Probit Classifiers with a Generalized Gaussian Scale Mixture Prior

Guoqing Liu[†] **Jianxin Wu**[†] **Suiping Zhou**[‡]
[†]Nanyang Technological University [‡]Teesside University
Singapore UK
liug0008@e.ntu.edu.sg jxwu@ntu.edu.sg S.Zhou@tees.ac.uk

Abstract

Most of the existing probit classifiers are based on sparsity-oriented modeling. However, we show that sparsity is not always desirable in practice, and only an appropriate degree of sparsity is profitable. In this work, we propose a flexible probabilistic model using a generalized Gaussian scale mixture prior that can promote an appropriate degree of sparsity for its model parameters, and yield either sparse or non-sparse estimates according to the intrinsic sparsity of features in a dataset. Model learning is carried out by an efficient modified maximum a posteriori (MAP) estimate. We also show relationships of the proposed model to existing probit classifiers as well as iteratively re-weighted l_1 and l_2 minimizations. Experiments demonstrate that the proposed method has better or comparable performances in feature selection for linear classifiers as well as in kernel-based classification.

1 Introduction

In binary classification, we are given a training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \{1, -1\}$ and N is the number of observations. The goal is to learn a mapping $y = f(\mathbf{x}; \boldsymbol{\omega})$ from the inputs to the targets based on $\mathcal{D}$, where $\boldsymbol{\omega}$ is the model parameter.

In this work, we pay attention to probit classifiers [Figueiredo, 2003; Kabán, 2007; Chen *et al.*, 2009], in which the corresponding mapping is specified by a likelihood model associated with priors over parameters of the model. Specifically, the $f(\mathbf{x}; \boldsymbol{\omega})$ is formulated by

$$f(\mathbf{x}; \boldsymbol{\omega}) = P(y = 1|\mathbf{x}) = \Psi\big(\boldsymbol{\Phi}(\mathbf{x})^T \boldsymbol{\omega}\big), \tag{1}$$

where $\Psi(\cdot)$ is a normal cumulative density function used as the link function; $\boldsymbol{\omega} \in \mathbb{R}^n$ is the unknown parameter vector of the likelihood model; $\boldsymbol{\Phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_n(\mathbf{x}))^T$ is a vector of n fixed functions of the input, usually called features, and $\boldsymbol{\Phi}(\mathbf{x})^T \boldsymbol{\omega}$ may contain the following well-known formulations:

- Linear classifiers, where $\boldsymbol{\Phi}(\mathbf{x}) = (1, x_1, \ldots, x_d)^T$ and $n = d + 1$ [Kabán, 2007].
- Nonlinear classifiers, in which $\phi_i(\mathbf{x}), i = 1, \ldots, n$, are fixed basis functions; usually $\phi_1(\mathbf{x}) = 1$.
- Kernel-based classifiers, in which $\boldsymbol{\Phi}(\mathbf{x}) = (1, K(\mathbf{x}, \mathbf{x}_1), \cdots, K(\mathbf{x}, \mathbf{x}_N))^T$, and $K(\mathbf{x}, \mathbf{x}_i)$ are Mercer kernel functions; here $n = N + 1$ [Tipping, 2001].

In previous work, sparsity of the learned model is always expected. In order to obtain a sparse $f(\mathbf{x}; \boldsymbol{\omega})$, previous work has adopted various sparsity-inducing priors over $\boldsymbol{\omega}$ in probabilistic modeling. Most of these priors belong to the Gaussian scale mixture (GSM) distributions: Laplacian (or Gaussian-exponential) distribution has been widely used as a sparsity-inducing prior in various contexts [Figueiredo, 2003; Kabán, 2007], based on which a parameter-free Gaussian-Jeffreys' prior was further proposed in [Figueiredo, 2003]; two versions of Student-*t* (or Gaussian-inverse gamma) priors were utilized in [Chen *et al.*, 2009; Tipping, 2001], respectively; more recently, [Caron and Doucet, 2008; Griffin and Brown, 2010] paid attention to a Gaussian-gamma distribution. Besides, [Garrigues and Olshausen, 2010] proposed a Laplacian scale mixture (LSM) distribution to induce group sparsity, and [Raykar and Zhao, 2010] proposed a discrete mixture prior which is partially non-parametric.

2 Sparsity Is *not* Always Desirable

Although most of the existing probit classifications are based on sparsity-oriented modeling, it is important to ask: is sparsity always desirable? Our answer is no, which is illustrated by a set of toy examples.

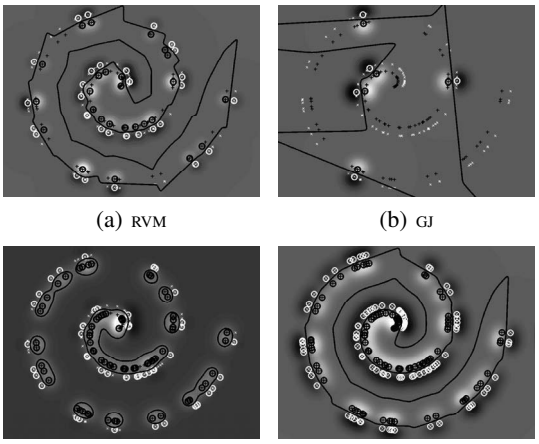
2.1 An Illustrative Example

Here we compare four kernel-based classifiers, including RVM with a Student-*t* prior in [Tipping, 2001], LAP with a Laplacian prior in [Figueiredo, 2003], GJ with a Gaussian-Jeffreys' prior also in [Figueiredo, 2003], and GGIG with a generalized Gaussian scale mixture (GGSM) prior that we will propose in Section 4. We use the Gaussian kernel in these experiments. These models are tested in two artificial datasets. The spiral dataset is generated by the Spider Toolbox[1], which can only be separated by spiral decision bound-

