Relevance tagging machine*

***D. A. Molchanov*[1], *D. A. Kondrashkin*[2], *and D. P. Vetrov*[2]**
dmolch111@gmail.com; kondra2lp@gmail.com; vetrovd@yandex.ru
[1]Lomonosov Moscow State University, GSP-1, Leninskie Gory, Moscow, Russia
[2]National Research University Higher School of Economics, 20 Myasnitskaya str., Moscow, Russia

In many classification or regression problems, there may be a lot of irrelevant features. Bayesian automatic relevance determination (ARD) is a popular approach to feature selection. However, the application area of this approach has been limited. In this paper, this approach is utilized in a more general case and it is applied to a binary classification problem with binary features. Also, a new binary classification model and a learning algorithm that can purge unwanted features from the model have been developed.

Keywords: *binary classification; feature selection; automatic relevance determination; sparse bayesian learning; variational lower bounds*

DOI: 10.21469/22233792.1.13.09

Машина релевантных тегов*

***Д. А. Молчанов*[1], *Д. А. Кондрашкин*[2], *Д. П. Ветров*[2]**
[1]Московский государственный университет им. М. В. Ломоносова, Москва, Россия
[2]Национальный исследовательский университет «Высшая школа экономики», Москва, Россия

При решении многих задач классификации или регрессии зачастую приходится сталкиваться с большим количеством нерелевантных признаков. Одним из известных способов решения задачи отбора признаков является метод, основанный на Байесовском подходе к выбору модели. Этот метод получил широкое распространение, однако область его применения была ограничена. В данной работе этот метод применяется для более широкого класса моделей и исследуется на примере задачи бинарной классификации с бинарными признаками. Также предложена новая модель для бинарной классификации данных и метод обучения этой модели, позволяющий автоматически убирать нерелевантные признаки.

Ключевые слова: *бинарная классификация; отбор признаков; автоматическое определение релевантности; вариационные нижние оценки*

DOI: 10.21469/22233792.1.13.09

1 Introduction

Feature selection is an important challenge that arises in most machine learning problems. There are different approaches to this task. One of them is to use predictive models that can automatically choose the most relevant features during the training procedure. For example, it can be done with LASSO (Least Absolute Shrinkage and Selection Operator) regression or other models that use L1-regularization to ensure sparsity. Bayesian ARD [1]) is another approach to developing such models. As an example, consider the Relevance Vector Machine (RVM), [2]. In case of regression, the RVM is a linear model with an ARD prior; $\boldsymbol{x}$ is an object; t is its target;

*This research is funded by RFBR grant #15-31-20596mol-a-ved, Microsoft Research, research initiative: Computer vision collaborative research in Russia, Skoltech SDP Initiative, applications A1 and A2.

$\boldsymbol{w}$ is a vector of model parameters or weights; and $\boldsymbol{\varphi}(\boldsymbol{x})$ is a vector of generalized features. The model definition is shown below:

$$\boldsymbol{w} = (w_1, \ldots, w_M)^{\mathrm{T}} \in \mathbb{R}^M; \quad \boldsymbol{\varphi}(\boldsymbol{x}) = (\varphi_1(\boldsymbol{x}), \ldots, \varphi_M(\boldsymbol{x}))^T \in \mathbb{R}^M, \quad t \in \mathbb{R};$$

$$y(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\varphi}(\boldsymbol{x}); \tag{1}$$

$$p(t \,|\, \boldsymbol{x}, \boldsymbol{w}, \beta) = \mathcal{N}(t \,|\, y(\boldsymbol{x}), \beta^{-1}); \tag{2}$$

$$p(\boldsymbol{w} \,|\, \boldsymbol{\alpha}) = \prod_{i=1}^{M} \mathcal{N}(w_i \,|\, 0, \alpha_i^{-1}) \tag{3}$$

and the following expression is the marginal likelihood function, also known as evidence [3]:

$$p(\boldsymbol{t} \,|\, \boldsymbol{X}, \boldsymbol{\alpha}, \beta) = \int p(\boldsymbol{t} \,|\, \boldsymbol{X}, \boldsymbol{w}, \beta) p(\boldsymbol{w} \,|\, \boldsymbol{\alpha})\, d\boldsymbol{w}\,.$$

Equation (2) defines the likelihood function for an object $\boldsymbol{x}$. Here, β is the noise precision, $\beta = \sigma^{-2}$, and $y(\boldsymbol{x})$ is the mean of the target function given by a linear model defined in (1). Expression (3) describes the prior over the weight parameters $\boldsymbol{w}$ (ARD prior). When the evidence of the model is maximized with respect to hyperparameters $\boldsymbol{\alpha}$, some of them go to infinity. The corresponding weight parameters will then have posterior distributions that are concentrated at zero; so, the corresponding basis functions $\varphi_i(\boldsymbol{x})$ are pruned out of the model. This effect is known as ARD effect and is explained and discussed in [1, 2] and [4, p. 349–353].

