Non-linear Regression for Bag-of-Words Data via Gaussian Process Latent Variable Set Model

Yuya Yoshikawa
Nara Institute of Science and Technology
8916-5, Takayama-cho, Ikoma-shi,
Nara, Japan
yoshikawa.yuya.yl9@is.naist.jp

Tomoharu Iwata
NTT Communication Science Laboratories, NTT Corporation
2-4, Hikaridai, Seika-cho,
Soraku-gun, Kyoto, Japan
iwata.tomoharu@lab.ntt.co.jp

Hiroshi Sawada
NTT Service Evolution Laboratories,
NTT Corporation
1-1 Hikari-no-Oka, Yokosuka-shi,
Kanagawa, Japan
sawada.hiroshi@lab.ntt.co.jp

Abstract

Gaussian process (GP) regression is a widely used method for non-linear prediction. The performance of the GP regression depends on whether it can properly capture the covariance structure of target variables, which is represented by kernels between input data. However, when the input is represented as a set of features, e.g. bag-of-words, it is difficult to calculate desirable kernel values because the co-occurrence of different but relevant words cannot be reflected in the kernel calculation. To overcome this problem, we propose a Gaussian process latent variable set model (GP-LVSM), which is a non-linear regression model effective for bag-of-words data. With the GP-LVSM, a latent vector is associated with each word, and each document is represented as a distribution of the latent vectors for words appearing in the document. We efficiently represent the distributions by using the framework of kernel embeddings of distributions that can hold high-order moment information of distributions without need for explicit density estimation. By learning latent vectors so as to maximize the posterior probability, kernels that reflect relations between words are obtained, and also words are visualized in a low-dimensional space. In experiments using 25 item review datasets, we demonstrate the effectiveness of the GP-LVSM in prediction and visualization.

1 Introduction

In many regression problems, the input is represented as a set of features. A typical example of such features is bag-of-words (BoW) representation, which is used for representing a document as a multiset of words appearing in the document while ignoring the order of the words. Gaussian process (GP) regression is a widely used method for such regression problems in various domains, e.g. natural language processing (Cohn and Specia 2013), time series analysis (Preotiuc-Pietro and Cohn 2013), computer vision (Kapoor et al. 2009) and data mining (Lampos and Aletras 2014). The performance of the GP regression generally depends on whether it can properly capture the covariance structure of target variables, which is represented by kernels between input data (e.g. documents). The GP regression for BoW representation has a major weakness that the co-occurrence of different but relevant words cannot be reflected in the kernel calculation. For example, when dealing with a problem of predicting ratings from item review texts, 'good' and 'excellent' are semantically similar and characteristic words for high rating reviews. Nevertheless, in the BoW representation, the two words might not affect the computation of the kernel value between the texts because many kernels, e.g. linear, polynomial and Gaussian RBF kernels, evaluate kernel values based on word co-occurrences in a document.

To overcome this weakness, we propose *a Gaussian process latent variable set model (GP-LVSM)*, which is a non-linear regression model effective for BoW data. Figure 1 illustrates the GP-LVSM. The GP-LVSM assumes that a latent vector is associated with each vocabulary term, and each document is represented as a distribution of the latent vectors for words appearing in the document. By using the framework of kernel embeddings (Smola et al. 2007), we can effectively represent the distributions without density estimation while preserving necessary information of distributions. In particular, the GP-LVSM maps each distribution into a reproducing kernel Hilbert space (RKHS), and generates a regression function from a Gaussian process with the covariance structures calculated by kernels between documents on the RKHS. The learning of the GP-LVSM is based on maximizing a posterior (MAP) estimation, which is performed by updating the latent vectors for words and other kernel parameters. The learned latent vectors for semantically similar words are located close to each other in the latent space, and we can obtain kernel values that reflect the semantics. As a result, the GP-LVSM can predict the target variables of unseen data using more rich and useful representation than BoW representation. Moreover, the GP-LVSM can be used as a supervised visualization method by plotting the two- or three-dimensional latent vectors for words.

In the experiments, we demonstrate the quantitative and qualitative effectiveness of the GP-LVSM on 25 item review datasets. First, we show that the GP-LVSM outperforms standard non-linear and linear regression methods in rating prediction. Then, we show that the performance of the GP-LVSM is robust for the dimensionality of the latent vectors for words, and we can obtain vector representations

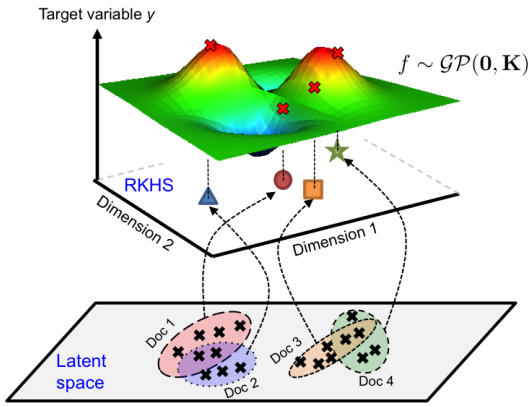

Figure 1: Illustration of GP-LVSM. Each word is represented as a latent vector denoted by '$\times$' in the latent space. The distributions of the documents are mapped into a reproducing kernel Hilbert space (RKHS). The target variables are expressed by a non-linear regression function generated from a Gaussian process.

for words on a quite low-dimensional space while achieving high prediction performance. Finally, we show that the GP-LVSM is also useful for visualizing words.

