Non-negative matrix factorization with smoothness and sparse penalties

Teodor Marinov, Matthew Francis-Landau, Ryan Cotterell

1 Problem formulation

In this project we consider a variant of the non-negative matrix factorization problem (NMF) [**?**]. The basic NMF problem is posed as follows

$$\begin{aligned} \underset{\mathrm{W}\in\mathbb{R}^{d\times k},\mathrm{H}\in\mathbb{R}^{k\times n}}{\text{minimize:}} \quad & \|\mathrm{X}-\mathrm{WH}\|_F^2 \\ \text{subject to:} \quad & \mathrm{W}_{i,j}\geq 0, \mathrm{H}_{i,j}\geq 0 \end{aligned} \tag{1}$$

where $\mathrm{X}\in\mathbb{R}^{d\times n}$ is some data matrix and k is given and fixed. This is a non-convex optimization problem. In [**?**] the authors suggest simple alternating multiplicative updates and claim that the proposed algorithm has a fixed point. In [**?**], however, it is indicated that the claim is wrong. Another approach to solving problem 1 is the following algorithm – initialize $\mathrm{W}_0, \mathrm{H}_0$ randomly, at step t set W_t to be the minimizer of

$$\begin{aligned} \underset{\mathrm{W}\in\mathbb{R}^{d\times k}}{\text{minimize:}} \quad & \|\mathrm{X}-\mathrm{W}_{t-1}\mathrm{H}_{t-1}\|_F^2 \\ \text{subject to:} \quad & \mathrm{W}_{i,j}\geq 0 \end{aligned} \tag{2}$$

and H_t to be the minimizer of

$$\begin{aligned} \underset{\mathrm{H}\in\mathbb{R}^{k\times n}}{\text{minimize:}} \quad & \|\mathrm{X}-\mathrm{W}_{t}\mathrm{H}_{t-1}\|_F^2 \\ \text{subject to:} \quad & \mathrm{H}_{i,j}\geq 0 \end{aligned} . \tag{3}$$

Proceed to carry out this alternating minimization approach until some stopping criteria is met e.g. $\|\mathrm{W}_t\mathrm{H}_t-\mathrm{W}_{t+1}\mathrm{H}_{t+1}\|_F^2<\epsilon$. In [**?**] it is shown that this algorithm is going to have a fixed point. Note that 2,3 are now constraint convex-optimization problems so one can choose their favorite method to solve them.

Usually NMF is applied to real-world problems where the W and H term have some interpretation – for example X can be the Fourier power spectogram of an audio signal where the m,n-th entry is the power of signal at time window n and frequency bin m. The assumption is that the observed signal is coming from a mixture of k static sound sources. Now each column of W can be interpreted as the average power spectrum of an audio source and each row of H can be interpreted as time-varying gain of a source. In practice the number of sources k is not known and we would like to also infer it from the data. This can be done by introducing an additional factor in the optimization problem which indicates the weight of a source in the mixture.

$$\begin{aligned} \underset{\mathrm{W}\in\mathbb{R}^{d\times d},\Theta\in\mathbb{R}^{d\times d},\mathrm{H}\in\mathbb{R}^{d\times n}}{\text{minimize:}} \quad & \|\mathrm{X}-\mathrm{W}\Theta\mathrm{H}\|_F^2+\lambda\|\Theta\|_1 \\ \text{subject to:} \quad & \mathrm{W}_{i,j}\geq 0, \mathrm{H}_{i,j}\geq 0, \Theta_{i,i}\geq 0, \Theta_{i\neq j}=0 \end{aligned} \tag{4}$$

In problem 4 Θ is introduced as the weight matrix for the mixture and an l_1 penalty is introduced to keep the number of "active" sources small. Such a NMF problem has been considered in [**?**] and a Bayesian approach is taken in solving it by specifying distributions over the elements of W, H and Θ. In our project

we directly try to solve a problem similar 4 with an additional penalty term which forces the columns of W to vary smoothly. To conclude the section we present the optimization problem:

$$\begin{aligned} \underset{\mathrm{W}\in\mathbb{R}^{d\times d},\Theta\in\mathbb{R}^{d\times d},\mathrm{H}\in\mathbb{R}^{d\times n}}{\text{minimize:}} \quad & \frac{1}{n}\left\|\mathrm{X}-\mathrm{W}\Theta\mathrm{H}\right\|_F^2+\lambda\left\|\Theta\right\|_1+\eta\sum_{i,j}\left(\mathrm{W}_{i,j}-\mathrm{W}_{i+1,j}\right)^2 \\ \text{subject to:} \quad & \mathrm{W}_{i,j}\geq 0,\mathrm{H}_{i,j}\geq 0,\Theta_{i,i}\geq 0,\Theta_{i\neq j}=0 \end{aligned} \tag{5}$$

2 Algorithm

Problem 5 is not a convex optimization problem, however, if one considers the 3 separate problems

$$\begin{aligned} \underset{\mathrm{W}\in\mathbb{R}^{d\times d}}{\text{minimize:}} \quad & \frac{1}{n}\left\|\mathrm{X}-\mathrm{W}\Theta\mathrm{H}\right\|_F^2+\eta\sum_{i,j}\left(\mathrm{W}_{i,j}-\mathrm{W}_{i+1,j}\right)^2 \\ \text{subject to:} \quad & \mathrm{W}_{i,j}\geq 0,\mathrm{H}_{i,j}\geq 0 \end{aligned} \tag{6}$$

$$\begin{aligned} \underset{\Theta\in\mathbb{R}^{d\times d}}{\text{minimize:}} \quad & \frac{1}{n}\left\|\mathrm{X}-\mathrm{W}\Theta\mathrm{H}\right\|_F^2+\lambda\left\|\Theta\right\|_1 \\ \text{subject to:} \quad & \Theta_{i,i}\geq 0,\Theta_{i\neq j}=0 \end{aligned} \tag{7}$$

$$\begin{aligned} \underset{\mathrm{H}\in\mathbb{R}^{d\times n}}{\text{minimize:}} \quad & \frac{1}{n}\left\|\mathrm{X}-\mathrm{W}\Theta\mathrm{H}\right\|_F^2 \\ \text{subject to:} \quad & \mathrm{H}_{i,j}\geq 0 \end{aligned} \tag{8}$$

each one is a convex optimization problem. What is more the objectives in 6 and 8 are differentiable and 7 is strongly convex assuming that $\mathrm{H}^\top\mathrm{W}$ is full rank. The proposed algorithm is now to solve each of the convex optimization problems separately in an alternating fashion. Pseudo code is given in algorithm 2.

