EE8231 Optimization Theory Course Project

Qian Zhao (zhaox331@umn.edu)

May 15, 2014

Contents

Abstract

Logistic regression and matrix factorization specifically SVD are both very popular in application areas of computational advertising and recommender system. This project basically can be divided into two parts. The first part is to test different optimization methods to solve sparse 2-class logistic regression. This is a convex non-smooth problem. Particularly, I'll try to answer the question of "how to handle the non-smooth part of the L1-regularizer in the objective function". The second part is to try to solve the convex approximation of the matrix completion problem specifically SVD and extend this approximation from the L2-norm loss function to hinge/logistic loss function which would combine these two parts together. Beyond traditional SVD, SVDFeature[1] or other similar model is proposed to handle auxiliary information. This project also tries to give a convex approximation to the auxiliary information augmented family of SVD models.

1 Introduction

Regarding sparse 2-class logistic regression in machine learning, different methods have been proposed to handle this problem especially the non-smooth L1-regularizer part, such as ADMM(Alternating Direction Method of Multipliers), COGD(Composite Objective Gradient Descent), CDN(Coordinate Descent Method Using One-dimensional Newton Directions), OWL-QN(Orthant-Wise Limited-memory Quasi-Newton method). There are also possibilities to transform this problem into an equivalent smooth constraint optimization problem and then approach it using interior-point methods(IPM). The question that this project answers is "what's the difference of these methods to solve the same problem, including convergence properties and performance etc."

Regarding matrix factorization (SVD) or low-rank matrix completion in classical form, it can be relaxed to a convex optimization problem through nuclear norm. This is true when the loss function is L2-Norm. What if the loss function is logistic loss which takes the same loss function as logistic regression does or hinge loss? Are there convex approximations to this problem? The current practice is using stochastic gradient descent or alternative least square. To handle given auxiliary information besides matrix entries, SVDFeature and other extensive models from SVD are put forward. The question is "are there convex approximations to this extended problem?" Lastly, how to combine logistic regression and matrix factorization in the same loss function while achieving a convex approximation.

2 Sparse 2-class Logistic Regression

2.1 Problem formulation

Let $\{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$ be the dataset under consideration, where $x_i \in R^d, y_i \in \{0, 1\}$. The objective function to be minimized is given by:

$$\text{minimize } f(w) = h(w) + g(w) = -\sum_{i=1}^{N}\{y_i w^T x_i - log(1 + exp(w^T x_i))\} + \lambda\|w\|_1. \quad \text{(P1)}$$

where $h(w) = -\sum_{i=1}^{N}\{y_i w^T x_i - log(1 + exp(w^T x_i))\}$ which is smooth and $g(w) = \lambda\|w\|_1$ which is non-smooth. The output of the modal is $\hat{y} = w^T x_i$ and then squeezed with a sigmoid function.

The non-smooth part $g(w)$ makes this problem harder to deal with because $f(w)$ has no gradient when $w = 0$. Note that the output of the model can also be written as $\hat{y} = w^T x_i + b$ adding a bias term. Sometimes b is included in w by adding a constant dimension in x. Unless specially pointed out, the model here doesn't explicitly use the bias term.

There are different ways to solve this problem. Before exploring them, a brief introduction to applications of this problem is mentioned below.

2.2 Applications

Sparse 2-class logistic regression is widely used in binary classification and ranking. For example, this model can be used in spam filtering where the output denotes spamming or not spamming. The texts are classified into these two categories based on the input features extracted from the texts. Another example is to rank the advertisements in computational advertising such as sponsored search system, because the output of this model can be directly used to fit the click-through rate(CTR) or the probability of clicking with sound probabilistic interpretation. This model can be used in recommender system as well where the system tries to predict users' preference on items based on their probability of liking the items. What's attractive about adding the L1 regularization is to get a sparse trained model which automatically gets rid of weights for non-descriptive features.

