Neural Comput & Applic (2003) 12: 212–219
DOI 10.1007/s00521-003-0384-x

ORIGINAL ARTICLE

Tom Heskes · Jan-Joost Spanjers · Bart Bakker
Wim Wiegerinck

Optimising newspaper sales using neural-Bayesian technology

Received: 1 October 2002 / Accepted: 7 August 2003 / Published online: 31 October 2003

Abstract We describe a software system, called just enough delivery (JED), for optimising single-copy newspaper sales, based on a combination of neural and Bayesian technology. The prediction model is a huge feedforward neural network, in which each output corresponds to the sales prediction for a single outlet. Input-to-hidden weights are shared between outlets. The hidden-to-output weights are specific to each outlet, but linked through the introduction of priors. All weights and hyperparameters can be inferred using (empirical) Bayesian inference. The system has been tested on data for several different newspapers and magazines. Consistent performance improvements of 1 to 3% more sales with the same total amount of deliveries have been obtained.

Keywords Bayesian inference · Feed forward · Multi-task learning · Neural networks · Time-series prediction

1 Symbols

The following symbols are represented in the text:

$\sigma_i(t)$	Actual sales of outlet i at time (edition) t
$y_i(t)$	Predicted sales
$\epsilon_i(t)$	Noise component in actual sales
$x_{ik}(t)$, $x_i(t)$	Inputs (explanatory variables)
D_i, D	All sales data
I_i, I	All input data
A_{ij}, $\boldsymbol{A}_i$, A	Hidden-to-output weights and biases (bias: $j = 0$)
B_{jk}	Input-to-hidden weights and biases (bias: $k = 0$)
$\boldsymbol{M}$	Prior mean of the hidden-to-output weights
Σ	Prior covariance between hidden-to-output weights
σ_i, σ	Standard deviation of noise component
Λ	All parameters shared between outlets
$\Delta_i\ (t)$	Nonstationary correction term
$v_i(t)$	Dynamic noise component
$\delta(t)$	Systematic noise component

2 Optimising newspaper sales

With declining sales in recent years, single-copy newspaper and magazine sales are getting more and more attention. One of the issues is how to distribute the newspapers as efficiently as possible. Newspaper sales are extremely irregular, but, since deliveries have to be determined on a daily basis and for a huge number of outlets, almost any performance improvement is worth the effort. Advanced statistical tools can help to obtain these improvements. In this article, we describe our software solution called just enough delivery (JED), which is built around a combination of neural and Bayesian methodology. An earlier version of JED is in operation at De Telegraaf, a major Dutch newspaper company. The latest version, which already has been tested on data from several other international editors, is currently (July 2002) running in shadow and is expected to go online within a couple of months.

JED has been developed for newspapers and magazines that have the right of return: the outlet returns the unsold copies and only has to pay for the ones sold. The financial accounting involved ensures that there is a reliable registration of the number of returned copies, at the level of the individual outlets and (usually) on a daily basis. The corresponding continuous flow of information is sketched in Fig. 1.

T. Heskes (✉) · J.-J. Spanjers · B. Bakker · W. Wiegerinck
SMART Research BV and SNN,
University of Nijmegen,
Geert Grooteplein 21, 6525 EZ Nijmegen,
The Netherlands
E-mail: tom@snn.kun.nl

Current systems for steering single-copy newspaper sales can be roughly subdivided into two main classes: rule-based and adaptive. The rule-based systems contain simple user-specified rules of the type "if the outlet has been sold out the last two weeks, increase the delivery with two copies." Sometimes exceptions are built in, e.g., for outlets that are presumed to be seasonally dependent. Most adaptive systems are also very basic and usually contain some version of exponential smoothing: the delivery is a smoothed version of the recently observed sales, with a few extra copies added.

Building more elaborate adaptive systems in which newspaper sales are matched against explanatory variables is difficult due to the high risk of over fitting. It is easy to come up with quite a number of (possibly) explanatory variables, e.g., recent sales figures, sales figures from last year, seasonal variables, information about competitors, holidays, weather, news content, and so on. But when straightforwardly trained on the data for a single outlet (usually about two to three years of data, i.e., 100 to 150 examples when newspapers for different days of the weeks are treated independently), serious overfitting is almost inevitable.

A Bayesian solution has been proposed recently in [1]. Here the models trained on individual outlets are automatically regularised using the so-called evidence framework [2]. In this article we present an alternative use of Bayesian technology that tries to explicitly take into account the similarities between different outlets such as to let them "learn from each other". This is implemented in two ways: by choosing a specific neural architecture and by introducing sensible priors on the weights of this neural network. Formulating this in a probabilistic framework, we describe how Bayesian methodology can be applied for inference, prediction and making decisions.