Algorithm 1 Alternating minimization meta algorithm for problem 5

Input: $\mathrm{X},\mathrm{W}_0,\mathrm{H}_0,\Theta_0,\epsilon$
Output: $\mathrm{W}_T,\mathrm{H}_T,\Theta_T$
  while $\left\|\mathrm{W}_{t-1}\mathrm{H}_{t-1}\Theta_{t-1}-\mathrm{W}_t\mathrm{H}_t\Theta_t\right\|_F^2>\epsilon$ **do**

$$\mathrm{W}_{t+1}:=\underset{\mathrm{W}\in\mathbb{R}^{d\times d}}{\operatorname{argmin}} \quad \frac{1}{n}\left\|\mathrm{X}-\mathrm{W}\Theta_t\mathrm{H}_t\right\|_F^2+\eta\sum_{i,j}\left(\mathrm{W}_{i,j}-\mathrm{W}_{i+1,j}\right)^2$$

$$\text{subject to} \quad \mathrm{W}_{i,j}\geq 0,\mathrm{H}_{i,j}\geq 0$$

$$\mathrm{H}_{t+1}:=\underset{\mathrm{H}\in\mathbb{R}^{d\times n}}{\operatorname{argmin}} \quad \frac{1}{n}\left\|\mathrm{X}-\mathrm{W}_{t+1}\Theta_t\mathrm{H}\right\|_F^2$$

$$\text{subject to} \quad \mathrm{H}_{i,j}\geq 0$$

$$\Theta_{t+1}:=\underset{\Theta\in\mathbb{R}^{d\times d}}{\operatorname{argmin}} \quad \frac{1}{n}\left\|\mathrm{X}-\mathrm{W}_{t+1}\Theta\mathrm{H}_{t+1}\right\|_F^2+\lambda\left\|\Theta\right\|_1$$

$$\text{subject to} \quad \Theta_{i,i}\geq 0,\Theta_{i\neq j}=0$$

  end while

The main focus of our project is now to solve each of the problems 6,7,8 by using different algorithms explored in class, comparing our empirical observations with the derived convergence results. The algorithms we choose to compare are Projected Gradient/Subgradient Descent, Simple Dual Averaging and Augmented Lagrangian. For Projected Gradient/Subgradient Descent we both experiment with fixed step size and decreasing step size as $\frac{1}{t}$. We are also going to assume that all the minimizers of the above problems are in some compact set – it is not hard to imagine that this holds true, for example consider minimizing the objective in 6. If we let $\left\|\mathrm{W}\right\|_F$ go to infinity for fixed Θ,H and X the objective is going to go to infinity and

thus $\|\mathrm{W}\|_F$ must be bounded so we can assume that there exists optimal W^* is in some bounded closed ball with respect to the Frobenius norm. Thus we can restrict our attention on solving the optimization problems on the intersection of closed set with a compact set i.e. a compact set. Thus we can assume the existence of at least one minimizer of each of the optimization problems 6,7 and 8. A similar argument holds for Θ and H.

2.1 Subgradients for problems 6,7,8

If f denotes the respective objective of problems 6,7 and 8 then gradients and an element of the subdifferential of 7 is given by

$$\begin{aligned}
\nabla f(\mathrm{W}) &= \frac{2}{n}\left(\mathrm{W}\Theta\mathrm{H}-\mathrm{X}\right)\left(\Theta\mathrm{H}\right)^\top + \eta\tilde{\mathrm{W}} \text{ where} \\
\tilde{\mathrm{W}}_{i,j} &= 2\left(2\mathrm{W}_{i,j}-\mathrm{W}_{i+1,j}-\mathrm{W}_{i-1,j}\right), \\
\tilde{\mathrm{W}}_{1,j} &= 2\left(\mathrm{W}_{1,j}-W_{2,j}\right), \\
\tilde{\mathrm{W}}_{d,j} &= 2\left(\mathrm{W}_{d,j}-\mathrm{W}_{d-1,j}\right)
\end{aligned} \tag{9}$$

$$\nabla f(\mathrm{H}) = \frac{2}{n}\left(\mathrm{W}\Theta\right)^\top\left(\mathrm{W}\Theta\mathrm{H}-\mathrm{X}\right) \tag{10}$$

$$\partial f(\Theta) \ni \left(\frac{2}{n}\mathrm{W}^\top\left(\mathrm{W}\Theta\mathrm{H}-\mathrm{X}\right)\mathrm{H} + \lambda \mathrm{sign}\left(\Theta\right)\right)\odot \mathrm{I} \tag{11}$$

where $\odot$ denotes the Hadamard product and "sign" is the sign function applied element wise to Θ. The derivation in 11 holds because Θ is always constraint to be a diagonal matrix.

3 Projected Gradient Descent

3.1 Fixed step size

TODO: include experiments and comment on comparison with the theory

For this part of the project a modified version of **Algorithm 1** from lecture slides 4 is used with different choices of fixed step size α_k. The difference with the algorithm given in lecture 4 is the stopping criteria – as already discussed in class checking if the norm of the gradient is close to 0 will not work well for objectives including l_1 penalty term, instead we choose to stop our procedure either after a fixed number of steps (in our experiments this is 200 when solving problems 6 and 7 and 500 when solving problem 8) or if the distance between consecutive iterates becomes less than ϵ (where $\epsilon \in [10^{-4}, 10^{-5}]$). As discussed in class this is usually not a good stopping criteria unless the objective is differentiable with L-Lipschitz continuous derivatives. Luckily both the objectives in 6 and 8 are differentiable with Lipschitz continuous gradients which we show now.

