Block coordinate descent algorithm for L1/L2 Regularized Logistic Regression

Abstract

The strucutre learning algorithm proposed for discrete graphical models described in Ravikumar et al.[3] involves solving an L1/L2 Regularized Logistic Regression problem. In this note, we describe the Block coordinate descent algorithm being used for solving this problem. This algorithm is an adaptation of the algorithm proposed in Meier et al.[1]. Early results from experiments which use this algorithm are described in a course project report[4].

1 The discrete graphical model and its parametrization

Consider a discrete pairwise graphical model describing a probability distribution over p variables, each of which can take one of m discrete values. In this report, we use a non-minimal parametrization for the graphical model; and we assume that the node potentials are all constant:

$$Pr(x|\Theta) = \prod_{(i,j)\in E} \phi_{i,j}(x_i, x_j) \propto \exp(\sum_{(i,j)\in E} \Theta_{i,j,x_i,x_j}), \tag{1}$$

This can also be written as

$$Pr(x|\Theta) = \exp(\sum_{(i,j)\in E,(l,k)\in\{1,..m\}^2} \Theta_{i,j,l,k} I[x_i = l, x_j = k]),$$

but we often resort to the former, more succinct notation.

Thus, a probability distribution is completely specified by $\Theta \in R^{p\times p\times m\times m}$, with $\forall (i,j) \notin E : \Theta_{i,j,:,:} = 0$. Note that, not being a minimal parametrization, there exist several sets of variables which describe exactly the same probability distribution. For example, one can replace the parameter array $\Theta_{i,j} \in R^{m\times m}$ corresponding to edge (i,j) with $\Theta'_{i,j} = \Theta_{i,j} + k11^T$ and still describe the same probability distribution. Minimal representations are described in Ravikumar[2] and Ravikumar et al.[3].

2 The neighborhood learning algorithm

2.1 The optimization problem

The corresponding conditional probability distribution for the node i is

$$Pr(X_i = x_i | X_{/i} = x_{/i}, \Theta) = \frac{\exp(\sum_{j \in V - \{i\}} \Theta_{i,j,x_i,x_j})}{\sum_l \exp(\sum_{j \in V - \{i\}} \Theta_{i,j,l,x_j})}. \quad (2)$$

Above, it is assumed that $\forall j \notin \Gamma(i), \forall k : \Theta_{i,j,x_i,k} = 0$.

Given n observations $S = \left\{ x^{(j)} \right\}$, and viewing x_i as a response variable whose value depends on the indicator variables like $I[x_j = k]$, we use this to construct a negative log likelihood function:

$$nll_i(\Theta_{i,:,:,:} | x^{(k)}) = - \sum_{j \in V - \{i\}} \Theta_{i,j,x_i^{(k)},x_j^{(k)}} + \log[\sum_l \exp(\sum_{j \in V - \{i\}} \Theta_{i,j,l,x_j^{(k)}})].$$

To determine the neighborhood $\Gamma(i)$ of node i, we solve the following problem:

$$\arg\min_{\Theta_{i,:,:,:}} \left\{ n^{-1} \sum_{k=1}^{n} nll_i(\Theta_{i,:,:,:} | x^{(k)}) + \lambda \sum_j \| \Theta_{i,j,:,:} \|_2 \right\}. \quad (3)$$

2.2 Gradient and the Hessian

We now describe the computation of the gradient $\nabla_\Theta nll_i(\Theta_{i,:,:,:} | x^{(k)}) = G \in R^{p \times m \times m}$.

$$\forall v \in V - \{i\} :$$

$$G_{v,x_i,x_v} = -1 + (\sum_l \exp(\sum_{j \in V - \{i\}} \Theta_{i,j,l,x_j^{(k)}}))^{-1} exp(\sum_{j \in V - \{i\}} \Theta_{i,j,x_i^{(k)},x_j^{(k)}}))$$

$$G_{v,k \neq x_i,x_v} = (\sum_l \exp(\sum_{j \in V - \{i\}} \Theta_{i,j,l,x_j^{(k)}}))^{-1} exp(\sum_{j \in V - \{i\}} \Theta_{i,j,k,x_j^{(k)}}))$$

$G_{i,j,k} = 0$ is used for all other i, j, k. $n^{-1} \sum_{k=1}^{n} \nabla_\Theta nll_i(\Theta_{i,:,:,:} | x^{(k)})$ is then used in the coordinate descent algorithm.