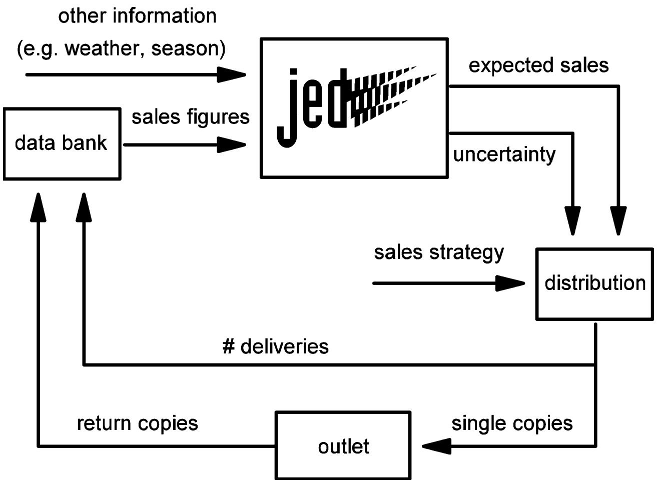


Fig. 1 Sketch of the flow of information for a single outlet. The model is updated based on the latest available information. Given the current values for the explanatory variables, the model does not only predict the upcoming sales, but also the uncertainty of this prediction. Combined with the company's strategy, e.g., a focus on reducing returns or rather reducing sell-outs, the "optimal" delivery can be computed. The return copies are collected and stored in the database

In this article, we will try to highlight the underlying concepts, assumptions, and applied technology. Mathematical details can be found in [3, 4, 5].

3 The model

3.1 Multi-task learning

To be able to use Bayesian machinery, we have to set up a probabilistic model, specifying how the data is generated as well as prior probabilities on the parameters of the model. Our model assumption is that the observed sales $s_\mathrm{i}(t)$ for outlet i and edition t is given by the model output $y_\mathrm{i}(\mathbf{x}_\mathrm{i},(t))$ (to be specified below) with additional Gaussian noise $\epsilon_\mathrm{i}(t)$ of standard deviation σ_i:

$$s_i(t) = y_i(\mathbf{x}_i(t)) + \varepsilon_i(t) \tag{1}$$

Strictly speaking, this model is incorrect, since newspaper sales are always discrete and positive, and can never be larger than the amount of deliveries. Especially the latter restriction is important, since, when not taken into account, this would lead to a structural underestimation of the sales that can be obtained. A practical, heuristic solution is to correct the observed sales figures in case of a sell-out. The corrected sales figure is then taken to be the expected sales that could have been obtained with "infinite deliveries" given that the sales are at least the actual amount of deliveries. This number can be computed using the sales model (Eq. 1). A more elegant solution is to explicitly take the sell-out into account as partially observable (sales of at least the amount of deliveries) [3], similar to the incorporation of censored patients in survival analysis. For notational convenience, we will act here as if the heuristic corrections are applied. Furthermore, all sales data is rescaled such that the observed sales per outlet have a zero mean and unit standard deviation and, after this rescaling, we take the same noise standard deviation $\sigma_\mathrm{i} = \sigma$ for all outlets.

Through definition of the model output $y_\mathrm{i}(\mathbf{x}_\mathrm{i},(t))$, we can take into account the (possible) effect of explanatory variables $x_\mathrm{ik}(t)$. Here k numbers the variables (e.g., temperature or recent sales) and the index i indicates that the inputs $x_\mathrm{ik}(t)$ are specific to outlet i. The prediction model is sketched in Fig. 2: a feedforward neural network with a single layer of hidden units. In math, the network output $y_\mathrm{i}(t)$ given inputs $\mathbf{x}_\mathrm{i}(t)$ reads:

$$y_i(t) = y_i(\mathbf{x}_i(t)) = \sum_j A_{ij} g\left(\sum_k B_{jk} \mathbf{x}_{ik}(t) + B_{j0}\right) + A_{i0} \tag{2}$$

where $g(\cdot)$ is typically a sigmoidal function like the hyperbolic tangent. It is important to note that where

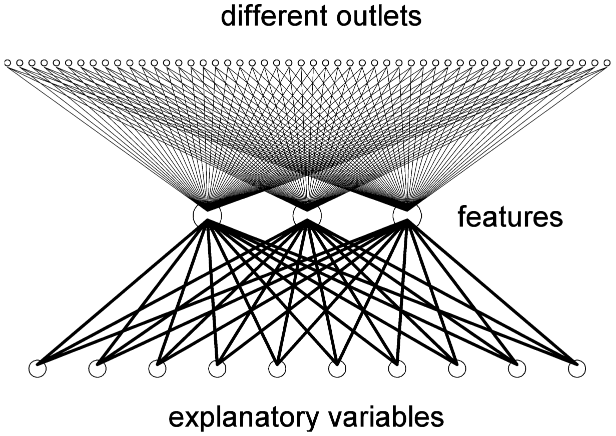


Fig. 2 Neural network architecture. Input information is propagated through a bottleneck of hidden units to the outputs, each representing a particular outlet

the hidden-to-output weights A_{ij} are specific to outlet i, the input-to-hidden weights B_{jk} are shared between outlets, even although the inputs $\mathbf{x}_{\mathrm{i}}(t)$ are outlet-specific. This makes the so-called multi-task learning approach [6, 7] different from a standard regression task in which all outputs are correlated with the same input patterns. In this particular application, the hidden units form a kind of bottleneck and transform the high-dimensional input space (typically 20 to 30 inputs) to a lower dimensional feature space (about 2 to 6 hidden units). These are supposed to represent features that are optimal for the overall task of predicting sales for the particular newspaper or magazine. Since the data for all tasks can be used to learn this transformation (see below), the risk of overfitting this part of the network is greatly reduced.