However, this effect is usually studied on models with Gaussian prior. The present authors propose to extend this approach and use another family of distributions. In this paper, a binary classification problem with binary features is considered as an example. A new probabilistic model has been developed for this task and beta prior distribution has been used to reproduce ARD effect.

2 Model of Relevance Tagging Machine

2.1 Probabilistic model

Consider a binary classification problem of objects that have binary features (tags). Let $(\boldsymbol{x}_i, t_i)_{i=1}^n$ be the training set, where $\boldsymbol{x}_i$ is the object described by a binary vector, $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{id})^{\mathrm{T}}$, d denotes the number of tags, and $t_i \in \{0, 1\}$ is the class label. In this notation, $x_{ij} = 1$ if object $\boldsymbol{x}_i$ has tag j and $x_{ij} = 0$ otherwise.

Under the assumption that all tags affect the class label independently, we define the probabilistic model of relevance tagging machine (RTM):

$$\mathsf{P}(t = 1 \,|\, x_j = 1) = q_j;$$

$$\mathsf{P}(t = 1 \,|\, \boldsymbol{x}, \boldsymbol{q}) = \prod_{j=1}^{d} q_j^{x_j} \times \left(\prod_{j=1}^{d} q_j^{x_j} + \prod_{j=1}^{d} (1 - q_j)^{x_j} \right)^{-1}$$

where $\boldsymbol{q} = (q_1, \ldots, q_d)^{\mathrm{T}}$ are the model parameters, which are responsible for the tags' influence on the class label.

2.2 Bayesian Automatic Relevance Determination approach

Similarly to the RVM, follow a traditional Bayesian ARD approach to feature selection. The basic idea is to treat parameters $\boldsymbol{q}$ as random variables and place independent priors over them.

As the domain of q_j is $[0, 1]$, it is natural to use beta distribution over q_j. Also, as both classes are meant to be of the same importance, symmetrical distribution is used:

$$q_j \sim \mathrm{Beta}(\alpha_j + 1, \alpha_j + 1),\ \alpha_j \in [0, +\infty).$$

Here, $\alpha_j = 0$ corresponds to the uniform distribution over q_j, so that there is no regularization of q_j. Contrary, if α_j tends to plus infinity, the variance of q_j tends to zero and that implies $q_j = 0.5$. It means that the jth tag is removed from the model:

$$\mathsf{P}(t \,|\, \boldsymbol{x}, \boldsymbol{q} = (q_1, \ldots, q_{j-1}, q_j = 0.5, q_{j+1}, \ldots, q_d)^{\mathrm{T}}) = \mathsf{P}(t \,|\, \boldsymbol{x}, \boldsymbol{q} = (q_1, \ldots, q_{j-1}, q_{j+1}, \ldots, q_d)^{\mathrm{T}}) = \mathsf{P}(t \,|\, \boldsymbol{x}, \boldsymbol{q}^{\setminus j}).$$

Note that the case $\alpha_j \in (-1, 0)$ is not considered because in this case, maximum a posteriori (MAP) estimate of $\boldsymbol{q}$ would be more contrast (i.e. closer to 0 or 1) than maximum likelihood (ML) estimate. In case of the problem under investigation, it is unreasonable to believe that a tag is actually more relevant than it seems to be.

The posterior is written using Bayes' theorem:

$$\mathsf{P}(\boldsymbol{q} \,|\, \boldsymbol{X}, \boldsymbol{t}, \boldsymbol{\alpha}) = \frac{\mathsf{P}(\boldsymbol{t} \,|\, \boldsymbol{X}, \boldsymbol{q}) p(\boldsymbol{q} \,|\, \boldsymbol{\alpha})}{\int \mathsf{P}(\boldsymbol{t} \,|\, \boldsymbol{X}, \boldsymbol{q}) p(\boldsymbol{q} \,|\, \boldsymbol{\alpha}) d\boldsymbol{q}}\,. \tag{4}$$

2.3 Evidence maximization

The denominator in Eq. (4) is called the *evidence* [3] of the model. In general, a simple model has higher evidence than the complex one if they have the same prediction accuracy [4, p. 349–352]. In the presented case, evidence maximization is expected to set $\alpha_j = +\infty$ for the majority of irrelevant features:

$$E(\boldsymbol{\alpha}) = \int \mathsf{P}(\boldsymbol{t} \,|\, \boldsymbol{X}, \boldsymbol{q}) p(\boldsymbol{q} \,|\, \boldsymbol{\alpha})\, d\boldsymbol{q} \to \max_{\boldsymbol{\alpha}}\,.$$

However, in the described model, likelihood and prior are not the conjugate distributions; so, the evidence is intractable. It also cannot be efficiently estimated numerically, because numerical computation of multidimensional integrals is a very difficult and time-consuming task. Therefore, one needs some kind of approximation in order to maximize the evidence. In this paper, an approach that uses variational lower bounds for optimization is described.

