Greedy Alternating Optimization for FHIM

Sanjay Purushotham, Martin Renqiang Min

NEC Labs, Princeton, New Jersey, USA

Abstract

This report investigates and examines the greedy alternating optimization procedures used for solving the non-convex optimization problem of our **F**actorized **H**igh order **I**nteractions **M**odel (FHIM) model.

Keywords: l_1-regularization, linear regression, single task learning, high order feature interactions

1. Introduction

A major challenge in bioinformatics and medical informatics is to identify interpretable high-order interactions between different inputs of complex systems predictive of systems' outcomes. For example, many human diseases are manifested as the dysfunction of some pathways or functional gene modules because genes and proteins seldom perform their functions independently and disrupted patterns due to diseases are often more obvious at a pathway or module level. However, identifying reliable discriminative high-order gene interactions from genome-wide case-control studies for accurate disease diagnosis such as early cancer diagnosis is still a challenging problem, because we often have very limited patient samples but hundreds of millions of gene interactions to consider. Moreover, many human diseases are related to each other, so effectively using shared information between diseases can significantly boost our success rate of finding informative gene interactions as biomarkers compared to solving these individual problems separately. Our proposed model to solve this problem is general, and it can be used to identify any discriminative complex system input interactions that are predictive of system outputs given limited high-dimensional training data.

2. Problem Formulation

Consider a regression setup with a training set of n samples and p features, $\{(\mathbf{X}^{(i)}, y^{(i)})\}$, where $\mathbf{X}^{(i)} \in \mathbb{R}^p$ is the i^{th} instance (column) of the design matrix $\mathbf{X}$ $(p \times n)$, $y^{(i)} \in \mathbb{R}$ is the i^{th} instance of response variable $\mathbf{y}$ $(n \times 1)$, and $i = 1, \ldots, n$. To model the response in terms of the predictors, we can set up a linear regression model

$$y^{(i)} = \boldsymbol{\beta}^T \mathbf{X}^{(i)} + \epsilon^{(i)}, \tag{1}$$

or a logistic regression model

$$p(y^{(i)} = 1 | \mathbf{X}^{(i)}) = \frac{1}{1 + \exp(-\boldsymbol{\beta}^T \mathbf{X}^{(i)} - \beta_0)}, \tag{2}$$

where $\boldsymbol{\beta} \in \mathbb{R}^p$ is the weight vector associated with single features (also called main effects), $\boldsymbol{\epsilon} \in \mathbb{R}^n$ is a noise vector, and $\beta_0 \in \mathbb{R}$ is the bias term. In many practical fields such as bioinformatics and medical informatics, the main terms (the terms only involving single features) are not enough to capture complex relationship between the response and the predictors, and thus high-order interactions are necessary. In this paper, we consider regression models with both main effects and high-order interaction terms. Equation 3 shows a linear regression model with all pairwise interaction terms.

$$y^{(i)} = \boldsymbol{\beta}^T \mathbf{X}^{(i)} + \mathbf{X}^{(i)T} \mathbf{W} \mathbf{X}^{(i)} + \epsilon^{(i)}, \tag{3}$$

where $\mathbf{W}(p \times p)$ is the weight matrix associated with all the pairwise feature interactions. The corresponding loss function (the sum of squared errors) is as follows (we center the data to avoid an additional bias term),

$$L_{sqerr}(\boldsymbol{\beta}, \mathbf{W}) = \frac{1}{2} \sum_{i=1}^{n} ||y^{(i)} - \boldsymbol{\beta}^T \mathbf{X}^{(i)} - \mathbf{X}^{T(i)} \mathbf{W} \mathbf{X}^{(i)}||_2^2. \tag{4}$$

We can similarly write the logistic regression model with pairwise interactions as follows,

$$p(y^{(i)} | \mathbf{X}^{(i)}) = \frac{1}{1 + \exp(-y^{(i)}(\boldsymbol{\beta}^T \mathbf{X}^{(i)} + \mathbf{X}^{(i)T} \mathbf{W} \mathbf{X}^{(i)} + \beta_0))} \tag{5}$$

and the corresponding loss function (the sum of the negative log-likelihood of the training data) is,

$$L_{logistic}(\boldsymbol{\beta}, \mathbf{W}, \beta_0) = \sum_{i=1}^{n} \log(1 + \exp(-y^{(i)}(\boldsymbol{\beta}^T \mathbf{X}^{(i)} + \mathbf{X}^{(i)T} \mathbf{W} \mathbf{X}^{(i)} + \beta_0)). \quad (6)$$