Lemma 3.1. *The objective in problem 6 is differentiable with L-Lipschitz continuous gradients.*

Proof. Denote the objective in problem 6 by $f(\mathrm{W})$. Then $\nabla f(\mathrm{W}) = \frac{2}{n}\left(\mathrm{W}\Theta\mathrm{H}-\mathrm{X}\right)\left(\Theta\mathrm{H}\right)^\top + \eta\tilde{\mathrm{W}}$ where $\tilde{\mathrm{W}}_{i,j} = 2\left(2\mathrm{W}_{i,j}-\mathrm{W}_{i+1,j}-\mathrm{W}_{i-1,j}\right), \tilde{\mathrm{W}}_{1,j} = 2\left(\mathrm{W}_{1,j}-W_{2,j}\right), \tilde{\mathrm{W}}_{d,j} = 2\left(\mathrm{W}_{d,j}-\mathrm{W}_{d-1,j}\right)$. With this we have

$$\left\|\nabla f\left(\mathrm{W}_1-\mathrm{W}_2\right)\right\|_F = \left\|\frac{2}{n}\left(\left(\mathrm{W}_1-\mathrm{W}_2\right)\Theta\mathrm{H}\right)\left(\Theta\mathrm{H}\right)^\top + \eta\left(\tilde{\mathrm{W}_1}-\tilde{\mathrm{W}_2}\right)\right\|_F \leq \left(\frac{2}{n}\left\|\Theta\mathrm{H}\right\|_F^2 + 12\eta\right)\left\|\mathrm{W}_1-\mathrm{W}_2\right\|_F \tag{12}$$

where we used triangle inequality and bounded each of the $\left\|\left(\mathrm{W}_1\right)_{i,1:j}-\left(\mathrm{W}_2\right)_{i,1:j}\right\|_F \leq \left\|\mathrm{W}_1-\mathrm{W}_2\right\|_F$. □

The above lemma shows that the Lipschitz constant for the objective can indeed be very large as it depends on the product $\Theta\mathrm{H}$, however, in practice setting fixed step size $\alpha \leq 0.05$ seems to be in the range $(0, \frac{2}{L})$ which is when convergence for the algorithm is guaranteed. Sadly we can not guarantee strong convexity

or strict convexity for the objectives in 6 and 8 so the theorem which characterizes the best convergence rate is Theorem 1.9 in lecture slides 6. From our experiments we observe that our initial points W_0 and H_0 are roughly in the order of 10^3 and 10^5 from what we consider an optimal point and with $\alpha \sim 0.005$ we should have convergence roughly as $|f(\mathrm{W}_k) - f^*| \leq \frac{10^3}{0.005*k}$ and $|f(\mathrm{H}_k) - f^*| \leq \frac{10^5}{0.005*k}$. Here f denotes the respective objective function and f^* denotes the optimum objective value.

Surprisingly we can get linear convergence for 7 under mild assumptions that the matrix $\mathrm{H}^\top\mathrm{W}$ is full rank. Such a rate will follow from showing the next lemma.

Lemma 3.2. *Assume* $\mathrm{H}^\top\mathrm{W}$ *is full rank. Then the objective in 7 is strongly convex with strong convexity parameter* $\gamma < \frac{1}{n}\sigma_{min}(\mathrm{H})^2\sigma_{min}(\mathrm{W})^2$.

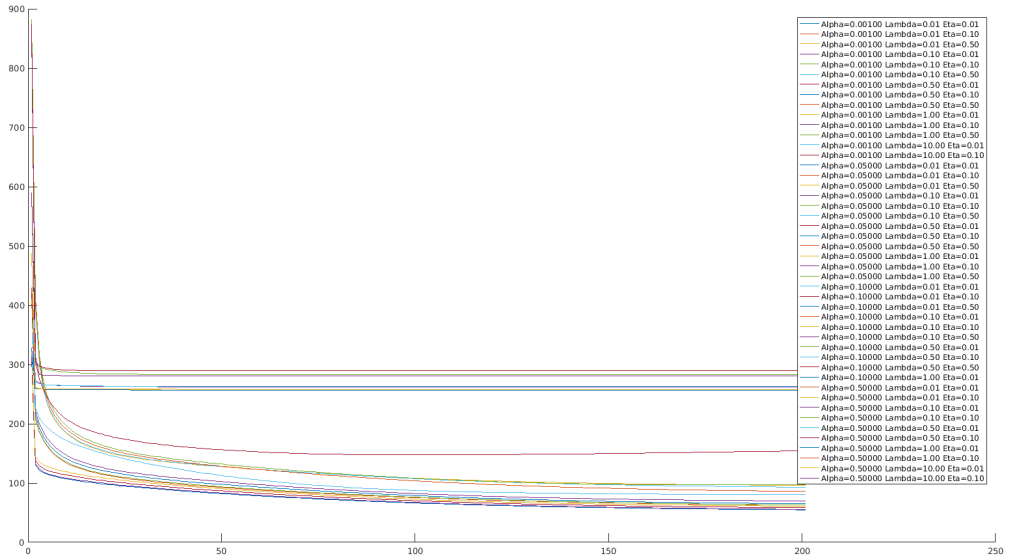
Proof. To show the objective in 7 is strongly convex we are going to show equivalently that $\frac{1}{n}\left\|\mathrm{W}\Theta\mathrm{H}\right\|_F^2$ is strongly convex under the given assumption. To do this we are going to use a second order condition for strong convexity i.e. the fact that the Hessian of the above function should be a positive definite form. Since the Hessian of $\frac{1}{n}\left\|\mathrm{W}\Theta\mathrm{H}\right\|_F^2$ is an order 4 tensor and we would not like to compute it we are going to use a little trick and vectorize $\mathrm{W}\Theta\mathrm{H}$. Let $vec(\mathrm{A})$ denote the vectorization of a matrix A by stacking its columns on top of each other. We use a famous equality $vec\left(\mathrm{W}\Theta\mathrm{H}\right) = \left(\mathrm{H}^\top \otimes \mathrm{W}\right) vec(\Theta)$ where $\otimes$ denotes the Kronecker product. If we denote $\mathrm{A} = \mathrm{H}^\top \otimes \mathrm{W}$ and $\mathrm{x} = vec(\Theta)$ then $\frac{1}{n}\left\|\mathrm{W}\Theta\mathrm{H}\right\|_F^2 = \frac{1}{n}\left\|\mathrm{Ax}\right\|_2^2$. The Hessian of $\frac{1}{n}\left\|\mathrm{Ax}\right\|_2^2$ equals $\frac{2}{n}\mathrm{A}^\top\mathrm{A}$. Now the strong convexity parameter of $\frac{1}{n}\left\|\mathrm{Ax}\right\|_2^2$ is characterized by the smallest singular value of the of $\mathrm{A}^\top\mathrm{A}$ which equals the smallest singular value squared of $\mathrm{A} = \mathrm{H}^\top \otimes \mathrm{W}$. From theorem 13.12 in [**?**] we know that the smallest singular value of $\mathrm{H}^\top \otimes \mathrm{W}$ is given by $\sigma_{min}(\mathrm{H})\sigma_{min}(\mathrm{W})$ which concludes the proof. □