[1]Spider machine learning toolbox, Jason Weston, Andre Elisseeff, Gökhan BakIr, Fabian Sinz, Available: `http://www.kyb.tuebingen.mpg.de/bs/people/spider/`

aries. The cross dataset samples from two equal Gaussian distributions with one being rotated by 90°, and an "X" type boundary is expected. Parameter setting will be detailed in Section 6.2.

Fig. 1 demonstrates the behaviors of four models on the spiral data. Each model makes prediction only based on kernel functions corresponding to the samples with black or white circles in the figure. RVM, as well as LAP and GJ, cannot recover the correct decision boundary. In this case there is few information redundancy among features (i.e., kernel function base), and almost all of the kernel functions are useful for prediction. Under this situation, models with sparsity-inducing priors would drop helpful information. The more sparsity-inducing priors one uses, the worse predictions one would obtain. The proposed GGIG with parameter $q = 2.0$[2] encourages non-sparse solutions. As a result, we lose few information, and obtain a good decision boundary. Different from the spiral data, there is much redundant information in the cross data. As shown in Fig. 2, comparing to RVM and LAP, our model generates a similar decision boundary by automatically choosing $q = 0.1$ which is sparsity-inducing, but uses less kernel functions. This is because the priors used by LAP and RVM are not sparsity-encouraging enough. In contrary, due to excessive sparsity-inducing of the Gaussian-Jeffreys' prior, GJ uses too few kernel functions.



(a) RVM (b) GJ (c) LAP (d) GGIG ($q = 2.0$)

Figure 1: Experimental results in the spiral data.

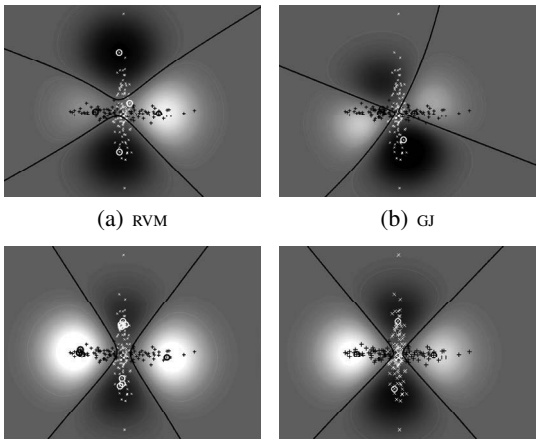
We believe *different problems need different degrees of sparsity*, which are determined by the information redundancy among features in the data. When features inherently encode orthogonal characterizations of a problem, enforcing sparsity would lead to the discarding of useful information and the degrading of the generalization ability. Only a proper degree of sparsity for a specific problem is profitable.

2.2 Adjusting Induced Sparsity of Priors

Thus, in probit classifiers, we want to *adjust the degree of induced sparsity from priors in a data-dependent way*. Usually, induced sparsity can be regulated by two factors: kurtosis and variance of a prior. A high kurtosis distribution has a sharper peak and longer, fatter tails. When its excess kurtosis is bigger than 0, a large fraction of parameters are expected to be zero, therefore it induces sparsity. This factor enables adjustment of sparsity by tuning the shape parameters of the prior distribution. Variance could also influence the degree of induced sparsity. In a zero mean prior, a small variance indicates that a parameter is distributed closely around zero, which encourages sparsity of the parameter.

We could view kurtosis of a prior as corresponding to the structure of a penalization used in regularization methods, which determines whether or not a sparse solution is encouraged in a problem. While variance is related to the trade-off parameter, which is used to re-scale the penalization, and balance the model fitness with regularization. Hence, both kurtosis and variance are important for controlling the degree of induced sparsity, and a mis-adjustment of either one may result in bad solutions.

[2] q is a shape parameter of the GGSM prior used in GGIG, which will be introduced in Section 3. It controls the sparsity level, and is determined automatically in our algorithm.



(a) RVM (b) GJ (c) LAP (d) GGIG ($q = 0.1$)

Figure 2: Experimental results in the cross data.

2.3 Our Contributions

We introduce here a family of generalized Gaussian scale mixture (GGSM) distributions for $\boldsymbol{\omega}$ in Section 3. Compared to GSM and LSM, an additional shape parameter q is used by GGSM, which is continuously defined in $(0,\ 2]$. Both kurtosis and variance of GGSM can be flexibly regulated by q, so does the induced sparsity. This family includes as special cases GSM and LSM when $q = 1$ and 2, respectively. This gives GGSM an opportunity to promote the appropriate degree of sparsity in a data-dependent way. The proposed model using a GGSM prior can yield either sparse or non-sparse estimates according to the sparsity among features in a dataset. The estimation of $\boldsymbol{\omega}$ with arbitrary $q \in (0, 2]$ is carried out by an efficient modified MAP algorithm, which is detailed in Section 4. Its convergence is also analyzed.

