Variational inference for Student-t MLP models

Hang T. Nguyen[a,*], Ian T. Nabney[a]

[a] *The Non-linearity and Complexity Research Group (NCRG), School of Engineering and Applied Science, Aston University, Aston Triangle, Birmingham, B4 7ET, UK*

Abstract

This paper presents a novel methodology to infer parameters of probabilistic models whose output noise is a Student-t distribution. The method is an extension of earlier work for models that are linear in parameters to nonlinear multi-layer perceptrons (MLPs). We used an EM algorithm combined with variational approximation, the evidence procedure, and an optimisation algorithm. The technique was tested on two regression applications. The first one is a synthetic dataset and the second is gas forward contract prices data from the UK energy market. The results showed that forecasting accuracy is significantly improved by using Student-t noise models.

Key words: Variational inference, Student-t noise, multilayer perceptrons, EM algorithm, forecast.

1. Introduction

In forecasting models, we generally assume that the data is corrupted by noise:

$$y_t = f(\mathbf{x}_t) + \varepsilon_t,$$

where ε_t is a zero-mean probability distribution. Normally, the noise is assumed to be Gaussian distribution either because of arguments derived from the central

*Corresponding authour: email address: thihangn@aston.ac.uk, Fax: +44 121 204 3685, Tel: +44 121 257 7718

limit theorem or just to simplify calculations. For example, the log likelihood of a Gaussian noise model is a quadratic function of the output variables. This leads to the fact that in the training process, we can easily estimate the maximum likelihood solution using optimisation algorithms. Software and frameworks for training machine learning models such as radial basis functions (RBF), MLP, and linear regression (LR) with Gaussian noise can be found in [1]. Conversely, other noise models are much less tractable. So why use the Student-t distribution?

In our previous work [2, 3], we used models with Gaussian noise to forecast gas and electricity forward prices in the UK energy market. In these experiments, the kurtosis, which is a measure of how outlier-prone a distribution is, of the residuals (i.e. the different between target and output of forecasting model) is between 16 and 17: the kurtosis of the Gaussian distribution is 3. Furthermore, $P(\mu - 3\sigma < r < \mu + 3\sigma) \approx 0.982$, where μ and σ are the mean and standard derivation of the residual respectively. The equivalent probability for a Gaussian distribution is 0.997; therefore, the residual distribution has heavy tails. This means that the residual distributions are much more outlier-prone than the Normal distribution. The large number of outliers can make the training process unreliable and error bar estimates inaccurate, because Gaussians are sensitive to outliers. It is clear that this data is not modelled well by a Gaussian distribution as has often been noted for financial data.

As a consequence, a Student-t distribution can be considered as a good alternative to a Gaussian because it is a fat-tailed distribution and is more robust. Moreover, the Student-t distribution family contains the Normal distribution as special case.

There are several previous studies of inference with Student-t models. Tipping and Lawrence proposed a framework for training an RBF model with fixed basis functions [4]. This study is a fully Bayesian treatment based on a varia-

tional approximation framework. A variational inference scheme was also used for unsupervised learning with mixture models: Bishop and Svensén presented an algorithm for automatically determining the number of components in a mixture of t-distribution using a Bayesian variational framework [5]. In order to obtain a tractable solution, it was assumed that the latent variables are independent, and thus posterior distributions of latent variables can be factorized. This means that the algorithm does not capture correlations among the latent variables. Archambeau and Verleysen introduced a new variational Bayesian learning algorithm for Student-t mixture models, in which they removed the assumption of variable independence [6]. Numerical experiments showed that their model had a greater robustness to outliers than Bishop and Svensén's method in [5].

This paper presents a novel methodology to infer parameters of Student-t probabilistic models. This methodology for MAP estimation is an extension of the technique introduced by Tipping and Lawrence [4], in which models are assumed to be linear in parameters. Both approaches are based on a variational approximation. The main advantage of our method is that it is not limited to models whose output is linearly dependent on model parameters. On the other hand, our approach provides only MAP estimates of parameters while Tipping and Lawrence give a fully Bayesian treatment in which predictions are made by integrating out all the parameters apart from those defining the t-distribution, which are optimised. Thus, although our algorithm can be applied to models that are linear in parameters, we would not expect it to outperform Tipping and Lawrence, so our discussion focusses on the MLP.

This paper is organised as follows. In Section 2, Student-t noise models are presented. Section 3 describes our inference technique for MLPs. Numerical results on two datasets are given in Section 4. Section 5 discusses some conclusions.

2. Student-t noise model

We assume that the output data is corrupted by noise with a Student-t distribution.

$$y_t = f(\mathbf{x}_t, \boldsymbol{\omega}) + \varepsilon_t,$$

where ε_t is a Student-t noise process, and $f(\mathbf{x}_t)$ is the output function of a forecast model, which can be a multi-layer perceptron (MLP), radial basis function (RBF), or linear regression (LR). In the case of MLP models, the output is non-linear in the parameters. Conversely, the output is linear in parameters when the model is LR or RBF. We are not investigating the case where the independent variables $\mathbf{x}_t$ are also noisy.

The Student-t distribution can be considered as a mixture of an infinite number of zero-mean Gaussians with different variances:

$$\begin{aligned} p(\varepsilon_t|c,d) &= \int_0^\infty p(\varepsilon_t|\beta_t)p(\beta_t|c,d)\ d\beta_t \\ &= \frac{d^c}{\Gamma(c)}\left(\frac{1}{2\pi}\right)^{1/2}\left[d+\frac{\varepsilon_t^2}{2}\right]^{-c-1/2}\Gamma(c+1/2), \end{aligned} \tag{1}$$

where

$$p(\varepsilon_t|\beta_t) = N(\varepsilon_t|0,\beta_t^{-1}),$$

$$p(\beta_t|c,d) = Gamma(\beta_t|c,d) = \frac{d^c}{\Gamma(c)}\beta_t^{c-1}\exp(-\beta_t d).$$

The mixture weight for a given β_t is specified by the Gamma distribution $p(\beta_t|c,d)$. $\nu = 2c$ is called the "number of degrees of freedom" and $\sigma = \sqrt{d/c}$ is the scale parameter of the distribution. The degrees-of-freedom parameter ν can be considered as a robustness tuning parameter [6]. When ν tends to infinity, this distribution converges to a Gaussian. Therefore, the Student-t noise model still contains the Gaussian as a special case when ν is very large.

3. MAP estimation for MLPs

The aim of our approach is to find maximum a posterior (MAP) estimates of network and noise model parameters. MAP estimation is not a fully Bayesian treatment because it finds the optimal parameters of the models instead of integrating over all unknown parameters. This is equivalent to the type-II maximum likelihood method [7].

In this paper, we will describe an EM algorithm for training a model with a Student-t noise model. This training framework can be use for both "non-linear in parameters" models and "linear in parameters" models.