Theorem 1.8 in lecture slides 4 now characterizes the linear convergence rate to a local neighborhood of the solution. To address our choice of stopping criteria, from Theorem 1.8 we know that $d(\Theta_{k+1}, \Theta^*)^2 < \alpha\frac{\kappa_g^2}{\gamma} + c^k d(\Theta_0, \Theta^*)^2$, where Θ^* is the optimal solution to 7, $c < 1$ depends on α and γ and κ_g is a bound on the norm of the elements in the sub-differential of the objective. For k large enough this implies that all of the Θ_k's are going to be contained in an open ball of fixed radius which is approximately $c^k d(\Theta_0, \Theta^*)^2$ – this implies that the distance between any two consecutive iterates $d(\Theta_k, \Theta_{k+1})^2$ is also going to be less than $\alpha\frac{\kappa_g^2}{\gamma}$. Since the convergence theory does not guarantee anything more, stopping our algorithm when $d(\Theta_k, \Theta_{k+1})^2$ becomes small enough seems acceptable. To be absolutely fair $d(\Theta_k, \Theta_{k+1})^2$ being small is only a necessary condition for convergence but not sufficient – it might happen that two consecutive iterates are close to each other, however, they are still not close to the optimal Θ^*. To alleviate this problem one might check that all the pair-wise distances between $\Theta_k, \Theta_{k+1}, ..., \Theta_{k+\tau}$ are small.

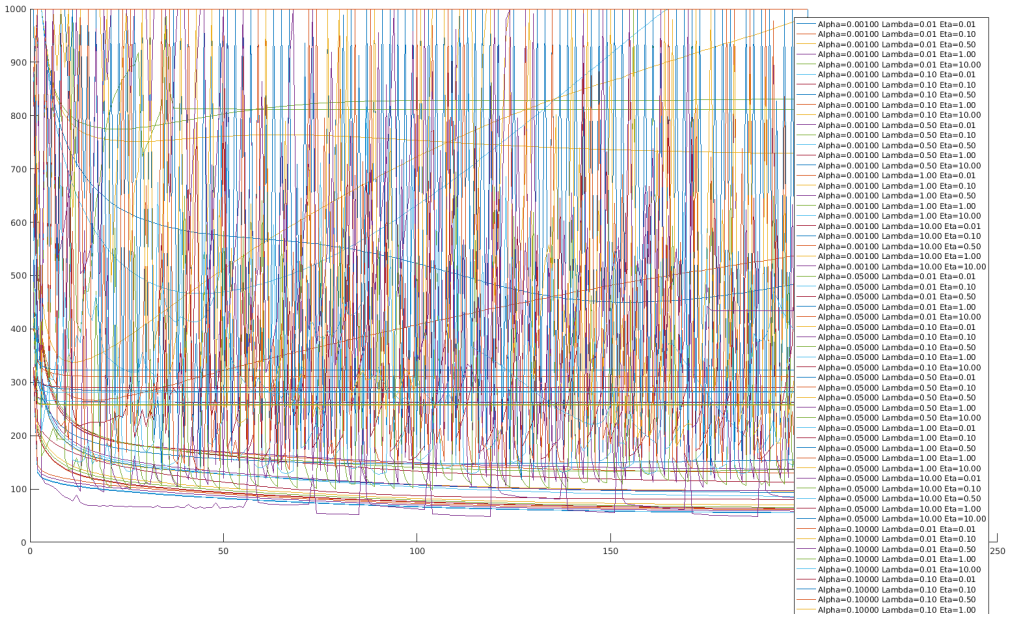
3.1.1 Fixed step size experiment results

All of our experiments are ran on a matrix $\mathrm{X} \in \mathbb{R}^{40\times 5000}$ constructed by sampling $\mathrm{W} \in \mathbb{R}^{40\times 40}, \Theta \in \mathbb{R}^{40\times 40}$ and $\mathrm{H} \in \mathbb{R}^{40\times 5000}$ from standard Gaussian and then thresholding so each of the matrices have non-negative entries. We also only keep the diagonal entries of Θ and finally set $\mathrm{X} := \mathrm{W}\Theta\mathrm{H}$. The number of non-zero entries in Θ is 18. We note that this is not the best construction of X we could have come up since neither Θ is very sparse nore is there any smoothness constraints on the columns of W. Thus as it is about to be shown the optimal solutions the the variants parametrized by η and λ of problem 5 we solve are not going to be close (in the sense of Frobenius norm) to the factors from which X was constructed. In fact for various settings of λ and η we achieve much better performance on the overall objective of 5 than the original W, Θ and H factors from which X was constructed. Our initializations for each of the tests is done in the same way the original factors of X were constructed. While this is by no means a good initialization we still see roughly the same performance of all of the methods we use for solving the sub-problems in 5 so initialization does not seem to play a major role in the final performance. We have performed several different experiments specifically for the projected gradient descent method with both increasing and decreasing step size. We begin by running each of the projected gradient descent algorithms on the separate optimization problems for 200 iterations when solving 6 and 7 and 500 iterations when solving 8. Each of these are ran a total of 250

times in the outer most loop of the alternating minimization algorithm. The respective plots are only given for every 10 iterations of the internal PGD loop. The parameters which yielded the smallest overall objective were $\lambda = 0.01$, $\eta = 0.01$ and $\alpha = 0.001$ was 54.1. This is no surprise as because X was not constructed with factors taking into account the smoothness and sparsity penalties when these penalties are enforced the least we should get the smallest loss. We chose to run the internal minimization projected gradient descent algorithm for only 200/500 iterations as the number of different values of λ, η and initial step size α we try together with the 200 iterations of the outer loop take quite a bit of time. The overall objective results can be seen in figure 1. The number of non-zero entrie sof the obtained Θ was 21.