3 Modeling

3.1 Likelihood

Given a training dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, let us consider the corresponding vector of hidden variables $\mathbf{z} = [z_1, \ldots, z_N]^T$,

where $z_i = \mathbf{\Phi}(\mathbf{x}_i)^T\boldsymbol{\omega} + \xi_i$, and $\{\xi_i\}_{i=1}^N$ is a set of independent zero-mean and unit-variance Gaussian samples. If we know $\mathbf{z}$, we could obtain a simple linear regression likelihood with unit noise variance $\mathbf{z}|\boldsymbol{\omega} \sim \mathcal{N}(\mathbf{z}|\mathbf{\Phi}\boldsymbol{\omega}, \mathbf{I})$, i.e.,

$$p(\mathbf{z}|\boldsymbol{\omega}) = (2\pi)^{-\frac{n}{2}} \exp\Big(-\frac{1}{2}\|\mathbf{z} - \mathbf{\Phi}\boldsymbol{\omega}\|_2^2\Big). \tag{2}$$

This enlightens us on the use of the expectation maximization (EM) algorithm to estimate $\boldsymbol{\omega}$ by treating $\mathbf{z}$ as missing data. Here $\mathbf{\Phi} = [\mathbf{\Phi}(\mathbf{x}_1), \ldots, \mathbf{\Phi}(\mathbf{x}_N)]^T$.

3.2 Generalized GSM Prior

We assume that the components of $\boldsymbol{\omega}$ are independent and identically distributed,

$$p(\boldsymbol{\omega}) = \prod_{i=1}^{n} p(\omega_i|q)$$

and each ω_i follows a GGSM prior, i.e.,

$$p(\omega_i|q) \quad = \quad \int \mathcal{GN}(\omega_i|\lambda_i, q)p(\lambda_i)d\lambda_i, \tag{3}$$

where $\mathcal{GN}(\omega_i|\lambda_i, q)$ denotes the generalized Gaussian distribution

$$\mathcal{GN}(\omega_i|\lambda_i, q) = \frac{q}{2\lambda_i^{\frac{1}{q}}\Gamma(\frac{1}{q})} \exp\{-\frac{|\omega_i|^q}{\lambda_i}\}, \tag{4}$$

with a variance parameter λ_i and a shape parameter $q \in (0,\, 2]$; λ_i is a positive random variable with probability $p(\lambda_i)$ which will be defined later.

Compared to GSM and LSM, an additional shape parameter q is used by GGSM, which is continuously defined in $(0,\, 2]$. It is easy to see that, when $q = 1, 2$, GGSM would be specialized as LSM and GSM, respectively. This gives us an opportunity to generate more appropriate priors than GSM and LSM in a data-dependent way. More specifically, the variance and kurtosis of $\mathcal{GN}(\omega_i|\lambda_i, q)$ are given by

$$\mathbb{K}ur(\omega_i) = \frac{\Gamma(\frac{5}{q})\Gamma(\frac{1}{q})}{\Gamma(\frac{3}{q})^2}, \quad \mathbb{V}ar(\omega_i) = \frac{\lambda_i^{\frac{2}{q}}\Gamma(\frac{3}{q})}{\Gamma(\frac{1}{q})}.$$

Both of them can be regulated by q. Hence, when associated with the same $p(\lambda_i)$, GGSM is more flexible than GSM and LSM, and a proper degree of sparsity may be induced with an appropriate q. In this work, q is learned in a data-dependent fashion on a separate validation set $\mathcal{Q}$. For a specific problem, we search the best q in $\mathcal{Q}$ by cross validation. In our experiments, we will use $\mathcal{Q} = \{0.1, 0.5, 1.0, 1.5, 2.0\}$.

As to priors for the variance parameter λ_i, conjugate or conditionally-conjugate distributions are always chosen due to their computational convenience. In the following sections, we focus on a special case of the above model, where $\lambda_i = \lambda$, $i = 1, \ldots, n$, and λ is imposed by an inverse gamma distribution, i.e., $\lambda \sim \mathcal{IG}(\lambda|a, b)$.

4 Estimation Method

The MAP estimation is a straightforward approach to learn parameters of the above probabilistic model; that is

$$\arg\max_{\boldsymbol{\omega}} p(\boldsymbol{\omega}, \mathbf{y}, \mathbf{z}, \lambda)$$

where $p(\boldsymbol{\omega}, \mathbf{y}, \mathbf{z}, \lambda) = p(\mathbf{z}|\boldsymbol{\omega})p(\boldsymbol{\omega}|\lambda)p(\lambda)p(\mathbf{y}|\mathbf{z})$. However, the resulting optimization is computationally challenging, unless $q = 1, 2$. To address this problem, we use a technique called the minorization method [Hunter and Lange, 2004] to propose a modified MAP estimation, where $p(\boldsymbol{\omega}, \mathbf{y}, \mathbf{z}, \lambda)$ is replaced by a $\underline{p}(\boldsymbol{\omega}, \tilde{\boldsymbol{\omega}}, \mathbf{y}, \mathbf{z}, \lambda)$ which minorizes $p(\cdot)$ and can be easily maximized. Function $\underline{p}(\cdot)$ satisfies two conditions for all $\boldsymbol{\omega} \in \mathbb{R}^n$:

$$p(\boldsymbol{\omega}, \mathbf{y}, \mathbf{z}, \lambda) \geq \underline{p}(\boldsymbol{\omega}, \tilde{\boldsymbol{\omega}}, \mathbf{y}, \mathbf{z}, \lambda), \quad \forall \tilde{\boldsymbol{\omega}} \in (\mathbb{R}^+)^n$$
$$p(\boldsymbol{\omega}, \mathbf{y}, \mathbf{z}, \lambda) = \underline{p}(\boldsymbol{\omega}, |\boldsymbol{\omega}|, \mathbf{y}, \mathbf{z}, \lambda),$$

and a variational parameter $\tilde{\boldsymbol{\omega}} \in (\mathbb{R}^+)^n$ is introduced. At each iteration, we first choose this parameter as the current value of $\boldsymbol{\omega}$, and find the optimal update for $\boldsymbol{\omega}$ by

$$\arg\max_{\boldsymbol{\omega}} \underline{p}(\boldsymbol{\omega}, \tilde{\boldsymbol{\omega}}, \mathbf{y}, \mathbf{z}, \lambda). \tag{5}$$

We then update $\tilde{\boldsymbol{\omega}}$ with the new $|\boldsymbol{\omega}|$. The algorithm is stopped until the distance between $|\boldsymbol{\omega}|$ and $\tilde{\boldsymbol{\omega}}$ is less than some threshold.

4.1 Derivation of $\underline{p}(\boldsymbol{\omega}, \tilde{\boldsymbol{\omega}}, \mathbf{y}, \mathbf{z}, \lambda)$

Constructing a proper $\underline{p}(\boldsymbol{\omega}, \tilde{\boldsymbol{\omega}}, \mathbf{y}, \mathbf{z}, \lambda)$ is important for the minorization learning. In this work, we let

$$\underline{p}(\boldsymbol{\omega}, \tilde{\boldsymbol{\omega}}, \mathbf{y}, \lambda, \mathbf{z}) \equiv p(\mathbf{z}|\boldsymbol{\omega})\underline{p}(\boldsymbol{\omega}, \tilde{\boldsymbol{\omega}}, \lambda)p(\lambda)p(\mathbf{y}|\mathbf{z}), \tag{6}$$

where $\underline{p}(\boldsymbol{\omega}, \tilde{\boldsymbol{\omega}}, \lambda)$ is a lower bound of $p(\boldsymbol{\omega}|\lambda)$, and is induced by the weighted arithmetic and geometric mean inequality[3]:

$$\underline{p}(\boldsymbol{\omega}, \tilde{\boldsymbol{\omega}}, \lambda) = C_{\lambda,q} \exp\Big(-\frac{q}{2\lambda}\big(\sum_{i=1}^{n} \tilde{\omega}_i^{q-2}\omega_i^2 + g(\tilde{\boldsymbol{\omega}})\big)\Big) \tag{7}$$

with $\tilde{\boldsymbol{\omega}} = (\tilde{\omega}_1, \cdots, \tilde{\omega}_n)^T, \quad \tilde{\omega}_i \in \mathbb{R}^+$, and

$$C_{\lambda,q} = \Big(\frac{q}{2\lambda^{\frac{1}{q}}\Gamma(\frac{1}{q})}\Big)^n, \quad g(\tilde{\boldsymbol{\omega}}) = \frac{2-q}{q}\sum_{i=1}^{n} \tilde{\omega}_i^q.$$

Since $p(\boldsymbol{\omega}|\lambda) \geq \underline{p}(\boldsymbol{\omega}, \tilde{\boldsymbol{\omega}}, \lambda)$, and $p(\boldsymbol{\omega}|\lambda) = \underline{p}(\boldsymbol{\omega}, \tilde{\boldsymbol{\omega}}, \lambda)$, if and only if $|\boldsymbol{\omega}| = \tilde{\boldsymbol{\omega}}$, we conclude that $\underline{p}(\boldsymbol{\omega}, \tilde{\boldsymbol{\omega}}, \mathbf{y}, \mathbf{z}, \lambda)$ is minorizing the original joint density function. Hereunto, we have generated a modified MAP estimation implemented by $\max_{\boldsymbol{\omega}, \tilde{\boldsymbol{\omega}}} \underline{p}(\boldsymbol{\omega}, \tilde{\boldsymbol{\omega}}, \mathbf{y}, \mathbf{z}, \lambda)$, which is equivalent to iteratively maximizing the following objective function with respect to $\boldsymbol{\omega}$ and $\tilde{\boldsymbol{\omega}}$:

$$h(\boldsymbol{\omega}, \tilde{\boldsymbol{\omega}}) = -\frac{1}{2}\|\mathbf{z} - \mathbf{\Phi}\boldsymbol{\omega}\|_2^2 - \frac{q}{2\lambda}\big(\sum_{i=1}^{n} \tilde{\omega}_i^{q-2}\omega_i^2 + g(\tilde{\boldsymbol{\omega}})\big).$$

[3] We consider the following inequality:

$$(\omega_i^2)^{\frac{q}{2}}(\tilde{\omega}_i^2)^{1-\frac{q}{2}} \leq \frac{q}{2}\omega_i^2 + (1 - \frac{q}{2})\tilde{\omega}_i^2,$$

which is a special case of the weighted arithmetic and geometric mean inequality.