Last but not least, we assume independently and identically distributed (iid) observations, yielding the data likelihood term:

$$P(D|A,B,I) = \prod_i \prod_t \sqrt{\frac{1}{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}(s_i(t) - y_i(t))^2\right] \tag{3}$$

with $y_i(t) = y_i(\mathbf{x}_i(t))$ as specified in Eq. 2. D stands for all observed sales figures $s_i(t)$, I for all explanatory variables $x_{ik}(t)$. The iid assumption is standard in many applications, but might be too strong for time-series data like newspaper sales. Furthermore, the assumption that the errors $\epsilon_i(t)$ for a particular edition are uncorrelated across different outlets i is also tricky, especially when considering aggregate figures. Corrections and alternatives will be discussed in a later section.

3.2 Prior information

In a maximum likelihood setting, we would simply try to find the weights (including biases) $\{A,B\}$ maximising the data likelihood (Eq. 3). In essence, this would yield a standard optimisation procedure with a backpropagation-type learning rule. This may seem fine for the input-to-hidden weights B, but leads to severe overfitting on the hidden-to-output weights for any reasonable number of hidden units. One option is to introduce regularisation, for example, by adding a so-called weight-decay term, the strength of which can be determined through cross-validation. In the Bayesian framework, such a regularising component can be introduced through definition of a prior. Here we propose to define a prior on the hidden-to-output weights $\frac{1}{2}\mathbf{A}_i = \{A_{i0}, A_{i1}, ...A_{ih}\}$:

$$P(\mathrm{A_i}|\mathbf{M},\Sigma) = \sqrt{\frac{1}{(2\pi)^{h+1}\det\Sigma}} \exp\left[-\frac{1}{2}(\mathbf{A}_i - \mathbf{M})^T \Sigma^{-1}(\mathbf{A}_i - \mathbf{M})\right] \tag{4}$$

with M a vector of length $h+1$, with h the number of hidden units, and Σ an $(h+1) \times (h+1)$ covariance matrix. This corresponds to a so-called exchangeability assumption (the same $\mathbf{M}$ for all tasks) and implements a tendency for similar hidden-to-output weights across tasks. Exactly how similar is determined by the covariance matrix Σ. $\mathbf{M}$ and Σ are referred to as hyperparameters. As for the input-to-hidden weights, we can use all available data to try and infer them (see below). Note that the standard approach in the evidence framework, which is also taken in [1], is to choose $\mathbf{M} = 0$ and Σ as spherical or at most diagonal.

The exchangeability assumption is rather strong. It means that, prior to the arrival of any (sales) data, there is no reason to distinguish between the different outlets. It is relatively straightforward to relax this assumption and consider outlet-dependent averages $\mathbf{M}_{\mathrm{i}}$ that depend on the characteristics of the outlet [4]. A simple example is to choose different means for outlets in different parts of the country. Another option is to have the different outlets cluster themselves into groups with different means and/or covariances. This can be implemented by introducing a mixture of Gaussians as the prior [5, 8]. In this article, we will stick to the simplest case of a single mean. Our model is fairly similar to a so-called multi-level model in statistics (see e.g., [9]), with the hidden-to-output weights A_{ij} playing the role of the "random effects" and the input-to-hidden weights B_{jk} and prior mean $\mathbf{M}$ making up the "fixed effects".

4 Bayesian inference

4.1 Empirical Bayes

In a full (hierarchical) Bayesian framework, we should also specify hyperpriors on the hyperparameters $\mathbf{M}$ and Σ, as well as on the input-to-hidden weights B and noise standard deviation σ. The multi-task setting, however, suggests an efficient alternative, called empirical Bayes [10]. Empirical Bayes consists of two steps. First, we try to find the "hyperparameters" Λ^* that maximise the likelihood of the data:

$$\Lambda^* = \underset{\Lambda}{\operatorname{argmax}}\, P(D|\Lambda, I) \tag{5}$$

In our case, Λ consists of all parameters that are shared between outlets, i.e.,

$$\Lambda = \{\mathbf{M}, \Sigma, B, \sigma\} \tag{6}$$

Given these maximum likelihood hyperparameters, we then approximate the distribution of the model parameters through:

$$\begin{aligned} P(\mathbf{A}_i|D, I) &\approx P(\mathbf{A}_i|\Lambda^*, D_i, I_i) \\ &\propto P(D_i|\mathbf{A}_\mathrm{i}, B^*, \sigma^* I^*) P(\mathbf{A}_i|\mathbf{M}^*, \Sigma^*) \end{aligned} \tag{7}$$

Empirical Bayes is an approximation of the full hierarchical Bayesian approach, asymptotically equivalent when the number of outlets goes to infinity. The huge number of outlets in the cases we are considering here (on the order of a few hundred at least), makes it an excellent approximation. The evidence framework described and applied in [1, 2] can be interpreted as an empirical Bayesian approach for single-task learning. In the single-task learning case, the argument for taking the maximum likelihood value of the hyperparameters is less obvious and has been the subject of a lively debate (see e.g., [11]).

The likelihood to be maximised in Eq. 5 involves a high-dimensional integral over the hidden-to-output weights A:

$$\begin{aligned} P(D|\Lambda, I) &= \prod_i \int d\mathbf{A}_i P(\mathbf{A}_i, D_i|\Lambda, I_i) \\ &= \prod_i \int d\mathbf{A}_i P(D_i|\mathbf{A}_i, B, \sigma, I_i) P(\mathbf{A}_i|\mathbf{M}, \Sigma) \end{aligned} \tag{8}$$

where D_i and I_i denote the data (observed sales and explanatory variables, respectively) for outlet i and the integrals are over the $(h + 1)$-dimensional vectors of hidden-to-output weights and biases $\mathbf{A}_i$. To find its maximum, we can make use of the (generalised) expectation-maximisation (EM) algorithm [12]. In the expectation step, we compute the probabilities $P(\mathbf{A}_i|\Lambda^{old}, D_i, I_i)$(in fact, we only need the means and covariances) of the hidden-to-output weights and biases given the current hyperparameters Λ^{old}using Eq. 7 with Λ^{old} substituted for Λ^*.