2.1. Our Approach

In this section, we propose an optimization-driven sparse learning framework to identify discriminative single features and groups of high-order interactions among input features for output prediction in the setting of limited training data. When the number of input features is huge (e.g. biomedical applications), it is practically impossible to explicitly consider quadratic or even higher-order interactions among all the input features based on simple lasso penalized linear regression or logistic regression. To solve this problem, we propose to factorize the weight matrix $\mathbf{W}$ associated with high-order interactions between input features to be a sum of K rank-one matrices for pairwise interactions or a sum of low-rank high-order tensors for higher-order interactions. Each rank-one matrix for pairwise feature interactions is represented by an outer product of two identical vectors, and each m-order ($m > 2$) tensor is represented by the outer product of m identical vectors. Besides minimizing the loss function of linear regression or logistic regression, we penalize the ℓ_1 norm of both the weights associated with single input features and the weights associated with high-order feature interactions. Mathematically, we solve the optimization problem to identify the discriminative single and pairwise interaction features as follows,

$$\{\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{a}}_k\} = \arg\min_{\boldsymbol{a}_k, \boldsymbol{\beta}} L_{sqerr}(\boldsymbol{\beta}, \mathbf{W}) + \lambda_\beta \|\boldsymbol{\beta}\|_1 + \sum_{k=1}^{K} \lambda_{a_k} \|\boldsymbol{a}_k\|_1 \quad (7)$$

where $\mathbf{W} = \sum_{k=1}^{K} \boldsymbol{a}_k \odot \boldsymbol{a}_k$, $\odot$ represents the tensor product/outer product, and $\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{a}}_k$ represent the estimated parameters of our model and let Q represent objective function of (7) . For logistic regression, we replace $L_{sqerr}(\boldsymbol{\beta}, \mathbf{W})$ in (7) by $L_{logistic}(\boldsymbol{\beta}, \mathbf{W}, \beta_0)$. We call our model Factorization-based High-order Interaction Model (FHIM).

Proposition 2.1. *The optimization problem in Equation 7 is convex in* $\boldsymbol{\beta}$ *and non-convex in* $\boldsymbol{a}_k$.

Because of the non-convexity property of our optimization problem, it is difficult to propose optimization algorithms which guarantee convergence to global optima. Here, we adopt a greedy alternating optimization methods to find the local optima for our problem. In the case of pairwise interactions, fixing other weights, we solve each rank-one weight matrix each time. Please note that our symmetric positive definite factorization of $\mathbf{W}$ makes this sub-optimization problem very easy. Moreover, for a particular rank-one weight matrix $\boldsymbol{a}_k \odot \boldsymbol{a}_k$, the nonzero entries of the corresponding vector $\boldsymbol{a}_k$ can be interpreted as the block-wise interaction feature indices of a densely interacting feature group. In the case of higher-order interactions, the optimization procedure is similar to the one for the pairwise interactions except that we have more rounds of alternating optimization. The parameter K of $\mathbf{W}$ is generally unknown in real datasets, thus, we greedily estimate K during the alternating optimization algorithm. In fact, the combination of our factorization formulation and the greedy algorithm is effective for estimating the interaction weight matrix $\mathbf{W}$. $\boldsymbol{\beta}$ is re-estimated when K is greedily added during the alternating optimization as shown in algorithm 1.

Algorithm 1 Greedy Alternating Optimization

1: Initialize $\boldsymbol{\beta}$ to $\mathbf{0}$, $K = 1$ and $\boldsymbol{a}_K = \mathbf{1}$
2: While (K==1) OR ($\boldsymbol{a}_{K-1} \neq \mathbf{0}$ for $K > 1$)
3: Repeat until convergence
4: $\beta_j^t = \arg\min_j Q(\beta_1^t, ..., \beta_{j-1}^t, \beta_{j+1}^{t-1}, \beta_p^{t-1}), \boldsymbol{a_k}^{t-1})$
5: $a_{k,j}^t = \arg\min_j Q((a_{k,1}^t, ..., a_{k,j-1}^t, a_{k,j+1}^{t-1}, a_{k,p}^{t-1}), \boldsymbol{\beta}^t)$
6: End Repeat
7: K = K + 1; $\boldsymbol{a}_K = \mathbf{1}$
8: End While
9: Remove $\boldsymbol{a}_K$ and $\boldsymbol{a}_{K-1}$ from $\boldsymbol{a}$.