2.3 Solving constrained optimization: IPM

(P1) can be transformed into an equivalent constraint optimization problem with a smooth objective function given by:

$$\text{minimize } f(w) = -\sum_{k=1}^{N} \{y_i w^T x_i - log(1 + exp(w^T x_i))\} + \lambda u \tag{P2}$$

$$\text{subject to: } \|w\|_1 \leq u$$

[2] consider a log barrier function so that (w, u) is an interior point of the feasible region:

$$\text{minimize } f(w, u) = t(\sum_{i=1}^{d} u_i + Ch(w)) - \sum_{j=1}^{d} log(u_j^2 - w_j^2) \tag{P3}$$

where $t > 0$ is a parameter. The "central path" formed by the unique minimizers for different t approaches an optimal solution as $t \to \infty$. For every iteration, IPM tries to solve P3 by finding a Newton direction, therefore enjoies a fast convergence. This will be shown in the experiment part. See [2] for more details about how IPM works. But it doesn't discuss details of their method's convergence. Note that the objective function to balance the fitting part h(w) and regularization part is delt with differently from P2 using C instead of λ. However, they have unique correspondence to each other.

2.4 Subgradient/Pseudo-gradient based method: OWL-QN

The L-BFGS limited-memory quasi-Newton method works very well for smooth and differentiable objective functions. [3] extends it to deal with objective functions with a non-smooth regularization part such as sparse logistic regression. Basically, what they did was extending gradient to pseudo-gradient which is a quite similar concept to subgradient. They also limit the algorithm's search

to the same orthant so that the non-smooth critical point is avoided. Once the next point crosses orthant, it's set to be zero. Pseudo-gradient is defined as the left partial derivative when the left partial derivative is greater than zero and as the right partial derivative when the right partial derivative is less than zero, otherwise it's zero. As the following equation shows:

$$\diamond f(x) = \begin{cases} \partial_i^- f(x), \text{if } \partial_i^- f(x) > 0 \\ \partial_i^+ f(x), \text{if } \partial_i^+ f(x) < 0 \\ 0, \text{otherwise.} \end{cases} \tag{2.1}$$

Since left partial derivative must be less than or equal to right partial derivative because of convexity, this function is well-defined. This is actually a narrowly defined subgradient. See [3] for more details about their algorithms. Convergence is proved.

2.5 Cyclic Coordinate Descent Method: CDN

As a coordinate descent method, CDN[4] fixes all but one variable of w and minimizes a one-variable function. That is,for every element in w, it tries to solve the following sub problem (P4) where k is the iteration. CDN is proposed based on BBR[5] which minimizes a quadratic function upper-bounding the function $l_j(z)$ in a trust region to approximately solve (P4). While trying to find an upper-bound of $l_j(z)$, BBR doesn't use the second derivative of $l_j(z)$ at 0, that is $l_j^{''}(0)$, CDN uses this information and is shown to gain faster convergence. See [4] for more details about their algorithm.

$$\text{minimize}_z l_j(z) = |w_j^{k,j} + z| - |w_j^{k,j}| + h_j(z; w^{k,j}) - h_j(0; w^{k,j}) \tag{P4}$$

2.6 Primal-Dual Method: ADMM

ADMM[6] is closely related to dual decomposition and method of multipliers. In constrained optimization, method of multipliers uses an augmented Lagrangian as the dual problem. Based on this, ADMM transformed the original problem (P1) into the following constrained optimization version.

$\text{minimize}_{x,z} h(x) + g(z)$

subject to $x = z$

The augmented Lagrangian is $L(x, z, y) = h(x) + g(z) + < y, x - z > + \frac{\rho}{2}\|x - z\|_2$. So the ADMM udpates are:

$x_{k+1} = argmin_x h(x) + < y_k, x - z_k > + \frac{\rho}{2}\|x - z_k\|_2$

$z_{k+1} = argmin_z \lambda|z|_1 + < y_k, x_{k+1} - z > + \frac{\rho}{2}\|x_{k+1} - z\|_2$

$y_{k+1} = y_k + \rho(x_{k+1} - z_{k+1})$

Although the second sub problem to achieve z_{k+1} is non-smooth, it has a closed-form solution which is usually called soft-threshold. By its optimality condition, zero should be in it's sub differential, and further gives the following solution.

$$z_{j,k+1} = \begin{cases} (y_{j,k} + \rho x_{j,k+1} - \lambda)/\rho, \text{if } y_{j,k} + \rho x_{j,k+1} > \lambda \\ (y_{j,k} + \rho x_{j,k+1} + \lambda)/\rho, \text{if } y_{j,k} + \rho x_{j,k+1} < -\lambda \\ 0, \text{otherwise.} \end{cases} \tag{2.2}$$

When h(x) is decomposible, the first sub problem to abtain x_{k+1} is also easily decomposed, which means you can do variable splitting. This is one of the biggest benefits of ADMM as claimed in [6]. In the experiment, the first sub problem is solved using limited memory quasi-Newton based on L-BFGS similar to what OWL-QN does in it's main problem. [6] also gives the proof of convergence in Appendix A.