3 Variational lower bounds for evidence maximization

Definition 1. *A variational lower bound on a function* $f(\boldsymbol{w})$, $\boldsymbol{w} \in M \subseteq \mathbb{R}^n$, *is a function* $g(\boldsymbol{w}, \boldsymbol{\xi})$, $\boldsymbol{w} \in M$, $\boldsymbol{\xi} \in M$, *with the following properties:*

$$g(\boldsymbol{w}, \boldsymbol{w}) = f(\boldsymbol{w})\ \forall \boldsymbol{w} \in M;$$
$$g(\boldsymbol{w}, \boldsymbol{\xi}) \leqslant f(\boldsymbol{w})\ \forall \boldsymbol{w} \in M,\ \forall \boldsymbol{\xi} \in M,$$

$\boldsymbol{\xi}$ *is called a variational parameter.*

Consider an optimization problem $f(\boldsymbol{w}) \to \max_{\boldsymbol{w}}$. If $g(\boldsymbol{w}, \boldsymbol{\xi})$ is a variational lower bound on $f(\boldsymbol{w})$, then this optimization problem can be solved in such coordinatewise optimization procedure:

$$\boldsymbol{w}^{k+1} = \arg\max_{\boldsymbol{w}} g(\boldsymbol{w}, \boldsymbol{\xi}^k); \quad \boldsymbol{\xi}^{k+1} = \boldsymbol{w}^{k+1}.$$

This optimization procedure is known as bound optimization algorithm or bound optimizer. Many popular optimization methods in machine learning and pattern recognition are the special cases of this algorithm. For instance, EM (expectation–maximization) algorithm and its extensions, generalized iterative scaling algorithm for maximum entropy models, nonnegative matrix factorization algorithm, and concave-convex procedure are the common examples of bound optimizers [5].

In general case, this lower bound may not be exact for any point $\boldsymbol{w}$ and variational parameters may be from a different space. In that case, the result of a similar optimization procedure

$$\boldsymbol{\xi}^{k+1} = \arg\max_{\boldsymbol{\xi}} g(\boldsymbol{w}^k, \boldsymbol{\xi}); \quad \boldsymbol{w}^{k+1} = \arg\max_{\boldsymbol{w}} g(\boldsymbol{w}, \boldsymbol{\xi}^{k+1}) \tag{5}$$

can be treated as an approximate solution of the original optimization task.

This approach is widely used in various optimization problems. The best feature of it is that there is no need to compute the original function $f(\boldsymbol{w})$. For example, a similar approach is used in [6] where it is applied to Bayesian logistic regression.

In case of RTM, this approach is applied to evidence maximization. A variational lower bound has been obtained on the evidence integrand and its integral has been used as a set of evidence lower bounds which can be used in an optimization procedure shown in (5).

Theorem 1. *Function* $\tilde{E}(\boldsymbol{\alpha}, \boldsymbol{H})$ *is a lower bound on RTM evidence for all* $\boldsymbol{H} \in (0,1)^{n\times d}$, $\boldsymbol{\alpha} \in [0, +\infty)^d$:

$$E(\boldsymbol{\alpha}) \geqslant \tilde{E}(\boldsymbol{\alpha}, \boldsymbol{H}) = \int \prod_{i=1}^{n} L_i(\boldsymbol{q}, \boldsymbol{\eta}_i) \prod_{j=1}^{d} p(q_j \mid \alpha_j)\, d\boldsymbol{q} =$$

$$= \left(\prod_{i=1}^{n} c_i(\boldsymbol{\eta}_i)\right) \prod_{j=1}^{d} \int \exp\left(\sum_{i:j\in Q_i} \tilde{c}_{ij}(\boldsymbol{\eta}_i) \left(\frac{1-q_j}{q_j}\right)^{|Q_i|(2t_i-1)}\right) p(q_j \mid \alpha_j)\, dq_j$$

$$\forall \boldsymbol{H} \in (0,1)^{n\times d},\ \forall \boldsymbol{\alpha} \in [0, +\infty)^d,$$

where

$$Q_i = \{j | x_{ij} = 1\};$$

$$c_i(\boldsymbol{\eta}_i) = \frac{\prod\limits_{j\in Q_i} \eta_{ij}^{t_i} (1-\eta_{ij})^{1-t_i}}{\prod\limits_{j\in Q_i} \eta_{ij} + \prod\limits_{j\in Q_i} (1-\eta_{ij})} \exp\left(\frac{\prod\limits_{j\in Q_i} \eta_{ij}^{1-t_i} (1-\eta_{ij})^{t_i}}{\prod\limits_{j\in Q_i} \eta_{ij} + \prod\limits_{j\in Q_i} (1-\eta_{ij})}\right);$$

$$\tilde{c}_{ij}(\boldsymbol{\eta}_i) = -\frac{\prod\limits_{j\in Q_i} \eta_{ij}^{t_i} (1-\eta_{ij})^{1-t_i}}{\prod\limits_{j\in Q_i} \eta_{ij} + \prod\limits_{j\in Q_i} (1-\eta_{ij})} \left(\frac{\eta_{ij}}{1-\eta_{ij}}\right)^{|Q_i|(2t_i-1)} |Q_i|^{-1};$$

and $\boldsymbol{H}$ *is the matrix of variational parameters and its* i*th row is equal to* $\boldsymbol{\eta}_i^{\mathrm{T}}$.