The GP-LVSM provides a general framework of solving regression problems for BoW data. Thus, the idea of the GP-LVSM can be applied to various machine learning problems, which have been solved based on GP regression such as multi-task learning (Bonilla, Chai, and Williams 2008) and active learning (Kapoor and Grauman 2007).

The rest of the paper is organized as follows. In Section 2, we review models and techniques related to the GP-LVSM. Section 3 introduces the framework of kernel embeddings of distributions, which is a key technique in the GP-LVSM. In Section 4, we explain the details of the GP-LVSM. In Section 5, we show the effectiveness and the properties of the GP-LVSM experimentally. Finally, we conclude with future work in Section 6.

2 Related Work

Topic models such as latent Dirichlet allocation (LDA) (Blei, Ng, and Jordan 2003) finds latent topic structures from BoW data. By learning the LDA, we obtain a low-dimensional and dense vector representation for each document. Supervised topic model (Blei and McAuliffe 2007) is a topic model of predicting target variables from documents, and uses the low-dimensional vectors for documents as features for prediction. We note that there are mainly two differences between the GP-LVSM and the supervised topic model, which would show that the GP-LVSM is better than the supervised topic model. The first one is that the GP-LVSM performs non-linear prediction, while the supervised topic model is linear prediction. The second one is that the GP-LVSM uses Vq parameters to represent a document, while the supervised topic model only uses K parameters, where V is the number of words in the document, q is the latent dimensionality for words and K is the number of topics. Generally, because of $Vq > K$, the GP-LVSM can capture the characteristic of the document in more detail than the supervised topic model.

The GP-LVSM is related to but different from the Gaussian process latent variable model (GP-LVM), which is used for dimension reduction (Lawrence 2004) and matrix factorization (Lawrence and Urtasun 2009). Given documents represented with bag-of-words, the GP-LVM learns a single latent vector for each document. Since the GP-LVM cannot obtain the latent vector of a new document, we cannot use it as a regression method. On the other hand, since the GP-LVSM learns a latent vector for each word, we can predict the target variable of a new document by using the representation of the document calculated from the latent vectors for words.

The GP-LVSM employs a framework of kernel embeddings of distributions for representing documents. The kernel embeddings have been used for extending kernel methods to distribution data. For example, the support measure machine is a method for kernel-based discriminative learning on distributions, which generalizes the support vector machine by kernel embeddings (Muandet et al. 2012). The one-class support measure machine is a group anomaly detection method that finds anomalous aggregated behaviors of objects (Muandet and Schölkopf 2013). The latent support measure machine is a generalization of the support measure machine, which can classify bag-of-words data (Yoshikawa, Iwata, and Sawada 2014). To the best of our knowledge, this paper is the first study that incorporates the kernel embeddings of distributions in Gaussian processes.

3 Kernel Embeddings of Distributions

In this section, we introduce the framework of the kernel embeddings of distributions. The kernel embeddings of distributions are to embed any probability distribution $\mathbb{P}$ on space $\mathcal{X}$ into a reproducing kernel Hilbert space (RKHS) $\mathcal{H}_k$ specified by kernel k, and the distribution is represented as element $m(\mathbb{P})$ in the RKHS. More precisely, when given distribution $\mathbb{P}$, the kernel embedding of the distribution $m(\mathbb{P})$ is defined as follows:

$$m(\mathbb{P}) := \mathbb{E}_{\mathbf{x}\sim\mathbb{P}}[k(\cdot, \mathbf{x})] = \int_{\mathcal{X}} k(\cdot, \mathbf{x}) d\mathbb{P} \in \mathcal{H}_k, \qquad (1)$$

where kernel k is referred to as *the embedding kernel*. It is known that kernel embedding $m(\mathbb{P})$ preserves the properties of probability distribution $\mathbb{P}$ such as mean, covariance and higher-order moments by using *characteristic kernels* (e.g. Gaussian RBF kernel) (Sriperumbudur and Gretton 2010).

In practice, although distribution $\mathbb{P}$ is unknown, we are given a set of samples $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ drawn from the distribution, where n is the number of samples. In this case, by interpreting sample set $\mathbf{X}$ as empirical distribution $\hat{\mathbb{P}} = \frac{1}{n}\sum_{i=1}^n \delta_{\mathbf{x}_i}(\cdot)$, where $\delta_{\mathbf{x}}(\cdot)$ is the Dirac delta function at point $\mathbf{x} \in \mathcal{X}$, empirical kernel embedding $m(\mathbf{X})$ is given by

$$m(\mathbf{X}) = \frac{1}{n}\sum_{i=1}^n k(\cdot, \mathbf{x}_i) \in \mathcal{H}_k, \qquad (2)$$

which can be approximated with an error rate of $||m(\mathbf{X}) - m(\mathbb{P})||_{\mathcal{H}_k} = O_p(n^{-\frac{1}{2}})$ (Smola et al. 2007). Unlike kernel density estimation, the error rate of the kernel embeddings is independent of the dimensionality of the given distribution.