In the maximisation step, we update the hyperparameters such as to increase the "full data loglikelihood":

$$\begin{aligned} H(\Lambda, \Lambda^{old} &= \int dA\, P(A|\Lambda^{old}, D, I) \log P(A, D|\Lambda, I) \\ &= \sum_i \int d\mathbf{A}_i P(\mathbf{A}_i|\Lambda^{old}, D_i, I_i) \log P(D_i|\mathbf{A}_i, B, \sigma, I_i) \\ &\quad \times P(\mathbf{A}_i|\mathbf{M}, \Sigma) \end{aligned} \tag{9}$$

It can be shown that an increase in $H(\Lambda, \Lambda^{old})$ implies that:

$$P(D|\Lambda, I) \geqslant P(D|\Lambda^{old}, I) \tag{10}$$

and thus that the EM algorithm converges to a (local) maximum of the likelihood $P(D|\Lambda, I)$. More details can be found in [3, 5].

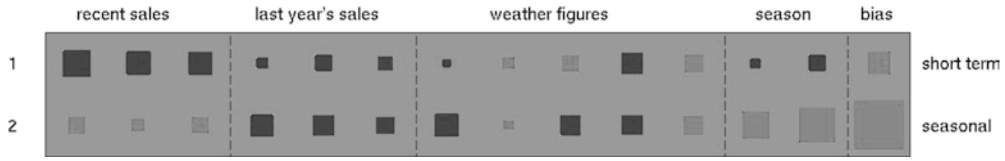
With a linear transfer function $g(\cdot)$ and some further simplifications, the integral in Eq. 8 happens to be analytically doable [4]. In this case, we can directly optimise the likelihood, which makes the optimisation an order of magnitude faster than with the EM algorithm. In practical applications, the fast algorithm can be used for an initial estimate, which is then fine-tuned with the more elaborate EM procedure.

4.2 An example

The Bayesian machinery does not only lead to improved performance (see below and [4] for a comparison with single-task and non-Bayesian techniques), but the results obtained also tend to make a lot of sense. An example is given in Fig. 3, showing a Hinton diagram of the input-to-hidden weights B_{jk} obtained when training the model with two hidden units on a set of 343 outlets, concerning 156 consecutive editions of Saturday's newspaper. The inputs include recent sales (four to six weeks in the past), last year's sales (51 to 53 weeks in the past), weather information (temperature, wind, sunshine, precipitation quantity and duration) and season (the cosine and the sine of the scaled week number). It can be seen that one hidden unit focuses on recent sales figures (referred to as "short term") and the other one on last year's sales and season (referred to as "seasonal").

The effect of the prior can be seen in Fig. 4. The left panel plots the maximum likelihood (ML) solutions for the hidden-to-output weights $\mathbf{A}_i$ of the different outlets; the right panel the maximum *a posteriori* (MAP) solutions. The definitions of these maximum likelihood and maximum *a posteriori* solutions are, respectively,

Fig. 3 A Hinton diagram of the hidden-to-input weights B_{jk}. Positive weights are white, negative weights are black. The absolute magnitude of each weight corresponds to the size of its square. Past sales figures are coded in inputs 1–6, inputs 7–11 represent weather information and 12–13 indicate the season. The rightmost squares represent the biases B_{i0} of the hidden units



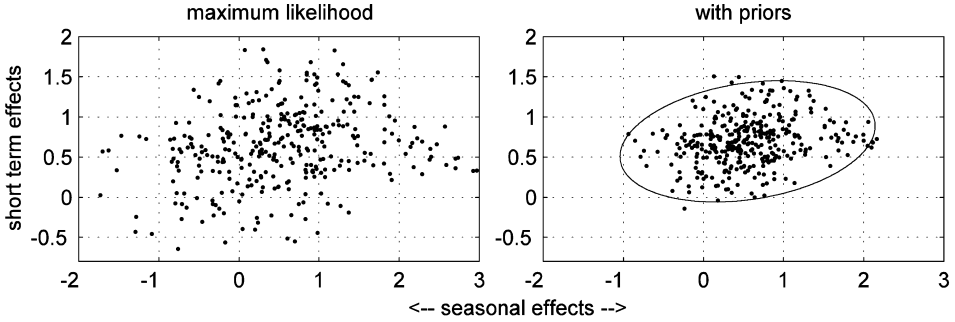


Fig. 4 The maximum likelihood values for the hidden-to-output weights (left panel), and their maximum *a posteriori* values (right panels). Each mark represents the value of the "short-term" and "seasonal" weights for one outlet. The ellipse in the right panel indicates the 95% confidence interval of the priors imposed on the weights

$$\begin{aligned}\mathbf{A}_i^{ML} &= \underset{\mathbf{A}_i}{\operatorname{argmax}}\, P(D_i|\mathbf{A}_i,B_i^*,\sigma^*,I_i)\\ \mathbf{A}_i^{MAP} &= \underset{\mathbf{A}_i}{\operatorname{argmax}}\ P(D_i|\mathbf{A}_i,B_i^*,\sigma^*,I_i)P(\mathbf{A}_i|\mathbf{M}^*,\Sigma^*).\end{aligned} \tag{11}$$

It can be seen that the regularised MAP solutions are closer together than the ML solutions. An important difference with a standard regularisation approach is that the map solutions are regressed towards the mean **M***, rather than towards zero. For example, the positive mean for the short-term effects implies that higher recent sales in general have a positive effect on current sales.