2.7 Proximal Gradient Method: COGD

The proximal gradient of f(w) with respect to $g(w)$ is defined as: $\hat{\nabla} f(w) = w - prox_g(w - \nabla h(w))$, where $prox_g(z) = \text{argmin}_y g(y) + \frac{1}{2}\|z - y\|^2$. The optimality condition for it is $\hat{\nabla} f(w) = 0$, that is, $w = prox_g[w - \alpha \nabla h(w)]$. So the update becomes:

$w_{k+1} = prox_g[w_k - \alpha^k \nabla h(w)] \Leftrightarrow$

$w_{k+1} = argmin_y g(y) + \frac{1}{2}\|w - \alpha^k \nabla h(w) - y\|^2 \Leftrightarrow$

$w_{k+1} = argmin_y g(y) + < \nabla h(w), y - w_k > + \frac{1}{2\alpha_k}\|y - w_k\|^2$

This algorithms is also called Composite Objective Gradient Descent(COGD). In the experiment, α_k is used as a unified parameter $\rho = \alpha_k$. It varies with different datasets. More details are given in the experiment section. From another perspective, this method converts the original harder problem to a sequence of easier sub problems which actually have a closed form solution. It can also be regarded as minimizing a quadratic upper bound of the original objective function. The proof of convergence can be given in three steps, which are sufficient decrease, local error bound and cost-to-go estimate. See [7] for more details. The solution for every iteration is given as follows.

$$w_{j,k+1} = \begin{cases} \rho(\frac{w_{j,k}}{\rho} - \nabla h(w_{j,k}) - \lambda), \text{if } \frac{w_{j,k}}{\rho} - \nabla h(w_{j,k}) > \lambda \\ \rho(\frac{w_{j,k}}{\rho} - \nabla h(w_{j,k}) + \lambda), \text{if } \frac{w_{j,k}}{\rho} - \nabla h(w_{j,k}) < -\lambda \\ 0, \text{otherwise.} \end{cases} \tag{2.3}$$

3 Matrix Completion With Hinge/Logistic Loss

3.1 Problem formulation

Let $(x_{g,1}, x_{u,1}, x_{v,1}, y_1), (x_{g,1}, x_{u,2}, x_{v,2}, y_2), ..., (x_{g,N}, x_{u,N}, x_{v,N}, y_N)$ be the dataset under consideration, where $x \in R^d$ and N is the number of the instances. y can be in {-1, 1} or {0, 1} depending on the loss function. The former one is for hinge loss and the latter on is for logistic loss. Either way, the output is binary. The problem here is as follows where D is a dictionary of factors and has dimention d x r. B is a dictionary of biases and has dimention d x 1. $g(D, B)$ is the regularization term on the variables D and B.

$$\text{minimize } L = \sum_{i=1}^{N} \{max\{1 - y_i(x_{g,i}^T B + x_{u,i}^T D^T D x_{v,i}\}, 0\} + g(D, B) \qquad \text{(P5, hinge loss)}$$

For logistic loss, the problem becomes:

minimize $L = -\sum_{i=1}^{N}\{y_i(x_{g,i}^T B + x_{u,i}^T D^T D x_{v,i}) - log(1 + exp(x_{g,i}^T B + x_{u,i}^T D^T D x_{v,i})) + g(D, B)\}$ (P6)

In the following discussion, in order to make it simple but general enough, hinge loss that is (P5) will be used. It can be easily applied to (P6). This is non-convex because of the quadratic term $D^T D$. The goal of this part of the project is to figure out a convex approximation to this problem. As we know, in standard matrix completion, assuming matrix $M \in R^{mxn}$ and a subset E of the entries are known, the problem is as follows where U and V are the two factored matrices with low rank r:

$$\text{minimize}_{x,y,z}\|Z - UV^T\|_2^2 \tag{P7}$$

subject to $Z_{i,j} = M_{i,j}, (i,j) \in E$

This standard matrix completion problem can be transformed into an equivalent form where P_E denotes projection to the entries in E:

$$\text{minimize}_Z rank(Z) + \lambda\|P_E(Z - M)\|^2 \tag{P8}$$

This can further be relaxed to a convex optimization problem as follows where $\|Z\|_*$ denotes the nuclear norm:

$$\text{minimize}_Z\|Z\|_* + \lambda\|P_E(Z - M)\|^2$$

(P5) is different from this standard matrix completion problem in two aspects. First is that the loss function is changed to hinge or logistic loss from L2-norm loss because of the binary output of the data. Second, the factorization is not in terms of two single matrices anymore. Instead, it is factorized into a product of two parts where each part is a linear combination of the columns of the dictionary D. This makes figuring out a convex approximation a little bit harder. Before going into that, a brief introduction of its applications are mentioned in the following section.

3.2 Applications

As we know, standard matrix completion problem is closely related to the Netflix problem where M contains the movie ratings from users. By compleing the rating matrix, users' preference on their unrated movies are predicted which can be used for movie recommendation. Sometimes, the rating is not given in real scale, rather it's a binary indicator that could be playing, browsing or other similar implicit feedback from users. In this case, binary matrix completion is used to model the data.

What's more, the data usually contain more than ratings like users' demographic information, items' attributes and other short-term or even session based context features in online applications. In order to utilize this information, matrix completion(SVD) is extended to be auxiliary information augmented like SVDFeature [1]. This model is more general and gain better predicting accuracy and thus very popular in real applications which makes it significant to figure out an efficient way of solving this problem illustrated in (P5). Correspondingly, x_u are relevant features about the users and x_v are relevant features about items. There are possible global features that may represent global bias in the data. They are given in x_g.

3.3 Convex approximation: matrix lifting

The current practice of solving matrix completion problem is Alternating Least Square (for L2-norm loss) or Stochastic Gradient Descent. This project only gives simulated numerical experiments using Alternating minimization compared with the convex approximation discussed here based on matrix lifting. For binary matrix completion problem without auxiliary information modeling, [8] has given a convex approximation solution. They also talked about the probabilistic interpretation of the logistic model. This section goes beyond it and gives a convex approximation for more general (P5).

By relacing the D^TD part with another higher rank matrix(matrix lifting), (P5) is transformed into the following equivalent form:

$$\text{minimize } L = \sum_{i=1}^{N}\{max\{1 - y_i(x_{g,i}^T B + x_{u,i}^T \tilde{D} x_{v,i}\}, 0\} + g(\tilde{D}, B) \tag{P9}$$

$$\text{subject to: } \tilde{D} = D^TD <=> \tilde{D} \succeq 0 \text{ and } rank(\tilde{D}) \leq r$$

In (P9), the only nonconvex part is the rank contraint on matrix $\tilde{D}$. Applying the same trick as the convex approximation of standard matrix completion problem based on nuclear norm, (P9) is relaxed to be the following convex optimization problem which is a SDP(Semi-definite programming). $\|\tilde{D}\|_*$ is the nuclear norm of $\tilde{D}$.

$$\text{minimize } L = \sum_{i=1}^{N}\{max\{1 - y_i(x_{g,i}^T B + x_{u,i}^T \tilde{D} x_{v,i}\}, 0\} + \lambda_1\|\tilde{D}\|_* + \lambda_2|B|_1 \tag{P10}$$

$$\text{subject to: } \tilde{D} \succeq 0$$

To evaluate this method, a comparison with alternating minimization is done in the numerical experiment. Note that, although alternating minimization is not convex as a whole and thus suffer from the possible local minimum, it has more power of modeling. For example, instead of using a quadratic term D^TD in (P5), it can be replaced with a bilinear term U^TV where two different dictionaries are used for user and item features.

4 Numerical Experiments

4.1 Sparse 2-class logistic regression

The numerical experiments done here are based on the previous work [4] where different optimization methods are compared for sparse logistic regression. In it, CDN wins. However, they didn't give comparison between CDN and OWL-QN and they didn't have ADMM and COGD then. To compare CDN/IPM with ADMM/COGD with the same baseline, ADMM and COGD are implemented in the project in C++ based on the OWL-QN package in [3]. Limited memory quasi-Newton with L-BFGS is used to solve the first sub problem of ADMM. Different metrics are illustrated including accuracy, gradient norm(g-norm), relative objective and non zero variables as the solving process goes. Because of the different balance ways in terms of C and λ in the objectives mentioned in the IPM subsection, all the metrics are properly scaled. Two datasets "duke" and "a2a" are used. The statistics for these two datasets are as follows. See [4] for more details.