3. Optimization Procedures

In this section, we explore different optimization procedures for improving our Greedy Alternating Optimization algorithm mentioned in Algorithm 1.

3.1. Initialization and Warm Start

Initialization: Initialization of $\boldsymbol{\beta}$ and a_k plays a key role in finding a good local optima for our optimization problem. A poor initialization leads to solution getting stuck in a bad local optima. Thus, we cleverly initialize our parameters as follows: Set $\boldsymbol{\beta}$ as $\boldsymbol{\beta}_{LASSO}$ or $\boldsymbol{\beta}_{L1LogReg}$ and $\mathbf{a}_k$ as $\mathbf{1}$. We could also initialize $\mathbf{a}_k$ with $\mathbf{a}_{OLS}$, however since we don't know the true low rank $\mathbf{K}$ of $\mathbf{W}$, we initialize $\mathbf{a}_k$ as 1 and estimate $\mathbf{K}$ greedily. Table 1 shows how different initializations of $\boldsymbol{\beta}$ affects the support recovery of $\mathbf{W}_{est}$ and $\boldsymbol{\beta}_{est}$. This table shows that initializing $\boldsymbol{\beta}$ with $\boldsymbol{\beta}_{Lasso}$ gives better results that initializing $\boldsymbol{\beta}$ with 0.

Warm-Start: Finding the optimal tuning parameters λ_β and λ_{a_k} is challenging and generally many researchers find the optimal parameters by grid search and cross-validation. We can use 'warm-start' strategy to improve the search performance. We can use the solution of optimization problem using one λ as the initialization for the optimization problem using a different λ. In our experiments, we found that the warm-start strategy reduces the iterations needed for convergence.

$\mathbf{n}$	$\mathbf{p}$	$\mathbf{K}$	$\boldsymbol{\beta}_{init}$	$\boldsymbol{\beta}_{est}$	$\mathbf{W}_{est}$	Time	$\boldsymbol{\beta}_{sparsity}$	$\mathbf{a}_{k,sparsity}$
1000	50	1	Lasso	1	0.7246	75.9	5	5
1000	50	1	0	1	0.6956	81.98	5	5
1000	50	3	Lasso	1	0.4719	52.27	5	5
1000	50	3	0	1	0.48	71.1	5	5
1000	50	5	Lasso	1	0.3576	91.82	5	5
1000	50	5	0	1	0.347	112.97	5	5
100	500	1	Lasso	0.632	0.013	270.00	25	25
100	500	1	0	0.616	0.008	540.63	25	25

Table 1: Performance of High order linear regression Off-Diagonal FHIM with different initialization of $\boldsymbol{\beta}_{init}$

3.2. Stopping Criterion for greedy optimization

In our experiments, we found that the stopping criterion of the alternating optimization algorithm directly affects the convergence rate and the converged solution. Due to the non-convexity of our greedy optimization problem, the decrease in objective function's value may not correspond to

the decrease in loss function's value. Thus, we introduce a new stopping criterion where our optimization algorithm keeps track of the decrease in both the loss function's value and the objective function's value along with the best loss function during the iterations. During the greedy estimate of K, we stop the addition of new $\mathbf{a}_k$, if and only if the loss function of the $\mathbf{a}_k$ is larger than the loss function of $\mathbf{a}_{k-1}$ for $k > 1$. When the stopping criterion is met, the parameters related to the best loss function is chosen as the parameters for the local optima.

3.3. Normalization

We observed in our experiments that the normalization of the loss function and the sub-gradient has a significant impact on the tuning parameters λ_β and λ_{a_k}. In particular, for the high order logistic regression, unnormalized loss function and unnormalized sub-gradient are helpful for choosing the tuning parameter better.

3.4. Fast Greedy Alternating Optimization

In algorithm 1, we optimized each $\mathbf{a}_k$ ($\forall k \in (1, K)$) during all the iterations of greedy alternating optimization. Below, we provide a new faster greedy algorithm where instead of optimizing each $\mathbf{a}_k$, we only optimize $\mathbf{a}_K$. We use the optimization procedures described in this section in our new fast greedy alternating optimization. Table 2 shows that only optimizing $\mathbf{a}_K$ is faster that optimizing each $\mathbf{a}_k$ in all iterations.