When $\boldsymbol{\omega}$ is fixed, $\tilde{\boldsymbol{\omega}}$ can be updated by the equation $\tilde{\boldsymbol{\omega}} = |\boldsymbol{\omega}|$, which is proven by differentiating $h(\boldsymbol{\omega}, \tilde{\boldsymbol{\omega}})$ and setting to zero; when $\tilde{\boldsymbol{\omega}}$ is fixed, we use the EM algorithm to solve a l_2 norm based convex optimization over $\boldsymbol{\omega}$ by treating $\mathbf{z}$ and λ as missing data.

4.2 Updating ω via Standard EM Algorithm

Now we give more details of the EM algorithm used to update $\boldsymbol{\omega}$ in the proposed algorithm.

Expectation Step

In the E-step, it begins with the functional $H(\boldsymbol{\omega}|\boldsymbol{\omega}^k)$, namely

$$H(\boldsymbol{\omega}|\boldsymbol{\omega}^k) = E_{\lambda,\mathbf{z}}\big(h(\boldsymbol{\omega}, \tilde{\boldsymbol{\omega}}^k)|\mathbf{y}, \boldsymbol{\omega}^k\big) \tag{8}$$

Here $\boldsymbol{\omega}^k$ is the estimate of $\boldsymbol{\omega}$ at k-th iteration. $\langle\cdot\rangle$ denotes the expectation of a random variable. And both $\langle\frac{1}{\lambda}\rangle^k$ and $\langle\mathbf{z}\rangle^k$ can be easily derived as follows:

$$\langle\frac{1}{\lambda}\rangle^k = \frac{\frac{n}{q} + a}{\sum_{i=1}^n |\omega_i^k|^q + b}, \tag{9}$$

and

$$\langle z_i\rangle^k = \boldsymbol{\Phi}(\mathbf{x}_i)\boldsymbol{\omega}^k + \frac{y_i\mathcal{N}\big(\boldsymbol{\Phi}(\mathbf{x}_i)\boldsymbol{\omega}^k|0,1\big)}{\Psi\big(y_i\boldsymbol{\Phi}(\mathbf{x}_i)\boldsymbol{\omega}^k\big)}. \tag{10}$$

We can rewrite Eq. 9 as

$$\big(\langle\frac{1}{\lambda}\rangle^k\big)^{-1} = \rho\cdot\frac{b}{a} + \big(1-\rho\big)\cdot\frac{\sum_{i=1}^n|\omega_i^k|^q}{\frac{n}{q}} \tag{11}$$

with $\rho = \frac{a}{\frac{n}{q}+a} \in (0,1)$. It is interesting to see that, the reciprocal of $\langle\frac{1}{\lambda}\rangle^k$ is a convex combination of a prior-dependent term and a data-dependent term, while ρ is the mixture parameter. In this work, we use a non-informative inverse gamma prior, i.e., $a = b = 10^{-3}$, to make the estimate depend more on the observations than on the prior knowledge. Besides, in implementation Eq. 9 is updated only based on $\omega_i^k \geq 10^{-4}$.

Maximization Step

Following the E-step, the M-step updates $\boldsymbol{\omega}$ by

$$\boldsymbol{\omega}^{k+1} = \big(\langle\frac{1}{\lambda}\rangle^k\cdot q\cdot\mathbf{V}^k + \boldsymbol{\Phi}^T\boldsymbol{\Phi}\big)^{-1}\boldsymbol{\Phi}^T\langle\mathbf{z}\rangle^k, \tag{12}$$

with $\mathbf{V}^k = diag((\tilde{\omega}_1^k)^{q-2},\ldots,(\tilde{\omega}_n^k)^{q-2})$. In order to avoid handling arbitrarily large numbers in the matrix $\mathbf{V}^k$, we rewrite the updating rule in Eq. 12 as

$$\boldsymbol{\omega}^{k+1} = \mathbf{U}\big(\langle\frac{1}{\lambda}\rangle^k\cdot q\cdot\mathbf{I} + \boldsymbol{\Phi}^T\boldsymbol{\Phi}\mathbf{U}\big)^{-1}\boldsymbol{\Phi}^T\langle\mathbf{z}\rangle^k,$$

where $\mathbf{U} = diag((\tilde{\omega}_1^k)^{2-q},\ldots,(\tilde{\omega}_n^k)^{2-q})$.

4.3 Monotonicity Analysis

The algorithm proposed above will be referred to as GGIG in this paper. The convergence of GGIG can be guaranteed by its monotonicity. Specifically, its objective function, i.e., $p(\boldsymbol{\omega}, \mathbf{y}, \mathbf{z}, \lambda)$, is non-decreasing between iterations; that is,

$$p(\boldsymbol{\omega}^k, \mathbf{y}, \mathbf{z}^k, \lambda^k) \leq p(\boldsymbol{\omega}^{k+1}, \mathbf{y}, \mathbf{z}^{k+1}, \lambda^{k+1}), \tag{13}$$

for $k = 0, 1, 2, \ldots$.