Given a data set $\mathbf{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)\}$, our goal is to optimise parameters of a predictive model (i.e. MLP, LR or RBF) using MAP. To simplify the notation, let $\Omega = \{\boldsymbol{\omega},\ c,\ d,\ \alpha\}$ be the set of parameters/hyperparameters of the model and noise. The posterior density of the parameters given a dataset $\mathbf{D}$ is given by

$$p(\boldsymbol{\Omega}|\mathbf{D}) = \frac{p(\mathbf{D}|\boldsymbol{\Omega})p(\boldsymbol{\Omega})}{p(\mathbf{D})},$$

where $p(\mathbf{D}|\boldsymbol{\Omega})$ is the dataset likelihood, $p(\boldsymbol{\Omega})$ is the prior, and $p(\mathbf{D})$ is evidence. Because the denominator does not affect the MAP solution, we can ignore this term: $p(\boldsymbol{\Omega}|\mathbf{D}) \propto p(\mathbf{D}|\boldsymbol{\Omega})p(\boldsymbol{\Omega})$. The likelihood and the prior are given by

$$p(\mathbf{D}|\boldsymbol{\Omega}) = p(\mathbf{D}|\boldsymbol{\omega},c,d) = \prod_{t=1}^{T} p(y_t|\mathbf{x}_t, \boldsymbol{\omega},c,d)$$

$$p(y_t|\mathbf{x}_t, \boldsymbol{\Omega}) = \frac{d^c}{\Gamma(c)} \left(\frac{1}{2\pi}\right)^{1/2} \left[d + \frac{(y_t - f(\mathbf{x}_t, \boldsymbol{\omega}))^2}{2}\right]^{-c-1/2} \Gamma(c + 1/2)$$

$$p(\boldsymbol{\Omega}) = p(\boldsymbol{\omega}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})p(c,d). \tag{2}$$

The weight prior $p(\boldsymbol{\omega}|\boldsymbol{\alpha})$ is a Gaussian. It is helpful to generalise the hyperparameter $\boldsymbol{\alpha}$ to multiple hyperparameters $\alpha_1, \ldots, \alpha_M$ corresponding to groups of

weights $\mathcal{W}_1, \ldots, \mathcal{W}_M$. In theory, we can create groupings of the weights in any way that we want. However, weights in an MLP are normally divided into four groups: first-layer weights, first-layer biases, second-layer weights, and second-layer biases. In addition, the first layer weights can be also divided into several groups: weights fanning out from a input variable are associated to a separate group. The latter grouping approach relates to automatic relevance determination (ARD) [8] and is used in our experiments. Denote group dimensions by $\mathbf{W}_1, \ldots, \mathbf{W}_M$ corresponding to the groups $\mathcal{W}_1, \ldots, \mathcal{W}_M$. Thus the dimension of $\boldsymbol{\omega}$ is $\mathbf{W} = \sum_{m=1}^{M} \mathbf{W}_m$.

$$p(\boldsymbol{\omega}|\boldsymbol{\alpha}) = \prod_{m=1}^{M} N(\mathcal{W}_m|0,\alpha_m^{-1}) = \prod_{m=1}^{M} \left(\frac{\alpha_m}{2\pi}\right)^{\mathbf{W}_m/2} \exp\left[\sum_{m=1}^{M}\left(-\frac{\alpha_m}{2}\sum_{\omega\in\mathcal{W}_m}\omega^2\right)\right] \quad (3)$$

There are many possible choices for the densities $p(\boldsymbol{\alpha})$ and $p(c,d)$, but for simplicity we assume that they are uniform distributions. Therefore, they will be ignored in the subsequent analysis. Hence

$$\begin{aligned}
\log p(\boldsymbol{\Omega}|\mathbf{D}) &\propto \log\left[p(\mathbf{D}|\boldsymbol{\Omega})p(\boldsymbol{\Omega})\right] \\
&= Tc\log d + T\log\frac{\Gamma(c+1/2)}{\Gamma(c)} - \frac{\mathbf{W}+T}{2}\log 2\pi \\
&\quad - \left(c+\frac{1}{2}\right)\sum_{t=1}^{T}\log\left[d+\frac{(y_t - f(\mathbf{x}_t,\boldsymbol{\omega}))^2}{2}\right] \\
&\quad + \sum_{m=1}^{M}\left(\frac{\mathbf{W}_m}{2}\log\alpha_m\right) - \sum_{m=1}^{M}\left(\frac{\alpha_m}{2}\sum_{\omega\in\mathcal{W}_m}\omega^2\right).
\end{aligned}$$

3.1. Variational approximation

The Student-t distribution of each observation y_t can be considered as a mixture of an infinite number of zero-mean Gaussians with inverse variance β_t. Let $\boldsymbol{\beta} = \{\beta_1, \beta_2, \ldots, \beta_T\}$; then

$$p(\mathbf{D}|\boldsymbol{\Omega}) = \int_0^\infty p(\mathbf{D},\boldsymbol{\beta}|\boldsymbol{\Omega})d\boldsymbol{\beta} = \int_0^\infty p(\mathbf{D}|\boldsymbol{\beta},\boldsymbol{\Omega})p(\boldsymbol{\beta}|\boldsymbol{\Omega})d\boldsymbol{\beta}, \quad (4)$$

where

$$p(\mathbf{D}|\boldsymbol{\beta},\boldsymbol{\Omega}) = \prod_{t=1}^{T} p(y_t|\beta_t,\boldsymbol{\Omega}) = \prod_{t=1}^{T}\left(\frac{\beta_t}{2\pi}\right)^{1/2}\exp\left\{-\frac{\beta_t}{2}(y_t - f(\mathbf{x}_t,\boldsymbol{\omega}))^2\right\} \quad (5)$$

$$p(\boldsymbol{\beta}|\boldsymbol{\Omega}) = \prod_{t=1}^{T} p(\beta_t|\boldsymbol{\Omega}) \ = \prod_{t=1}^{T} Gamma(\beta_t|c,d) = \prod_{t=1}^{T} \frac{d^c e^{-d\beta_t}\beta_t^{c-1}}{\Gamma(c)}. \tag{6}$$

It is difficult to optimise $p(\mathbf{D}|\boldsymbol{\Omega})$ directly, but optimising $p(\mathbf{D},\boldsymbol{\beta}|\boldsymbol{\Omega})$ is significantly easier. We use a variational method [9] to approximate the posterior $p(\mathbf{D}|\boldsymbol{\Omega})p(\boldsymbol{\Omega})$ as follows. An approximating distribution $q(\boldsymbol{\beta})$ for $p(\boldsymbol{\beta}|\mathbf{D},\boldsymbol{\Omega})$ is introduced: for every choice of $q(\boldsymbol{\beta})$, the following decompositions hold:

$$\begin{aligned} \log\left[p(\mathbf{D}|\boldsymbol{\Omega})p(\boldsymbol{\Omega})\right] &= \log\left[p(\mathbf{D},\boldsymbol{\beta}|\boldsymbol{\Omega})p(\boldsymbol{\Omega})\right] - \log p(\boldsymbol{\beta}|\mathbf{D},\boldsymbol{\Omega}) \\ \log\left[p(\mathbf{D}|\boldsymbol{\Omega})p(\boldsymbol{\Omega})\right] &= \mathcal{L}(q,\boldsymbol{\Omega}) + KL(q||p), \end{aligned} \tag{7}$$

where

$$\begin{aligned} \mathcal{L}(q,\boldsymbol{\Omega}) &= \int_0^\infty q(\boldsymbol{\beta}) \log\left\{\frac{p(\mathbf{D},\boldsymbol{\beta}|\boldsymbol{\Omega})p(\boldsymbol{\Omega})}{q(\boldsymbol{\beta})}\right\} d\boldsymbol{\beta} \\ &= \int_0^\infty q(\boldsymbol{\beta}) \log p(\mathbf{D},\boldsymbol{\beta}|\boldsymbol{\Omega})p(\boldsymbol{\Omega}) d\boldsymbol{\beta} - \int_0^\infty q(\boldsymbol{\beta}) \log q(\boldsymbol{\beta}) \ d\boldsymbol{\beta} \end{aligned} \tag{8}$$

$$KL(q||p) = -\int_0^\infty q(\boldsymbol{\beta}) \log\left\{\frac{p(\boldsymbol{\beta}|\mathbf{D},\boldsymbol{\Omega})}{q(\boldsymbol{\beta})}\right\} d\boldsymbol{\beta}. \tag{9}$$