(a) Projected gradient on experiments with well behaved results



(b) Projected gradient on all of our 64 hyperparameter experiments. (Or as we like to call it, modern art)

Figure 1: Experimental results from fixed step sized for projected gradient decent running $\alpha \in \{0.001, 0.05, 0.1, 0.5\}, \eta \in \{0.01, 0.1, 0.5, 1, 10\}, \lambda \in \{0.01, 0.1, 0.5, 1, 10\}$. For our experiments with larger step sizes, we observe that the obective values are not stable when plotting on the range of < 1000. However even looking on a large scale there still tend to be spikes with some of the methods when using larger η, λ as well. As with our experiments using other methods, we observe that we are able to find the lowest value of our objective when setting η and λ to small (0.01) values. This is expected since these behave as reguralarization on the matrices which we are able to find. Additionally, we note again that for $\lambda = \eta = 0$ we kown that there exists an exact solution to this problem.

Next we are going to plot the internal objective values for several different runs of projected gradient descent for each of the problems which optimize over H,Θ and W individually. We are going to restrict all our plots for the projected gradient descent section to the parameters which obtained the smallest overall objective on the experiments ran above. This is done since there are just too many combinations of parameters to plot for the number of different experiments we ran and there is no feasible way to choose meaningful results to present as most of the reasonable plots look very similar to each other. The detailed plots for H,Θ and W can be found in figure 2

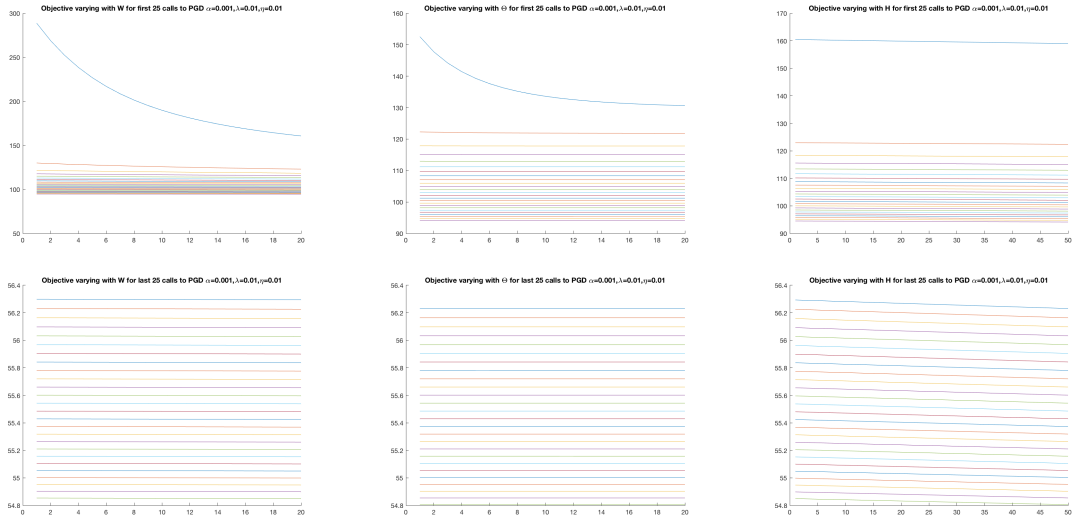


Figure 2: Detailed plots for each call to PGD with fixed step size for 6,7 and 8

We briefly comment on what is observed – first the objective between each call to PGD never increases which is very unexpected. This basically implies that the overall objective is always decreasing with every step of the suboptimization procedures. Secondly the behavior of the objectives for W and Θ is somewhat expected as both first decrease rapidly and at the final calls almost don't change. The more surprising observation referencing back to our theory discussion is the convergence rate for Θ. Since the objective is strongly convex we know that we should have linear convergence to a local neighbourhood of the optimal solution. The plots do not show this, in fact it seems that the objective for W converges quicker. Our final observation is that the objective for H does not change almost at all. We conjecture that this is dew to poor step size choice and that a step-size different step size choices for PGD applied to each of the 3 optimization sub-routines should fix the problem. Thus these are the next experiments we run. A final comment is that another interesting experiment to run would be to decrease the number of outer most loops of the alternating minimization procedure and increase the number innner loops for PGD for each problem so that the total number of iterations does not change and see if we observe the same overall objective. The next experiments we present something in between – we multiply the step-size α by 10/100 when doing PGD on the objective for H and 0.1/0.01 for Θ. We also increase the number of iterations to 1000 and 5000, however, the iterations of the outer most loop are kept to 200.

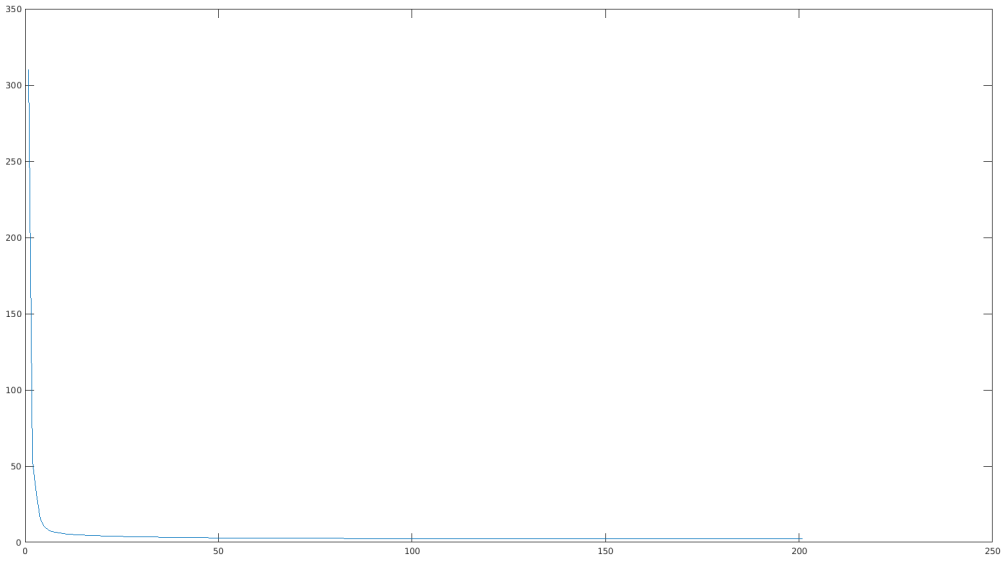


Figure 3: Overall objective for changing the alpha between different suboptimization problems (10 and 0.1)

