Logistic regression with weight grouping priors

M. Korzeń[a], S. Jaroszewicz[b], P. Klęsk[1a]

[a]*Department of Methods of Artificial Intelligence and Applied Mathematics, Westpomeranian University of Technology, Szczecin, Poland*
[b]*Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland; National Institute of Telecommunications, Warsaw, Poland*

Abstract

A generalization of the commonly used Maximum Likelihood based learning algorithm for the logistic regression model is considered. It is well known that using the Laplace prior (L^1 penalty) on model coefficients leads to a variable selection effect, when most of the coefficients vanish. It is argued that variable selection is not always desirable; it is often better to group correlated variables together and assign equal weights to them. Two new kinds of a-priori distributions over weights are investigated: Gaussian Extremal Mixture (GEM) and Laplacian Extremal Mixture (LEM) which enforce grouping of model coefficients in a manner analogous to L^1 and L^2 regularization. An efficient learning algorithm is presented, which simultaneously finds model weights and the hyperparameters of those priors. Examples are shown in the experimental part where the proposed a-priori distributions outperform Gauss and Laplace priors as well as other methods which take coefficient grouping into account, such as the elastic net. Theoretical results on parameter shrinkage and sample complexity are also included.

Keywords: logistic regression, regularization, weight grouping, attributes grouping

1. Introduction

Variable selection problem for linear models is considered one of the most important in statistical inference (Hesterberg et al., 2008). Recently, many new variable selection methods became popular, including stagewise selection (Hastie et al., 2009) and L^1-regularization techniques such as *Lasso* (Williams, 1994; Tibshirani, 1996; Mkhadri & Ouhourane, 2013) and *LARS* (Efron et al., 1996). However, variable selection is not always the best possible approach.

If the predictor variables are correlated, it is often more desirable to group correlated variables together and assign them equal or similar weights. A more detailed justification is given in Section 2, where it is argued that variable *averaging* may give much better results than variable selection.

In order to achieve such averaging, we devise a supervised learning algorithm maximizing the log-likelihood criterion with suitably chosen priors on model weights. Our priors correspond to mixtures of Gaussian or Laplace distributions which force the weights to cluster around the means of the mixture components. The priors work analogously to L^1 and L^2 regularization, such that the prior based on the Gaussian distribution forces the weights to lie close to their group averages, and the prior based on the Laplace distribution forces most weights to be *exactly equal* to their group averages. As the resulting optimization problem is nonconvex, we present an algorithm, similar to the EM approach, consisting of two repeated steps: (1) maximization of log-likelihood for the current assignment of variables to groups, and (2) reassignment of variables with identical or similar weights to appropriate groups. Theoretical properties of the proposed method, such as parameter shrinkage and sample complexity have also been analyzed.

The advantages of coefficient grouping and averaging have, of course, already been recognized by researchers, and several methods which allow for weight grouping in regression models have been proposed. We will now review those approaches and explain the differences from the method proposed in this paper.

[1]ul. Żołnierska 49, 71-210 Szczecin, Poland, tel. +48 91 4495556, fax. +48 91 4495990

Email addresses: mkorzen@wi.zut.edu.pl (M. Korzeń), s.jaroszewicz@ipipan.waw.pl (S. Jaroszewicz), pklesk@wi.zut.edu.pl (P. Klęsk)

Zou & Hastie (2005) introduce a method called *elastic net*, which combines L^1 and L^2 regularization. The L^1 term enforces variable selection, while the L^2 term introduces a 'grouping effect', thanks to which correlated variables tend to have similar coefficients. The grouping effect is, however, just a by-product of the regularization method used, and it is thus difficult to control its strength; typically it is not possible to enforce equal or approximately equal weights. In contrast, our method allows for direct control over the strength of the grouping effect (which we demonstrate theoretically) and coefficients of correlated variables can be forced to lie arbitrarily close to each other. Moreover, the prior based on the mixture of Laplace distributions allows for enforcing strict equality of most weights to their respective group averages.

A technique called *group Lasso* has been described in (Yuan & Lin, 2004; Kim et al., 2006) (and introduced earlier by Bakin (1999)), which extends Lasso by taking into account the group structure of variables. However the groups need to be specified in advance and incorporated into the regularization term. Our method, on the other hand, groups variables automatically. Moreover, group Lasso does not allow for direct control over the relative sizes of weights within groups, while our approach gives the analyst precise control of the grouping behavior.

If attributes are ordered in some natural way (e.g. in time series data), there is an interesting approach called *fused lasso*, where both large weight values and large differences between consecutive weights are penalized (Friedman et al., 2007; Tibshirani et al., 2005). The approach has been generalized to image data by requiring similar weights for variables corresponding to adjacent pixels. Our motivation is different, as we require whole groups of attributes to have similar weights, not just consecutive or adjacent ones.

The remaining part of the paper is organized as follows: we present the motivation in Section 2, give a detailed description of the proposed method in Section 3, describe the optimization algorithm in Section 4, and analyze the approach theoretically and experimentally in Sections 5 and 6. Section 7 concludes.

2. Motivation and a simplified model

Consider a simplified model with s independent hidden variables H_k, $k = 1, \ldots, s$, which are not observed directly, but which are directly correlated with the target variable Y. Instead of H_k, we can only observe a set of variables X_j, $j = 1, \ldots, n$, which are noisy observations of the H_k, each X_j depending on a single hidden variable H_k. More precisely, for every j, $X_j = H_k + \xi_j$ for some k, where ξ_j is a random noise term with zero mean and variance σ_k^2, equal for all X_j depending on a given H_k. Assume further, that $\mathrm{Cov}(\xi_j, H_k) = \mathrm{Cov}(\xi_j, Y) = 0$ for all values of j and k, and $\mathrm{Cov}(\xi_i, \xi_j) = 0$ for all $i \neq j$, but for all k, $\mathrm{Cov}(Y, H_k) \neq 0$. Our task is to construct a linear predictor of Y based on X_j's:

$$\hat{Y} = \sum_{j=1}^{n} \alpha_j X_j.$$

We now state a lemma, which underlies the main motivation of the paper. We first introduce some additional notation. Two indexing functions κ, $\jmath$ will be used: $\kappa(j)$ gives an index from $\{1, \ldots, s\}$ such that X_j is a noisy observation of $H_{\kappa(j)}$; $\jmath(k, j)$ gives an index from $\{1, \ldots, n\}$ of the j-th variable dependent on H_k. Further, let n_k denote the number of variables being the noisy observations of H_k.

Lemma 2.1. *Suppose H_k, $k = 1, \ldots, s$, are the independent hidden variables in the model described above, and $X_j = H_{\kappa(j)} + \xi_j$, $j = 1, \ldots, n$, are the observed variables. Assume all noise terms ξ_j have zero mean and variance $\sigma^2_{\kappa(j)}$ (equal for all noisy observations of H_k). Then, for any constants A_k and any numbers $\alpha_{\jmath(k,j)} \geq 0$ such that $\sum_{j=1}^{n_k} \alpha_{\jmath(k,j)} = 1$ (for all k) we have:*

$$E\left(\sum_{k=1}^{s} A_k \sum_{j=1}^{n_k} \alpha_{\jmath(k,j)} X_{\jmath(k,j)} - Y\right)^2 \geq E\left(\sum_{k=1}^{s} A_k \frac{1}{n_k} \sum_{j=1}^{n_k} X_{\jmath(k,j)} - Y\right)^2. \tag{1}$$

The proof can be found in Appendix A.1. The interpretation of the lemma is as follows: if there exists a set of independent hidden variables H_k, which are related to the target variable Y by a linear relationship

$$Y = A_1 H_1 + \ldots + A_s H_s + \xi, \tag{2}$$

where ξ is a noise term with zero mean, and a set of observed variables X_j (being noisy observations of the H_k's), then the best modeling approach is to form averages of observed variables in each group ($\frac{1}{n_k}\sum_{j=1}^{n_k} X_{J(k,j)}$) and to build a linear model based on these averages. Moreover, one can see that the averages reconstruct the hidden variables H_k and that the coefficients corresponding to the averages are good estimates of A_k's.

Unfortunately in practice we do not know which X_j's correspond to which hidden variables or how close their relationship is. In the following two sections we introduce an algorithm which allows for finding groupings of variables which presumably are noisy observations of the same hidden variable, and which automatically estimates how close the coefficient of each variable in a group should be to the group's average.

3. A-priori distributions inducing a weight grouping effect

In order to induce the desired coefficient grouping effect (which we want to achieve, having in mind the motivating lemma) we propose a supervised learning algorithm which maximizes the log-likelihood of the weights given data, but is equipped with suitably chosen a-priori distributions. In this section we propose two such priors, which we call the **GEM** (Gaussian Extremal Mixture) and the **LEM** (Laplace Extremal Mixture). Those priors are one of the main contributions of the paper.

To gain some intuition recall first the well known L^2 and L^1 regularization techniques, which are equivalent to imposing, respectively, Gaussian and Laplace priors on weights:

$$p_{\mathrm{G}}(w|\sigma) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{w^2}{2\sigma^2}\right),$$
$$p_{\mathrm{L}}(w|\lambda) = \frac{1}{2\lambda}\exp\left(-\frac{|w|}{\lambda}\right).$$

It is well known that the Gaussian prior forces weights to be small and cluster around zero, whereas the Laplace prior forces many of them to be *exactly equal* to zero (Cawley & Talbot, 2006). From our perspective such models can be regarded as a *single group* of variables with zero average weight (center). We now propose an extension of this model to *multiple groups* with arbitrary centers by introducing the following *extremal mixture* priors on each weight

GEM Gaussian Extremal Mixture model:

$$p^*_{\mathrm{GEM}}(w|\mathbf{c},\sigma) = q(\mathbf{c})\max_{k=1,\ldots,s}\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(w-c_k)^2}{2\sigma^2}\right), \tag{3}$$

LEM Laplace Extremal Mixture model:

$$p^*_{\mathrm{LEM}}(w|\mathbf{c},\lambda) = q(\mathbf{c})\max_{k=1,\ldots,s}\frac{1}{2\lambda}\exp\left(-\frac{|w-c_k|}{\lambda}\right), \tag{4}$$

where s stands for the number of components in the mixture (i.e. the number of groups) and $q(\mathbf{c})$ is a suitable normalizing factor. The value of c_k is the center of the k-th group; coefficients of variables in this group will cluster around this value. The joint prior on the whole weight vector $\mathbf{w}$ is the product of the priors on all w_i's.