The proof of Theorem 1 is provided in Appendix A.

The shapes of the evidence integrand and its lower bounds are shown in Figs. 1 and 2. Note that although the evidence lower bound can be computed as a product of d one-dimensional integrals, these integrals still have to be computed numerically. Also, note that although the number of variational parameters is nd, usually most of them are inessential and do not affect the value of this lower bound. A variational parameter η_{ij} is essential if and only if $x_{ij} = 1$.

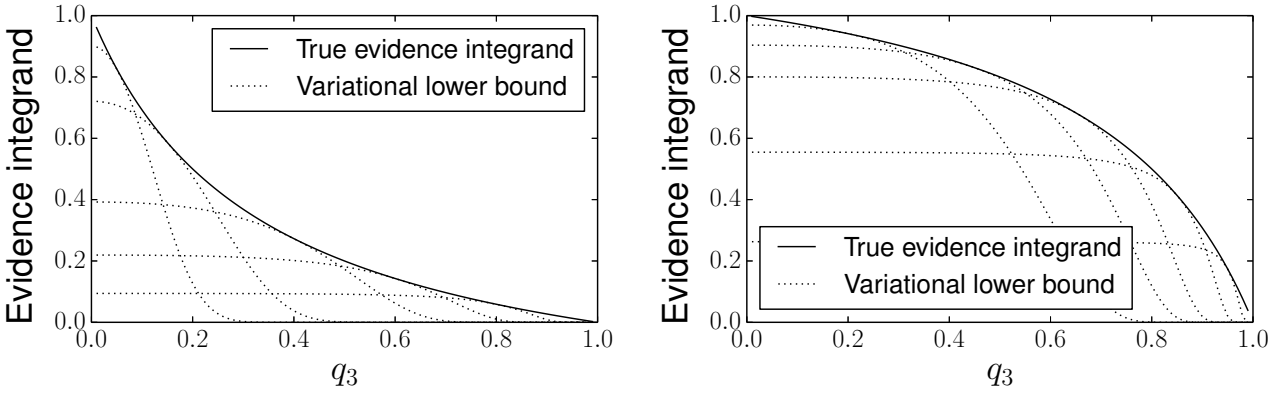

Figure 1 Evidence integrand variational lower bounds for a single object for different values of variational parameters η: (a) $t = 0$, $\boldsymbol{x} = (1, 0, 1)^{\mathrm{T}}$ and $q_1 = 0.8$; and (b) $t = 0$, $\boldsymbol{x} = (0, 1, 1)^{\mathrm{T}}$ and $q_2 = 0.2$

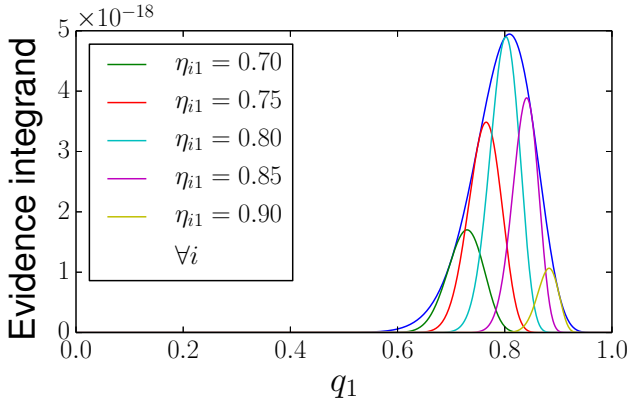

Figure 2 Evidence integrand lower bounds for the whole dataset

Therefore, there are only $n\tau$ essential variational parameters where τ is the average number of tags per object.

The evidence lower bound is optimized in an EM-like algorithm:

1) E-step: $\boldsymbol{H}^{\text{new}} = \arg\max_{\boldsymbol{H}} \log \tilde{E}(\boldsymbol{\alpha}^{\text{old}}, \boldsymbol{H})$; and
2) M-step: $\boldsymbol{\alpha}^{\text{new}} = \arg\max_{\boldsymbol{\alpha}} \log \tilde{E}(\boldsymbol{\alpha}, \boldsymbol{H}^{\text{new}})$.

These two steps are repeated until convergence.

On E-step, hyperparameters $\boldsymbol{\alpha}$ are fixed and variational parameters $\boldsymbol{H}$ are tuned to obtain the most accurate lower bound. On M-step, the best lower bound from the E-step is optimized with respect to hyperparameters $\boldsymbol{\alpha}$. The L-BFGS-B method [8] was used to handle optimization problems on both steps of algorithm. This method is called RTM-EM.

Let k_E and k_M be the number of iterations of L-BFGS-B on E-step and M-step, respectively. The complexity of one iteration is then equal to $O(n\tau k_E + dk_M)$ operations of numerical integration.