Name	#features	#instances	#non-zero entries
a2a	123	32,561	451,592
duke	7,129	44	306,547

Table 1: The statistics for "duke" and "a2a" dataset.

Following are the figures for different metrics. Note that the bias term b is included by adding a constant dimention in the input feature x as mentioned in section 2.1.

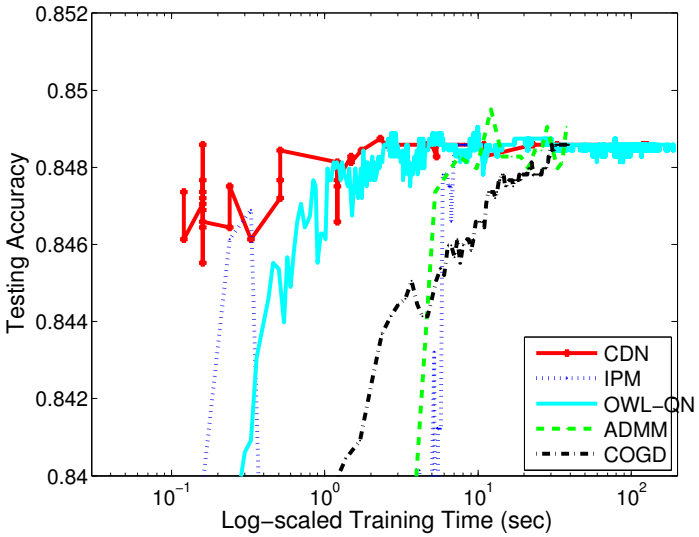

Figure 1: Testing accuracy on a2a dataset with log-scaled training time

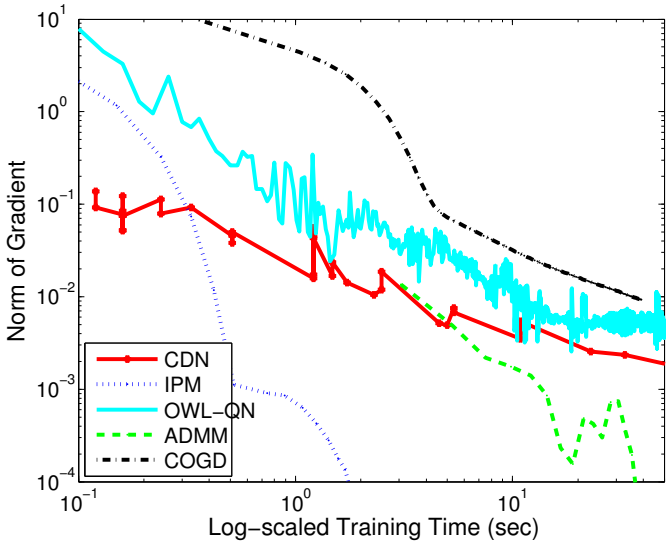

Figure 2: Gradient norm on a2a dataset with log-scaled training time

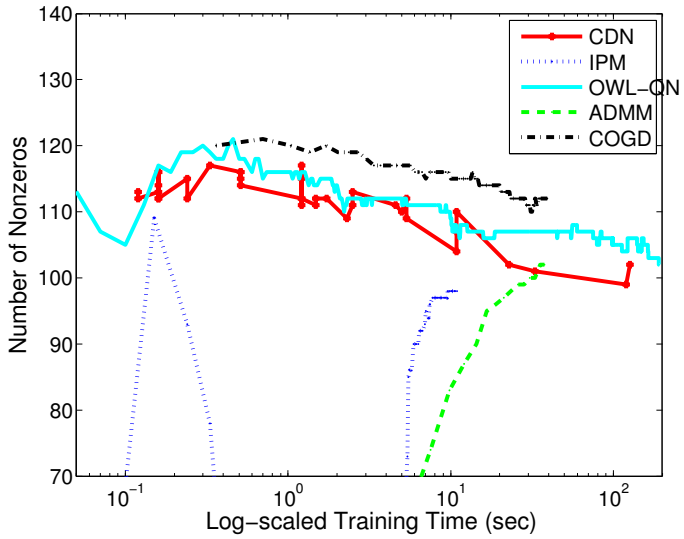

Figure 3: Number of nonzero variable on a2a dataset with log-scaled training time