4 Gaussian Process Latent Variable Set Model

In this section, we define the proposed model, the Gaussian Process Latent Variable Set Model (GP-LVSM), in detail. Then, we explain how the GP-LVSM is learned, and when a new input is given, how the GP-LVSM predicts the target variable of the input.

4.1 Model

Suppose that we are given a set of N training data $\mathcal{D} = \{(d_i, y_i)\}_{i=1}^N$, where d_i is a set of words appearing in the ith document and $y_i \in \mathbb{R}$ is its target variable. Here, d_i is bag-of-words with vocabulary set $\mathcal{V}$.

With the GP-LVSM, each word $v \in \mathcal{V}$ is represented by a q-dimensional latent vector $\mathbf{x}_v \in \mathbb{R}^q$, and the ith document is represented as a set of latent vectors for words appearing in the document $\mathbf{X}_i = \{\mathbf{x}_v\}_{v \in d_i}$. Then, using the framework of kernel embeddings described in Section 3, we can obtain representation of the ith document from $\mathbf{X}_i$ by $m(\mathbf{X}_i) = \frac{1}{|d_i|}\sum_{v \in d_i} k(\cdot, \mathbf{x}_v)$.

The GP-LVSM assumes the following regression model with Gaussian noise for a document and target pair (d_i, y_i):

$$f(\mathbf{X}_i) = \mathbf{w}^\top m(\mathbf{X}_i), \quad y_i = f(\mathbf{X}_i) + \epsilon, \tag{3}$$

where $\mathbf{w}$ is a weight vector of the regression and ϵ is a noise drawn from a Gaussian distribution with zero mean and precision parameter β, i.e., $\epsilon \sim \mathcal{N}(0, \beta^{-1})$.

We consider the probabilistic model for Eq. (3). Given a set of latent vectors $\mathbf{X} = \{\mathbf{x}_v\}_{v \in \mathcal{V}}$, weight vector $\mathbf{w}$, and a set of documents $\mathbf{d} = \{d_i\}_{i=1}^N$, the likelihood of target variables $\mathbf{y} = [y_1, y_2, \cdots, y_N]^\top$ is given by the following Gaussian distribution:

$$\begin{aligned} &p(\mathbf{y}|\mathbf{w}, \mathbf{X}, \mathbf{d}, \beta, \gamma) \\ &= \prod_{i=1}^N \frac{1}{\sqrt{2\pi\beta^{-1}}} \exp\left(-\frac{\beta}{2}(y_i - f(\mathbf{X}_i))^2\right), \end{aligned} \tag{4}$$

where γ is a parameter of embedding kernel k. We analytically marginalize out weight vector $\mathbf{w}$ by assuming the following Gaussian prior distribution with zero mean and precision parameter α:

$$p(\mathbf{w}|\alpha) = \frac{1}{\sqrt{2\pi\alpha^{-1}}} \exp\left(-\frac{\alpha}{2}\mathbf{w}^\top\mathbf{w}\right). \tag{5}$$

By doing the marginalization, we do not need to explore the optimal $\mathbf{w}$ in a potentially infinite dimensional space. The marginal likelihood of target variables $\mathbf{y}$ is also a Gaussian distribution, which can be obtained analytically because likelihood Eq. (4) and prior Eq. (5) are both Gaussian distributions. As a result, the marginal likelihood is given by

$$\begin{aligned} &p(\mathbf{y}|\mathbf{X}, \mathbf{d}, \alpha, \beta, \gamma) \\ &= \int p(\mathbf{y}|\mathbf{w}, \mathbf{X}, \mathbf{d}, \beta) p(\mathbf{w}|\alpha) \mathrm{d}\mathbf{w} \\ &= p(\mathbf{y}|\mathbf{0}, \alpha^{-1}\mathbf{M}\mathbf{M}^\top + \beta^{-1}\mathbf{I}), \end{aligned} \tag{6}$$

where $\mathbf{M} = [m(\mathbf{X}_1), m(\mathbf{X}_2), \cdots, m(\mathbf{X}_N)]^\top$. The mean and the covariance are derived by using $\mathbb{E}[\mathbf{y}] = \mathbf{M}\mathbb{E}[\mathbf{w}] = \mathbf{0}$ and $\mathbb{E}[\mathbf{y}\mathbf{y}^\top] = \mathbf{M}\mathbb{E}[\mathbf{w}\mathbf{w}^\top]\mathbf{M}^\top = \alpha^{-1}\mathbf{M}\mathbf{M}^\top$, respectively.