We now compute the diagonal elements of the Hessian. To do this, we describe the matrix $H \in R^{p \times m \times m}$, where $H_{i,j,k} = \frac{\partial^2 nll_i(x)}{\partial \Theta_{i,j,k}^2}$. The second order derivatives of $n^{-1} \sum_{k=1}^{n} nll_i(\Theta_{i,:,:,:} | x^{(k)})$ is then computed using this. This computation is easily done together with the computation of the gradient.

$$H_{v,k,x_v} = -(\sum_l \exp(\sum_{j \in V - \{i\}} \Theta_{i,j,l,x_j^{(k)}}))^{-2} exp(2 \sum_{j \in V - \{i\}} \Theta_{i,j,k,x_j^{(k)}}))$$
$$+ (\sum_l \exp(\sum_{j \in V - \{i\}} \Theta_{i,j,l,x_j^{(k)}}))^{-1} exp(\sum_{j \in V - \{i\}} \Theta_{i,j,k,x_j^{(k)}})).$$

Observe from the above that $1 \geq H_{v,x_r,x_v} \geq 0$.

2.3 The block coordinate descent algorithm

The algorithm is specified as Algorithm 1.

Algorithm 1 Block coordinate descent algorithm

Input: $\Theta_i^{(0)}$, Sample set $S = \left\{x^{(r)}\right\}$ of n points, node i whose neighborhood is to be determined, λ, tol.
Output: Θ_i.

$\Theta_i \leftarrow \Theta_i^{(0)}$
loop
  for all $v \in V - \{i\}$ **do**
    Compute $\bar{nll}(\Theta_i) = n^{-1} \sum_k nll_i(\Theta_{i,:,:,:}|x^{(k)})$.
    Compute $G = \nabla_{\Theta_i} \bar{nll}(\Theta_i)$. [We only need $G_{v,:,:}$ actually.]
    Together with G, compute the array H' with $H'_{i,j,k} = \frac{\partial^2 \bar{nll}(\Theta_i)}{\partial \Theta_{i,j,k}^2}$. [We only need $H'_{v,:,:}$ actually.]
    $h_v \leftarrow -\max\left\{H', 10^{-5}\right\}$. [Actually, need to do $h_v \leftarrow -\max H'_{v,:,:}$.]
    $T \leftarrow -G_{v,:,:} - h_v \Theta_{v,:,:}$.
    $D \leftarrow p \times m \times m$ array of zeros.
    if $\|T\|_F \geq \lambda$ **then**
      $D_{v,:,:} \leftarrow -\Theta_i$.
    else
      $D_{v,:,:} \leftarrow -h_v^{-1}[-G_v - \lambda \frac{T}{\|T\|_F}]$.
    end if
    if $\max|D| \geq tol$ **then**
      Get α from line search along D with $\alpha_0 = 2, \delta = 0.75, \sigma = .01$.
      $\Theta_{i,v} \leftarrow \Theta_{i,v} + \alpha D_v$.
    end if
  end for
  Return if decrease in objective value is less than *tol*.
end loop

References

[1] Lukas Meier, Sara van de Geer, and Peter Buhlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):53–71, February 2008. ISSN 1369-7412. doi: 10.1111/j.1467-9868.2007.00627.x. URL `http://dx.doi.org/10.1111/j.1467-9868.2007.00627.x`.

[2] Pradeep Ravikumar. Private communication, 2010.

[3] Pradeep Ravikumar, M. J. Wainwright, and J. Lafferty. High-dimensional ising model selection using l1-regularized logistic regression. *Annals of Statistics*, 2009.

[4] Vishvas Vasuki. Learning discrete graphical models using l1/l2 regularized logistic regression. Project report prepared for the course 'Sparsity, structure and algorithms.'.