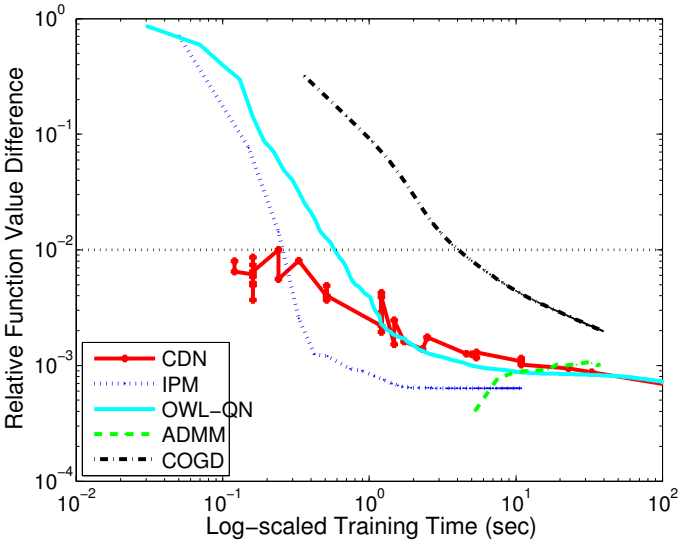

Figure 4: Relative objective on a2a dataset with log-scaled training time

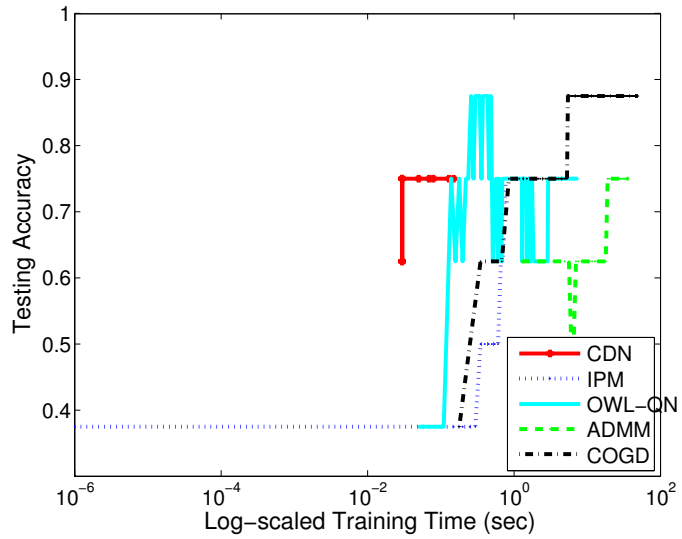

Figure 5: Testing accuracy on duke dataset with log-scaled training time

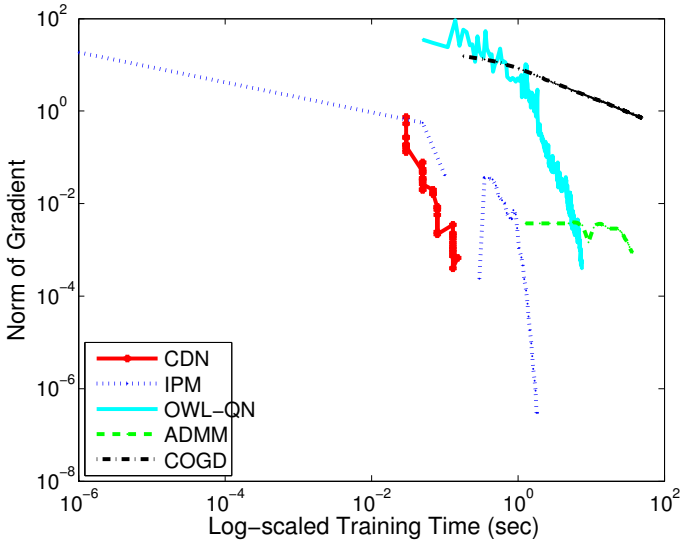

Figure 6: Gradient norm on duke dataset with log-scaled training time

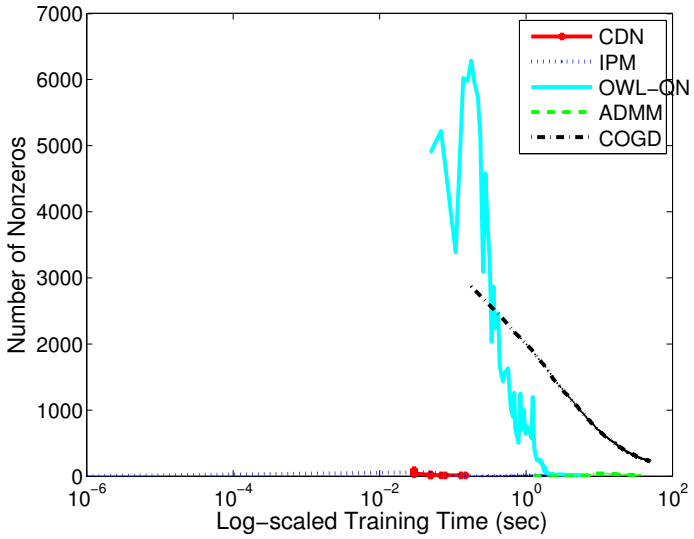

Figure 7: Number of nonzero variable on duke dataset with log-scaled training time

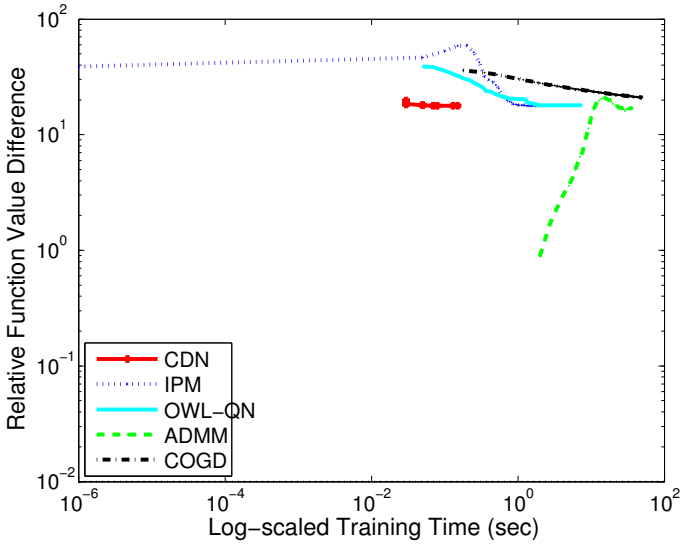

Figure 8: Relative objective on duke dataset with log-scaled training time

4.2 Matrix completion with hinge/logistic loss

The methods to solve (P5) relacing $D^T D$ with $U^T V$ and P(10) are implemented in Matlab based on package "CVX"[9]. Because of the performance directly using CVX semidefnite programming, this experiment is done on a simulated small dataset where the size of the factor dictionary is 10(meaning there are 10 features for both users and items) and the size of the bias dictionary is 15. 50 instances are randomly generated. x_u, x_v, x_g all have sparsity of 0.1 while generated. The output is randomly assigned 1 or -1 with equal ratio. In both problems, the parameter to control

the regularization on bias dictionay B is 0.01. For (P5), L2-norm regularization is used on U and V and the coefficient is 0.01 as well. 10 rounds of alternating is done to capture a rough idea about the optimizing process of alternating minimization. r=2 for matrices U and V.

Following are the training results for alternating minimization to solve (P5):

Round	#obj	#accuracy
1	18.8745	0.8000
2	18.2044	0.8000
3	18.8745	0.8000
4	18.2044	0.8000
5	18.8745	0.8000
6	18.2044	0.8000
7	18.8745	0.8000
8	18.2044	0.8000
9	18.8745	0.8000
10	18.2044	0.8000

Table 2: Training process of alternating minimization for (P5).

For (P10), the following results are obtained for different nuclear norm regularizers.

Nuclear norm regularizer	#non-zero eigen value	accuracy
1	1(value: 2.78)	0.80
1e-2	1(value: 20.424)	0.80
1e-4	1(value: 236.37)	0.78
1e-6	1(value: 2270.59)	0.80
1e-8	9(values: 22398.00, 2.48, 2.48,...)	0.80

Table 3: Results for (P10) when nuclear norm regularizer is different.