Complexity of RTM-EM is too high; so, a simplification is suggested. In RTM-EM, on E-step of EM algorithm, an attempt to obtain the best possible value of variational parameters was

Table 1 Relevance determination performance on synthetic data

Noise	RTM-MAP-EM	RTM-EM	RVM	L1-LR
Percentage of removed irrelevant tags				
Random	99.64%	99.46%	99.10%	85.63%
Correlated	84.44%	88.90%	84.65%	100.00%
Percentage of removed genuine tags				
Random	4.50%	4.68%	2.63%	3.54%
Correlated	2.50%	4.34%	1.04%	2.50%

made. Instead of that, a variational lower bound on the evidence integrand was used that is exact at its point of maximum. E-step will then look like this:

$$\boldsymbol{\eta}_i^{\text{new}} = \boldsymbol{q}^{\text{MAP}} = \arg\max_{\boldsymbol{q}} \mathsf{P}(\boldsymbol{t} \mid \boldsymbol{X}, \boldsymbol{q}) p(\boldsymbol{q} \mid \boldsymbol{\alpha}^{\text{old}}) \; \forall i.$$

Note that all objects share the same set of variational parameters; so, there are only d of them: $\boldsymbol{\eta}_i = \boldsymbol{\eta}_k$ for all $i, k = 1, \ldots, n$. This method was named RTM-MAP-EM. Its complexity is $O(dk_M)$ operations of numerical integration.

It was experimentally shown that RTM-MAP-EM also purges irrelevant (both noisy and correlated) tags and has comparable accuracy with RTM-EM algorithm. Also, both EM-based methods remove the majority of irrelevant tags on early steps. It means that only several steps of EM-algorithm are needed for feature selection. Further optimization will just tune the remaining hyperparameters.

The complete algorithm of RTM-MAP-EM is provided in Appendix B.

4 Experiments

4.1 Synthetic data

The ability of the presented methods to remove irrelevant features on a synthetic dataset was studied and compared to two classic feature selection models — RVM, where a similar idea is applied to linear regression, and L1-regularized logistic regression. There were 500 objects and 50 features. The data consisted of genuine tags that were used to generate the class label, and two types of irrelevant tags: random tags and tags that were correlated to some of the genuine tags. Relevance determination accuracy is shown in Table 1.

Both EM-based methods successfully remove nearly all random features and most correlated features. The RTM-MAP-EM, RTM-EM, and RVM give comparable results and detect random features better than L1-regularized logistic regression (L1–LR). However, L1–LR provided the best results in removing correlated tags.

4.2 Sentiment analysis

Also, the methods were tested on a real task: sentiment analysis problem [9]. In this problem, objects are sentences and the task is to classify them into positive and negative ones. A bag of words representation was used (each tag represents a word; $x_{ij} = 1$ if object $\boldsymbol{x}_i$ contains the jth word from the dictionary). There were 1000 train objects, 411 test objects, and 1869 features. There were 11 tags per objects in average. Test set classification accuracy is shown in Table 2.

Table 2 Prediction performance on sentiment analysis dataset

Method	Prediction accuracy
RTM-MAP-EM	0.9659
RVM	0.9586
L1-LR	0.9708
RF	0.9416
GBDT	0.9683
SVM	0.9683

The present method (RTM-MAP-EM) was compared to different state-of-the-art classifiers like the RVM, L1-LR, Random Forest (RF), gradient boosting over decision stumps (GBDT), and SVM.

The present method provides prediction accuracy that is comparable to classical methods. It also provides a way to sort features with respect to their importance: RTM-MAP-EM chose about 70 tags to be relevant and removed everything else; L1-LR chose about 120 tags; and the RVM chose about 230 tags. A histogram of weights of most relevant words for these methods is shown in Figs. 3–5.