In equation (7), the second component $KL(q||p)$ is the Kullback-Leibler divergence between $q(\boldsymbol{\beta})$ and $p(\boldsymbol{\beta}|\mathbf{D},\boldsymbol{\Omega})$. It is clear that $KL(q||p) \geq 0$, with equality if and only if $q(\boldsymbol{\beta}) = p(\boldsymbol{\beta}|\mathbf{D},\boldsymbol{\Omega})$. Therefore, $\mathcal{L}(q,\boldsymbol{\Omega}) \leq \log p(\mathbf{D}|\boldsymbol{\Omega})$, i.e. $\mathcal{L}(q,\boldsymbol{\Omega})$ is a lower bound on $\log\left[p(\mathbf{D}|\boldsymbol{\Omega})p(\boldsymbol{\Omega})\right]$.

3.2. EM for optimising the posterior

Based on the decomposition in equation (7), we use an EM algorithm to maximise $p(\mathbf{D}|\boldsymbol{\Omega})\ p(\boldsymbol{\Omega})$. The two following steps are repeated:

- **E-step**: fix $\boldsymbol{\Omega}$ and maximise $\mathcal{L}(q,\boldsymbol{\Omega})$ with respect to $q(\boldsymbol{\beta})$. The lower bound can be seen as a negative Kullback-Leibler divergence between $q(\boldsymbol{\beta})$ and a distribution which is proportional to $p(\mathbf{D},\boldsymbol{\beta}|\boldsymbol{\Omega})p(\boldsymbol{\Omega})$. Thus maximising $\mathcal{L}(q,\boldsymbol{\Omega})$

is equivalent to minimising this Kullback-Leibler divergence. The lower bound is maximised when $q(\boldsymbol{\beta}) \propto p(\mathbf{D}, \boldsymbol{\beta}|\boldsymbol{\Omega})p(\boldsymbol{\Omega}) = p(\mathbf{D}|\boldsymbol{\beta}, \boldsymbol{\Omega})p(\boldsymbol{\beta}|\boldsymbol{\Omega})p(\boldsymbol{\Omega})$. Discarding terms that are independent of β, we have:

$$\log q(\boldsymbol{\beta}) \propto \sum_{t=1}^{T} \log \{p(y_t|\boldsymbol{\omega}, \beta_t)p(\beta_t|c, d)\}$$

$$\begin{aligned} \log q(\beta_t) &\propto \log \{p(y_t|\boldsymbol{\omega}, \beta_t)p(\beta_t|c, d)\} \\ &\propto (c - \frac{1}{2}) \log \beta_t \; - \left[d + \frac{1}{2}(y_t - f(\mathbf{x}_t, \boldsymbol{\omega}))^2\right] \beta_t + const. \end{aligned}$$

The above equation shows that $\log q(\beta_t)$ is a linear combination of $\log \beta_t$ and β_t. Therefore, $q(\boldsymbol{\beta})$ is a product of Gamma distributions with the following parameters:

$$q(\boldsymbol{\beta}) = \prod_{t=1}^{T} Gamma(\beta_t|\widetilde{c}, \widetilde{d}_t) \tag{10}$$

$$\widetilde{c} = c + \frac{1}{2}, \; \widetilde{d}_t = d + \frac{1}{2}(y_t - f(\mathbf{x}_t, \boldsymbol{\omega}))^2. \tag{11}$$

Note that the method in [4] estimated posterior distributions of parameters $\boldsymbol{\omega}, \boldsymbol{\alpha}$, and $\boldsymbol{\beta}$. In order to obtain a tractable solution for these distributions, they assumed that $\boldsymbol{\omega}, \boldsymbol{\alpha}$, and $\boldsymbol{\beta}$ are a posteriori separable, such that $q(\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = q_{\boldsymbol{\omega}}(\boldsymbol{\omega})q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha})q_{\boldsymbol{\beta}}(\boldsymbol{\beta})$. In our work, this assumption changes since we estimate the distribution of $\boldsymbol{\beta}$ only; the other parameters (i.e. $\boldsymbol{\omega}$ and $\boldsymbol{\alpha}$) are optimised in the M-step (which is equivalent to delta function for each parameter vector).

- **M-step**: fix $q(\boldsymbol{\beta})$ using equations (10) and (11), and maximise $\mathcal{L}(q, \boldsymbol{\Omega})$ with respect to $\boldsymbol{\Omega}$. In equation (8), the first component is the expectation of a complete-data log likelihood. The second component is the entropy of $q(\boldsymbol{\beta})$

and does not depend on $\boldsymbol{\Omega}$. Therefore, we can ignore this component in the subsequent analysis:

$$\mathcal{L}(q, \boldsymbol{\Omega}) = \int_0^\infty q(\boldsymbol{\beta}) \log \{p(\mathbf{D}, \boldsymbol{\beta}|\boldsymbol{\Omega}) p(\boldsymbol{\Omega})\} \, d\boldsymbol{\beta} = \langle \log \{p(\mathbf{D}, \boldsymbol{\beta}|\boldsymbol{\Omega}) p(\boldsymbol{\Omega})\} \rangle_{q(\beta)} . \quad (12)$$

We now describe how this optimisation can be done in the following section.

3.3. Optimising the lower bound of log posterior

Firstly, we have to compute the lower bound $\mathcal{L}(q, \boldsymbol{\Omega})$.

$$\begin{aligned} \log \{p(\mathbf{D}, \boldsymbol{\beta}|\boldsymbol{\Omega}) p(\boldsymbol{\Omega})\} &= \sum_{t=1}^{T} \log \{p(y_t|\boldsymbol{\omega}, \beta_t) p(\beta_t|c, d)\} + \log p(\boldsymbol{\omega}|\boldsymbol{\alpha}) \\ &\quad + \log p(\boldsymbol{\alpha}) + \log p(c, d). \end{aligned} \quad (13)$$

$q(\boldsymbol{\beta})$ is defined by equation (10). The densities $p(\boldsymbol{\alpha})$ and $p(c, d)$ are assumed to be uniform distributions. Therefore, they will be ignored in the subsequent analysis.

$$\mathcal{L}(q, \boldsymbol{\Omega}) = \sum_{t=1}^{T} \langle \log \{p(y_t|\boldsymbol{\omega}, \beta_t) p(\beta_t|c, d)\} \rangle_{q(\boldsymbol{\beta})} + \langle \log p(\boldsymbol{\omega}|\boldsymbol{\alpha}) \rangle_{q(\boldsymbol{\beta})} .$$