To see this, we firstly notice that

$$p(\boldsymbol{\omega}^k, \mathbf{y}, \mathbf{z}^k, \lambda^k) = \underline{p}(\boldsymbol{\omega}^k, \tilde{\boldsymbol{\omega}}^k, \mathbf{y}, \mathbf{z}^k, \lambda^k), \tag{14}$$

which can be derived from the properties of the surrogate function $\underline{p}(\cdot)$. Then, since $\boldsymbol{\omega}^k, \mathbf{z}^k$ and λ^k are updated by a standard EM algorithm, it is clear that

$$\underline{p}(\boldsymbol{\omega}^k, \tilde{\boldsymbol{\omega}}^k, \mathbf{y}, \mathbf{z}^k, \lambda^k) \leq \underline{p}(\boldsymbol{\omega}^{k+1}, \tilde{\boldsymbol{\omega}}^k, \mathbf{y}, \mathbf{z}^{k+1}, \lambda^{k+1}). \tag{15}$$

This relationship comes from the monotonicity of the EM algorithm. Continue this way and again consider the properties of the surrogate function $\underline{p}(\cdot)$, we have

$$\begin{aligned} p(\boldsymbol{\omega}^k, \mathbf{y}, \mathbf{z}^k, \lambda^k) &\leq \underline{p}(\boldsymbol{\omega}^{k+1}, \tilde{\boldsymbol{\omega}}^k, \mathbf{y}, \mathbf{z}^{k+1}, \lambda^{k+1}) \\ &\leq \underline{p}(\boldsymbol{\omega}^{k+1}, \tilde{\boldsymbol{\omega}}^{k+1}, \mathbf{y}, \mathbf{z}^{k+1}, \lambda^{k+1}) \\ &= p(\boldsymbol{\omega}^{k+1}, \mathbf{y}, \mathbf{z}^{k+1}, \lambda^{k+1}). \end{aligned}$$

Thus, Eq. 13 is established. For a bounded sequence of the objective function values $\{p(\boldsymbol{\omega}^k, \mathbf{y}, \mathbf{z}^k, \lambda^k)\}$, GGIG converges monotonically to a stationary value.

5 Related Work

In this section, we show the relationships of our model to other existing approaches. Many of these approaches are special cases of GGIG.

5.1 Probit Classifiers with GSM Priors

[Figueiredo, 2003] proposed probit classifiers using a hierarchical Laplacian distribution, and we refer to it as LAP. Specifically, LAP considers that each ω_i has an independent zero-mean Gaussian prior $\omega_i|\tau_i \sim \mathcal{N}(\omega_i|0,\tau_i)$ with its own variance τ_i, and that each τ_i has an exponential hyper-prior with a variance parameter γ. Using standard MAP estimation, LAP obtains

$$\boldsymbol{\omega}^{k+1} = \big(\gamma\cdot\mathbf{V}_{lap}^k + \boldsymbol{\Phi}^T\boldsymbol{\Phi}\big)^{-1}\boldsymbol{\Phi}^T\langle\mathbf{z}\rangle^k, \tag{16}$$

with $\mathbf{V}_{lap}^k = diag((\omega_1^k)^{-1},\ldots,(\omega_n^k)^{-1})$. Comparing the above rule with Eq. 12, it is not difficult to see that, GGIG can be specialized as LAP by letting $q = 1, a \gg N+1$ and $b = \frac{a}{\gamma}$. Actually, generalized Gaussian distribution with $q = 1$ plays the same role as the hierarchical Laplacian prior, while setting a to very large values makes $\frac{1}{\lambda}$ heavily peaked at the value $\frac{a}{b} = \gamma$.

To eliminate the parameter γ in LAP, Figueiredo extended the hierarchical Laplacian prior to be a parameter-free Gaussian-Jeffreys' prior, which used an improper Jefferys' distribution as the hyper-prior. We here refer to this model as GJ. Its model parameter $\boldsymbol{\omega}$ is updated by

$$\boldsymbol{\omega}^{k+1} = \big(\mathbf{V}_{gj}^k + \boldsymbol{\Phi}^T\boldsymbol{\Phi}\big)^{-1}\boldsymbol{\Phi}^T\langle\mathbf{z}\rangle^k, \tag{17}$$

with $\mathbf{V}_{gj}^k = diag((\omega_1^k)^{-2},\ldots,(\omega_n^k)^{-2})$. Although the GJ model avoids pre-specifying γ, it always leads to an exaggerated sparsification, which would be interpreted by the following limit:

$$\lim_{q\to 0}\langle\frac{1}{\lambda}\rangle^k\cdot q\cdot\mathbf{V}^k = \frac{n}{n+b}\mathbf{V}_{gj}^k, \tag{18}$$

where $\langle \frac{1}{\lambda} \rangle^k \cdot q \cdot \mathbf{V}^k$ is the only difference between Eq. 12 and Eq. 17. GJ is a special case of GGIG with $q \to 0$ and $b \ll n$. Thus, it is not surprising that GJ is likely to provide over-sparse solutions.

A version of the hierarchical Student-*t* prior was utilized in probit classifiers in [Chen *et al.*, 2009], which is referred to as STU here. It gave a similar rule of GJ in Eq. 17.

5.2 Models with a LSM Prior

[Garrigues and Olshausen, 2010] proposed a LSM prior in sparse coding models of natural images. With $q = 1$, the GGSM prior used in our model is the factorial version of the LSM prior in the paper. An important difference between LSM and GGSM is that, the former can only encourage sparse solutions, while the latter with different values of q can induce both sparsity and non-sparsity.