5 Operation and evaluation

5.1 Online operation

The hyperparameters Λ describe the overall characteristics of all prediction tasks. Theoretical and empirical evidence [7, 4] indicates that data for on the order of a few hundred outlets is sufficient to estimate them: the performance hardly gets better when more outlets are considered. It also seems reasonable to assume that these characteristics under normal conditions do not change very rapidly. Updating these once every few years is then sufficient. The characteristics at the more local level of the individual outlets, represented by the hidden-to-output weights $\mathbf{A}_i$, may change much more frequently and better be updated on a weekly or at least monthly basis. In practice, all hyperparameters are computed online, based on a representative set of outlets. After that, the hyperparameters are kept fixed, and the outlets operate independently, regularly updating the (probability distribution of the) hidden-to-output weights $\mathbf{A}_\mathrm{i}$.

The posterior distribution of these weights takes into account the last two to three years of data, with the most recent examples replacing the oldest ones. This sliding-window approach implicitly still assumes that the sales data can be considered roughly stationary on a time scale of a few years. This may be fine for the hidden-to-output weights A_{ij} (for $j \geq 1$, i.e., excluding the biases A_{i0}). For example, we expect an outlet's sensitivity to weather circumstances or seasonal effects to be more or less constant over a few years. Nonstationarity seems to have the largest impact on the average sales, i.e., on the biases A_{i0}. This average may change quite a lot on a time scale smaller than a few years. Examples are an increase due to new construction in the vicinity of the outlet, or a decrease due to the arrival of a strong competitor. These are typical nonstationary effects that are hard to predict in advance. The naive sliding-window approach, in these situations, leads to a structural underestimation and overestimation, respectively. To take nonstationarity in the bias into account, we add a correction term to the (stationary) prediction of the average sales $y_i(t)$:

$$\tilde{y}_i(t) = y_i(t) + \Delta_i(t) \text{ with } \Delta_i(t) = \Delta_i(t-1) + v_i(t) \tag{12}$$

a random-walk equation for the correction $\Delta_i(t)$ with Gaussian white noise $v_i(t)$. This random-walk equation describes a simple dynamic model. Keeping track of the distribution of $\Delta_i(t)$ using Bayesian update equations is straightforward [13]. A limiting case of the resulting procedure leads to exponential smoothing, but the dynamic linear model has many principled advantages (coherence, interpretability and validity even for just a few data points). In short, the incorporation of this nonstationary model component makes the system very robust against unexpected and unpredictable changes, while it hardly bothers the performance of the system if no such changes take place.

5.2 Optimal delivery

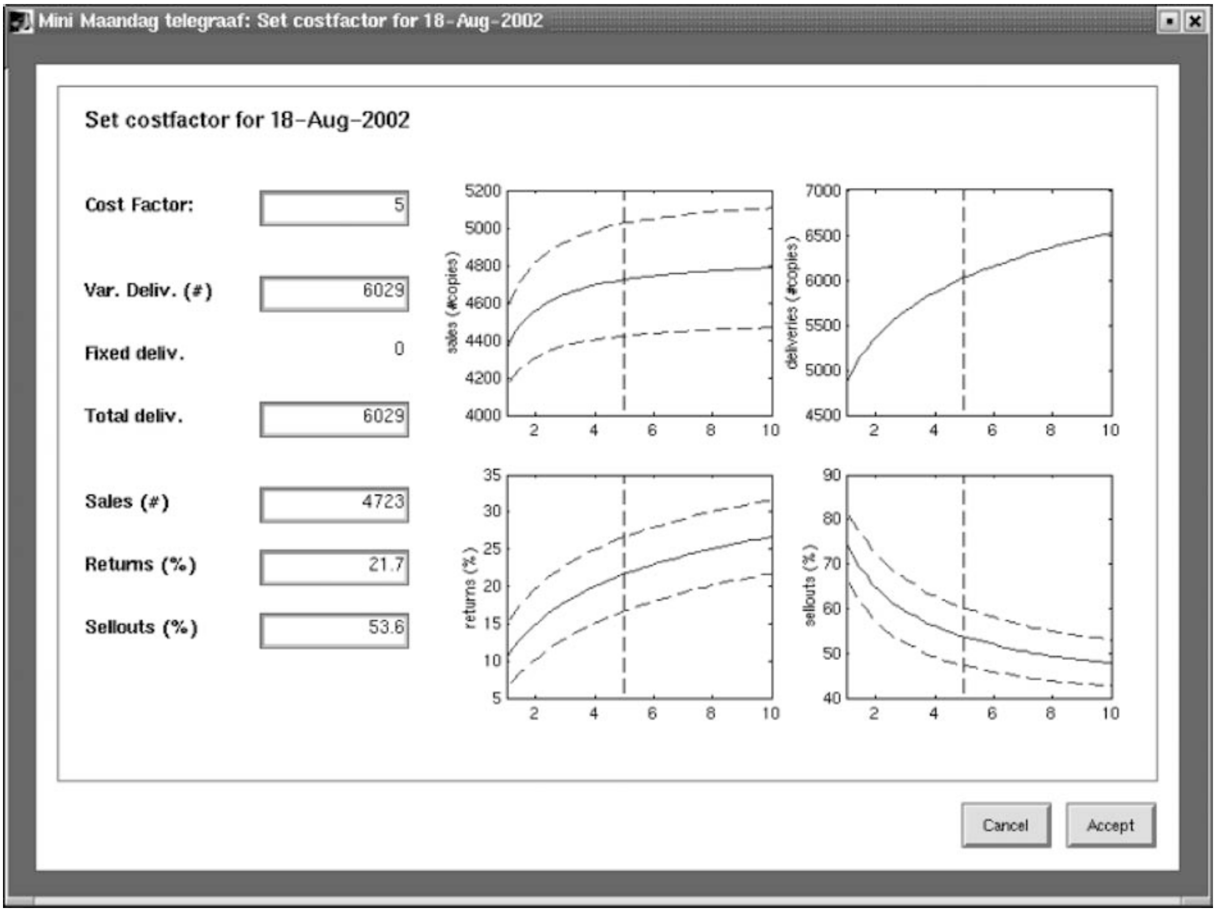
Suppose that we have to determine the delivery for an outlet i given input $\mathbf{x}_i(t)$, based on all available data D and I up to edition t. Following our probabilistic model, the sales distribution reads (the random-walk component $\Delta_i(t)$ can be easily taken into account as well, but is omitted here for notational convenience):