The (i, j) element of $\mathbf{M}\mathbf{M}^\top$ is inner product $\langle m(\mathbf{X}_i), m(\mathbf{X}_j)\rangle_{\mathcal{H}_k}$ of the kernel embeddings for ith and jth documents on RKHS $\mathcal{H}_k$ specified by embedding kernel k. The value of the inner product denotes the similarity between their documents. From Eq. (2), the inner product is given by

$$\begin{aligned} &\langle m(\mathbf{X}_i), m(\mathbf{X}_j)\rangle_{\mathcal{H}_k} \\ &= \left\langle \frac{1}{|d_i|}\sum_{s \in d_i} k(\cdot, \mathbf{x}_s), \frac{1}{|d_j|}\sum_{t \in d_j} k(\cdot, \mathbf{x}_t) \right\rangle_{\mathcal{H}_k} \\ &= \frac{1}{|d_i||d_j|}\sum_{s \in d_i}\sum_{t \in d_j} k(\mathbf{x}_s, \mathbf{x}_t). \end{aligned} \tag{7}$$

Using the inner product, we define kernels between documents. For each pair of document indexes (i, j), the kernel value between their documents is calculated as follows:

$$K_{ij} = \alpha^{-1}\langle m(\mathbf{X}_i), m(\mathbf{X}_j)\rangle_{\mathcal{H}_k} + \beta^{-1}\delta_{ij}, \tag{8}$$

where δ_{ij} is a function that returns 1 if i is equal to j and 0 otherwise. By defining $\mathbf{K}$ as a Gram matrix such that ith row and jth column is K_{ij}, marginal likelihood Eq. (6) can be rewritten as the following Gaussian distribution with zero mean and covariance $\mathbf{K}$.

$$\begin{aligned} &p(\mathbf{y}|\mathbf{X}, \mathbf{d}, \alpha, \beta, \gamma) \\ &= \frac{1}{(\sqrt{2\pi})^N\sqrt{\det\mathbf{K}}} \exp\left(-\frac{1}{2}\mathbf{y}^\top\mathbf{K}^{-1}\mathbf{y}\right). \end{aligned} \tag{9}$$

Choice of kernels between documents. Although Eq. (7) is a *linear* kernel between documents, we can extend it to *non-linear* kernels. An example of such kernels is the Gaussian RBF kernel with parameter $\lambda > 0$ between i and j documents, which is given by

$$\begin{aligned} &\exp\left(-\frac{\lambda}{2}||m(\mathbf{X}_i) - m(\mathbf{X}_i)||^2_{\mathcal{H}_k}\right) \\ &= \exp\Big(-\frac{\lambda}{2}(\langle m(\mathbf{X}_i), m(\mathbf{X}_i)\rangle_{\mathcal{H}_k} \\ &- 2\langle m(\mathbf{X}_i), m(\mathbf{X}_j)\rangle_{\mathcal{H}_k} + \langle m(\mathbf{X}_j), m(\mathbf{X}_j)\rangle_{\mathcal{H}_k})\Big), \end{aligned} \tag{10}$$

where the inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}_k}$ is calculated by Eq. (7). In the sections below, we use linear kernel Eq. (7).

4.2 Learning

We estimate the parameters of the GP-LVSM, latent vectors $\mathbf{X}$, precision parameters α and β, and kernel parameter γ.

For latent vectors $\mathbf{X}$, we place a Gaussian prior with zero mean and precision parameter ρ: $p(\mathbf{X}|\rho) \propto \prod_{v \in \mathcal{V}} \exp(-\frac{\rho}{2}||\mathbf{x}_v||_2^2)$. Then, the parameter estimation is performed by maximizing the following logarithm of the posterior of the parameters:

$$\begin{aligned} \mathcal{L}(\Theta) &= \log p(\mathbf{y}|\mathbf{X}, \mathbf{d}, \alpha, \beta, \gamma) + \log p(\mathbf{X}|\rho) \\ &\propto -\frac{1}{2}\mathbf{y}^\top\mathbf{K}^{-1}\mathbf{y} - \frac{1}{2}\log\det\mathbf{K} - \frac{\rho}{2}\sum_{v \in \mathcal{V}}||\mathbf{x}_v||_2^2, \end{aligned} \tag{11}$$

where, $\Theta = \{\mathbf{X}, \alpha, \beta, \gamma\}$ is a set of parameters to be estimated.

To maximize Eq. (11), we use the quasi-Newton method, which is a gradient-based optimization method (Liu and Nocedal 1989). For each word $v \in \mathcal{V}$, the gradient with respect to $\mathbf{x}_v$ can be calculated by

$$\frac{\partial \mathcal{L}(\Theta)}{\partial \mathbf{x}_v} = \sum_{i=1}^{N} \sum_{j=1}^{N} \left(\frac{\partial \mathcal{L}(\Theta)}{\partial \mathbf{K}} \right)_{ij} \frac{\partial K_{ij}}{\partial \mathbf{x}_v} - \rho \mathbf{x}_v. \quad (12)$$

