting in combination with transient models should be done more carefully.

5 Conclusions

We have investigated the effectiveness of K-nearest neighbour algorithms for forecasting call volumes in call centers, based on two distance functions (the Euclidean distance and the Pearson correlation distance). From a statistical point of view, the nearest neighbour algorithm yields significantly more accurate predictions than the conventional method

n_d	SL_d	n_d^{CV}	$\mathrm{SL}_d^{\mathrm{CV}}$	n_d^*	SL_d^*
89	29.5783	91	30.9722	90.5	34.7052
84	27.8924	88	27.6744	105	32.7986
70	27.8958	86	12.1418	74	29.2221
102	29.0204	139	29.2838	134	29.8620
92	29.9510	95.5	30.0246	95.5	30.2343
117	29.7751	148	33.9807	142	34.6535

Table 3: Numerical experiments for staffing with transient models.

that simply takes average over the historical data. Depending on the characteristics of the call arrival function in a specific interval, preference is given to one of the distance functions, resulting in a hybrid forecasting algorithm that mixes forecasting methods over the intervals. Additionally, the nearest neighbour algorithm does not require data preparation and captures the correlation structures typically found in call center data.

When the forecasting algorithms are viewed upon from a staffing perspective, the method of staffing and the way the service level is calculated is relevant for the performance. When the staffing is based on independent periods, the average waited service level under the hybrid nearest neighbour algorithm performs very well. When the staffing is based on a transient model, however, no firm conclusion can be drawn. The error in the forecast in earlier periods have a big influence on the staffing in later periods. Hence, dynamic forecasting when using transient models warrants more research.

References

[1] C.G. Atkeson, A.W. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11:11–73, 1997.

[2] A.N. Avradimis, A. Deslauriers, and P. L'Ecuyer. Modeling daily arrivals to a telephone call center. *Management Science*, 50:896–908, 2004.

[3] S. Bhulai and G.M. Koole. A queueing model for call blending in call centers. *IEEE Transactions on Automatic Control*, 48:1434–1438, 2003.

[4] L. Brown, N. Gans, A. Mandelbaum, A. Sakov, H. Shen, S. Zeltyn, and L. Zhao. Statistical analysis of a telephone call center: A queueing-science perspective. *Submitted*, 2002.

[5] T.M. Cover and P.E. Hart. Nearest neighbour pattern classification. *IEEE Transactions on Information Theory*, 13:21–27, 1967.

[6] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. Wiley, 1973.

[7] A.K. Erlang. Solutions of some problems in the theory of probabilities of significance in automatic telephone exchanges. *Electroteknikeren*, 13:5–13, 1917.

[8] N. Gans, G. Koole, and A. Mandelbaum. Telephone call centers: Tutorial, review, and research prospects. *Manufacturing and Service Operations Management*, 5:79–141, 2003.

[9] N. Gans and Y.-P. Zhou. A call-routing problem with service-level constraints. *Operations Research*, 51:255–271, 2003.

[10] W.K. Grassmann. Transient solutions in Markovian queueing systems. *Computers & Operations Research*, 4:47–53, 1977.

[11] L. Green and P. Kolesar. The pointwise stationary approximation for queues with non-stationary arrivals. *Management Science*, 37:84–97, 1991.

[12] L. Green, P. Kolesar, and J. Soares. Improving the SIPP approach for staffing service systems that have cyclic demands. *Operations Research*, 49:549–564, 2001.

[13] S. Halfin and W. Whitt. Heavy-traffic limits for queues with many exponential servers. *Operations Research*, 29:567–587, 1981.

[14] A. Ingolfsson, E. Cabral, and X. Wu. Combining integer programming and the randomization method to schedule employees. *Working paper*, 2003.

[15] A. Ingolfsson, M.A. Haque, and A. Umnikov. Accounting for time-varying queueing effects in workforce scheduling. *European Journal of Operational Research*, 139:585–597, 2002.

[16] G. Jongbloed and G.M. Koole. Managing uncertainty in call centers using Poisson mixtures. *Applied Stochastic Models in Business and Industry*, 17:307–318, 2001.

[17] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[18] S.G. Steckley, S.G. Henderson, and V. Mehrotra. Service system planning in the presence of a random arrival rate. *Submitted*, 2004.

[19] J. Yoo. *Queueing models for staffing service operations*. PhD thesis, University of Maryland, 1996.