From equations (3), (5), and (6), we have

$$\begin{aligned} \langle \log \{p(y_t|\boldsymbol{\omega}, \beta_t) p(\beta_t|c, d)\} \rangle_{q(\boldsymbol{\beta})} &= \left(c - \frac{1}{2}\right) \langle \log \beta_t \rangle_{q(\boldsymbol{\beta})} - \frac{(y_t - f(\mathbf{x}_t, \boldsymbol{\omega}))^2}{2} \langle \beta_t \rangle_{q(\boldsymbol{\beta})} \\ &\quad - d \langle \beta_t \rangle_{q(\boldsymbol{\beta})} + c \log d - \log \Gamma(c) - \frac{1}{2} \log(2\pi) \end{aligned}$$

$$\langle \log p(\boldsymbol{\omega}|\boldsymbol{\alpha}) \rangle_{q(\boldsymbol{\beta})} = \sum_{m=1}^{M} \left(\frac{\mathbf{W}_m}{2} \log \alpha_m \right) - \frac{\mathbf{W}}{2} \log (2\pi) - \sum_{m=1}^{M} \left(\frac{\alpha_m}{2} \sum_{\omega \in \mathcal{W}_m} \omega^2 \right),$$

where $\langle \log \beta_t \rangle_{q(\boldsymbol{\beta})} = \langle \log \beta_t \rangle_{p(\beta_t|\widetilde{c}, \widetilde{d}_t)} = \psi(\widetilde{c}) - \log \widetilde{d}_t$ and $\langle \beta_t \rangle_{q(\boldsymbol{\beta})} = \langle \beta_t \rangle_{p(\beta_t|\widetilde{c}, \widetilde{d}_t)} = \widetilde{c}/\widetilde{d}_t$, with $\psi(\cdot)$ the "psi" or "digamma" function, defined as $\psi(x) = \partial / \partial x \left[\log \Gamma(x)\right]$ [10]. The lower bound is given by (constant components are ignored for simplicity):

$$\begin{aligned} \mathcal{L}(q, \boldsymbol{\Omega}) &= \left(c - \frac{1}{2}\right) \sum_{t=1}^{T} \left(\psi(\widetilde{c}) - \log \widetilde{d}_t \right) - \frac{1}{2} \sum_{t=1}^{T} \frac{\widetilde{c}}{\widetilde{d}_t} (y_t - f(\mathbf{x}_t, \boldsymbol{\omega}))^2 \\ &\quad - d \sum_{t=1}^{T} \frac{\widetilde{c}}{\widetilde{d}_t} + T c \log d - T \log \Gamma(c) \\ &\quad + \sum_{m=1}^{M} \left(\frac{\mathbf{W}_m}{2} \log \alpha_m \right) - \sum_{m=1}^{M} \left(\frac{\alpha_m}{2} \sum_{\omega \in \mathcal{W}_m} \omega^2 \right). \end{aligned} \quad (14)$$

We partition the parameters into three groups $\{c, d\}$, $\{\boldsymbol{\omega}\}$, and $\{\boldsymbol{\alpha}\}$, and optimise each group in turn with the others held fixed.

3.3.1. Optimise $\{c, d\}$

We can use a nonlinear optimisation algorithm (e.g. scaled conjugate gradient (SCG) [11]) to find an optimal solution for c, d. Derivatives of the lower bound with respect to c, d are given by:

$$\begin{aligned}\frac{\partial \mathcal{L}(q, \boldsymbol{\Omega})}{\partial c} &= \sum_{t=1}^{T} \left(\psi(\widetilde{c}) - \log \widetilde{d}_t\right) + T \log d - T\psi(c) \\ \frac{\partial \mathcal{L}(q, \boldsymbol{\Omega})}{\partial d} &= -\sum_{t=1}^{T} \frac{\widetilde{c}}{\widetilde{d}_t} + T\frac{c}{d},\end{aligned}$$

with constraints $c, d > 0$. These constraints can be enforced by a substitution: $c = \exp(\widehat{c})$, $d = \exp(\widehat{d})$. Derivatives of the lower bound with respect to $\widehat{c}$, $\widehat{d}$ are given by:

$$\begin{aligned}\frac{\partial \mathcal{L}(q, \boldsymbol{\Omega})}{\partial \widehat{c}} &= \frac{\partial \mathcal{L}(q, \boldsymbol{\Omega})}{\partial c}\frac{\partial c}{\partial \widehat{c}} = c\left[\sum_{t=1}^{T} \left(\psi(\widetilde{c}) - \log \widetilde{d}_t\right) + T \log d - T\psi(c)\right] \\ \frac{\partial \mathcal{L}(q, \boldsymbol{\Omega})}{\partial \widehat{d}} &= \frac{\partial \mathcal{L}(q, \boldsymbol{\Omega})}{\partial d}\frac{\partial d}{\partial \widehat{d}} = d\left[-\sum_{t=1}^{T} \frac{\widetilde{c}}{\widetilde{d}_t} + T\frac{c}{d}\right].\end{aligned}$$

3.3.2. Optimise $\boldsymbol{\omega}$

We now can consider optimisation of $\mathcal{L}(q, \boldsymbol{\Omega})$ with respect to $\boldsymbol{\omega}$ using a nonlinear optimisation algorithm such as SCG. The relevant partial derivative of lower bound $\mathcal{L}(q, \boldsymbol{\Omega})$ is given by:

$$\begin{aligned}\frac{\partial \mathcal{L}(q, \boldsymbol{\Omega})}{\partial \omega_i} &= -\frac{1}{2}\frac{\partial}{\partial \omega_i}\left[\sum_{t=1}^{T} \frac{\widetilde{c}}{\widetilde{d}_t}(y_t - f(\mathbf{x}_t, \boldsymbol{\omega}))^2\right] - \widehat{\alpha}_i\omega_i \\ &= -\sum_{t=1}^{T}\left\{\frac{\widetilde{c}}{\widetilde{d}_t}\frac{\partial}{\partial \omega_i}\left[\frac{1}{2}(y_t - f(\mathbf{x}_t, \boldsymbol{\omega}))^2\right]\right\} - \widehat{\alpha}_i\omega_i,\end{aligned}$$

where $\widehat{\boldsymbol{\alpha}} = [\widehat{\alpha}_1, \widehat{\alpha}_2, \ldots, \widehat{\alpha}_{\mathbf{W}}]$, $\widehat{\alpha}_i = \alpha_m$ if $i \in \mathcal{W}_m$, $i = 1, \ldots, \mathbf{W}$ and $m = 1, \ldots, M$. The term $\partial/\partial\omega_i\left[(y_t - f(\mathbf{x}_t, \boldsymbol{\omega}))^2/2\right]$ is the derivative of the mean square error function (MSE) for models with Gaussian noise. Equations for this derivative are presented in [1].

3.3.3. Optimise $\boldsymbol{\alpha}$

Our objective is to estimate the most probable value of $\boldsymbol{\alpha}$, in other word we maximise $p(\boldsymbol{\alpha}|\mathbf{D})$. The following procedure is derived from the standard evidence procedure [12]. The main difference is that the scalar hyperparameter β in the standard evidence procedure is replaced by a T-dimensional vector $\boldsymbol{\eta}$, to be derived below. This vector is fixed and defined by equation (11) while β in the standard evidence procedure is optimised simultaneously with $\boldsymbol{\alpha}$. In addition, we generalise the hyperparameter $\boldsymbol{\alpha}$ to multiple hyperparameters $\alpha_1, \ldots, \alpha_M$ corresponding to groups of weights $\mathcal{W}_1, \ldots, \mathcal{W}_M$, so it is consistent to the above sections.

$$p(\boldsymbol{\alpha}|\mathbf{D}) = \frac{p(\mathbf{D}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})}{p(\mathbf{D})}$$

The distribution $p(\mathbf{D}|\boldsymbol{\alpha})$ is called the evidence for $\boldsymbol{\alpha}$. Because the denominator does not affect the optimisation solution and $p(\boldsymbol{\alpha})$ is assumed to be uniform, these terms are ignored in the subsequent analysis. This means that we have to maximise the evidence $p(\mathbf{D}|\boldsymbol{\alpha})$ with respect to $\boldsymbol{\alpha}$. Firstly, we have to compute $p(\mathbf{D}|\boldsymbol{\alpha})$.