The first factor $\frac{\partial \mathcal{L}(\Theta)}{\partial \mathbf{K}}$ is the gradient of $\mathcal{L}(\Theta)$ with respect to Gram matrix $\mathbf{K}$, which is given by

$$\frac{\partial \mathcal{L}(\Theta)}{\partial \mathbf{K}} = \frac{1}{2}\mathbf{K}^{-1}\mathbf{y}\mathbf{y}^\top\mathbf{K}^{-1} - \frac{1}{2}\mathbf{K}^{-1}, \quad (13)$$

where we note that the form of the gradient is independent of the choice of the embedding kernel. The second factor in Eq. (12), $\frac{\partial K_{ij}}{\partial \mathbf{x}_v}$, is the gradient of the kernel with respect to $\mathbf{x}_v$, which varies by the choice of the embedding kernel. An example of the embedding kernel is a Gaussian RBF embedding kernel with parameter $\gamma > 0$, which is defined as

$$k_\gamma(\mathbf{x}_s, \mathbf{x}_t) = \exp\left(-\frac{\gamma}{2}||\mathbf{x}_s - \mathbf{x}_t||_2^2\right). \quad (14)$$

In this case, the gradient of K_{ij} with respect to $\mathbf{x}_v$ is given by

$$\begin{aligned} \frac{\partial K_{ij}}{\partial \mathbf{x}_v} &= \frac{\alpha^{-1}}{|d_i||d_j|} \sum_{s \in d_i} \sum_{t \in d_j} k_\gamma(\mathbf{x}_s, \mathbf{x}_t) \quad (15) \\ &\times \begin{cases} \gamma(\mathbf{x}_t - \mathbf{x}_s) & (v = s \wedge v \neq t) \\ \gamma(\mathbf{x}_s - \mathbf{x}_t) & (v = t \wedge v \neq s) \\ \mathbf{0} & (v = t \wedge v = s) \end{cases} \end{aligned}$$

As with the estimation of latent vectors $\mathbf{X}$, α, β and γ can be estimated using the chain rule of Eq. (12). The gradients of the kernel with respect to α and β are given by

$$\frac{\partial K_{ij}}{\partial \alpha} = -\frac{\alpha^{-2}}{|d_i||d_j|} \sum_{s \in d_i} \sum_{t \in d_j} k(\mathbf{x}_s, \mathbf{x}_t), \quad (16)$$

$$\frac{\partial K_{ij}}{\partial \beta} = -\beta^{-2}\delta_{ij}, \quad (17)$$

which are independent of the choice of the embedding kernel. When the embedding kernel is Gaussian RBF kernel Eq. (14), the gradient with respect to kernel parameter γ is given by

$$\frac{\partial K_{ij}}{\partial \gamma} = -\frac{\alpha^{-1}}{2|d_i||d_j|} \sum_{s \in d_i} \sum_{t \in d_j} k_\gamma(\mathbf{x}_s, \mathbf{x}_t)||\mathbf{x}_s - \mathbf{x}_t||_2^2. \quad (18)$$

Using these gradients, we can obtain a local solution of the parameters by continuing to update the parameters in order until the improvement of Eq. (11) is converged. The computational cost to calculate the gradient for each word vector $\mathbf{x} \in \mathbf{X}$ is $O(N^2W^2q)$, where W is the average number of words in documents. However, when one wants to use large training data, by using stochastic gradient descent, the computational cost can be reduced to $O(W^2q)$.

Table 1: Specification of datasets. N_{tr} is the number of training data, N_{te} is the number of test data and $|\mathcal{V}|$ is the maximum number of vocabularies in training data. The number of development data is equal to N_{tr}.

	N_{tr}	N_{te}	$\lvert\mathcal{V}\rvert$
apparel	1,000	7,064	1,449
automotive	200	324	918
baby	800	2,635	1,250
beauty	800	1,274	1,747
books	1,000	9,927	1,953
camera	1,000	5,338	1,434
cell phones & service	300	409	1,501
computer & video games	600	1,550	2,000
dvd	1,000	9,892	2,184
electronics	1,000	9,883	1,341
gourmet food	400	756	1,713
grocery	500	1,612	1,565
health & personal care	1,000	5,154	2,165
jewelry & watches	500	951	1,313
kitchen & housewares	1,000	9,855	1,161
magazines	700	2,745	1,695
music	1,000	9,870	1,716
musical instruments	100	127	542
office products	100	220	569
outdoor living	400	781	1,141
software	500	1,375	1,759
sports & outdoors	900	3,859	1,360
tools & hardware	30	49	155
toys & games	1,000	9,947	1,883
video	1,000	9,878	2,012

4.3 Prediction

When a prediction is required, we can use the standard formula for prediction by a Gaussian process regression (Rasmussen and Williams 2005). Given a new document d_* consisting of words in $\mathcal{V}$, the predictive target variable y_* is given by

$$y_* = \mathbf{k}_*^\top \mathbf{K}^{-1}\mathbf{y}, \quad (19)$$

where $\mathbf{k}_*$ is a vector whose element is a kernel value between the new document and a training document, that is,

$$\mathbf{k}_* = [K_{*1}, K_{*2}, \cdots, K_{*N}]^\top. \quad (20)$$

Intuitively, the prediction is given by a weighted sum of training target variables $\mathbf{y}$, where the weights are calculated by kernel values between training documents.