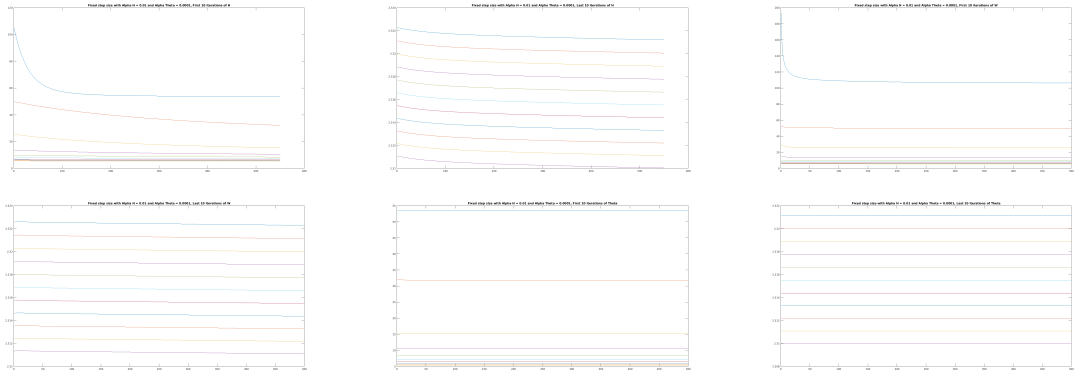


Figure 4: Plots of W, H, Θ optimized on the first and last 10 iterations of projected gradient decent with different values of α for each sub problem

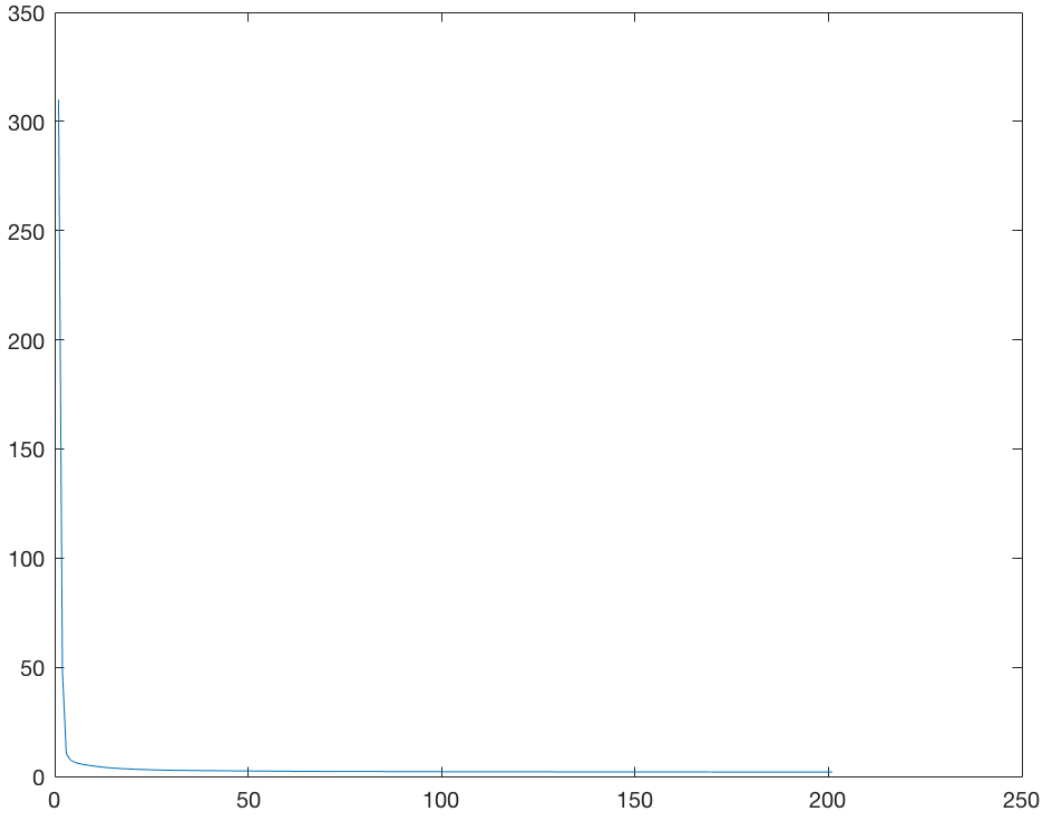


Figure 5: Overall objective for changing the alpha between different suboptimization problems (100 and 0.01)

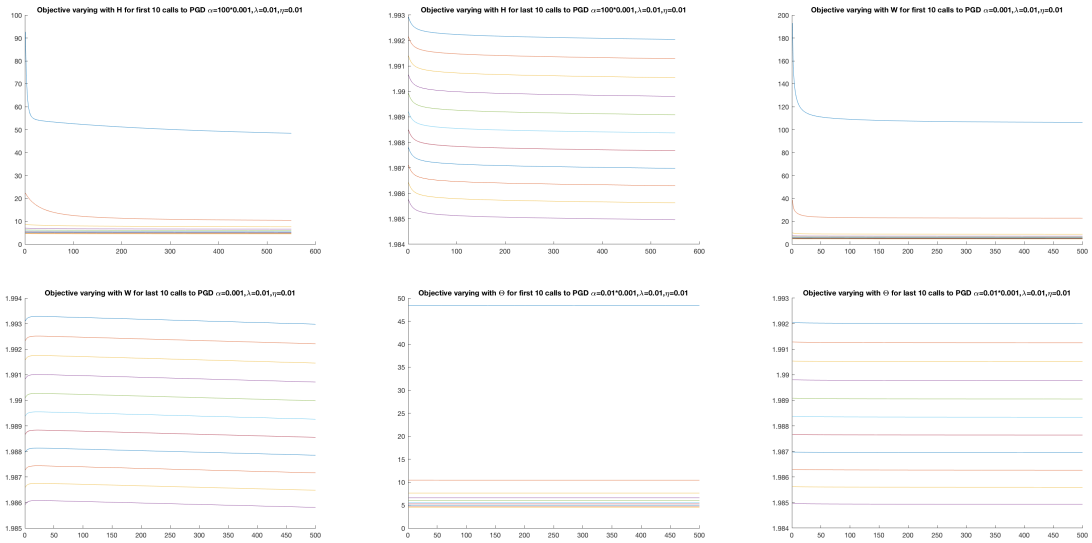


Figure 6: Plots of W, H, Θ optimized on the first and last 10 iterations of projected gradient decent with different values of α for each sub problem