5.3 Iteratively Re-weighted l_1 and l_2 Minimization

From Eq. 8, it is clear that the EM progress is equivalent to a version of the iteratively re-weighted l_2 minimization [Chartrand and Yin, 2008]. Consider a special case of GGSM, where each component ω_i uses different λ_i, and $q = 1$. Then inference in the model can be implemented by solving the following MAP sequence:

$$\boldsymbol{\omega}^{k+1} = \arg\max_{\boldsymbol{\omega}} -\frac{1}{2}\langle \|\mathbf{z} - \boldsymbol{\Phi}\boldsymbol{\omega}\|_2^2 \rangle^k - \sum_{i=1}^{n} \frac{1+a}{|\omega_i^k| + b}|\omega_i|,$$

which is equivalent to the update proposed in the iteratively re-weighted l_1 minimization [Candès *et al.*, 2008]. Hence, our model provides a probabilistic interpretation for the two iteratively re-weighted algorithms.

6 Experiments

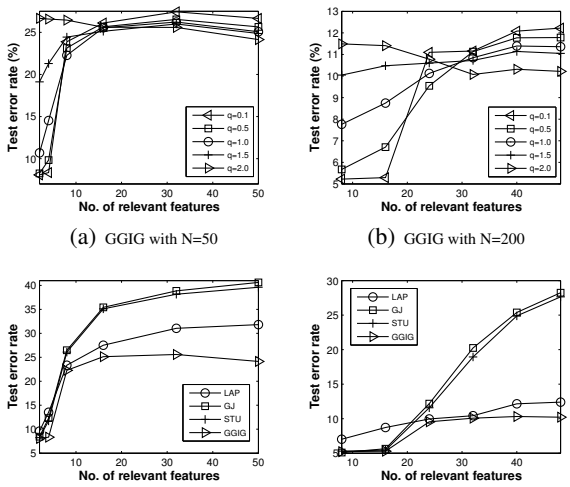
We now empirically study the behaviors and prediction performance of the proposed GGIG method. We first focus on linear classifiers, where we have full control over the distribution of the relevant information among features in order to shed light on the appropriateness of sparse and non-sparse. Then we pay attention to kernel-based classifiers.

6.1 Feature Selection for Linear Classifiers

Here $\boldsymbol{\Phi}(\mathbf{x}) = (1, x_1, \ldots, x_d)^T$ in Eq. 1, and our method may be seen as the combination of a learning algorithm and a feature selection step.

We consider synthetic data having 50-dimensional i.i.d. features. The data are generated using $\boldsymbol{\omega} = [3, \ldots, 3, 0, \ldots, 0]^T$, where the number of relevant features n_f provides a range of testing conditions by varying in the set $\{2, 4, 8, 16, 32, 50\}$. We use independent zero-mean and unit-variance Gaussians to draw a data matrix $\mathbf{X}$, and use a Gaussian with mean $\mathbf{X}\boldsymbol{\omega}$ and unit-variance to draw $\mathbf{y}$, which are then thresholded at zero to provide the class labels $\mathbf{z}$. In this group of experiments, the training set size is varied in $\{50, 200\}$. For each training set, a test set containing 3000 samples is also generated from the same model as the corresponding test set.

We test GGIG with different values of $q \in \mathcal{Q}$. Fig. 3(a) and 3(b) show the obtained results in cases of $N = 50$ and 200. We can see that, GGIG with different q obtain quite different performances for a problem. As n_f is changed from 2 to 50, the corresponding optimal q is increased from 0.1 to 2.0, which coincides well with our expectation. When n_f is relatively small, the data has large sparsity in the features. So the model needs a prior that is sparsity-encouraging with a large kurtosis, which means small values of q are better. In contrary, when n_f is large, most features are relevant to the decision-making, and few information is redundant, thus enforcing sparsity would lead to the discarding of useful information. As a result, priors should have a low kurtosis, i.e. large values of q are preferred.



(a) GGIG with N=50 (b) GGIG with N=200

(c) Comparison in N=50 (d) Comparison in N=200

Figure 3: Experimental results from feature selection

The above experiments confirm that different classification problems need to be imposed by different degrees of sparsity, which are determined by the information redundancy among their features. They also demonstrate that a proper degree of induced sparsity can be provided in GGSM by tuning the shape parameter q from cross validation.

Next we compare GGIG with other existing probit classifiers, including LAP, GJ, as well as STU. To find the optimal $\gamma^\star$ in LAP, we do 5-fold cross validation within $\{0.01, 0.04, 0.08, 0.1, 0.4, 0.8, 1, 4, 8, 10\}$. Parameters of STU are specified referring to [Chen *et al.*, 2009]. From the experimental results in Fig. 3(c) and 3(d), we can see that GGIG gives lower averaged error rate than any other models in all cases. As we analyzed above, this is because the proposed GGSM-based model utilizes appropriate priors in probabilistic modeling, which could induce proper degrees of sparsity for various problems.

6.2 Kernel-based Classifiers

We then show experiments of the kernel-based classifiers. In the following experiments, the Gaussian kernel is used.

Toy datasets

In this part, we give more details on the toy experiments in Section 2. We compare GGIG, RVM, LAP and GJ on the spiral and cross datasets, respectively. About parameter setting, we select kernel width used in each algorithm through 5-fold

Table 1: Average Error Rate on Two Toy Datasets.