Since the GP-LVSM provides the posterior distribution of the predictive target variable, we can calculate the variance of the predictive value, which is given by

$$\sigma_*^2 = K_{**} - \mathbf{k}_*^\top \mathbf{K}^{-1}\mathbf{k}_*. \quad (21)$$

This variance σ_*^2 can be used for measuring the confidence of the prediction: a smaller variance indicates a higher confidence for the prediction.

5 Experiments

In this section, we demonstrate the effectiveness of the GP-LVSM in prediction and visualization.

Table 2: Prediction errors (RMSE) and their standard deviations. Values in bold typeface are better than the others. The 'Average' row indicates the average errors and their standard deviations on all the datasets.

	GP-LVSM	GP regression	Ridge	Lasso	Elastic net
apparel	0.885 ± 0.015	0.936 ± 0.069	0.889 ± 0.018	0.898 ± 0.018	$\mathbf{0.868 \pm 0.019}$
automotive	$\mathbf{0.958 \pm 0.049}$	1.002 ± 0.075	0.972 ± 0.058	1.036 ± 0.059	0.989 ± 0.074
baby	$\mathbf{0.874 \pm 0.034}$	0.985 ± 0.039	0.933 ± 0.025	0.891 ± 0.031	0.919 ± 0.055
beauty	$\mathbf{0.890 \pm 0.030}$	0.938 ± 0.052	0.936 ± 0.019	0.914 ± 0.029	0.935 ± 0.044
books	$\mathbf{0.923 \pm 0.024}$	0.979 ± 0.023	1.111 ± 0.022	0.941 ± 0.027	0.937 ± 0.025
camera & photo	$\mathbf{0.870 \pm 0.013}$	0.962 ± 0.024	0.957 ± 0.010	0.898 ± 0.015	0.894 ± 0.019
cell phones & service	$\mathbf{0.843 \pm 0.034}$	0.942 ± 0.028	0.968 ± 0.038	0.943 ± 0.028	0.905 ± 0.026
computer & video games	$\mathbf{0.847 \pm 0.019}$	0.881 ± 0.021	0.987 ± 0.025	0.898 ± 0.019	0.932 ± 0.023
dvd	$\mathbf{0.942 \pm 0.040}$	0.968 ± 0.044	1.117 ± 0.035	0.945 ± 0.036	0.951 ± 0.038
electronics	$\mathbf{0.854 \pm 0.010}$	0.927 ± 0.045	0.983 ± 0.019	0.890 ± 0.012	0.882 ± 0.016
gourmet food	$\mathbf{0.931 \pm 0.045}$	0.947 ± 0.065	0.983 ± 0.048	0.986 ± 0.041	0.993 ± 0.052
grocery	0.927 ± 0.045	0.925 ± 0.082	0.940 ± 0.036	$\mathbf{0.922 \pm 0.049}$	0.953 ± 0.068
health & personal care	0.883 ± 0.019	$\mathbf{0.877 \pm 0.058}$	0.946 ± 0.015	0.902 ± 0.024	0.888 ± 0.022
jewelry & watches	$\mathbf{0.900 \pm 0.049}$	0.954 ± 0.082	0.924 ± 0.045	0.900 ± 0.043	0.945 ± 0.055
kitchen & housewares	$\mathbf{0.860 \pm 0.017}$	0.922 ± 0.065	0.956 ± 0.011	0.884 ± 0.008	0.873 ± 0.009
magazines	$\mathbf{0.835 \pm 0.022}$	0.882 ± 0.046	0.895 ± 0.019	0.877 ± 0.016	0.898 ± 0.028
music	0.960 ± 0.052	0.977 ± 0.065	1.128 ± 0.045	$\mathbf{0.954 \pm 0.064}$	0.956 ± 0.064
musical instruments	$\mathbf{0.966 \pm 0.144}$	1.025 ± 0.188	0.978 ± 0.128	1.031 ± 0.185	1.023 ± 0.189
office products	1.033 ± 0.108	1.041 ± 0.105	$\mathbf{1.025 \pm 0.114}$	1.108 ± 0.088	1.076 ± 0.113
outdoor living	$\mathbf{0.882 \pm 0.036}$	0.920 ± 0.074	0.952 ± 0.037	0.960 ± 0.050	0.955 ± 0.033
software	$\mathbf{0.806 \pm 0.014}$	0.915 ± 0.064	0.927 ± 0.015	0.883 ± 0.024	0.870 ± 0.025
sports & outdoors	$\mathbf{0.875 \pm 0.015}$	0.949 ± 0.048	0.950 ± 0.013	0.887 ± 0.015	0.896 ± 0.022
tools & hardware	0.892 ± 0.165	$\mathbf{0.884 \pm 0.260}$	0.918 ± 0.273	1.107 ± 0.256	0.974 ± 0.272
toys & games	$\mathbf{0.846 \pm 0.030}$	0.879 ± 0.050	0.908 ± 0.021	0.865 ± 0.019	0.851 ± 0.022
video	$\mathbf{0.844 \pm 0.027}$	0.867 ± 0.027	0.975 ± 0.019	0.891 ± 0.033	0.887 ± 0.029
Average	$\mathbf{0.893 \pm 0.052}$	0.939 ± 0.047	0.970 ± 0.064	0.936 ± 0.068	0.930 ± 0.054

5.1 Datasets and settings

For evaluation, we use 25 item review datasets obtained from Amazon.com, where each dataset corresponds to an item category on Amazon.com. Each review is represented with bag-of-words without short, low-frequency and stop words, and is associated with a rating ranging from $\{1, 2, \cdots, 5\}$. In our experiments, we use the bag-of-words as input document d and the standardized value of the rating as target variable y. Table 1 shows the specification of the datasets. For each dataset, we randomly choose five sets of training, development and test data from the whole of the dataset.