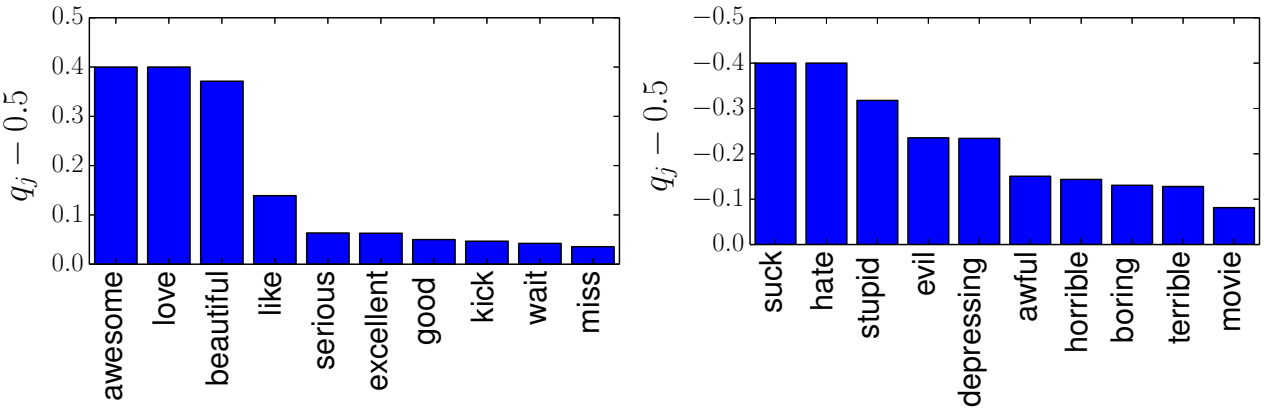

Figure 3 $q_j - 0.5$ for most relevant tags according to RTM-MAP-EM

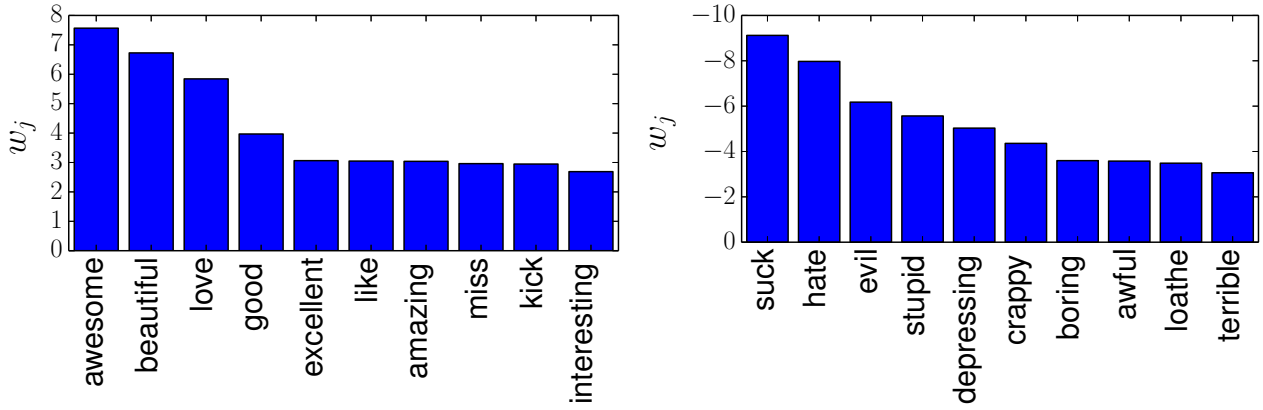

Figure 4 Weights of linear model tuned by L1-LR

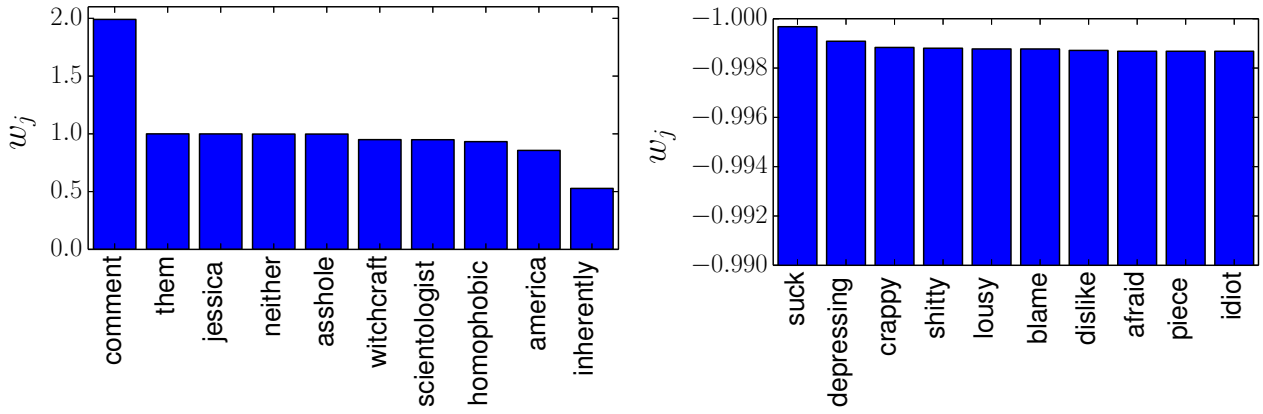

Figure 5 Weights of linear model tuned by RVM

The RVM chose a lot of rare words to represent the negative class (words "crappy," "shitty," "lousy," "blame," "afraid," "piece," and "idiot" have less than seven occurrences in the dataset) and the words from the positive class does not look relevant at all. The RVM failed to solve the relevance determination problem on this dataset.

The words chosen by RTM-MAP-EM and logistic regression are quite intuitive in case of this problem. the present model, it isn't true. Most of them are very emotional. Top-20 words chosen by logistic regression are almost the same as top-20 words, chosen by the present method. However, the present model provided a more sparse solution with comparable prediction performance. Therefore, the present method proved to be better at relevance determination than logistic regression and the RVM on this dataset.