$$p(\mathbf{D}|\boldsymbol{\alpha}) = \int_{-\infty}^{\infty} p(\mathbf{D}|\boldsymbol{\omega})p(\boldsymbol{\omega}|\boldsymbol{\alpha})\ d\boldsymbol{\omega}. \tag{15}$$

In the E-step, $\mathcal{L}(q, \boldsymbol{\Omega})$ is maximised with respect to $q(\boldsymbol{\beta})$, in other word we minimise $KL(q||p)$ with respect to $q(\boldsymbol{\beta})$. In this case $KL(q||p) \approx 0$, thus $\log\left[p(\mathbf{D}|\boldsymbol{\Omega})p(\boldsymbol{\Omega})\right] \approx \mathcal{L}(q, \boldsymbol{\Omega})$. Therefore, $\log p(\mathbf{D}|\boldsymbol{\Omega})$ can be defined by equation (14) without the last two terms (which are derived from the component $\log p(\boldsymbol{\Omega})$). Ignoring the components which are independent of $\boldsymbol{\omega}$, we obtain:

$$\log p(\mathbf{D}|\boldsymbol{\omega}) = -\frac{1}{2}\sum_{t=1}^{T} \frac{\widetilde{c}}{\widetilde{d}_t}(y_t - f(\mathbf{x}_t, \boldsymbol{\omega}))^2 + const. \tag{16}$$

Substitute equations (3) and (16) to (15), we have:

$$p(\mathbf{D}|\boldsymbol{\alpha}) \propto \prod_{m=1}^{M} \left(\frac{\alpha_m}{2\pi}\right)^{\mathbf{W}_m/2} \int_{-\infty}^{\infty} \exp\left(-S(\boldsymbol{\omega})\right) d\boldsymbol{\omega}, \tag{17}$$

where

$$S(\boldsymbol{\omega}) = \boldsymbol{\eta}' E_{\mathbf{D}}(\boldsymbol{\omega}) + \boldsymbol{\alpha}' E_w(\boldsymbol{\omega}), \tag{18}$$

where $\boldsymbol{\eta}$, $\boldsymbol{\alpha}$, $E_{\mathbf{D}}(\boldsymbol{\omega})$, and $E_w(\boldsymbol{\omega})$ are column vectors, $\boldsymbol{\eta}'$ and $\boldsymbol{\alpha}'$ are the transposes of $\boldsymbol{\eta}$ and $\boldsymbol{\alpha}$ respectively, and $\boldsymbol{\eta}' E_{\mathbf{D}}(\boldsymbol{\omega})$ are the inner product of $\boldsymbol{\eta}$ and $E_{\mathbf{D}}(\boldsymbol{\omega})$:

$$E_{\mathbf{D}}(\boldsymbol{\omega}) = \left[E_{\mathbf{D}}^1(\boldsymbol{\omega}), \ldots, E_{\mathbf{D}}^T(\boldsymbol{\omega})\right]', \; E_{\mathbf{D}}^t(\boldsymbol{\omega}) = \frac{1}{2}\left(y_t - f(\boldsymbol{\omega}, \mathbf{x}_t)\right)^2$$

$$\boldsymbol{\eta} = \left[\frac{\widetilde{c}}{\widetilde{d}_1}, \ldots, \frac{\widetilde{c}}{\widetilde{d}_T}\right]'$$

$$E_w(\boldsymbol{\omega}) = \left[E_w^1(\boldsymbol{\omega}), \ldots, E_w^M(\boldsymbol{\omega})\right]', \; E_w^m(\boldsymbol{\omega}) = \frac{1}{2}\sum_{\omega \in \mathcal{W}_m} \omega^2$$

$$\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_M]'.$$

Note that in the equation of the overall error function $S(\boldsymbol{\omega})$, the scalar hyperparameter β in the standard evidence procedure is replaced by a T-dimensional vector $\boldsymbol{\eta}$, and hyperparameter $\boldsymbol{\alpha}$ is generalised to multiple hyperparameters $\alpha_1, \ldots, \alpha_M$. To evaluate the integral in equation (17), we do the same procedure as described in [12]. Finally, we obtain the log evidence as follows:

$$\log p(\mathbf{D}|\boldsymbol{\alpha}) = \sum_{m=1}^{M}\left(\frac{\mathbf{W}_m}{2}\log\alpha_m\right) - \frac{1}{2}\log\|\mathbf{A}\| - \boldsymbol{\eta}' E_{\mathbf{D}}(\boldsymbol{\omega}_{MP}) - \boldsymbol{\alpha}' E_w(\boldsymbol{\omega}_{MP}) + const,$$

where $\boldsymbol{\omega}_{MP}$ is local minimum of the $S(\boldsymbol{\omega})$ (of course, it is the local maximum of lower bound $\mathcal{L}(q, \boldsymbol{\Omega})$ as well). Matrix $\mathbf{A}$ is the Hessian of the overall error function:

$$\mathbf{A} = \sum_{t=1}^{T} \boldsymbol{\eta}_t \nabla\nabla E_{\mathbf{D}}^t(\boldsymbol{\omega}_{MP}) + \mathrm{diag}(\widehat{\boldsymbol{\alpha}}),$$

where $\boldsymbol{\eta}_t = \widetilde{c}/\widetilde{d}_t$, $\widehat{\boldsymbol{\alpha}}$ is a $\mathbf{W}$-dimensional vector: $\widehat{\boldsymbol{\alpha}} = [\widehat{\alpha}_1, \widehat{\alpha}_2, \ldots, \widehat{\alpha}_{\mathbf{W}}]$, $\widehat{\alpha}_i = \alpha_m$ if $i \in \mathcal{W}_m$, $i = 1, \ldots, \mathbf{W}$ and $m = 1, \ldots, M$. $\mathrm{diag}(\widehat{\boldsymbol{\alpha}})$ is a diagonal matrix with the elements of $\widehat{\boldsymbol{\alpha}}$ on the main diagonal.

Let us return to our main objective, which is to optimise $\log p(\mathbf{D}|\boldsymbol{\alpha})$. The first step is to compute its partial derivative with respect to $\boldsymbol{\alpha}$. The most difficult term is the log of the matrix determinant $\|\mathbf{A}\|$. Let $\lambda_1, \ldots, \lambda_{\mathbf{W}}$ be the eigenvalues of the data Hessian $H = \sum_{t=1}^{T} \boldsymbol{\eta}_t \nabla\nabla E_{\mathbf{D}}^t(\boldsymbol{\omega}_{MP})$. Then $\mathbf{A}$ has eigenvalues $\lambda_1 + \widehat{\alpha}_1, \ldots, \lambda_{\mathbf{W}} + \widehat{\alpha}_{\mathbf{W}}$, and

$$\begin{aligned} \frac{\partial}{\partial \alpha_m} \ln \|\mathbf{A}\| &= \frac{\partial}{\partial \alpha_m} \ln \left(\prod_{i=1}^{\mathbf{W}} (\lambda_i + \widehat{\alpha}_i) \right) = \frac{\partial}{\partial \alpha_m} \sum_{i=1}^{\mathbf{W}} \ln (\lambda_i + \widehat{\alpha}_i) \\ &= \sum_{i \in \mathcal{W}_m} \frac{1}{\lambda_i + \alpha_m} = \sum_{i \in \mathcal{W}_m} \left(A^{-1}\right)_{ii}, \, m = 1, \ldots, M. \end{aligned}$$

The derivative of the log evidence with respect to α_m is:

$$\frac{\partial}{\partial \alpha_m} \log p(\mathbf{D}|\boldsymbol{\alpha}) = -E_w^m(\boldsymbol{\omega}_{MP}) - \frac{1}{2} \sum_{i \in \mathcal{W}_m} \frac{1}{\lambda_i + \alpha_m} + \frac{\mathbf{W}_m}{2\alpha_m}.$$

Equating this to zero and rearranging give an implicit equation for α_m

$$\alpha_m = \frac{\gamma_m}{2E_w^m(\boldsymbol{\omega}_{MP})}, \, m = 1, \ldots, M, \tag{19}$$

where

$$\gamma_m = \sum_{i \in \mathcal{W}_m} \frac{\lambda_i}{\lambda_i + \alpha_m}, \tag{20}$$

is a measure of the *number of well-determined* parameters; see section 10.4 in [13].