For comparison, we use four non-linear and linear regression methods: Gaussian Process (GP) regression, Ridge (Hoerl and Kennard 1970), Lasso (Tibshirani 1996) and Elastic net (Zou and Hastie 2005). With the GP regression, we use a Gaussian RBF kernel with additive noise term as follows:

$$K_{ij} = \alpha^{-1} \exp\left(-\frac{\gamma}{2}||\mathrm{vec}(d_i) - \mathrm{vec}(d_j)||_2^2\right) + \beta^{-1}\delta_{ij}, \tag{22}$$

where $\mathrm{vec}(\cdot)$ is a function that returns a vector with vocabulary length, and vth element of the vector is the frequency of the vth word in the given set. Parameters α, β and γ are estimated so as to maximize the marginal likelihood of the GP regression. Ridge (Hoerl and Kennard 1970), Lasso (Tibshirani 1996) and Elastic net (Zou and Hastie 2005) are standard linear regression models with different regularizers. We choose the parameters for these regularizers so as to minimize the prediction errors on development data. With the GP-LVSM, we learned the model with latent dimensionality $q \in \{1, 2, 4, 6, 8, 10\}$ and regularizer parameter $\rho \in \{10^{-2}, 10^{-1}, \cdots, 10^{2}\}$, and chose the optimal q and ρ so as to minimize the prediction errors on development data.

5.2 Prediction performance

Table 2 shows the prediction errors of ratings on test data. On 19 of 25 datasets, the GP-LVSM outperforms the other methods. On average of the prediction errors on all datasets, the GP-LVSM is the best method. This result indicates the GP-LVSM is robust and can perform better prediction than the other methods.

Next, we investigate how the choice of latent dimensionality q and regularizer parameter ρ of the GP-LVSM affects the prediction performance. Figure 2 shows the prediction errors of the GP-LVSM when varying the latent dimensionality q. Here, the regularizer parameter ρ was fixed at $\rho = 10$ to eliminate the effect of ρ. As shown in the figure, even with a very small latent dimensionality, the GP-LVSM achieves low prediction error. Even though q is relatively high, the errors are nearly unchanged compared to that of the best latent dimensionality. Thus, the performance of the GP-LVSM is robust for the dimensionality of the latent vectors for words,

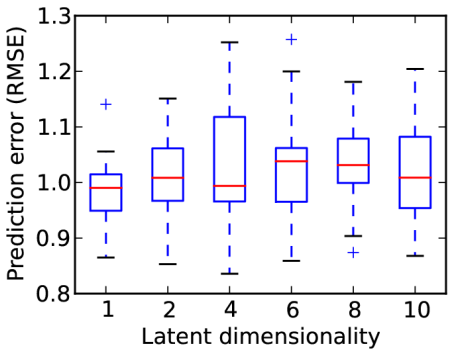

Figure 2: Prediction errors of GP-LVSM when varying latent dimensionality. The regularizer parameter is fixed at $\rho = 10$.

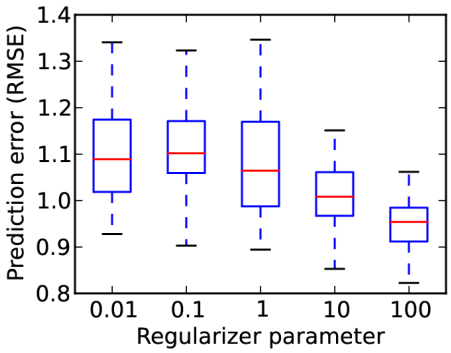

Figure 3: Prediction errors of GP-LVSM when varying regularizer parameter. The latent dimensionality is fixed at $q = 2$.

and we can obtain vector representations for words on a quite low dimensional space while achieving high prediction performance. Figure 3 shows the prediction errors when varying the regularizer parameter ρ. As opposed to the latent dimensionality, the predictive performance is sensitive to the choice of ρ. These results indicate that the GP-LVSM can archive the high predictive performance by focusing only on tuning the best ρ.