The normalizing factors $q(\mathbf{c})$ will be dropped in subsequent discussions, giving the following unnormalized priors:

$$p_{\mathrm{GEM}}(w|\mathbf{c},\sigma) = \max_{k=1,\ldots,s}\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(w-c_k)^2}{2\sigma^2}\right) \propto p^*_{\mathrm{GEM}}(w|\mathbf{c},\sigma), \tag{5}$$
$$p_{\mathrm{LEM}}(w|\mathbf{c},\lambda) = \max_{k=1,\ldots,s}\frac{1}{2\lambda}\exp\left(-\frac{|w-c_k|}{\lambda}\right) \propto p^*_{\mathrm{LEM}}(w|\mathbf{c},\lambda). \tag{6}$$

Since $q(\mathbf{c})$ does not depend on w, dropping it in the computations related to the weight update is not problematic. In Section 4.3 we justify this simplification also in the context of updating cluster centers.

The hyperparameters σ and λ are responsible for the spread of mixture components. For notational simplicity, let us denote by γ the suitable spread coefficient for each distribution, i.e. $\gamma = 1/(2\sigma^2)$ for the Gaussian and $\gamma = 1/\lambda$ for

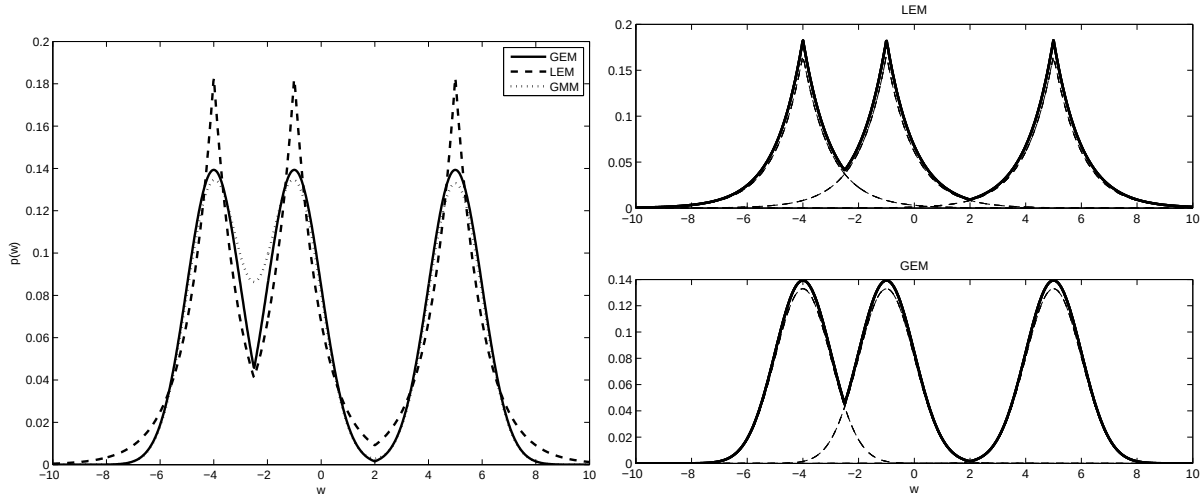


Figure 1: A-priori distributions on model coefficients inducing a weight grouping effect. For comparison, the classical mixture of Gaussians (labeled GMM) is included in the left figure. The two figures on the right show the contributions of the mixture components to the LEM and GEM densities. Note that the actual density (thick black line) is slightly above the maximum of the components (thin dashed lines); this is the result of their overlap, which is reflected in the normalizing factor $q(\mathbf{c})$ in Formulas 3 and 4.

the Laplace prior. This way, γ is a unified penalty coefficient. From now on, all priors shall be parametrized by the vector of their centers $\mathbf{c} = (c_1, \ldots, c_s)^T$ and their spread γ. Examples of densities of the proposed priors are depicted in Figure 1.

It is worth noting that instead of considering extremal mixtures it is also possible to consider classical mixtures: $p_{\mathrm{GMM}}(w|\mathbf{c}, \sigma) = \sum_{k=1}^{s} q_k \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(w-c_k)^2}{2\sigma^2}\right)$ for the Gaussian and $p_{\mathrm{LMM}}(w|\mathbf{c}, \lambda) = \sum_{k=1}^{s} q_k \frac{1}{2\lambda} \exp\left(-\frac{|w-c_k|}{\lambda}\right)$ for the Laplace distribution. These, however, lead to optimization difficulties. Using the EM-like algorithm yields a solution with all the group centers converging to a single point. Moreover, our experiments did not show any advantages of those mixtures over GEM and LEM.

Now, let us introduce some notation related to logistic regression. Let $D = (\mathbf{x}_i, d_i)_{i=1,\ldots,m}$ be a dataset, where $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in})^T \in \mathbb{R}^n$ are input vectors, $d_i \in \{0, 1\}$ are class labels, and m is the size of the dataset. The predicted variable y is interpreted as the probability that the class variable is equal to one. Typically, in this setting, the logistic regression model is used

$$y(\mathbf{x}; \mathbf{w}, b) = \frac{1}{1 + e^{-(\mathbf{w}^T\mathbf{x} + b)}}. \tag{7}$$

Although we focus mainly on the logistic model, most results remain equally valid for linear regression and for other Generalized Linear Models.

By the Bayes theorem, the posterior density of the weights given the data and the hyperparameters of the priors is

$$p(\mathbf{w}, b|D, \mathbf{c}, \gamma) \propto p(D|\mathbf{w}, b, \mathbf{c}, \gamma) p(\mathbf{w}, b|\mathbf{c}, \gamma),$$

where $p(\mathbf{w}|\mathbf{c}, \gamma) = \prod_{j=1}^{n} p(w_j|c, \gamma)$ and $p(w_j|\mathbf{c}, \gamma)$ is either a GEM or a LEM, and $p(b|\mathbf{c}, \gamma) = 1$. We assume that spreads of all mixture components are equal and fixed, while the centers are unknown and allowed to vary. The intercept b is not be regularized.

The next section describes an algorithm which, combining the proposed prior distributions with the simplified model described in Section 2, allows for fitting the hyperparameters of the priors to data. As the resulting optimization problem is nonconvex, we present an algorithm which works similarly to the EM approach and consists of two repeated steps: (1) maximization of the log-likelihood for the current assignment of variables to groups, and (2) reassignment of variables with identical or similar weights to appropriate groups. Theoretical properties of the proposed method, such as parameter shrinkage and sample complexity, are analyzed in the following sections.

4. Optimization algorithm and implementation details

The simplified model described in Section 2 poses some practical difficulties for a learning algorithm. Even if we assume that the hidden variables do exist, we know neither the number s of such variables, nor the assignments of observed variables X_j to hidden variables H_k. In other words we do not know how to partition the observed variables into groups. In this section we a present a learning algorithm which, together with the a-priori distributions on weights introduced in Section 3, allows for automatic grouping of weights (and hence also variables) based on the training data.

As mentioned in the introduction, the learning algorithm we propose is similar to the unsupervised Expectation Maximization (EM) algorithm. It consists of two steps: maximization of log-likelihood for the current grouping of weights, and re-assignment of weights with the same or similar values. Let us begin the presentation by describing the first step. A comparison (Minka, 2003) of various learning algorithms for logistic regression shows that the conjugate gradient descent based approach works quite well for logistic regression with the L^2 penalty. Carefully implemented gradient methods can also be used for learning with the L^1 penalty. We thus use this technique to learn models involving the proposed prior distributions on weights.

In order to use such optimization methods, we need to compute the gradient of the risk function, and perform linear optimization in the direction of the gradient. We describe both tasks in turn.

4.1. Computing the gradient

Let us write down the a-posteriori density of the weights and its logarithm

$$\begin{aligned} p(\mathbf{w}, b|D, \mathbf{c}, \gamma) &\propto M(\mathbf{w}, b|D, \mathbf{c}, \gamma) \qquad (8)\\ &= \prod_{i=1}^{m} y(\mathbf{x}_i; \mathbf{w}, b)^{d_i}(1 - y(\mathbf{x}_i; \mathbf{w}, b))^{1-d_i} \prod_{j=1}^{n} p(w_j|\mathbf{c}, \gamma),\\ Q(\mathbf{w}, b) &= \log M(\mathbf{w}, b|D, \mathbf{c}, \gamma)\\ &= \sum_{i=1}^{m}\Big(d_i \log y(\mathbf{x}_i; \mathbf{w}, b) + (1 - d_i)\log(1 - y(\mathbf{x}_i; \mathbf{w}, b))\Big)\\ &\quad + \sum_{j=1}^{n} \log\Big(p(w_j|\mathbf{c}, \gamma)\Big)\\ &= Q_D(\mathbf{w}) + Q_W(\mathbf{w}, \mathbf{c}). \qquad (9) \end{aligned}$$

The function to be maximized consists of two terms: Q_D, the log-likelihood of weights given the *data* and Q_W, the logarithm of the prior density on *weights*. The gradient of Q_D can be found in most texts on logistic regression. Moreover, one can show (see Appendix A.2) that the gradient of Q_W with respect to the weights is equal to

$$\frac{\partial \log(p_{\mathrm{GEM}})}{\partial w_j} = -2\gamma(w_j - c_{\kappa(j)}), \qquad (10)$$

$$\frac{\partial \log(p_{\mathrm{LEM}})}{\partial w_j} = -\gamma \operatorname{sgn}(w_j - c_{\kappa(j)}). \qquad (11)$$

4.2. Directional optimization

Once we know the gradient, the centers c_k, and the assignment of variables to those centers, we perform a line search to optimize the weights in the direction of the gradient. For the GEM model the minimized function is smooth along the direction of the gradient. A more difficult case arises for the LEM model, where the function is not smooth as there exist points where the derivative is not defined. We use the condition for the existence of an optimum of Q at a given point described in (Shevade & Keerthi, 2003) and (Cawley & Talbot, 2006). At the optimum, the following statement needs to hold

$$\frac{\partial \log(Q)}{\partial w_j} = \frac{\partial \log(Q_D)}{\partial w_j} - \gamma \operatorname{sgn}(w_j - c_{\kappa(j)}) = 0, \qquad (12)$$

which leads to conditions $w_j = c_{\kappa(j)}$, if $\left|\frac{\partial \log(Q_D)}{\partial w_j}\right| < \gamma$ and $w_j \neq c_{\kappa(j)}$, if $\left|\frac{\partial \log(Q_D)}{\partial w_j}\right| = \gamma$, see Cawley & Talbot (2006) for details. If the first condition is met, the corresponding weight is set exactly equal to $c_{\kappa(j)}$. In our implementation, if the following two conditions are met:

$$|w_j - c_{\kappa(j)}| < 10^{-8}, \tag{13}$$

$$\left|\frac{\partial \log(Q_D)}{\partial w_j}\right| < \gamma, \tag{14}$$

we assume that the j-th coordinate of the gradient is equal to zero and set $w_j = c_{\kappa(j)}$.

4.3. Update rule for cluster centers

Now we want to find the update rule for the coefficient group centers c_k, $k = 1, \ldots, s$, which maximizes Q for our extremal models GEM and LEM.