5 Discussion

For sparse logistic regression problem, from the above figures, we can see that CDN still has convergence advantage over other methods as claimed in [4]. Relatively, CDN converges faster than any other methods. However, for the metric of gradient norm, IPM performs better compared with CDN on "a2a" dataset where the number of features is smaller than the number of instances. It's obvious that IPM is not stable for metrics of testing accuracy, number of nonzeros on both "a2a" and "duke" datasets during optimization. OWL-QN has consistently performed a little bit worse than CDN for all metrics.

ADMM and COGD are converging slower compared with other methods. Especially, COGD converges very slow and takes much longer time. During running the optimizing process, COGD takes less time for every iteration but needs thousands of iterations to converge. However, COGD gains the best accuracy on "duke" dataset. More datasets are needed to validate this phenomenon. Because ADMM is alternatingly optimizing the primal and dual, the objective of ADMM computed

with Z variable for the L1-norm regularization is acutally increasing as the training time goes. The duality gap is decreasing to zero. In this experiment, ADMM has no advantage over CDN and OWL-QN, but because ADMM is naturally distributed, other experiments in parallel environment needs to be done to figure out the possible advantage of ADMM over other methods as claimed in [6]. COGD is very easy to implement because its subproblem has closed-form solution, but gives poor performance compared with other methods. In terms of number nonzeros, ADMM and IPM has advantage for "a2a" dataset and CDN and ADMM has advantage for "duke" dataset. For both datasets, OWL-QN initially gets a lot of nonzeros, then gradually decreases the number. However, because "a2a" dataset has small number of features, CDN generally performs better in terms of nonzeros.

For binary matrix completion problem, although the convex optimization after matrix lifting has nice convexity, it's hard to control the nuclear norm regularizer. The number of the nonzero eigen values of matrix $\tilde{D}$ varies dramatically as the regularizer differs. Alternating minimization achieves minimum in the first round of alternating and gets the best accuracy that both methods could possibly get. This shows that alternating minimization works well even it's not convex optimization. The regularizer needs to be tuned for different rounds so that the optimization is not biases to the first several rounds.

When matrix completion is using a logistic loss, sparse logistic regression becomes a special case of matrix completion where the data don't have user and item features. The global bias B resembles the weight w in sparse logistic regression. This makes it very easy to combine the two parts talked about in this report through a unified logistic loss function. But, aparently more general matrix completion is a harder problem to solve and thus put forward challenges to the current working methods for sparse logistic regression. Future work will be done on effciently training SVDFeature etc. or general matrix completion problem using different optimizatioin methods.

References

[1] Chen, Tianqi, et al. "SVDFeature: a toolkit for feature-based collaborative filtering." The Journal of Machine Learning Research 13.1 (2012): 3619-3622.

[2] Kwangmoo Koh, Seung-Jean Kim, and Stephen Boyd. An interior-point method for large-scale l1-regularized logistic regression. Journal of Machine Learning Research, 8:15191555, 2007.

[3] Andrew, Galen, and Jianfeng Gao. "Scalable training of L 1-regularized log-linear models." Proceedings of the 24th international conference on Machine learning. ACM, 2007.

[4] Yuan, Guo-Xun, Kai-Wei Chang, Cho-Jui Hsieh, and Chih-Jen Lin. A comparison of optimization methods and software for large-scale l1-regularized linear classification. The Journal of Machine Learning Research 9999 (JMLR 2010): 3183-3234.

[5] Alexandar Genkin, David D. Lewis, and David Madigan. Large-scale Bayesian logistic regression for text categorization. Technometrics, 49(3):291304, 2007.

[6] Boyd, Stephen, et al. "Distributed optimization and statistical learning via the alternating direction method of multipliers." Foundations and Trends in Machine Learning 3.1 (2011): 1-122.

[7] Schmidt, Mark W., Nicolas Le Roux, and Francis Bach. Convergence Rates of Inexact Proximal-Gradient Methods for Convex Optimization. Advances in Neural Information Processing Systems (NIPS. 2011).

[8] Davenport, Mark A., et al. "1-bit matrix completion." arXiv preprint arXiv:1209.3672 (2012).

[9] Grant, Michael, Stephen Boyd, and Yinyu Ye. "CVX: Matlab software for disciplined convex programming." (2008).