$$\begin{aligned}P(s_i|\mathbf{x}_i(t),D,I) &= \int d\Lambda \int d\mathbf{A}_i P(s_i|\mathbf{A}_i,\mathbf{x}_i(t),\Lambda)\\ &\quad \times P(\mathbf{A}_i|D_i,I_i,\Lambda)P(\Lambda|D,I)\\ &\approx \int d\mathbf{A}_i P(s_i|\mathbf{A}_i,\mathbf{x}_i(t),I_i,\Lambda^*)P(\mathbf{A}_i|D_i,I_i,\Lambda^*)\end{aligned} \tag{13}$$

where the approximation follows from the empirical Bayesian assumption that for all practical purposes we can consider the maximum-likelihood solution Λ^* for the hyperparameters. The integration over the hidden-to-output weights, rather than taking their most probable value, takes into account their uncertainty. Bayesian decision theory (see e.g., [10]) can now be applied to choose the delivery l_i^* that maximises the expected utility (note that here we assume that yield and costs are a linear function of the number of copies: any other rational choice can be handled in a similar way):

$$U(l_i|\mathbf{x}_i(t), D, I) = \int_0^{l_i} ds_i s_i P(s_i|\mathbf{x}_i(t), D, I) \times \text{yield per copy} - l_i \times \text{costs per copy}. \tag{14}$$

In a first approximation, this optimal delivery only depends on the expected sales, the width of the distribution (the "uncertainty" in Fig. 1), and a so-called cost factor which is defined as the ratio between yield and costs per copy. The setting of this cost factor is up to the company and depends on the company's "sales strategy": the higher the cost factor the more expansive the sales strategy, the lower the more conservative. In practice, determining this cost factor is far from trivial, but operators usually have some gut feeling of appropriate values (typically around 5 for newspapers and somewhat lower for magazines). An alternative is to tune this cost factor such that another set point is met, e.g., a desired total amount of deliveries or a particular (expected) percentage of returns (see Fig. 5). Keeping the same cost factor for all outlets then guarantees that the same utility criterion is optimised for all of them.



Mini Maandag telegraaf: Set costfactor for 18-Aug-2002

Set costfactor for 18-Aug-2002

Cost Factor:	5
Var. Deliv. (#)	6029
Fixed deliv.	0
Total deliv.	6029
Sales (#)	4723
Returns (%)	21.7
Sellouts (%)	53.6

Fig. 5 A sheet visualising the impact of the cost factor (the horizontal axis) on aggregate sales, deliveries, returns and sell-outs. Solid lines indicated expected values, dashed lines error bars. It allows the operator to implement a different sales strategy by choosing a different cost factor

5.3 Aggregate sales

In writing down the data likelihood, we made the assumption that the errors (difference between observed and predicted sales) $\epsilon_i(t)$ are uncorrelated across outlets for a particular edition t. This assumption may not hurt a lot when considering the predictions for individual outlets, but is very dangerous when aggregate sales figures are involved and especially when estimating error bars on these aggregate values [13]. For example, with uncorrelated noise the relative accuracy of the total sales of n outlets would scale with $1/\sqrt{n}$ and thus vanish for large n. This is, obviously, not the situation in practice: there are factors (such as, for example, news content) that may have a relatively small effect on the sales of