The lowest objective value for 5 we get when running the 5000 itertations per suboptimization problem with rescaling factors 10 and 0.1 is 2.31 and with rescaling factors 100 and 0.01 is 1.985. The sparsity for Θ was 16 for the 100 and 0.01 rescaling factors. As we can see both the rescaling and not keeping step-size constant across optimization problems helps with improving the objective significantly. This mainly seems to be due to the objective for 8 actually decreasing at a nice rate when the step size for PGD was increased. One might think that this suggests that we should have anyway increased the fixed step-sizeas 0.001 is too small, however, this is not the case as we can see from the orange line in the first plot of 1. Technically one can still make the argument that the plot for the larger step size was done only for a few iterations and if we can increase the number of iterations it might outperform the smaller step size. Finally we observe that after decreasing the step-size, PGD makes almost no progress at any of the applications to 7. It is probably the case that the step size is too small. We also ran the same experiment with 1000 iterations per suboptimization problem and the best overall objective we founds for rescaling factors 10 and 0.1 was 55.49 and for rescaling factors 100 and 0.01 was 8.7182 with sparsity for Θ 17. We conclude that changing the step size significantly improves the quality of the best solution overall for the objective in 5. We also ran an experiment where we stopped each of the PGD algorithms after distance between consecutive iterates became smaller than $5 * 10^{-5}$. The best objective value we found was 54.8 and the sparsity of Θ was 21. We make a final remark that for some of the parameter choices as can be seen from the second plot in figure 1 the gradient updates become very unstable with gradient norms growing very quickly to infinity. This happened mainly when at least one of the regularizer paramaters was large together with a moderate to large value of the fixed step-size.

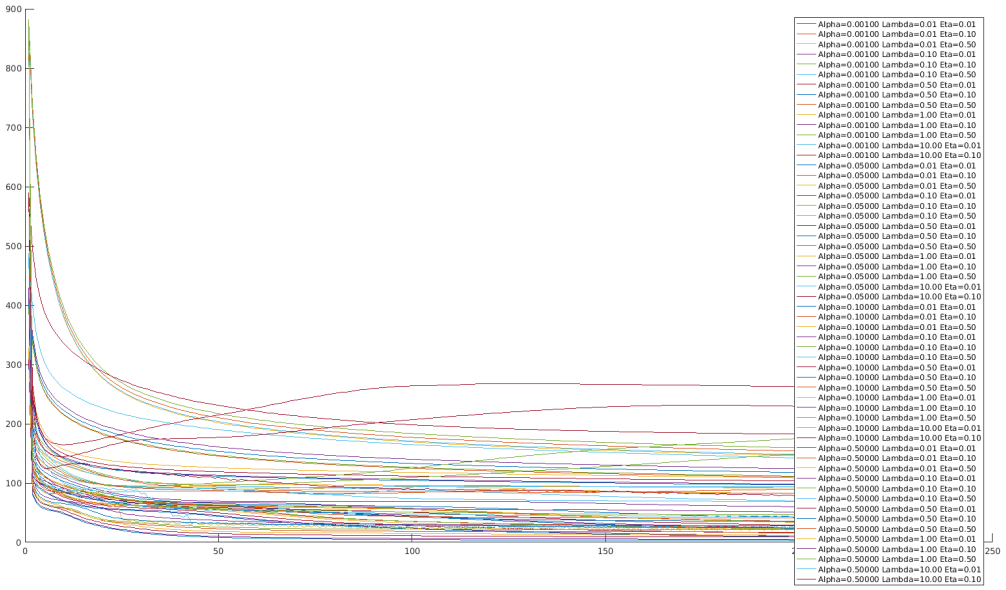
3.2 Decreasing step size

None of the convergence results in lecture slides 6 hold any longer for projected subgradient descent with decreasing step size. However, we can still characterize the convergence in terms of objective and iterates for step size decreasing as $\frac{1}{t}$. From Theorem 1.11 in lecture slides 4 we know that the iterates for projected gradient descent for problems 6,7 and 8 will converge to an optimal point (given such exists). For 6 and 8 we can use Lemma 1.3 in lecture slides 6 to show that this would imply convergence in objective. From Lemma 1.3 in lecture slides 6 we know that if f has an L-Lipschitz continuous gradient we have $f(x_k) - f(x^*) \leq \langle \nabla f(x^*)(x_k - x^*) \rangle + \frac{L}{2} \|x_k - x^*\|^2$. Using Cauchy-Schwartz inequality we have $\nabla f(x^*)^\top (x_k - x^*) \leq \|\nabla f(x^*)\| \|x_k - x^*\|$