3.4. Summary of training process

1. Chose initial values for $\widehat{c}$, $\widehat{d}$, and $\boldsymbol{\omega}$.
2. Update parameters of distribution $q(\boldsymbol{\beta})$ using equations (10) and (11).
3. Optimise the lower bound $\mathcal{L}(q, \boldsymbol{\Omega})$ w.r.t $\{\boldsymbol{\omega}, \widehat{c}, \widehat{d}, \boldsymbol{\alpha}\}$: partition these parameters into three groups $\left\{\widehat{c}, \widehat{d}\right\}$, $\{\boldsymbol{\omega}\}$, and $\{\boldsymbol{\alpha}\}$, and optimise each group with the others held fixed:
   (a) Optimise $\widehat{c}$, $\widehat{d}$ using scaled conjugate gradient.
   (b) Optimise $\boldsymbol{\omega}$ using scaled conjugate gradient.

(c) Optimise $\boldsymbol{\alpha}$ using equation (19).

(d) Repeat steps (a), (b) and (c) until convergence.

4. Repeat steps 2 and 3 until convergence.

We chose to terminate when either none of the changes at each update to $\boldsymbol{\omega}$ or $\log \alpha_m$ were greater than some threshold, here 10^{-6}, or a maximum number of iterations have been exceeded, depending on whichever occurs first.

4. Experimental results

The numerical experiments demonstrate the advantages and disadvantages of Student-t models over Gaussian ones. We tested on two forecasting tasks. The first is on a synthetic dataset which is similar to that in [4]. The second is a real life application to forecast forward gas prices in the UK market, provided by E.ON.

4.1. Model evaluation

In order to evaluate prediction performance of models, we computed three error measures: normalised mean squared error (NMSE), mean absolute percentage error (MAPE), and mean absolute error (MAE) which are defined by

$$\begin{aligned} e_{NMSE} &= \frac{\sum_{t=1}^{T} (y_t - \widehat{y}_t)^2}{\sum_{t=1}^{T} (y_t - E[y])^2} \\ e_{MAPE} &= \frac{1}{T} \sum_{t=1}^{T} \left| \frac{y_t - \widehat{y}_t}{y_t} \right| \times 100\% \\ e_{MAE} &= \frac{1}{T} \sum_{t=1}^{T} |y_t - \widehat{y}_t| , \end{aligned}$$

where y is the target data of test set, $\widehat{y}$ is the forecast, $E[y]$ is the mean of y, and T is the number of observations in the test set.

4.2. Synthetic data

We generated a dataset from the function $\text{sinc}(x) = (\sin(x))/x$ with additive Student-t noise. The dataset includes a training set (100 points at equally spaced intervals in $[-10, 10]$) and a test set (80 equally spaced noise-free points in $[-10, 10]$). The additive noise is drawn from a zero-mean Student-t distribution with one degree of freedom ($\nu = 1$) and scale parameter $\sigma = 0.02$.

We compared the prediction performance of four models: Gaussian MLP, our proposal Student-t MLP, Gaussian RBF, and Student-t RBF. The Gaussian MLP model was trained with the evidence procedure and the Student-t RBF was trained with the Tipping and Lawrence algorithm. Software for training Gaussian MLP/RBF can be found in [1]. Both Gaussian and Student-t RBF models had nine centres equally spaced in $[-10, 10]$ and fixed Gaussian basis functions $\phi_m(\mathbf{x}) = \exp\left\{-\left[(\mathbf{x} - \mathbf{x}_m)/2.0\right]^2\right\}$. Both MLP models had 5 hidden units and tanh activation functions. In training MLP Student-t, the maximum number of iteration for algorithms in step (3.a), (3.b), (3.c), (3.d), and (4) were 80, 40, 6, 5, and 40 respectively

Figure 1(a) shows shows the development of the log posterior $\log p(\boldsymbol{\Omega}|\mathbf{D})$ (ignoring the constant terms) during training of the Student-t MLP, indicating that our algorithm converges. Figure 1(b) shows the inferred noise distribution of Student-t and Gaussian MLP, compared with the true additive noise. The inferred noise distribution of the Student-t MLP is close to the real noise while that of the Gaussian MLP model is far from the real noise. This implies that the Student-t MLP model is capable of successfully learning noise parameters.

Figure 2 shows prediction results of the four models. In both MLP and RBF cases, models with Student-t noise outperform Gaussian noise. Table 1 provides prediction accuracy information, averaged over 10 trials. The table shows that the Student-t noise models are significantly better than Gaussian models. For

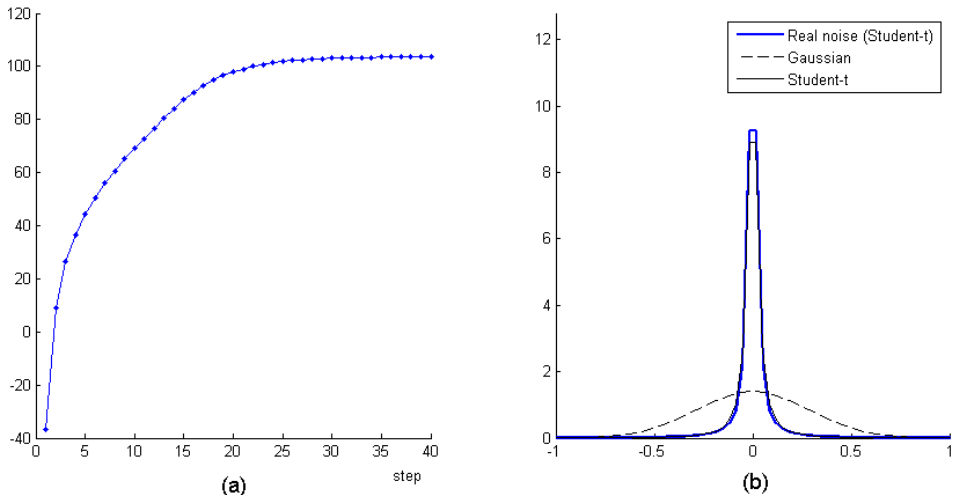

Figure 1: Results on synthetic dataset. (a) Development of negative log posterior $p(\mathbf{\Omega}|\mathbf{D})$ (ignoring the constant terms) in training Student-t MLP model. (b) The inferred noise distributions in Student-t MLP model and Gaussian MLP model, and the true noise distribution.

example, NMSE of the Student-t MLP model is 0.00373 while the equivalent value for the Gaussian MLP model is 0.19891. This proves the robustness to outliers of Student-t models.