Unfortunately, unlike the update rule for weights $\mathbf{w}$, the update rule for the centers cannot be described in purely Bayesian terms. The normalizing factors $q(\mathbf{c})$ in (3) and (4) depend on the centers $c_1, \ldots, c_s$, making such an update computationally difficult, which led us to simply ignoring those factors above. With this simplification, the update of cluster centers is much easier but does not have a strict Bayesian interpretation. We will now describe several arguments supporting the use of penalties given in Equations 5 and 6 instead of true Bayesian priors.

First, one can ignore the Bayesian interpretation and recast the problem within the *cost-penalty framework* with the cost equal to the log-likelihood of the data $Q_D(\mathbf{w})$ and penalty equal to (5) or (6). Note, that the same exact simplification is made by the k-means clustering algorithm and works well in practice.

One way to regain the Bayesian formulation also for updating the cluster centers is to include an additional improper prior on $\mathbf{c}$ proportional to $1/q^n(\mathbf{c})$. While introducing such a prior does seem like a kludge, the analogy to the k-means algorithm, again, lends it credibility.

A more elegant justification is offered in (Kulis & Jordan, 2012) or (Spurek & Tabor, 2012), where it is shown that the Bayesian interpretation of the k-means algorithm is recovered, when the within-cluster variance (corresponding to γ in our priors) tends to zero. This is equivalent to distances between cluster centers increasing to infinity. To see how this relates to our case, look at the right part of Figure 1. The dependence of the normalizing factor $q(\mathbf{c})$ on $\mathbf{c}$ is caused by the overlapping of the mixture components. Notice however that when the components become very narrow, or the centers very spread out, this overlap disappears and the normalizing factor q becomes independent of $\mathbf{c}$. Thus, Equations 5 and 6 are an approximation of true priors for the centers.

A yet another justification can be obtained by noticing that $\frac{1}{s} \leq q(\mathbf{c}) < 1$. Now, maximizing $Q_D + Q_W$ is equivalent to maximizing $Q_D + Q_W - n \log s$, which in turn is a lower bound for $Q_D + \sum_{i=1}^{n} \log p^*_{\mathrm{GEM}}$ or $Q_D + \sum_{i=1}^{n} \log p^*_{\mathrm{LEM}}$ respectively. So, after dropping the factors $q(\mathbf{c})$ we are in fact maximizing a lower bound on the true a-posteriori distribution.

After this justification we now proceed to the derivation of the update rule for cluster centers. It is easy to see, using an argument analogous to that for the derivatives with respect to weights, that the derivatives of Q_W with respect to c_k are

$$\frac{\partial \log(p_{\mathrm{GEM}})}{\partial c_k} = 2\gamma \sum_{j=1}^{n_k} (w_{\jmath(k,j)} - c_k), \tag{15}$$

$$\frac{\partial \log(p_{\mathrm{LEM}})}{\partial c_k} = \gamma \sum_{j=1}^{n_k} \operatorname{sgn}(w_{\jmath(k,j)} - c_k). \tag{16}$$

where the function $\jmath$ was defined in Section 2 and n_k is the number of weights in the k-th group.

For the Gaussian extremal model, using the condition $\partial \log(p_{\mathrm{GEM}})/\partial c_k = 0$, we obtain the following update rule for c_k

$$c_k = \frac{1}{n_k} \sum_{j=1}^{n_k} w_{\jmath(k,j)} = \operatorname{mean}(w_{\jmath(k,\cdot)}). \tag{17}$$

We remark that this is exactly the same update step as is used in the k-means clustering algorithm (here s-means).

For the Laplace extremal model, the condition $\partial \log(Q_{\mathrm{LEM}})/\partial c_k = 0$ leads to c_k being a point such that $\sum_{j=1}^{n_k} \operatorname{sgn}(w_{J(k,j)} - c_k) = 0$. Thus, equal numbers of weights have to lie on both sides of c_k. Note that this condition cannot always be met and in some cases c_k is not unique. But typically, the following equation holds

$$c_k = \operatorname{median}(w_{J(k,\cdot)}), \tag{18}$$

which is the update step of the s-medians clustering algorithm.

4.4. The final algorithm

The algorithm uses a modified conjugate gradients method. The difference is, that after each linear optimization step for the weights, we compute new assignments of weights to the centers, as well as new centers using Formulas 17 and 18. Both updates ensure that Q increases during each step.

The algorithm is presented in Figure 2. Note that during each iteration, the assignments of weights to groups (centers) can obviously change, and therefore one should not regard the indexing functions κ and J as fixed.

In Figure 3 we illustrate the evolution of the weights for increasing penalty γ for an artificial dataset described in Section 6. One can immediately see that the weights shrink towards the centers of their respective clusters and that the LEM prior enforces equality of most weights to their centers. See also Figure 8 in Section 6. An extensive experimental evaluation can also be found in that section.

```
1:  function [w, b]=groupingREG(D, s)
2:      w := 0;  b := 0;   i := 0;   c := 0
3:      while i ≤ i_max do
4:          [g_w, g_b] := gradient(Q(w, b))
5:          [q_w, q_b] := [q_w, q_b] + β[g_w, g_b]      % the conjugate direction
6:          [w, b] = arg max_t Q(w + t · q_w, b + t · q_b)      % line search in the conjugate direction
7:          update c_k, k = 1, ..., s using the s-means (or s-medians) update step, Equations (17) or (18)
8:          if norm(g_w) < 1e − 8 then
9:              break;
10:         end if
11:         i := i + 1
12:     end while
13: end function
```

Figure 2: The conjugate gradients based learning algorithm, s is the assumed number of weight clusters.

5. Theoretical properties

In this section we present analogues of shrinkage theorems given in (Zou & Hastie, 2005), as well as theorems on the sample complexity of the proposed method.

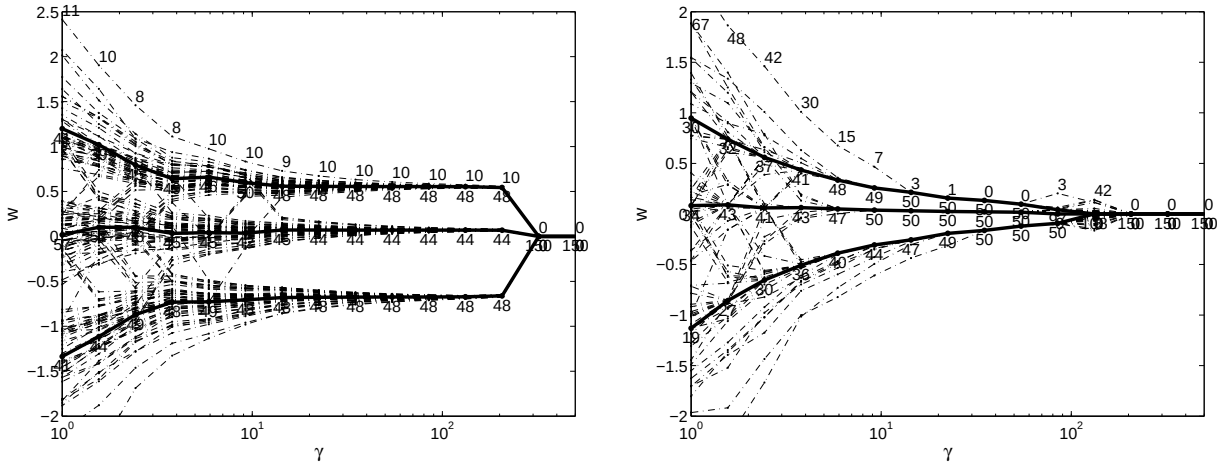
5.1. Shrinkage theorems

The following two shrinkage theorems are analogues of Theorem 2 from (Zou & Hastie, 2005) and guarantee that with a suitably chosen penalty coefficient, weights for highly correlated attributes shrink to become close (for GEM) or exactly equal (for LEM) to their group centers.

Theorem 5.1. *Consider the logistic regression model with the GEM prior. Let $x_{\cdot j}$, $j = 1, \ldots, n$, be standardized data columns (with zero means and $\|x_{\cdot j}\|_2 = 1$). Further, let w_{j_1}, w_{j_2} be optimal weights of two variables $x_{\cdot j_1}$ and $x_{\cdot j_2}$ which are assigned to the same cluster, i.e. $\kappa(j_1) = \kappa(j_2)$. Further let the sample correlation coefficient between $x_{\cdot j_1}$ and $x_{\cdot j_2}$ be ρ. Then*

$$|w_{j_1} - w_{j_2}| \leq \frac{1}{\sqrt{2}\gamma} \sqrt{1-\rho} L(0), \tag{19}$$

where $L(0)$ is some constant less than the sample size.



(GEM) (LEM)

Figure 3: The evolution of the weights for GEM and LEM priors with increasing γ penalty. The thick lines show the group centers and the dashed lines the evolution of individual weights. The numbers next to the thick lines denote how many weights are exactly equal to their respective centers (or within 5 standard deviations from the center for GEM), and the numbers above all the lines give the total number of weights not equal to their center.

Theorem 5.2. *Consider the logistic regression model with the LEM prior. Let $x_{\cdot j}$, $j = 1, \ldots, n$, be standardized data columns (with zero means and $\|x_{\cdot j}\|_2 = 1$). Let* $\mathbf{w}$ *be a weight vector maximizing the LEM criterion (Equation 9), and $w_{J(k,j)}$, $j = 1, \ldots, n_k$, be weights assigned to the cluster with center c_k. Then all weights within the k-th group can be made equal to c_k, if one of following conditions is met*

1. $\frac{\gamma}{\sqrt{1-\rho_{ij}}} > \sqrt{2}L(0)$ *for all x_i, x_j in the k-th group, where ρ_{ij} is the sample correlation coefficient between x_i and x_j,*
2. $\gamma > n_k L(0)$.

The proofs can be found in Appendix A.3. For Theorem 5.2 it is clear that condition 2 implies condition 1. However, if γ is not large enough for condition 2 to be met, then condition 1 states that sufficiently large correlations can still shrink the weights to their cluster's center.

5.2. The covering number and sample complexity for a simplified version of LEM

In the previous section we showed that in the LEM model, with correlated variables and suitable values of the regularization parameter γ, one can force most of the weights to be equal to their respective cluster centers. Let us now consider a *simplified* variant of the LEM algorithm, in which the centers $c_1, \ldots, c_s$ are adjustable, but the assignments of weights to groups are fixed. More formally, this means that the indexing functions κ, J are fixed throughout the algorithm.