5 Concluding Remarks

Most of previous work on Bayesian ARD approach consider only Gaussian prior. The authors demonstrate that other appropriate priors may also work well. It means that Bayesian ARD approach might be more broad than it was considered before and is not limited to the usage of Gaussian prior. Also, a method to solve a binary classification problem with binary features is suggested and an experimental comparison which shows that the present model is comparable to the state-of-the-art methods of classification and feature selection is provided. The experiments show that the present model provides better feature selection results than the classic feature selection models like RVM and L1-LR.

Appendix A

Proof of Theorem 1

Derive a variational lower bound on the likelihood function for a single object $\boldsymbol{x}$. Let Q be the set of its tags: $Q = \{j|x_j = 1\}$. After some transformations and a change of variables, a convex function is obtained and its tangent is used as its variational lower bound:

$$\mathsf{P}(t \,|\, \boldsymbol{x}, \boldsymbol{q}) = \frac{\prod\limits_{j\in Q} q_j^t (1-q_j)^{1-t}}{\prod\limits_{j\in Q} q_j + \prod\limits_{j\in Q} (1-q_j)} = \left(1 + \prod_{j\in Q} \left(\frac{1-q_j}{q_j}\right)^{2t-1}\right)^{-1}$$

as $t \in \{0, 1\}$;

$$s_j := \left(\frac{1-q_j}{q_j}\right)^{|Q|(2t-1)};$$

so,

$$\log \mathsf{P}(t \,|\, \boldsymbol{x}, \boldsymbol{q}) = -\log\left(1 + \left(\prod_{j\in Q} s_j\right)^{1/|Q|}\right). \tag{6}$$

As $s_j > 0$, the geometric mean $\left(\prod\limits_{j\in Q} s_j\right)^{1/|Q|}$ is concave with respect to $\boldsymbol{s}$ [7, p. 74]. As $f(x) = -\log x$ is convex and nonincreasing, the whole expression on the right part of (6) is convex with respect to $\boldsymbol{s}$ [7, p. 84]. Therefore, its tangent is its variational lower bound and after making inverse change of variables and taking the exponent, one obtains a variational lower bound on $\mathsf{P}(t \,|\, \boldsymbol{x}, \boldsymbol{q})$.

The variational lower bound on the likelihood of an object $\boldsymbol{x}_i$ from the training set looks as follows:

$$\mathsf{P}(t_i \,|\, \boldsymbol{x}_i, \boldsymbol{q}) \geqslant L_i(\boldsymbol{q}, \boldsymbol{\eta}_i) = c_i(\boldsymbol{\eta}_i) \exp\left(\sum_{j\in Q_i} \tilde{c}_{ij}(\boldsymbol{\eta}_i) \left(\frac{1-q_j}{q_j}\right)^{|Q_i|(2t_i-1)}\right)$$

where

$$Q_i = \{j|x_{ij} = 1\};$$

$$c_i(\boldsymbol{\eta}_i) = \frac{\prod\limits_{j\in Q_i} \eta_{ij}^{t_i}(1-\eta_{ij})^{1-t_i}}{\prod\limits_{j\in Q_i} \eta_{ij} + \prod\limits_{j\in Q_i} (1-\eta_{ij})} \exp\left(\frac{\prod\limits_{j\in Q_i} \eta_{ij}^{1-t_i}(1-\eta_{ij})^{t_i}}{\prod\limits_{j\in Q_i} \eta_{ij} + \prod\limits_{j\in Q_i} (1-\eta_{ij})}\right);$$

$$\tilde{c}_{ij}(\boldsymbol{\eta}_i) = -\frac{\prod\limits_{j\in Q_i} \eta_{ij}^{t_i}(1-\eta_{ij})^{1-t_i}}{\prod\limits_{j\in Q_i} \eta_{ij} + \prod\limits_{j\in Q_i} (1-\eta_{ij})} \left(\frac{\eta_{ij}}{1-\eta_{ij}}\right)^{|Q_i|(2t_i-1)} |Q_i|^{-1}.$$

The following equation concludes the proof:

$$E(\boldsymbol{\alpha}) = \int \mathsf{P}(\boldsymbol{t} \,|\, \boldsymbol{X}, \boldsymbol{q}) p(\boldsymbol{q} \,|\, \boldsymbol{\alpha})\, d\boldsymbol{q} \geqslant \int \prod_{i=1}^{n} L_i(\boldsymbol{q}, \boldsymbol{\eta}_i) \prod_{j=1}^{n} p(q_j \,|\, \alpha)\, d\boldsymbol{q} = \tilde{E}(\boldsymbol{\alpha}, \boldsymbol{H}).$$

Note that each object has its own set of variational parameters. As $\boldsymbol{\eta}_{ij}$ is dummy if $q_j \notin Q_i$, there are $\sum_{i,j} x_{ij}$ essential variational parameters.