The biggest disadvantage of our presented method is that it is computationally expensive. The average running time for Student-t MLP model for this case study was 525 (seconds), which is much longer than the others. (We ran experiments

Models	NMSE	MAPE	MAE	Running time (s)
Gaussian MLP	0.19891	163.230%	0.10712	1.0423
Student-t MLP	0.00373	29.544%	0.01144	545.2500
Gaussian RBF	0.19118	148.590%	0.07964	0.0078
Student-t RBF	0.01013	41.905%	0.02777	13.5190

Table 1: Errors and running time of forecast methods for synthetic dataset.

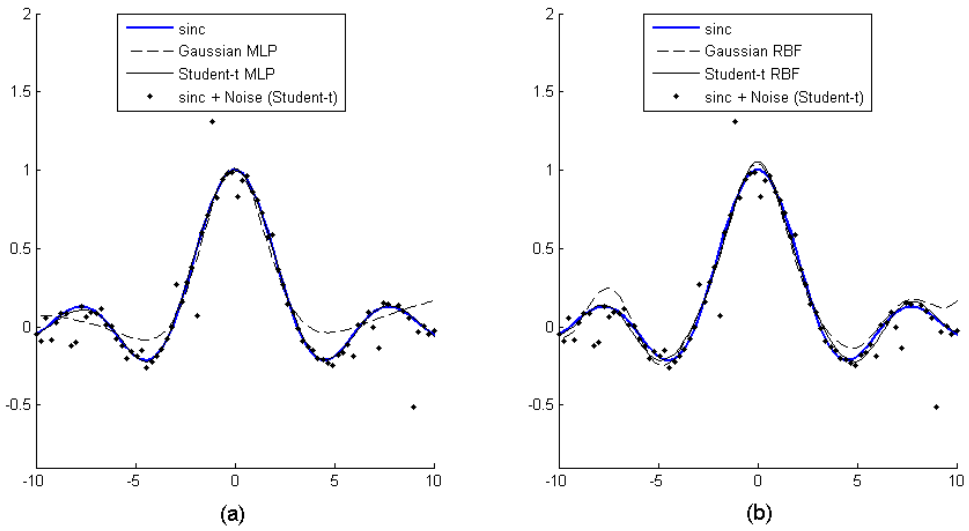

Figure 2: Synthetic dataset. (a) Data and predictions of the MLP models. (b) Data and predictions of the RBF models.

with code written in Matlab on a computer with Duo Core 1.66GHz CPU, RAM 1.5GB).

4.3. Gas forward price

The second experiment was to predict prices of monthly gas forward products. The monthly gas product is a forward contract for supplying gas in a single month in the future. In the UK energy market, it is possible to trade gas from one to six months ahead. There are six months of daily price data (approximately 130 data points) for each monthly gas product. For example, the July 2006 gas product can be traded from 03 Jan 2006 to 30 Jun 2006. To evaluate the behaviour of our method, 72 sub-datasets were used and we computed the average prediction results. We divided each forward product into three parts: each part was a test set of a sub-dataset. Each test set was associated to one training set which includes data of several forward products. Of course, all observations in a training set

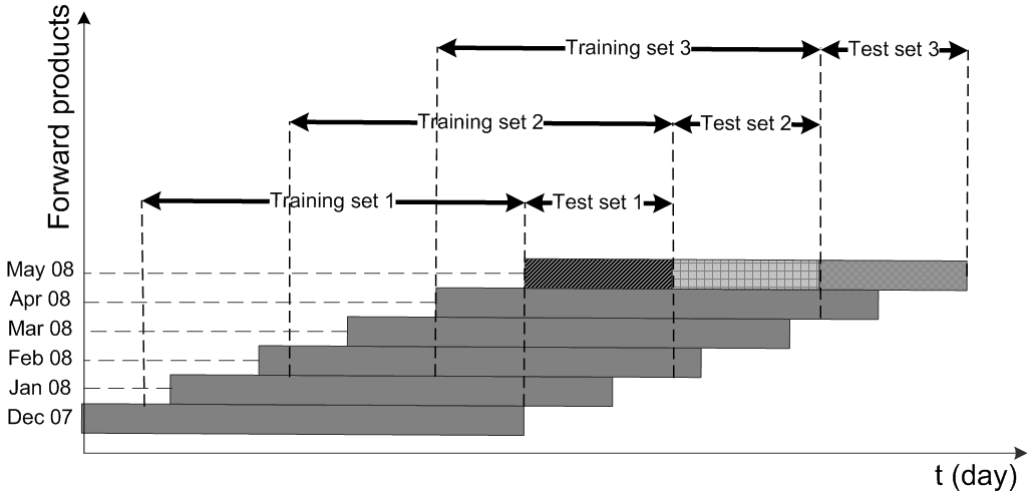

Figure 3: Allocation of training sets and test sets for gas price forecasts.

occurred before any in the associated test set. Figure 3 shows an example of how training sets and test sets were allocated. Data of the May-08 product was divided into three parts which were three test sets. In sub-dataset 1, the test set is the first 1/3 of the May-08 product, the training set includes data from Apr-08, Mar-08, Feb-08, and Jan-08 products.

Beside electricity demand and monthly forward gas prices, we were provided with a large number of exogenous variables which were potential candidates for inputs. However, only some of them are relevant. Using irrelevant variables as input will reduce the performance of the forecasting models. Therefore, in the training phase, automatic relevance determination (ARD) for Gaussian MLP was used to select relevant input variables [8]. The selected variables were $p_{t-1}, p_{t-2}, p^w_{t-1}, p^w_{t-2}$ for the first 36 sub-datasets and $p_{t-1}, p_{t-2}, p^s_{t-1}, p^s_{t-2}$ for the remaining, where p is the price of a monthly gas forward product, p^w and p^s are the price of a one-winter-ahead and one-summer-ahead forward products respectively, which are seasonal forward contracts for supplying gas in one winter/summer ahead. These variables were used as inputs for both Gaussian and Student MLPs.

Variables were normalised to zero mean and unit variance, then forecast time series were converted back to the original domain. The errors were computed on the original time series domain. We used MLPs with tanh activation functions and 7 hidden units. Determining the number of hidden units is based on the *number of well-determined* parameters during training the model by evidence procedure; see equation (20).

A random walk (RW) model was used as a benchmark to evaluate performance of forecast models. A RW is given by: $y_t = y_{t-1} + \varepsilon_{t-1}$, where ε is a zero-mean noise. The model predicts that tomorrow price will be equal to today price. We also computed the improvement ratio (IR) of errors of a method compared with corresponding errors of the benchmark model. For example, the IR of NMSE of a model M comparing with NMSE of the RW is given by:

$$IR_{NMSE}(M) = \frac{e_{NMSE}(RW) - e_{NMSE}(M)}{e_{NMSE}(RW)} \times 100\%.$$

Figure 4 shows the IR_{NMSE} of Student-t/Gaussian MLP models for 72 gas contract sub-datasets. The Student-t MLP model generally outperforms Gaussian MLP model. It obviously shows that IR_{NMSE} is around $[-20\%, 40\%]$ except datasets number 45, 46, 47, 62, and 65. On these irregular sub-datasets, the Gaussian MLP had extremely bad results while the Student-t MLP provided better performance. Figure 5 shows histograms of IR_{NMSE} for 72 sub-datasets. The number of these irregular sub-dataset was small, only 5 out of 72 sub-datasets. However, they contributed significantly to the overall results and make the average IR_{NMSE} of Gaussian MLP model worse. Table 2 shows average errors for all 72 sub-datasets. The improvement ratio of NMSE of Gaussian model is only 0.94% while the equivalent quantity of Student-t model is 10.11%. This proves that the Student-t model is more robust to outliers. It is superior to Gaussian models even in this real dataset where the noise is not expected to be an exact Student-t distribution.