For the remaining part of this section it is convenient to rewrite the linear part of the logistic model (7) as follows

$$\sum_{j=1}^{n} w_j x_j + b = \sum_{j=1}^{n} (\theta_j + c_{\kappa(j)}) x_j + b = \sum_{j=1}^{n} \theta_j x_j + \sum_{k=1}^{s} c_k \sum_{j=1}^{n_k} x_{J(k,j)} + b, \tag{20}$$

where θ_j is the deviation of the weight w_j from its cluster center. We note, that it is the sum of these deviations that is in fact constrained by regularization. Let $\boldsymbol{\theta}$ denote the vector $(\theta_1, \ldots, \theta_n)$.

Lemma 5.3. *Assume all variables in the training dataset are* normalized*, i.e. $\|x_{\cdot j}\|_\infty = 1$. Let F be the set of linear functions given in* (20)*. An optimal function from this set will be selected using the simplified LEM algorithm with fixed assignments of weights to centers. We use L^1-regularization which is equivalent to the restriction $\|\boldsymbol{\theta}\|_1 \leq a$ for*

some constant a. Assume bounded centers and bias term, $|c_k| \leq K_c, |b| \leq K_b$. Let $l_F = \{l_f(x,d) : f \in F, d \in \{0,1\}\}$ be the set of log-loss functions, where

$$l_f(x,d) = l(f(x),d) = -\Big(d \log \frac{1}{1+\exp(-f(x))} + (1-d)\log\Big(1 - \frac{1}{1+\exp(-f(x))}\Big)\Big). \tag{21}$$

Then, the uniform covering number for l_F in the d_1 metric ($d_1(a,b) = \frac{1}{m}\sum_{i=1}^{m}|a_i - b_i|$, see e.g. (Anthony & Bartlett, 1999)) can be bounded as follows

$$\mathcal{N}_1(\epsilon, l_F, m) \leq (2n+1)^{\lceil 4a^2/\epsilon^2 \rceil} \cdot e(s+2)\left(\frac{4(nK_c + K_b)e}{\epsilon}\right)^{s+1}. \tag{22}$$

The proof can be found in Appendix A.4. For the definition of the covering numbers we refer the reader to e.g. (Anthony & Bartlett, 1999; Zhang, 2002).

An interesting consequence of the lemma is that the covering number for the simplified LEM scales exponentially only with s, i.e. the number of groups, whereas it scales only polynomially with the number of attributes n. This fact is a consequence of the regularization method used. Note also, that typically $s \ll n$.

Covering numbers play an important role in the uniform convergence results and sample complexity bounds. The next result is an adopted version of a theorem given in (Ng, 2004, Theorem 3.1). The original version pertains to the standard L^1-regularization, here we extend it to the simplified LEM model. Note that in the theorem we reformulate the problem in terms of constrained minimization.

Theorem 5.4. *Assume all variables in the training dataset are normalized, i.e. $\|x_{\cdot j}\|_\infty = 1$. Let $(\mathbf{c}^* + \boldsymbol{\theta}^*, b^*)$ represent the optimal vector of weights and the optimal bias term for a regression problem described by an unknown probability distribution P from which the data is drawn. Suppose there exist only r indices $1 \leq j_1, j_2, \ldots, j_r \leq n$ for which components $\theta^*_{j_q}$ $(q = 1, \ldots, r)$ are non-zero. Moreover, for all $q = 1, \ldots, r$, $|\theta^*_{j_q}| \leq K_{\boldsymbol{\theta}^*}$. Consider a simplified LEM algorithm which finds*

$$\min_{\boldsymbol{\theta},\mathbf{c},b}\left(-\sum_{i=1}^{m} d_i \log y(x_i; \boldsymbol{\theta}+\mathbf{c}, b) + (1-d_i)\log(1 - y(x_i; \boldsymbol{\theta}+\mathbf{c}, b))\right) \tag{23}$$

subject to the constraint

$$\sum_{j=1}^{n} |w_j - c_{\kappa(j)}| = \sum_{j=1}^{n} |\theta_j| \leq a \tag{24}$$

and runs within a cross-validation procedure with $a = 0, 1, 2, 4, \ldots, C$ in order to find the best value for the regularization parameter a (every value of γ in the original problem corresponds to some value of a). Let $\epsilon > 0, \delta > 0, C > 0$ be fixed. Assume a bounded bias term $|b| \leq K_b$ and bounded centers $|c_k| \leq K_c, k = 1, \ldots, s$. Then, in order to guarantee that with probability at least $1 - \delta$, the parameters $(\widehat{\boldsymbol{\theta}} + \widehat{\mathbf{c}}, \widehat{b})$ returned by the learning algorithm are nearly as good as $(\boldsymbol{\theta}^* + \mathbf{c}^*, b^*)$ i.e. that:

$$\int_{X\times\{0,1\}} l_{(\widehat{\boldsymbol{\theta}}+\widehat{\mathbf{c}},\widehat{b})}(x,d)dP(x,d) \leq \int_{X\times\{0,1\}} l_{(\boldsymbol{\theta}^*+\mathbf{c}^*,b^*)}(x,d)dP(x,d) + \epsilon, \tag{25}$$

it suffices that $C > rK_{\boldsymbol{\theta}^*}$ and that

$$m(\epsilon,\delta) = \Omega\Big(n^2 \cdot (s\log(n) + s\log(s)) \cdot poly(r, K_{\boldsymbol{\theta}^*}, \log(1/\delta), 1/\epsilon)\Big). \tag{26}$$

By $l_{(\boldsymbol{\theta}+\mathbf{c},b)}$ we mean the loss function corresponding to the logistic function with parameters $(\boldsymbol{\theta}+\mathbf{c}, b)$. To prove the theorem it is sufficient to apply our bound on the covering number for the simplified LEM (given in Lemma 5.3) and to substitute it into the proof of Theorem 3.1 in (Ng, 2004).

Looking qualitatively at the theorem, and comparing it to the result in (Ng, 2004), we note that the sample complexity is unfavorable due to the n^2 factor. We would prefer the sample complexity to scale only logarithmically with n. In the following section we present an argument that, under certain assumptions about the data and the behavior of the learning algorithm, the unfavorable factor disappears.

5.3. Attributes averaging and reduction of sample complexity in the simplified model

Recall the *simplified model* and the motivation Lemma 2.1 from Section 2. In it we stated that the averages of observed variables $1/n_k \sum_{j=1}^{n_k} X_{J(k,j)}$ are good estimates of the hidden variables H_k, and that their coefficients are good estimates of A_k in the linear model $Y = A_1 H_1 + \cdots + A_s H_s$. Suppose we correctly guessed the number s, and that the learning algorithm discovered correct variables groupings in line with the shrinkage theorems.

The linear part of the logistic model (7) can be rewritten as

$$\begin{aligned}\sum_{j=1}^{n} w_j x_j + b &= \sum_{j=1}^{n} (\theta_j + c_{\kappa(j)}) x_j + b \\ &= \sum_{j=1}^{n} \theta_j x_j + \sum_{k=1}^{s} c_k n_k \frac{1}{n_k} \sum_{j=1}^{n_k} x_{J(k,j)} + b \\ &= \sum_{j=1}^{n} \theta_j x_j + \sum_{k=1}^{s} c_k n_k \bar{x}_{J(k,\cdot)} + b, \end{aligned} \tag{27}$$

where $\bar{x}_{J(k,\cdot)}$ denotes the average taken over variables in the k-th group. For agreement with the motivation lemma let us ignore bias term (without loss of generality), and assume a sufficiently small a in the regularization constraint $\sum_{j=1}^{n} |\theta_j| \leq a$, such that the first term in (27), $\sum_{j=1}^{n} \theta_j x_j$, becomes negligible compared to the other terms. Now, one can see that the remainder

$$\sum_{k=1}^{s} c_k n_k \bar{x}_{J(k,\cdot)}$$

corresponds to the linear combination $\sum_{k=1}^{s} A_k 1/n_k \sum_{i=1}^{n_k} X_{J(k,j)}$ of averages in the motivation lemma. Therefore the numbers $c_k n_k$ are estimates of A_k. Hence, it is easy to see that if one increases the number of observed variables and thus the cardinalities n_k, then absolute values of the centers $|c_k|$ must decrease, so that the product $c_k n_k \approx A_k$ remains approximately constant. Moreover by increasing n_k we obtain more accurate estimates of H_k and A_k. This observation is useful for reducing the order of sample complexity in Theorem 5.4, which includes an unfavorable n^2 factor. Looking at the proof of the theorem (Appendix A.4), one notes that the unfavorable n^2 term arises (via Hoeffding inequality) due to the range $[-nK_c - K_b, nK_c + K_b]$ to which the set of functions

$$\left\{ h(x; c, b) = \sum_{k=1}^{s} c_k \sum_{j=1}^{n_k} x_{J(k,j)} + b \colon |c_k| \leq K_c, |b| \leq K_b \right\}$$

maps. This range is pessimistic, and since, with normalized (in the L^∞ norm) data, we have

$$\sum_{k=1}^{s} c_k \sum_{j=1}^{n_k} x_{J(k,j)} = \sum_{k=1}^{s} c_k n_k \bar{x}_{J(k,\cdot)} \leq \sum_{k=1}^{s} |c_k| n_k = \left\| (c_1 n_1, \ldots, c_s n_s) \right\|_1, \tag{28}$$

then it is easy to see that the range can be narrowed to $[-\|A\|_1(1+\epsilon), \|A\|_1(1+\epsilon)]$, where $A = (A_1, \ldots, A_s)$, and ϵ denotes the error of approximation of the linear coefficients A_k by cluster centers; ϵ decreases at the rate $\sim 1/\sqrt{n}$ by the Central Limit Theorem. Therefore, the range *does not* depend on n and the sample complexity for the simplified LEM (under the stated assumptions) can be reduced to

$$m(\epsilon, \delta) = \Omega\Big((s \log(n) + s \log(s)) \cdot \mathrm{poly}(r, K_{\theta^*}, \log(1/\delta), 1/\epsilon) \Big). \tag{29}$$

6. Experiments

In the experiments we use a general model evaluation scheme shown in Figure 4. The core of model evaluation is the *resampling procedure* with factor α. It works in the following way:

1. Split the data D into a training set L, which contains randomly chosen $\lfloor \alpha \cdot |D| \rfloor$ examples and a test set T containing the remaining examples, $T = D \setminus L$.

2. Fit the model on the training set L, evaluate it on the test set T.

The models we consider require choosing a proper regularization coefficient γ. Both the evaluation and the choice of the hyperparameter have been carried out in the same way using resampling, but the choice of the hyperparameter is done in a separate loop, see Figure 4. All models have been fitted and evaluated on the same data subsets. The number of centers has been chosen arbitrarily as 3, but automatic selection could easily be included in the cross-validation framework.