DATA	GGIG	LAP	RVM	GJ
Spiral	**0.60(198.2)**	11.60(108.5)	8.92(56.2)	30.91(13.8)
Cross	**9.62(4.0)**	11.41(12.8)	11.20(7.5)	13.27(3.2)

Table 2: Average Error Rate on Four Real Datasets.

MODEL	SOLAR	GERMAN	THYROID	TITANIC
GGIG	**34.85**%	23.82%	**4.00%**	**21.60**%
SVM	35.98%	23.89%	4.98%	22.10%
RVM	35.19%	**23.77%**	5.06%	23.00%
LAP	36.66%	24.51%	4.74%	23.12%
GJ	38.02%	24.89%	4.66%	23.36%

cross validation within $\{0.1, 0.5, 1.0, \ldots, 10.0\}$. Parameters in RVM can be adapted by itself. Other parameters are specified following the way as before.

Fig. 1 and Fig. 2 (in page 2) demonstrate the behaviors of each model. As we discussed before, the spiral data is a non-sparse scenario, which contains few redundance information. The cross data is a sparse scenario, and high degrees of sparsity need to be induced from priors. Using cross validation, the GGIG method successfully determined $q = 2.0$ and $q = 0.1$ for these two datasets, respectively. Table 1 shows the average error rate over 50 independent runs. The quantity in bracket is the average number of used kernel functions for each model. We can see GGIG clearly outperforms other methods, especially in the spiral data.

Real datasets

To demonstrate the performances of GGIG further, we compare different algorithms on four benchmark datasets. These algorithms include GGIG, LAP, GJ, as well as SVM and RVM.

The four datasets have been preprocessed by Rätsch et al. to do binary classification tests[4], including Solar, German, Thyroid and Titanic. We optimize parameters following the way in [Ratsch *et al.*, 2001]. For SVM, the trade-off parameter C is searched in set $\{f \times 10^g\}$ with $f \in \{1, 3\}$ and $g \in \{-6, \ldots, 6\}$. Kernel width and other required parameters are specified as before. Table 2 reports the error rate of these models. GGIG outperforms other classifiers in three of these datasets, and is only a little worse than RVM in the German data.

7 Conclusions

In this paper, we begin with a set of toy experiments, which suggests that different problems need different degrees of sparsity. To induce an appropriate degree of sparsity for a specific problem, we propose a GGSM prior in the probabilistic modeling of the probit classifications. Comparing to the previous GSM and LSM priors, we can flexibly adjust the induced sparsity from the GGSM prior, and proper degrees of sparsity can be promoted by tuning the shape parameter q in a data-dependent way. The model learning with arbitrary $q \in (0, 2]$ is carried out by an efficient modified MAP algorithm. And we also analyze in detail relationships of the proposed method to other previous approaches.

[4] `http://ida.first.fraunhofer.de/projects/bench/benchmarks.htm`

Several questions still remain in this proposed model. Although it updates parameters using closed form formulations, a low convergence rate inheriting from the EM algorithm increases its computational cost. One potential way to address this issue is using successive over-relaxation [Yu, 2010]. Besides, another problem is how to guarantee a good local optimum due to the multi-modality of the posterior. We believe the ϵ-regularization recently proposed in [Chartrand and Yin, 2008] would be helpful.

References

[Candès *et al.*, 2008] E. Candès, M. Wakin, and S. Boyd. Enhancing sparsity by reweighted l_1 minimization. *Journal of Fourier Analysis and Applications*, 14:877–905, 2008.

[Caron and Doucet, 2008] F. Caron and A. Doucet. Sparse Bayesian nonparametric regression. In *25th International Conference on Machine Learning*, 2008.

[Chartrand and Yin, 2008] R. Chartrand and W. Yin. Iteratively reweighted algorithms for compressive sensing. In *33rd International Conference on Acoustics, Speech, and Signal Processing*, 2008.

[Chen *et al.*, 2009] H. Chen, P. Tino, and X. Yao. Probabilistic classification vector machines. *IEEE Transactions on Neural Networks*, 20(6):901–914, 2009.

[Figueiredo, 2003] M. Figueiredo. Adaptive sparseness for superived learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1150–1159, 2003.

[Garrigues and Olshausen, 2010] P. Garrigues and B. Olshausen. Group sparse coding with a Laplacian scale mixture prior. In *Advances in Neural Information Processing Systems 24*, 2010.

[Griffin and Brown, 2010] J. Griffin and P. Brown. Inference with normal-gamma prior distributions in regression problems. *Bayesian Analysis*, 5(1):171–188, 2010.

[Hunter and Lange, 2004] D. Hunter and K. Lange. A tutorial on MM algorithms. *The American Statistician*, 58:30–37, 2004.

[Kabán, 2007] A. Kabán. On Bayesian classification with Laplace priors. *Pattern Recognition Letters*, 28(10):1271–1282, 2007.

[Ratsch *et al.*, 2001] G. Ratsch, T. Onoda, and K. Muller. Soft margins for AdaBoost. *Machine Learning*, 42(3):287–320, 2001.

[Raykar and Zhao, 2010] V. Raykar and L. Zhao. Nonparametric prior for adaptive sparsity. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010.

[Tipping, 2001] M. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

[Yu, 2010] Y. Yu. Monotonically overrelaxed EM algorithms. Technical report, Department of Statistics, University of California, Irvine, 2010.