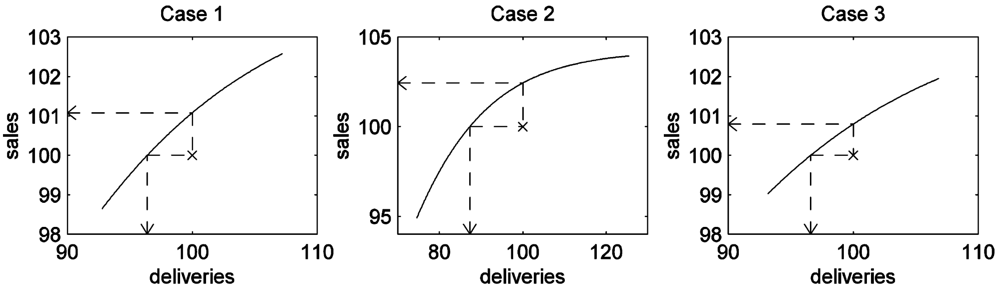


Fig. 6 The model's performance in three case studies. The total amount of actual deliveries, issued by the company's current system, and observed sales, based on these deliveries, are both indexed to 100 (crosses). Solid lines give the sales that could have been obtained if the model's suggestions had been followed for different amounts of deliveries. Dashed lines indicate the sales (↓) and delivery break-even points (←)

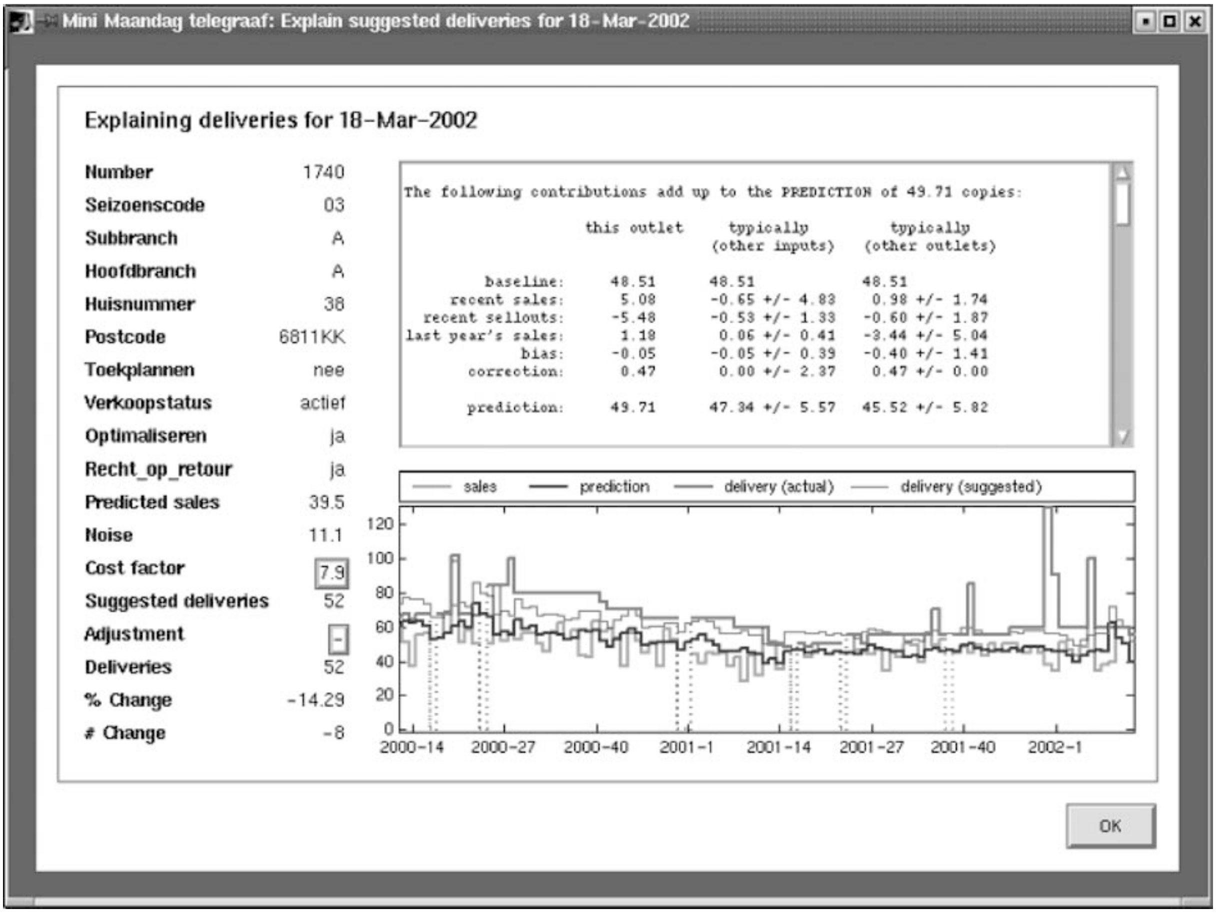
individual outlets, but, since this effect tends to point in the same direction for all outlets, cannot be neglected at the aggregate level. One way to incorporate this is to model the error $\epsilon_i(t)$ as:

$$\varepsilon_i(t) = \delta(t) + \tilde{\varepsilon}_i(t) \tag{15}$$

with $\delta(t)$ an edition-specific noise component, the same for all outlets, and $\tilde{\varepsilon}_i(t)$ a component specific to both edition and outlet. The relative importance of these two components can be estimated within the same empirical Bayesian framework [4]. For the newspaper data that we have encountered the variance of the systematic edition-specific component is between 5% and 15% of the total variance.

Adapting the sales model (Eq. 1) in this way, we can combine the individual sales predictions and estimate expected sales, return and sell-out figures at the aggregate level with the appropriate error bars. As an example, different settings of the cost factor are given in Fig. 5 (here for a set of 500 outlets). The operator can use these estimates to tune the cost factor to a specified (expected) performance, e.g., to choose the cost factor such as to achieve an expected sales of 4200 copies.