Algorithm 2 Fast Greedy Alternating Optimization

1: Initialize $\boldsymbol{\beta}$ to $\boldsymbol{\beta}_{LASSO}$, $K = 1$ and $\boldsymbol{a}_K = \mathbf{1}$
2: While (K==1) OR ($\boldsymbol{a}_{K-1} \neq \mathbf{0}$ for $K > 1$)
3: Repeat until convergence
4: $a^t_{K,j} = \arg\min_j Q((a^t_{K,1}, ..., a^t_{K,j-1}, a^{t-1}_{K,j+1}, a^{t-1}_{K,p}), \boldsymbol{\beta}^{t-1})$
5: $\beta^t_j = \arg\min_j Q(\beta^t_1, ..., \beta^t_{j-1}, \beta^{t-1}_{j+1}, \beta^{t-1}_p), \boldsymbol{a_K}^t)$
6: End Repeat
7: K = K + 1; $\boldsymbol{a}_K = \mathbf{1}$
8: End While
9: Return $\boldsymbol{a}_K$ and $\boldsymbol{\beta}$ which has the least loss function.

$\mathbf{n}$	$\mathbf{p}$	$\mathbf{K}$	$\boldsymbol{\beta}_{init}$	Updating $\mathbf{a}_k$	$\boldsymbol{\beta}_{est}$	$\mathbf{W}_{est}$	Time	$\boldsymbol{\beta}_{sparsity}$	$\mathbf{a}_{k,sparsity}$
1000	50	1	Lasso	update all $\mathbf{a}_k$	1	0.6956	88.38	5	5
1000	50	1	Lasso	update only $\mathbf{a}_K$	1	0.7826	80.57	5	5
1000	50	3	Lasso	update all $\mathbf{a}_k$	1	0.4340	120.87	5	5
1000	50	3	Lasso	update only $\mathbf{a}_K$	1	0.4265	107.77	5	5
1000	50	5	Lasso	update all $\mathbf{a}_k$	1	0.3893	154.12	5	5
1000	50	5	Lasso	update only $\mathbf{a}_K$	1	0.3932	126.98	5	5

Table 2: Performance of High order linear regression Off-Diagonal FHIM with different ways to update $\mathbf{a}_k$

4. Off-Diagonal FHIM

In section 2, high order regression problem formulations (eqn. (3) and (eqn. 5) were considered with $\mathbf{W}$ as a $(p \times p)$ matrix with all possible pairwise interactions including the self-interactions of variables. Below we define new problem formulations for high order regression problems with only pairwise interactions between the variables (i.e. no self interactions are included in the model). Thus, $\mathbf{W}$ becomes an off-diagonal $(p \times p)$ matrix (all diagonal elements are zeros) and is denoted by $\mathbf{W}_{OD}$.

$$y^{(i)} = \boldsymbol{\beta}^T \mathbf{X}^{(i)} + \mathbf{X}^{(i)T} \mathbf{W}_{OD} \mathbf{X}^{(i)} + \epsilon^{(i)}, \tag{8}$$

where $\mathbf{W}_{OD}(p \times p)$ is the weight matrix associated with only the pairwise feature interactions. The corresponding loss function (the sum of squared errors) is as follows (we center the data to avoid an additional bias term),

$$L_{sqerr}(\boldsymbol{\beta}, \mathbf{W}_{OD}) = \frac{1}{2} \sum_{i=1}^{n} ||y^{(i)} - \boldsymbol{\beta}^T \mathbf{X}^{(i)} - \mathbf{X}^{T(i)} \mathbf{W}_{OD} \mathbf{X}^{(i)}||_2^2. \tag{9}$$

We can similarly write the logistic regression model with pairwise interactions as follows,

$$p(y^{(i)}|\mathbf{X}^{(i)}) = \frac{1}{1 + \exp(-y^{(i)}(\boldsymbol{\beta}^T \mathbf{X}^{(i)} + \mathbf{X}^{(i)T} \mathbf{W}_{OD} \mathbf{X}^{(i)} + \beta_0))} \tag{10}$$

and the corresponding loss function (the sum of the negative log-likelihood of the training data) is,

$$L_{logistic}(\boldsymbol{\beta}, \mathbf{W}_{OD}, \beta_0) = \sum_{i=1}^{n} \log(1 + \exp(-y^{(i)}(\boldsymbol{\beta}^T \mathbf{X}^{(i)} + \mathbf{X}^{(i)T} \mathbf{W}_{OD} \mathbf{X}^{(i)} + \beta_0)). \quad (11)$$

Our FHIM for the high order regression formulations with $\mathbf{W}_{OD}$ is termed as Off-Diagonal FHIM.