1: **for all** $\alpha \in \{0.01, 0.02, \ldots 0.5\}$ **do**
2: **for all** model $\in \{\text{LEM}, \text{GEM}, L^1, L^2, \ldots\}$ **do**
3: choose regularization parameter using a resampling procedure with factor α.
4: **end for**
5: **for** $i = 1, \ldots, 50$ **do**
6: choose random data-subsets L and T using resampling with factor α
7: **for all** $\{\text{LEM}, \text{GEM}, L^1, L^2, \ldots\}$ **do**
8: fit each model on L and evaluate it on T
9: **end for**
10: **end for**
11: **end for**

Figure 4: Experimental model evaluation scheme with resampling.

6.1. Datasets used in the experiments

The experiments were performed on one artificial and three real, publicly available datasets. The artificial dataset (with 150 attributes) was generated using the following MATLAB code:

```
m = 2000;      % size of the dataset
n = 50;        % number of attributes in each group
s = 3;         % number of hidden variables
H = randn(2000, s);         % hidden variables
Y = H * [-5; -1; 3]' > 0;   % output
for k = 1 : s
    X = [X [H(:,k) * ones(1, n) + randn(m, n))]]; % observed data
end
```

The dataset corresponds to the idealized situation described in Section 2 with $s = 3$ hidden variables, each assigned $n = 50$ noisy observation variables. The variance of the noise is relatively high, close to the level of the signal. Two thousand data records were generated.

The three publicly available datasets are: `Waveform-5000`, `KDD2004CUP-physics` and `Reuters-21578`. The waveform dataset is an artificial benchmark introduced in (Breiman et al., 1984) consisting of 5000 instances and 40 attributes. The `KDD2004CUP-physics` is a dataset related to accelerator physics with 50000 records and 80 attributes (see: `http://osmot.cs.cornell.edu/kddcup/datasets.html`). The Reuters dataset is a well known text analysis benchmark, details are given below.

6.2. Linear Regression

Although the main focus of the paper is logistic regression, we begin by showing some experiments with linear regression on the artificial dataset.

We compare our two models with LEM and GEM priors to: Lasso (L^1), ridge (L^2) and unregularized (L) linear regression models (in fact we use the `pinv` command in MATLAB which involves a form of weak, tolerance based, regularization). Additionally, we compare with a reference model (Ref.) which is a linear model trained on estimates of the hidden variables obtained by averaging over known (in our controlled experiment) groups of attributes being observations of a single hidden variable. Recall that, by Lemma 2.1, this is the best possible model.

The results are presented in the Figure 5. One can see that, in order to achieve good accuracy, GEM and LEM require fewer samples than other approaches. About a 100 examples ($\alpha = 0.05$) for the GEM model (about 200 for

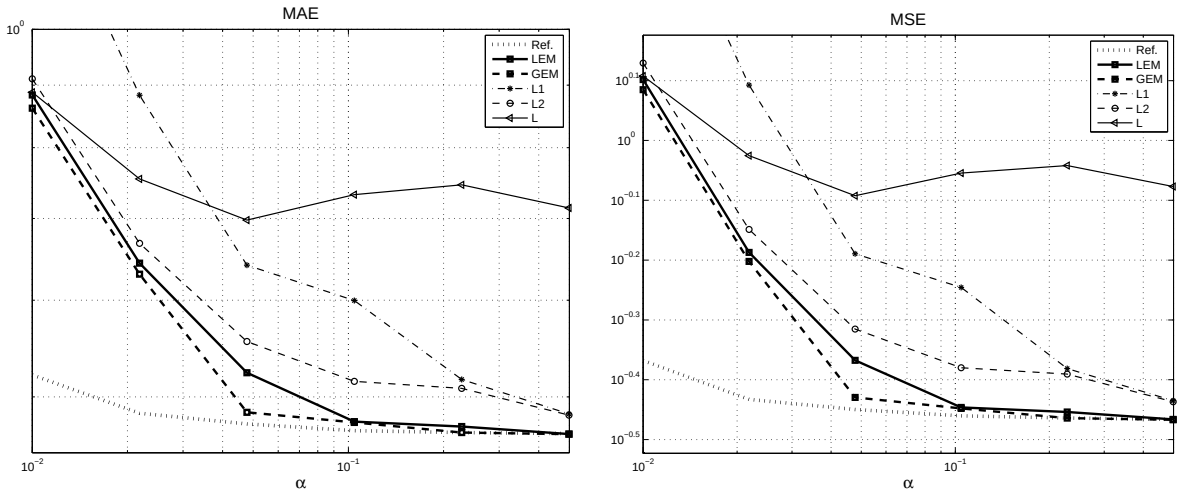


Figure 5: Linear regression. Comparison of LEM, GEM and other algorithms on an artificial data set. Mean Absolute Error (MAE) and Mean Squared Error (MSE) are shown as functions of the sample size. The results were averaged over 50 iterations.

LEM) are sufficient to recognize the hidden groups and reach an error level comparable with the theoretical reference model (dotted curve in the figure). Obviously, if the sample size grows arbitrarily large than all models converge to the reference model, which is partially visible in the plots for all models except the unregularized one.

Note, that the L^1 regularization (Lasso) which induces a variable selection effect (Ng, 2004) performs especially poorly on this dataset. Other models, such as those including an L^2 regularization term perform much better due to the weight averaging effect such terms induce (Zou & Hastie, 2005). The example clearly demonstrates that variable selection is not always the best possible course of action.

6.3. Classification

Analogous results for the classification task are presented in the Figure 6. The following linear classifiers were included in the comparison: LR blogreg, penalized logistic regression (Cawley & Talbot, 2006) using software available from `http://theoval.cmp.uea.ac.uk/~gcc/cbl/blogreg/`, elasticnet implementation using the `glmnet` R package (Friedman et al., 2009), LR netlab classical logistic regression implemented in the `netlab` package: `http://www.ncrg.aston.ac.uk/netlab/index.php`, msereg Matlab's neural network toolbox using a 1-layer network with `trainscg` learning method and `msereg` criterion, the libSVM library (Chang & Lin, 2001) with linear kernel, LibLinear with the default SVM model in the library (Fan et al., 2008) and the C constant determined using cross-validation.

It can be seen that in terms of accuracy as well as the Area Under the ROC Curve (AUC) both our models perform very well and either outperform the other approaches, or stay very close to the top performers.

6.4. Detailed analysis of the Reuters-21578 dataset

In order to assess the interpretability of the groupings produced by our methods we applied them to the `Reuters-21578` dataset, the attributes of which can be easily interpreted. The dataset (Lewis et al., 2004) is a collection of newswire documents from the year 1987 (21578 instances) on which different kinds of text categorizations can be performed. We use this dataset to predict whether a document concerns the USA or not (the class attribute is `places='USA'`). Each attribute corresponds to a single word present in or absent from the document. As one can expect, there are many clear correlations between attributes (e.g. the words 'states' and 'united' frequently occur together).

We retained only the 1000 most frequent word-attributes. The whole list of attributes consists of about 38000 unique words, most of them occurring in very few documents. The number 1000 was chosen arbitrarily, the least frequent word ('warrants') appeard in 237 documents (that is in about 1.1% of all documents). The most frequent words were: 'the' (15553 documents) and 'of' (15438). Note that we did not perform stop-words removal, stemming,

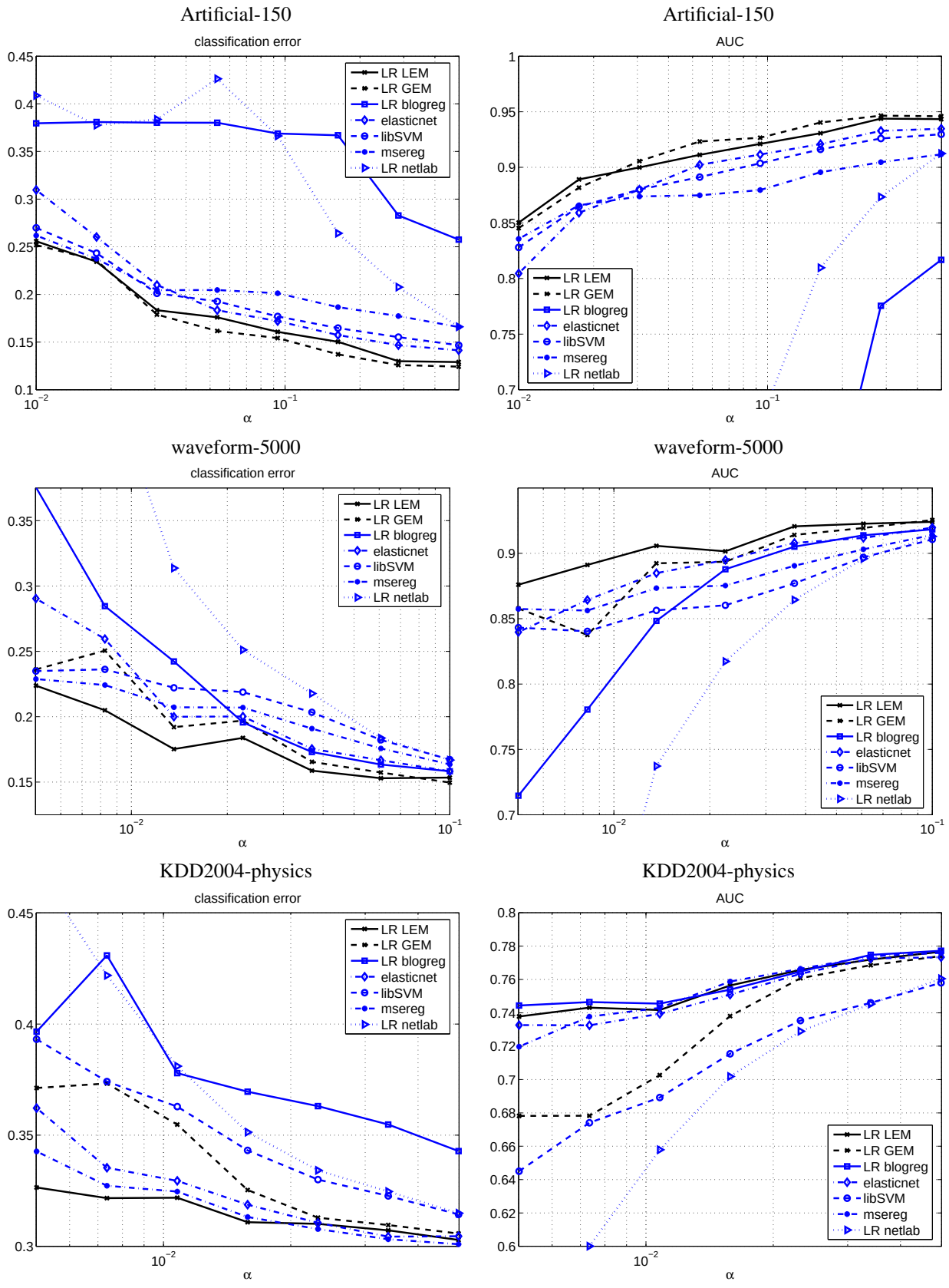


Figure 6: Logistic regression. Comparison of LEM, GEM and other algorithms on different datasets. The results were averaged over 50 iterations.

or any other kind of preprocessing. The reason for limiting the number of attributes is the interpretability of the results, the algorithms scale linearly in the number of attributes and can easily be applied the whole dataset with 38000 attributes.