Fig. 7 Sheet visualising the properties of a particular outlet and explaining the predicted sales in terms of the explanatory variables and how this is translated into a suggested delivery. Based on this information, the operator can judge whether to accept or change the suggested delivery



Mini Maandag telegraaf: Explain suggested deliveries for 18-Mar-2002

Explaining deliveries for 18-Mar-2002

Number	1740
Seizoenscode	03
Subbranch	A
Hoofdbranch	A
Huisnummer	38
Postcode	6811KK
Toekplannen	nee
Verkoopstatus	actief
Optimaliseren	ja
Recht_op_retour	ja
Predicted sales	39.5
Noise	11.1
Cost factor	7.9
Suggested deliveries	52
Adjustment	-
Deliveries	52
% Change	-14.29
# Change	-8

```
The following contributions add up to the PREDICTION of 49.71 copies:

                     this outlet     typically          typically
                                   (other inputs)    (other outlets)

         baseline:     48.51       48.51             48.51
     recent sales:      5.08       -0.65 +/- 4.83     0.98 +/- 1.74
  recent sellouts:     -5.48       -0.53 +/- 1.33    -0.60 +/- 1.87
last year's sales:      1.18        0.06 +/- 0.41    -3.44 +/- 5.04
             bias:     -0.05       -0.05 +/- 0.39    -0.40 +/- 1.41
       correction:      0.47        0.00 +/- 2.37     0.47 +/- 0.00

       prediction:     49.71       47.34 +/- 5.57    45.52 +/- 5.82
```

5.4 A performance evaluation

The model's suggestions can be compared with the company's deliveries for a range of different strategies, implemented by changing the cost factor. Results obtained in three different case studies are shown in Fig. 6. The graphs report test performance (the company returned the results only after our model had suggested the deliveries) based on more than 500 outlets followed for at least 13 consecutive weeks. The company's own amounts are indexed to 100, shown by the crosses, i.e., the axes can be interpreted as percentages relative to the company's performance. Of special interest are the delivery and sales break-even points, indicated by the dashed lines. Here the cost factor is chosen such that our model uses exactly the same total amount of deliveries or yields the same sales results, respectively. It can be seen that the sales improvement with same total amount of deliveries ranges from about 1% (case 1 and 3) to almost 3% (case 2). The results for the sales break-even points are even more impressive: the same sales can be reached with more than 3% (case 1 and 3) to even 13% (case 2) less deliveries.

6 Discussion

In this article we have described the methodology behind JED, a software system for optimising the distribution of newspapers and magazines. JED is based upon a probabilistic framework and uses Bayesian inference whenever possible to manipulate these probabilities. Considering the size of the problem and its daily occurrence, the ongoing challenge is to find sensible approximations that speed up the calculations, without hurting the performance. One of our current interests is to integrate the (empirical) Bayesian approach for multitask learning with the Bayesian methodology for time-series analysis. So-called dynamic hierarchical models [14] are in the right direction, but lack a bottleneck of feature units and do not quite contain the time dependencies that are appropriate for newspaper sales.

Prediction performance is important, but definitely not the whole story behind a successful product. We have also put quite some effort in algorithms that provide insight in the model and its decisions, examples of which are given in Figs. 5 and 7. Also with regard to these "secondary" tools, the underlying probabilistic framework helps a lot to increase the coherence between the tools and model components and makes them easier to apply and interpret.

References

1. Ragg T, Menzel W, Baum W and Wigbers M (2002) Bayesian learning for sales rate prediction for thousands of retailers. Neurocomputing 43:127–144
2. MacKay D (1995) Probable networks and plausible predictions—a review of practical Bayesian methods for supervised neural networks. Network 6:469–505
3. Heskes T (1998) Solving a huge number of similar tasks: a combination of multi-task learning and a hierarchical Bayesian approach. In: Proceedings of the International Conference on Machine Learning, Morgan Kaufmann, San Mateo, CA
4. Heskes T (2000) Empirical Bayes for learning to learn. In: Langley P (ed) Proceedings of the Seventeenth International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA
5. Bakker B, Heskes T (2003) Task clustering and gating for Bayesian multitask learning. J Mach Learn Res 4:83–99
6. Caruana R (1997) Multitask learning. Mach Learn 28:41–75
7. Baxter J (1997) A Bayesian/information theoretic model of learning to learn via multiple task sampling. Mach Learn 28:7–39
8. Cadez I, Ganey S and Smyth P (2000) A general probabilistic framework for clustering individuals. Technical report, University of California, Irvine, CA
9. Bryk A, Raudenbusch S (1992) Hierarchical linear models. Sage, Newbury Park, UK
10. Robert C (1994) The Bayesian choice: a decision-theoretic motivation. Springer, Berlin Heidelberg New York
11. Wolpert D (1993) On the use of evidence in neural networks. In: Hanson S, Cowan J and Giles L (eds) Advances in neural information processing systems 5, Morgan Kaufmann, San Mateo, CA
12. Dempster A, Laird N and Rubin D (1977) Maximum likelihood from incomplete data via the EM algorithm. J Roy Stat Soc B 39:1–38
13. West M, Harrison J (eds) (1977) Bayesian forecasting and dynamic models. Springer, Berlin Heidelberg New York
14. Gamerman D, Migon H (1993) Dynamic hierarchical models. J Roy Stat Soc B 55:629–642