5.3 Visualization

Finally, we show that the GP-LVSM can visualize words using two- or three-dimensional latent vectors for words. In our experiments, since we predict the ratings from item reviews, it is expected that positive and negative words for the items are separated from each other. Figure 4 shows the visualization result of the latent vectors for words, which are trained on a 'software' dataset. Here, the regularizer parameter is fixed at $\rho = 0.1$. For understandability, we selected positive and negative words based on Loughran and McDonald Financial Sentiment Dictionaries [1], and visualized

[1] http://www3.nd.edu/~mcdonald/Word_Lists.html

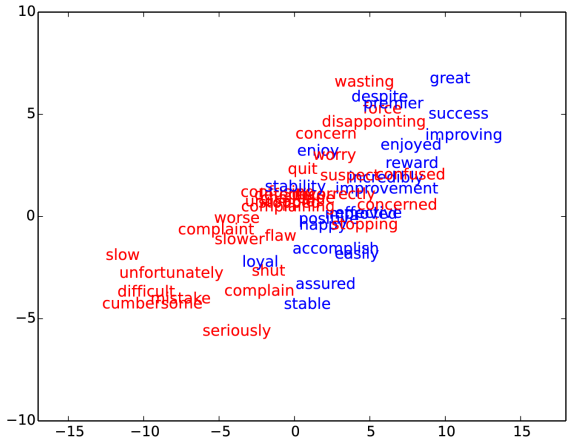

Figure 4: Visualization of latent vectors for words trained on 'software' dataset. Words in blue are positive words while words in red are negative words.

their latent vectors with blue and red colors. As shown in the figure, positive and negative words tend to gather in different regions. Therefore, 'great' and 'cumbersome', which are characteristic words in positive and negative polarity are far away from each other.

6 Conclusion

We have proposed a non-linear regression model for bag-of-words data, which we call a Gaussian process latent variable set model (GP-LVSM). The GP-LVSM represents each word as a latent vector, and each document as a distribution of the latent vectors for words appearing in the document. Then, the GP-LVSM maps each distribution into a reproducing kernel Hilbert space (RKHS) by using the framework of kernel embeddings of distributions, and generates a regression function from a Gaussian process with the covariance structures calculated by kernels between documents on the RKHS. Since the GP-LVSM can reflect the relations between words based on their latent vectors to the kernel values between documents, the GP-LVSM can improve the regression performance. In our experiments, we have shown that the GP-LVSM outperforms conventional linear and non-linear regression methods on the rating prediction using 25 item review datasets, and is useful for visualizing words by using the learned latent vectors for the words.

In future work, we will employ stochastic gradient descent to reduce the computational costs of the learning. Then, we will further confirm the effectiveness of the GP-LVSM by applying it to a varied domain of data, not limited to text.

Acknowledgment

This work was supported by JSPS Grant-in-Aid for JSPS Fellows (259867).

References

Blei, D. M., and McAuliffe, J. D. 2007. Supervised Topic Models. In *NIPS*.

Blei, D. M.; Ng, A. Y.; and Jordan, M. 2003. Latent Dirichlet Allocation. *The Journal of Machine Learning Research* 3(4-5):993–1022.

Bonilla, E.; Chai, K.; and Williams, C. 2008. Multi-task Gaussian Process Prediction. In *NIPS*.

Cohn, T., and Specia, L. 2013. Modelling Annotator Bias with Multi-task Gaussian Processes: An Application to Machine Translation Quality Estimation. In *ACL*.

Hoerl, A., and Kennard, R. 1970. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics* 12(1):55–67.

Kapoor, A., and Grauman, K. 2007. Active Learning with Gaussian Processes for Object Categorization. *ICCV*.

Kapoor, A.; Grauman, K.; Urtasun, R.; and Darrell, T. 2009. Gaussian Processes for Object Categorization. *International Journal of Computer Vision* 88(2):169–188.

Lampos, V., and Aletras, N. 2014. Predicting and Characterising User Impact on Twitter. In *ACL*, 405–413.

Lawrence, N., and Urtasun, R. 2009. Non-linear Matrix Factorization with Gaussian Processes. In *ICML*.

Lawrence, N. 2004. Gaussian Process Latent Variable Models for Visualisation of High Dimensional Data. In *NIPS*.

Liu, D. C., and Nocedal, J. 1989. On the Limited Memory BFGS Method for Large Scale Optimization. *Mathematical Programming* 45(1-3):503–528.

Muandet, K., and Schölkopf, B. 2013. One-Class Support Measure Machines for Group Anomaly Detection. In *UAI*.

Muandet, K.; Fukumizu, K.; Dinuzzo, F.; and Schölkopf, B. 2012. Learning from Distributions via Support Measure Machines. In *NIPS*.

Preotiuc-Pietro, D., and Cohn, T. 2013. A Temporal Model of Text Periodicities Using Gaussian Processes. In *EMNLP*, number October, 977–988.

Rasmussen, C. E., and Williams, C. K. I. 2005. *Gaussian Processes for Machine Learning*. The MIT Press.

Smola, A.; Gretton, A.; Song, L.; and Schölkopf, B. 2007. A Hilbert Space Embedding for Distributions. *Algorithmic Learning Theory*.

Sriperumbudur, B., and Gretton, A. 2010. Hilbert Space Embeddings and Metrics on Probability Measures. *The Journal of Machine Learning Research* 11:1517–1561.

Tibshirani, R. 1996. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B* 58(1):267–288.

Yoshikawa, Y.; Iwata, T.; and Sawada, H. 2014. Latent Support Measure Machines for Bag-of-Words Data Classification. In *NIPS*.

Zou, H., and Hastie, T. 2005. Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society: Series B* 301–320.