Results of an analysis similar to those in the previous section are presented in Figure 7 with the fraction of training data α varying between 10^{-3} and 10^{-1} (corresponding to data sets between 22 and 2159 instances). As one can see, the GEM weight grouping algorithm gave the best results for accuracy for a wide range of fractions α. Regarding the other algorithms, for small training samples good results have been achieved with L^1-regularized methods, and for larger learning samples with models using the L^2 penalties (SVMs also use the L^2 penalty). This behavior is coherent with theoretical properties of those models. For the AUC criterion the situation was similar, with the GEM model being an overall winner.

In Figure 8 we show histograms of weight coefficiensts of all 1000 attributes for $\alpha = 0.1$ (2158 learning instances) obtained using various linear model learning methods. One can immediately see that the proposed GEM and LEM approaches produced clear groups of attributes, with LEM forcing many weights to be exactly equal. Other approaches did not induce such a grouping effect.

Because each attribute has a natural meaning, we can analyze the groups of attributes obtained by the GEM and LEM algorithms. Looking at their respective histograms (Figure 8) we see three clear groups of coefficient values: positive, aroud zero, and negative. We can expect that attributes in the positive group are positively correlated with class '1' (i.e. the message is related to the USA), attributes in the negative group are positively correlated with class '0' (not related to the USA), and the third group should contain words which are neutral with respect to the class.

All groups of attributes contain words whose meaning seems either neutral or is not clearly related to the class, a typical example being stop-words such as 'the', 'a', 'at', 'about', etc. We ignore such words in the listings below. On the other hand, each group contains a large number of words whose membership in the group rises no doubts. Below we present a sample of word-attributes for each group selected using the GEM model (as on this dataset it achieved better accuracy than LEM).

Some of the word-attributes that belong to the negative coefficient group (should be correlated with the 'non-USA' class) are:

```
agency, analysts, australia, australian, bank, bankers, banks, billion, bond, brazil, britain, british,
canada, canadian, capital, china, crude, currency, economic, energy, england, europe, european, export,
exports, finance, firms, foreign, france, francs, french, german, gold, government, holdings, import,
industrial, industry, international, intervention, investment, london, ltd, mark, market, marks,
minister, ministry, oil, option, paris, pct, petroleum, plc, political, reporters, resources, rules,
sell, selling, shareholders, shares, shr, south, spokesman, sugar, swiss, target, tax, tender, total,
west, world, worth
```

There is a total of 246 words in this group. It can be seen that most of the words (like names of capital cities, countries or currencies, words denoting imported goods or international market terms) are clearly related to countries other than the USA and have been placed in the correct group. Note, for example, the abbreviation 'ltd' used in the names of non-American companies.

Some word-attributes in the positive group, which should be positively correlated with the 'USA' class are:

```
administration, agriculture, american, association, avg, business, chairman, co, commission, committee,
companies, company, computer, congress, contract, corp, court, cts, days, debentures, debt, department,
dlrs, dollar, farm, federal, gain, growth, house, inc, income, increase, insurance, its, loan, mths,
officer, plant, power, president, product, production, program, qtly, quarter, reagan, securities,
senate, service, shrs, soviet, states, stock, stocks, system, texas, time, trade, trading, trust, u,
unit, united, units, vs, washington, york
```

This group includes 110 words. It is hard to argue against most of those words being USA-related. The words include president's name, names of several states and cities, words related to US administration, stock market, currency etc. The words 'corp' and 'co'—parts of the names of many American companies are also correctly placed in this group. Interestingly, the word 'house' does appear in this group, but 'white', being very general, ended up in the 'neutral' group.

The neutral group contains 644 words such as 'institute', 'january', 'life', most of which cannot easily be assigned to any of the two classes. Overall, the groups are easily interpretable and related to the true meanings of words.

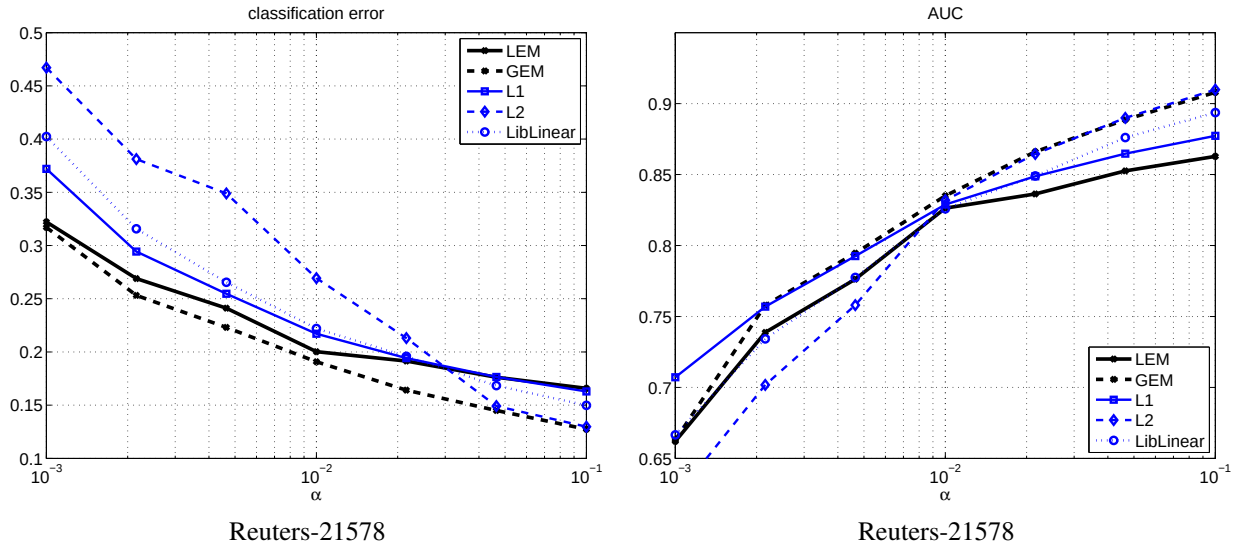


Figure 7: Comparison of LEM, GEM and other algorithms (L^1- and L^2-regularized logistic regression and linear SVM) on the `Reuters-21578` database for varying sample sizes. The results were averaged over 50 iterations.

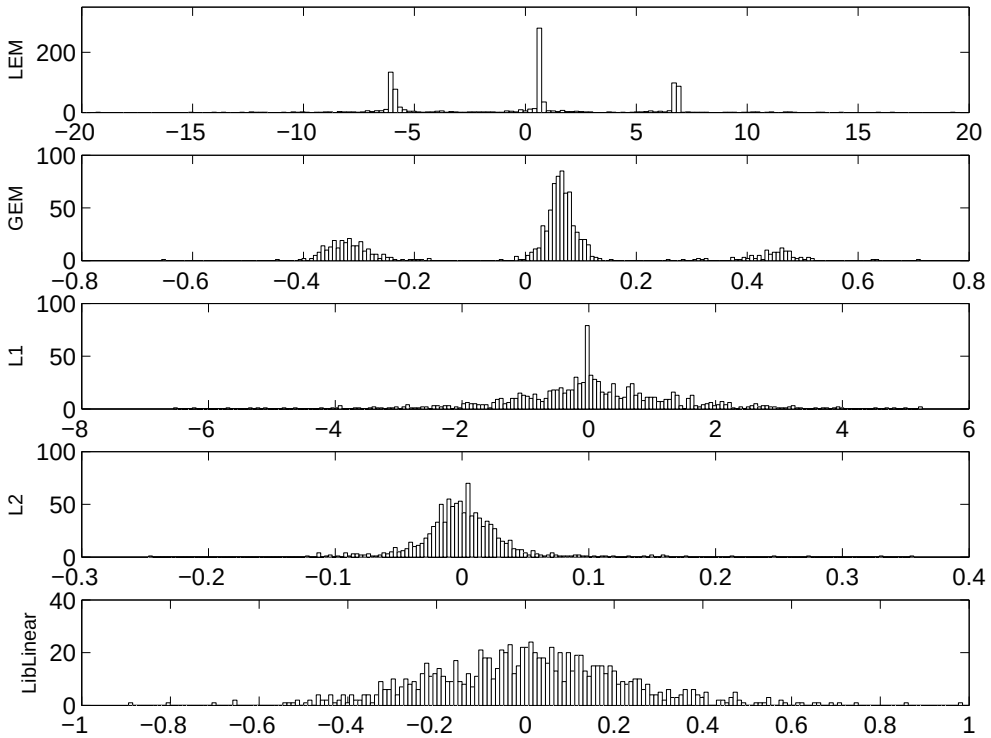


Figure 8: Comparison of the distributions of weights produced by LEM, GEM, regularized logistic regressions (L^1, L^2) and linear SVM, on the `Reuters-21578` database.

7. Conclusions and future research

In the paper we study learning algorithms for linear/logistic regression which perform automatic grouping of attributes. The main conclusion of the paper is that for datasets with correlated attributes (even with a large number of such attributes) attribute selection is often a less useful approach than attribute grouping (and averaging), which often leads to better learning results. Note that the attribute selection problem is a subset of the attribute grouping problem, since one of the group centers may be very close to zero (see e.g. Figure 3).

We present two a-priori mixture distributions (GEM, LEM) over weights that induce the desired grouping of correlated attributes. This fact is demonstrated experimentally and by means of shrinkage theorems. An effective learning algorithm which finds the weights, the group centers, and the assignment of weights to centers is presented. Experiments have verified high accuracy of the proposed approach usually exceeding that of other linear modeling methods.

We note that for data with independent attributes our approach might give worse results than other methods. However, one should realize that truly independent attributes occur very rarely in practice, typically only in planned experiments. In real settings, correlations practically always exist, especially in large data sets. On the other hand, if we choose a suitable regularization parameter via cross-validation, our approach (LEM or GEM) will be able to choose models (through small values of the regularization parameter) which do not induce unnecessary grouping.

We have also demonstrated low sample complexity for a simplified variant of the LEM prior, which uses a fixed assignment of attributes to groups. The study of sample complexity without this simplification requires a more careful analysis and will be a subject of future research.

Acknowledgements

This work has been partly financed by the Polish Ministry of Science and Higher Education from research funds for the years 2010–2012. Research project no.: N N516 424938.