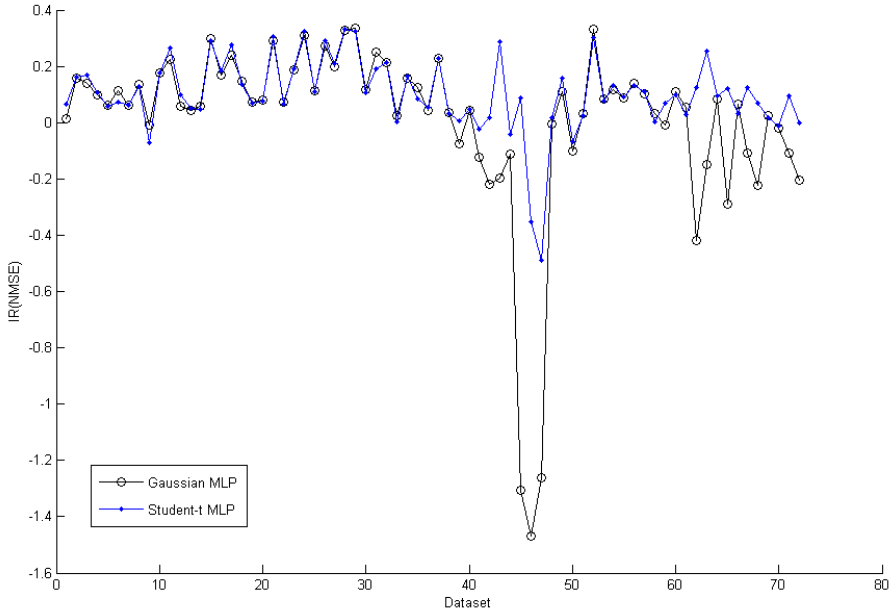

Figure 4: IR_{NMSE} of Student-t/Gaussian MLP models for 72 gas contract sub-datasets.

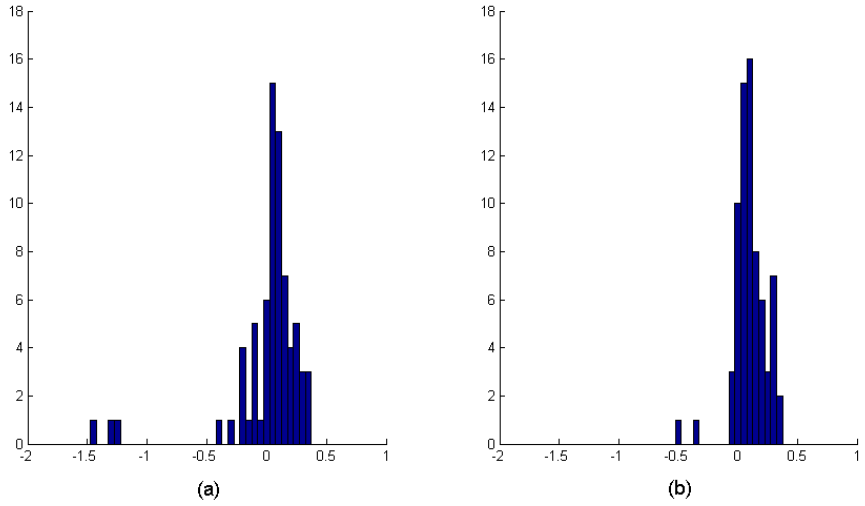

Figure 5: Histograms of IR_{NMSE} of 72 gas contract sub-datasets. (a) Gaussian MLP; (b) Student-t MLP

Models	IR(NMSE)	NMSE	MAPE	MAE
RW	0.00%	0.17566	1.823%	0.70766
Gaussian MLP	0.94%	0.17767	1.903%	0.71375
Student-t MLP	**10.11%**	**0.16389**	**1.763%**	**0.67884**

Table 2: Average errors of forecasting models for 72 gas forward price sub-datasets.

5. Conclusions

This paper presents a novel methodology for inferring parameters of Student-t probabilistic models. It was shown that it does not require the assum prior of a linear dependence of the output on model parameters. Removing this assumption leads to the fact that we can apply our framework to a large range of machine learning models. In particular, we can solve the inference problem of Student-t MLP model which cannot be solved by the previous methodologies in the literature.

It was shown experimentally that the Student-t models is less sensitive to outliers than Gaussian models. The Student-t models provide better predictions in both the synthetic data (where additive noise is a Student-t distribution) and the real data of gas forward price in the UK market (where noise has a heavy-tailed distribution but would not normally expected to be exactly a Student-t distribution).

The limitation of our presented method is its computational expense. It takes much longer to run than the other methods. However, in some real life applications, such as day-ahead price/demand energy prediction, this running time is acceptable considering the improved results.

6. Acknowledgments

Hang T. Nguyen would like to thank to E.ON for their financial support. The authors are grateful to Greg Payne, David Turner, Sally Friend, John Bateman, Matthew Cullen, Nick Sillito, Stuart Griffiths, Daniel Crispin and David Jones from E.ON for providing the datasets, valuable advice and supports. Some parts of program have been written using source code from the NETLAB toolbox (available at `http://www.ncrg.aston.ac.uk/netlab/index.php`).

References

[1] I.T. Nabney, NETLAB: Algorithms for Pattern Recognition, (Springer, Great Britain, 2002).

[2] H.T. Nguyen and I.T. Nabney, Combining the wavelet transform and forecasting models to predict gas forward prices, in: Proc. ICMLA'08, the 2008 Seventh International Conference on Machine Learning and Applications, (IEEE Computer Society, 2008) 311-317.

[3] H.T. Nguyen and I.T. Nabney, Energy forward price prediction with a hybrid adaptive model, in: CIFEr'09, IEEE Symposium on Computational Intelligence for Financial Engineering, (2009) 66-71.

[4] M.E. Tipping and N.D. Lawrence. Variational inference for Student-t models: Robust Bayesian interpolation and generalised component analysis, Neurocomputing, 69 (2005) 123-141.

[5] C.M. Bishop and M. Svensén, Robust Bayesian mixture modelling, Neurocomputing, 64(2005) 235-252.

[6] C. Archambeau and M. Verleysen, Robust Bayesian clustering, Neural Networks, 20(1) (2007) 129-138.

[7] J. Berger, Statistical Decision Theory and Bayesian Analysis (Springer, 1985).

[8] D.J.C. MacKay, Bayesian method for backprop network, in E. Domany, J. van Hemmen, and K. Schulten ed., Models of Neural Networks, III, (Springer, 1994) 211-254.

[9] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola and L. K. Saul. An introduction to variational methods for graphical models, Machine Learning, 37 (1999) 183-233.

[10] M. Abramowitz and I. A. Stegun, Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, (Dover, New York, 1964).

[11] M.F. Møller, A scaled conjugate gradient algorithm for fast supervised learning, Neural Networks, 6(4) (1993) 525-533.

[12] D.J.C. MacKay, Bayesian interpolation, Neural Computation, 4(3) (1992) 415-447.

[13] C.M. Bishop, Neural Networks for Pattern Recognition, (Oxford University Press, 1995).