Appendix B

RTM-MAP-EM algorithm

Algorithm 1 RTM-MAP-EM

Require: training set $(\boldsymbol{X}, \boldsymbol{t})$; maximum number of iterations T; tolerance ε
Ensure: tuned vector of hyperparameters $\boldsymbol{\alpha}$

$\boldsymbol{\alpha}^0 \leftarrow (1, \ldots, 1)^{\mathrm{T}}$ // *Initial value of hyperparameters*
$\boldsymbol{\eta}^0 \leftarrow \arg\max_{\boldsymbol{q}} \sum_{i=1}^{n} \log \mathsf{P}(t_i \,|\, \boldsymbol{x}_i, \boldsymbol{q}) + \log \mathsf{P}(\boldsymbol{q} \,|\, \boldsymbol{\alpha}^0)$ // $\boldsymbol{\eta} = \boldsymbol{q}^{\mathrm{MAP}}$
for $k = 0$ to T
  // *E-step:*
  $\boldsymbol{\eta}^k \leftarrow \arg\max_{\boldsymbol{q}} \sum_{i=1}^{n} \log \mathsf{P}(t_i \,|\, \boldsymbol{x}_i, \boldsymbol{q}) + \log \mathsf{P}(\boldsymbol{q} \,|\, \boldsymbol{\alpha}^{k-1})$ // $\boldsymbol{\eta} = \boldsymbol{q}^{\mathrm{MAP}}$
  $\boldsymbol{H}^k \leftarrow (\boldsymbol{\eta}^k, \ldots, \boldsymbol{\eta}^k)^T$
  // *M-step:*
  $\boldsymbol{\alpha}^k \leftarrow \arg\max_{\boldsymbol{\alpha}} \log \tilde{E}(\boldsymbol{\alpha}, \boldsymbol{H}^k)$
  if $\|\boldsymbol{\alpha}^k - \boldsymbol{\alpha}^{k-1}\|^2 < \varepsilon$ **then**
    break
$\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha}^k$
return $\boldsymbol{\alpha}$

Some notes about implementation:

- q_j and η_{ij} are bound to $[0.1, 0.9]$ for all i, j. Otherwise, computations tend to be unstable;
- $k_E = k_M = 10$ in all experiments;
- α_j is bound to $[0, 1000]$ for all j. If $\alpha_j \geqslant 900$, it is considered to be infinite: $\alpha_j := +\infty$; and
- there seems to be no need to wait till full convergence; for RTM-MAP-EM, $T = 20$ was enough in both experiments.

References

[1] MacKay, D., and R. Neal. 1994. Automatic relevance determination for neural networks. Cambridge University. Technical Report.

[2] Tipping, M. E. 2001. Sparse Bayesian learning and the relevance vector machine. *J. Machine Learning Res.* 1:211–244.

[3] MacKay, D. 1992. Bayesian interpolation. *Neural Computation* 4:415–447.

[4] Bishop, C. M. 2006. *Pattern recognition and machine learning*. New York, NY: Springer. 738 p.

[5] Salakhutdinov, R., S. Roweis, and Z. Ghahramani. 2002. On the convergence of bound optimization algorithms. *19th Conference on Uncertainty in Artificial Intelligence Proceedings*. 10:509–516.

[6] Jaakkola, T. S., and M. I. Jordan. 2000. Bayesian logistic regression: A variational approach. *Stat. Comput.* 10:25–37.

[7] Boyd, S., and L. Vandenberghe. 2004. *Convex optimization*. Cambridge: Cambridge University Press. 716 p.

[8] Byrd, R., P. Lu, and J. Nocedal. 1995. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Stat. Comput.* 16(5):1190–1208.

[9] Kaggle in Class. 2011. UMICH SI650 — Sentiment Classification. Available at: `http://inclass.kaggle.com/c/si650winter11/data` (accessed December 29, 2015).

Received June 14, 2015

Литература

[1] *MacKay D., Neal R.* Automatic relevance determination for neural networks // Cambridge University, 1994. Technical Report.

[2] *Tipping M. E.* Sparse Bayesian learning and the relevance vector machine // J. Machine Learning Res., 2001. No. 1. P. 211–244.

[3] *MacKay D.* Bayesian interpolation // Neural Computation, 1992. No 4. P. 415–447.

[4] *Bishop C. M.* Pattern recognition and machine learning. — New York, NY, USA: Springer, 2006. 738 p.

[5] *Salakhutdinov R., Roweis S., Ghahramani Z.* On the convergence of bound optimization algorithms // 19th Conference on Uncertainty in Artificial Intelligence Proceedings, 2002. No. 10. P. 509–516.

[6] *Jaakkola T. S., Jordan M. I.* Bayesian logistic regression: A variational approach // Stat. Comput., 2000. No. 10. P. 25–37.

[7] *Boyd S., Vandenberghe L.* Convex optimization. — Cambridge: Cambridge University Press, 2004. 716 p.

[8] *Byrd R., Lu P., Nocedal J.* A limited memory algorithm for bound constrained optimization // SIAM J. Sci. Stat. Comput., 1995. Vol. 16, No. 5. P. 1190–1208.

[9] Kaggle in Class. UMICH SI650 — Sentiment Classification. 2011. `http://inclass.kaggle.com/c/si650winter11/data`.

Поступила в редакцию 14.06.2015