Appendix A. Proofs

Appendix A.1. Proof of Lemma 2.1

Proof. Denote $\mathrm{Cov}(Y, H_k) = \rho_k$, and let $\sigma^2_{H_k}$ and σ^2_Y denote the variances respectively of the hidden variables and of the decision variable.

Consider the mean squared error $E\left(\sum_{k=1}^{s} A_k \sum_{j=1}^{n_k} \alpha_{J(k,j)} X_{J(k,j)} - Y\right)^2$ as a function of $\alpha_{J(k,j)}$. Note that

$$Z = E\left(\sum_{k=1}^{s} A_k \sum_{j=1}^{n_k} \alpha_{J(k,j)} X_{J(k,j)} - Y\right) = \sum_{k=1}^{s} A_k \sum_{j=1}^{n_k} \alpha_{J(k,j)} EX_{J(k,j)} - EY$$
$$= \sum_{k=1}^{s} A_k \sum_{j=1}^{n_k} \alpha_{J(k,j)} EH_k - EY = \sum_{k=1}^{s} A_k EH_k - EY$$

does not depend on $\alpha_{J(k,j)}$. Also,

$$E\left(\sum_{k=1}^{s} A_k \sum_{j=1}^{n_k} \alpha_{J(k,j)} X_{J(k,j)} - Y\right)^2 = \mathrm{Var}\left(\sum_{k=1}^{s} A_k \sum_{j=1}^{n_k} \alpha_{J(k,j)} X_{J(k,j)} - Y\right) + Z^2 \tag{A.1}$$

so to minimize the MSE it suffices to minimize the variance term in the equation above.

We have:

$$
\begin{aligned}
&\mathrm{Var}\left(\sum_{k=1}^{s} A_k \sum_{j=1}^{n_k} \alpha_{J(k,j)} X_{J(k,j)} - Y\right) \\
&\quad = \sum_{1\le k,l\le s} A_k A_l \sum_{1\le j,i\le n_k} \alpha_{J(k,j)}\alpha_{J(l,i)} \operatorname{Cov}\left(H_k + \xi_{J(k,j)}, H_l + \xi_{J(l,i)}\right) \\
&\qquad - 2\sum_{k=1}^{s} A_k \sum_{j=1}^{n_k} \alpha_{J(k,j)} \operatorname{Cov}\left(H_k + \xi_{J(k,j)}, Y\right) + \mathrm{Var}(Y) \\
&\quad = \sum_{k=1}^{s} A_k^2 \left(\mathrm{Var}(H_k) + \sigma^2_{\kappa(j)}\right) \sum_{j=1}^{n_k} \alpha^2_{J(k,j)} \\
&\qquad - 2\sum_{k=1}^{s} A_k \operatorname{Cov}(H_k, Y) \sum_{j=1}^{n_k} \alpha_{J(k,j)} + \mathrm{Var}(Y) \\
&\quad \ge \sum_{k=1}^{s} A_k^2 \left(\sigma^2_{H_k} + \sigma^2_{\kappa(j)}\right) \sum_{j=1}^{n_k} \left(\frac{1}{n_k}\right)^2 - 2\sum_{k=1}^{s} A_k \rho_k + \sigma_Y^2 \\
&\quad = \mathrm{Var}\left(\sum_{k=1}^{s} A_k \frac{1}{n_k} \sum_{j=1}^{n_k} X_{J(k,j)} - Y\right).
\end{aligned}
$$

The second equality follows from the assumed independencies, the inequality follows from Jensen's inequality for the convex function x^2: $\frac{1}{n_k}\sum_{j=1}^{n_k} \alpha^2_{J(k,j)} \ge \left(\frac{1}{n_k}\sum_{j=1}^{n_k} \alpha_{J(k,j)}\right)^2$, which combined with the fact $\sum_{j=1}^{n_k} \alpha_{J(k,j)} = 1$ yields $\sum_{j=1}^{n_k} \alpha^2_{J(k,j)} \ge \frac{1}{n_k} = \sum_{j=1}^{n_k} \left(\frac{1}{n_k}\right)^2$. Therefore, setting $\alpha_{J(k,j)} = 1/n_k$ minimizes the mean squared error. □

Remark on the general case of unequal noise variances. In Lemma 2.1 we assumed equal noise variances $\sigma^2_{\kappa(j)}$ for observed variables within each group. It is possible to consider a more general case with arbitrary noise variances $\sigma^2_{\xi_j}$, $j = 1, \ldots, n$. The minimization of the mean squared error then requires minimizing the following expression

$$
\sum_{k=1}^{s} A_k^2 \sigma^2_{H_k} \sum_{j=1}^{n_k} \alpha^2_{J(k,j)} \sigma^2_{\xi_j} - 2\sum_{k=1}^{s} A_k \rho_k + \sigma_Y^2. \tag{A.2}
$$

That the solution in this case is

$$
\alpha_{J(k,j)} = \frac{1/\sigma^2_{\xi_{J(k,j)}}}{\sum_{i=1}^{n_k} 1/\sigma^2_{\xi_{J(k,i)}}}. \tag{A.3}
$$

To see this, use constrained optimization with the Lagrangian $Q(\alpha, \lambda) = \sum_j \alpha_j^2 \sigma_j^2 + \lambda(\sum_j \alpha_j - 1)$, where λ is a Lagrange multiplier. The equation $\partial Q/\partial \alpha_j = 0$ yields $\alpha_j = -\lambda/(2\sigma_j^2)$. Summing over j and using $\sum_j \alpha_j = 1$ gives $\lambda = -2/\sum_j (1/\sigma_j^2)$, the result follows.

This result is intuitive — averaging within groups can be improved by assigning greater weights to variables with smaller variance. Unfortunately, the practical difficulty in applying the result lies in the fact that noise variances are unknown. Therefore, in the paper we do not focus on the best possible choice of numbers $\alpha_{J(k,j)}$, but rather on the more basic problem of discovering the correct grouping of variables. If this can be achieved, we apply simple averaging based on Lemma 2.1.

Appendix A.2. The gradient of Q_W

We begin by rewriting the expression for Q_W

$$\begin{aligned}\log(p_{\text{GEM}}) &= \sum_{j=1}^{n} \log\left(\max_{k=1,\ldots,s} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(w_j - c_k)^2}{2\sigma^2}\right)\right) \\ &= -\sum_{i=1}^{n} \min_{k=1,\ldots,s} \frac{(w_j - c_k)^2}{2\sigma^2} + C \\ &= -\gamma \sum_{j=1}^{n} \min_{k=1,\ldots,s} \left(w_j - c_k\right)^2 + C \\ &= -\gamma \sum_{j=1}^{n} \left(w_j - c_{\kappa(j)}\right)^2 + C. \end{aligned} \tag{A.4}$$

$$\begin{aligned}\log(p_{\text{LEM}}) &= \sum_{j=1}^{n} \log\left(\max_{k=1,\ldots,s} \frac{1}{\lambda} \exp\left(-\frac{|w_j - c_k|}{\lambda}\right)\right) \\ &= -\sum_{j=1}^{n} \min_{k=1,\ldots,s} \frac{|w_j - c_k|}{\lambda} + C \\ &= -\gamma \sum_{j=1}^{n} \min_{k=1,\ldots,s} |w_j - c_k| + C = -\gamma \sum_{j=1}^{n} |w_j - c_{\kappa(j)}| + C. \end{aligned} \tag{A.5}$$

Now

$$\frac{\partial \log(p_{\text{GEM}})}{\partial w_j} = -2\gamma(w_j - c_{\kappa(j)}), \tag{A.6}$$

$$\frac{\partial \log(p_{\text{LEM}})}{\partial w_j} = -\gamma \,\text{sgn}(w_j - c_{\kappa(j)}). \tag{A.7}$$

Appendix A.3. Proofs of shrinkage theorems

Proof of theorem 5.1. Let $y(\mathbf{w}, b)$ denote the vector of values predicted for each data point as a function of $\mathbf{w}$ and b, and let $\mathbf{d}$ be the decision column in the training data. Consider the conditions for the optimality of Q with respect to w_{j_1} and w_{j_2}

$$\frac{\partial Q_{GEM}}{\partial w_{j_1}} = {x_{\cdot j_1}}^T(\mathbf{d} - y(\mathbf{w}, b)) - 2\gamma(w_{j_1} - c_{\kappa(j_1)}) = 0, \tag{A.8}$$

$$\frac{\partial Q_{GEM}}{\partial w_{j_2}} = {x_{\cdot j_2}}^T(\mathbf{d} - y(\mathbf{w}, b)) - 2\gamma(w_{j_2} - c_{\kappa(j_2)}) = 0. \tag{A.9}$$

Subtracting equations (A.8) and (A.9) gives:

$$\begin{aligned}|w_{j_1} - w_{j_2}| &= \frac{1}{2\gamma}|(x_{\cdot j_1} - x_{\cdot j_2})^T(\mathbf{d} - y(\mathbf{w}, b))| \\ &\leq \frac{1}{2\gamma}\|x_{\cdot j_1} - x_{\cdot j_2}\| \cdot \|(\mathbf{d} - y(\mathbf{w}, b))\| \\ &\leq \frac{1}{2\gamma}\sqrt{2(1-\rho)}L(0), \end{aligned} \tag{A.10}$$

where $L(0) = \|(\mathbf{d} - y(\mathbf{0}, 0))\| \geq \|(\mathbf{d} - y(\mathbf{w}, b))\|$ is a constant independent of $\mathbf{w}$ and b. □

Proof of theorem 5.2. Let $y(\mathbf{w}, b)$ denote the vector of values predicted for each data point as a function of $\mathbf{w}$ and b. First notice that for two different weights w_{j_1}, w_{j_2} such that $\operatorname{sgn}(w_{j_1} - c_k)\operatorname{sgn}(w_{j_2} - c_k) < 0$, at an optimum of Q we have

$$\frac{\partial Q_{\mathrm{LEM}}}{\partial w_{j_1}} = {x_{\cdot j_1}}^T(\mathbf{d} - y(\mathbf{w}, b)) + \gamma = 0, \tag{A.11}$$

$$\frac{\partial Q_{\mathrm{LEM}}}{\partial w_{j_2}} = {x_{\cdot j_2}}^T(\mathbf{d} - y(\mathbf{w}, b)) - \gamma = 0. \tag{A.12}$$

Subtraction of equations (A.11) and (A.12) yields:

$$1 < 2 = \frac{1}{\gamma}|(x_{\cdot j_1} - x_{\cdot j_2})^T(\mathbf{d} - y(\mathbf{w}, b))| \leq \frac{\sqrt{2}}{\gamma}\sqrt{1-\rho}L(0). \tag{A.13}$$

What gives a contradiction if $\frac{\gamma}{\sqrt{1-\rho}} > \sqrt{2}L(0)$ for some values ρ and γ.

Next, consider the group of weights belonging to c_k. At an optimum of Q we have:

$$\frac{\partial Q_{\mathrm{GEM}}}{\partial w_{J(k,j)}} = {x_{\cdot J(k,j)}}^T(\mathbf{d} - y(\mathbf{w}, b)) - \gamma \operatorname{sgn}(w_{J(k,j)} - c_k) = 0, \text{for all } j = 1, \ldots, n_k \tag{A.14}$$

Summing over j we get

$$\begin{aligned}\left|\sum_{j=1}^{n_k} \operatorname{sgn}(w_{J(k,j)} - c_k)\right| &= \frac{1}{\gamma}\left|\sum_{j=1}^{n_k} x_{\cdot J(k,j)}^T(\mathbf{d} - y(\mathbf{w}, b))\right| \\ &\leq \frac{1}{\gamma}\|\mathbf{d} - y(\mathbf{w}, b)\| \sum_{j=1}^{n_k} \|x_{\cdot J(k,j)}\| = \frac{1}{\gamma} n_k L(0).\end{aligned} \tag{A.15}$$

Recall that c_k is the median weights belonging to a given cluster, i.e.

$$0 \leq \left|\sum_{j=1}^{n_k} \operatorname{sgn}(w_{J(k,j)} - c_k)\right| \leq \lceil n_k/2 \rceil - 1$$

and consider two cases:

1. $\left|\sum_{j=1}^{n_k} \operatorname{sgn}(w_{J(k,j)} - c_k)\right| = 0,$
2. $\left|\sum_{j=1}^{n_k} \operatorname{sgn}(w_{J(k,j)} - c_k)\right| = N \geq 1, \quad N \leq \lceil n_k/2 \rceil - 1.$

In case 1 the same number of weights lie on both sides of the center c_k. Taking (A.13) into account, for sufficiently large values of ρ and γ we conclude that $w_{J(k,j)} = c_k$ for all $j = 1, \ldots, n_k$. In case 2, using (A.15) we have

$$\frac{N}{n_k} \leq \frac{1}{\gamma}L(0), \tag{A.16}$$

which, if γ is sufficiently large, leads to a contradiction. Combining (A.13) and (A.16) we obtain the desired result. □

Appendix A.4. Proofs for the simplified LEM results

Proof of lemma 5.3. Assume $\|x\|_\infty \leq 1$ and consider the following sets of functions:

$$G = \left\{g(x; \boldsymbol{\theta}) = \sum_{j=1}^{n} \theta_j x_j \colon \|\boldsymbol{\theta}\|_1 \leq a\right\}, \tag{A.17}$$

$$H = \left\{h(x; \mathbf{c}, b) = \sum_{k=1}^{s} c_k \sum_{j \in S(k)} x_j + b \colon |c_k| \leq K_{\mathbf{c}}, |b| \leq K_b\right\}, \tag{A.18}$$

$$F = \left\{f = g + h \colon g \in G, h \in H\right\}. \tag{A.19}$$

$$l_F = \left\{l_f(x, d) \colon f \in F, d \in \{0, 1\}\right\}, \tag{A.20}$$

where $S(k)$ is the set of indices of variables in the k-th cluster and l_f represents the log-loss function

$$l_f(x,d) = l(f(x),d) = -\Big(d\log\frac{1}{1+\exp(-f(x))} + (1-d)\log\Big(1-\frac{1}{1+\exp(-f(x))}\Big)\Big). \tag{A.21}$$

For the set G, following (Zhang, 2002, Theorem 3) (via Maurey's lemma) we have that

$$\mathcal{N}_1(\epsilon, G, m) \le (2n+1)^{\lceil a^2/\epsilon^2\rceil}, \tag{A.22}$$

where $\|\boldsymbol{\theta}\|_1 \le a$ and $\|x\|_\infty \le 1$. On the other hand for the set H, noticing that the pseudodimension $\mathrm{Pdim}(H) = s+1$, due to (Anthony & Bartlett, 1999, Theorem 18.4) we have that:

$$\mathcal{N}_1(\epsilon, H, m) \le \mathcal{M}_1(\epsilon, H, m) \le e(s+2)\left(\frac{2(nK_{\mathbf{c}}+K_b)e}{\epsilon}\right)^{s+1}, \tag{A.23}$$

using the fact that $h\colon X \to [-nK_{\mathbf{c}}-K_b, nK_{\mathbf{c}}+K_b]$. $\mathcal{M}_1$ denotes the uniform packing number, where $X = [-1,1]^n$.

Now, fix the sample $x = \{x_1,\dots,x_m\}$ and consider two functions $f_1, f_2 \in F$ (with parameters $\boldsymbol{\theta}_1, \mathbf{c}_1, b_1$ for f_1 and $\boldsymbol{\theta}_2, \mathbf{c}_2, b_2$ for f_2). The d_1 distance between points $(f_1)_{|x} = (f_1(x_1),\dots,f_1(x_m))$ and $(f_2)_{|x}$ is

$$d_1\left((f_1)_{|x},(f_2)_{|x}\right) = \frac{1}{m}\sum_{i=1}^m |\boldsymbol{\theta}_1^T x_i + \mathbf{c}_1^T x_i + b_1 - (\boldsymbol{\theta}_2^T x_i + \mathbf{c}_2^T x_i + b_2)| \le \frac{1}{m}\sum_{i=1}^m |\boldsymbol{\theta}_1^T x_i - \boldsymbol{\theta}_2^T x_i| + \frac{1}{m}\sum_{i=1}^m |\mathbf{c}_1^T x_i + b_1 - (\mathbf{c}_2^T x_i + b_2)|. \tag{A.24}$$

It follows that given any $(\epsilon/2)$-cover $\widehat{G}$ for the set $G_{|x}$ and any $(\epsilon/2)$-cover $\widehat{H}$ for the set $H_{|x}$ there exists an ϵ-cover for $F_{|x}$ generated by all pairs $\left(\widehat{g},\widehat{h}\right), \widehat{g}\in\widehat{G}, \widehat{h}\in\widehat{H}$, with cardinality not greater than $\#\widehat{G}\cdot\#\widehat{H}$. In other words for any point $f_{|x}$ in $F_{|x}$ it is possible to find a pair $\left(\widehat{g},\widehat{h}\right)$ such that $d_1\left((\widehat{g}+\widehat{h})_{|x}, f_{|x}\right) < \epsilon$. Hence $\forall x \in X^m$:

$$\begin{aligned}\mathcal{N}(\epsilon, F_{|x}, d_1) &\le \mathcal{N}_1(\epsilon/2, G, m)\cdot\mathcal{N}_1(\epsilon/2, H, m)\\ &= (2n+1)^{\lceil 4a^2/\epsilon^2\rceil}\cdot e(s+2)\left(\frac{4(nK_{\mathbf{c}}+K_b)e}{\epsilon}\right)^{s+1}.\end{aligned} \tag{A.25}$$

To conclude the proof it suffices to note that the functions l_f satisfy the Lipshitz condition with constant $L=1$, since $\left|\frac{d}{dt}l(t,d)\right| \le 1$, therefore

$$\max_{z\in(X\times\{0,1\})^m} \mathcal{N}(\epsilon, (l_F)_{|z}, d_1) \le \mathcal{N}_1(\epsilon, F, m). \tag{A.26}$$

□

It is worth noting, that if assignments of weights to centers of groups were not assumed fixed, then the bound on the covering number would have to be multiplied by an additional term. This term would be determined by the richness of the space of possible assignments; e.g. for the case of two groups $s=2$, the term would be of order $O(2^n)$. Transforming this remark into a sample complexity result (see (29), where $\log\mathcal{N}_1(\cdot)$ is computed) gives a sample complexity of $\Theta(n\log n)$.

References

Anthony, M., & Bartlett, P. L. (1999). *Neural Network Learning: Theoretical Foundations*. Cambridge University Press.

Bakin, S. (1999). *Adaptive Regression and Model Selection in Data Mining Problems*. Ph.D. thesis Australian National University.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth.

Cawley, G. C., & Talbot, N. L. C. (2006). Gene selection in cancer classification using sparse logistic regression with bayesian regularisation. *Bioinformatics*, *22*, 2348–2355.

Chang, C. C., & Lin, C. J. (2001). *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (1996). Least angle regression. *Annals of Statistics*, *32*, 407–451.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., & Lin, C.-J. (2008). LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research*, *9*, 1871–1874.

Friedman, J., Hastie, T., Hofling, H., & Tibshirani, R. (2007). Pathwise coordinate optimization. *The Annals of Applied Statistics*, *2*, 302–332.

Friedman, J., Hastie, T., & Tibshirani, R. (2009). *Regularization Paths for Generalized Linear Models via Coordinate Descent*. Technical Report Department of Statistics, Stanford University.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

Hesterberg, T., Choi, N. H., Meier, L., & Fraley, C. (2008). Least angle and l_1 penalized regression: A review. *Statistics Surveys*, *2*, 61–93.

Kim, Y. et al. (2006). Blockwise sparse regression. *Statistica Sinica*, *16*, 375–390.

Kulis, B., & Jordan, M. (2012). Revisiting k-means: New algorithms via bayesian nonparametrics. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*. Edinburgh, Scotland.

Lewis, D. D., Yang, Y., Rose, T. G., & Li, F. (2004). RCV1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, *5*, 361–397.

Minka, T. P. (2003). *A comparison of numerical optimizers for logistic regression*. Technical Report Dept. of Statistics, Carnegie Mellon Univ.

Mkhadri, A., & Ouhourane, M. (2013). An extended variable inclusion and shrinkage algorithm for correlated variables. *Comput. Stat. Data Anal.*, *57*, 631–644.

Ng, A. Y. (2004). Feature selection, L1 vs. L2 regularization, and rotational invariance. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning* (p. 78). New York, NY, USA: ACM.

Shevade, S. K., & Keerthi, S. S. (2003). A simple and efficient algorithm for gene selection using sparse logistic regression. *Bioinformatics*, *19*, 2246–2253.

Spurek, P., & Tabor, J. (2012). Cross-entropy clustering. *CoRR*, *abs/1210.5594*. `http://arxiv.org/abs/1210.5594`.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, *58*, 267–288.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., & Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society Series B*, (pp. 91–108).

Williams, P. M. (1994). Bayesian Regularisation and Pruning using a Laplace Prior. *Neural Computation*, *7*, 117–143.

Yuan, M., & Lin, Y. (2004). *Model selection and estimation in regression with grouped variables*. Technical Report 1095 Department of Statistics, University of Wisconsin, Madison, WI.

Zhang, T. (2002). Covering number bounds of certain regularized linear function classes. *Journal of Machine Learning Research*, *2*, 527–550.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *J. R. Statist. Soc. B*, *67*, 301–320.