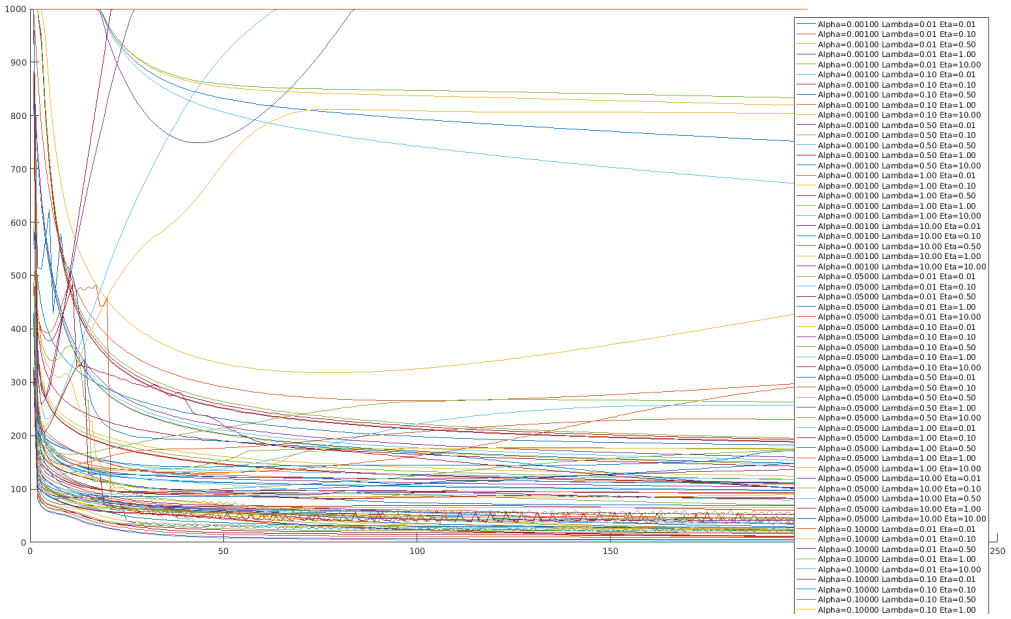
thus $f(x_k) - f(x^*) \leq \|x_k - x^*\| \left(\|\nabla f(x^*)\| + \frac{L}{2}\|x_k - x^*\|\right)$ and since $\|\nabla f(x^*)\|$ is bounded this shows convergence in objective. By triangle inequality we have $\|x_k - x^*\| \leq \|x_k - x_{k+1}\| + \|x_{k+1} - x^*\|$ and thus $f(x_k) - f(x^*) \leq \|x_k - x_{k+1}\| \left(\|\nabla f(x^*)\| + \frac{L}{2}\|x_k - x^*\|\right) + \|x_{k+1} - x^*\| \left(\|\nabla f(x^*)\| + \frac{L}{2}\|x_k - x^*\|\right) \leq \|x_k - x_{k+1}\| c_1 + \tilde{\epsilon}$ where $c_1 = \|\nabla f(x^*)\| + \frac{L}{2}\|x_k - x^*\|$ and $\tilde{\epsilon} = \|x_{k+1} - x^*\| \left(\|\nabla f(x^*)\| + \frac{L}{2}\|x_k - x^*\|\right)$. Clearly for k large enough $\tilde{\epsilon}$ is as small as we would like and c_1 is bounded thus for k large enough $\|x_{k+1} - x_k\| < \epsilon$ implies $f(x_k) - f(x^*)$ is small. This should somewhat justify our stopping criteria for 6 and 8. None of the above derivations, however, hold for 7 in fact the only thing we can argue is that the iterates Θ_k are going to converge to Θ^*. This is quite disappointing compared to the results we were able to obtain for a constant step size. In practice as we can see from the experiments we still get satisfactory results.

3.2.1 Decreasing step size experiment results

We run the same experiments for decreasing step size as we did for fixed step size. The step size decreases at rate $\frac{1}{\sqrt{k}}$. Contrary to the worse theoretical guarantees which we have for PGD with decreasing step size for our objectives in 6,7 and 8 we see much better results even when each internal optimization loop is ran for 200/500 iterations. The best resuts were obtained with intial step-size $\alpha = 0.5$ and parameters for the smoothness and sparseness penalties $\eta = 0.01$ and 0.01. We have already discussed why these smoothness and sparseness parameters perform best. The overall objective decreased to 3.26 and the sparsity of Θ was 13.



(a) Projected gradient on experiments with well behaved results for decreasing step size



(b) Projected gradient on all of our 64 hyperparameter experiments for decreasing step size.

Figure 7: Experimental results from decreasing step sized for projected gradient decent running $\alpha \in \{0.001, 0.05, 0.1, 0.5\}, \eta \in \{0.01, 0.1, 0.5, 1, 10\}, \lambda \in \{0.01, 0.1, 0.5, 1, 10\}$. With the decreasing step size, we observe that the objectives of even our worst behaved functions are still much smother then in the fixed step size case. We note that in the fixed step size case, that once $\alpha < 0.1$ for large λ and η most of the objectives became well behaved, and thus for the decreasing step size case, we have that even with $\alpha_0 = 0.5$, we are reaching these smaller values of α within 25 steps.

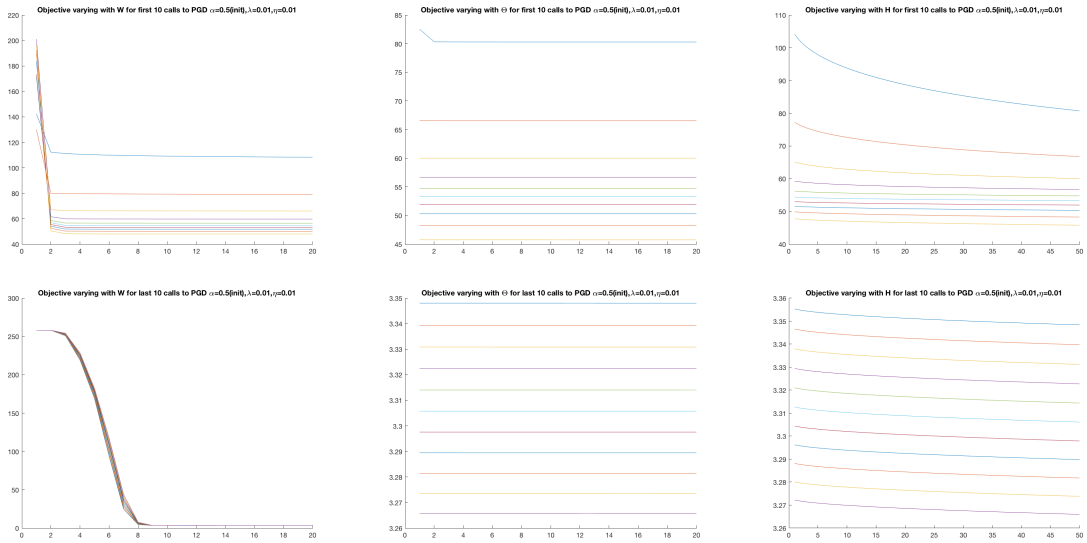


Figure 8: Plots of W, H, Θ optimized on the first and last 10 iterations of projected gradient decent (decreasing step size as $\frac{1}{\sqrt{k}}$

From figure 8 we can see a couple of differences compared to the fixed step size case – first the objective for H decreases a lot more and smoothly, secondly the objective for Θ stays almost the same and finally during the last 10 calls to PGD with decreasing step size to solve problem 6 the objective for W keeps restarting from the same place. We also notice that the objective for W decreases much quicker than for the fixed step size which suggests convergence rate faster than $\frac{1}{\sqrt{k}}$ which are the only guarantees we have from the theory we studied.

4 Simple Dual Averaging

In this section we compare the SDA given in lecture slides 4 as **Algorithm 3**. We also address details in the implementation, convergence theory and stopping criteria used.

4.1 Algorithm and implementation

Lower case bold letters denote matrices (contrary to standard convention) and the norm is the Frobenius norm together with the associated standrard inner product for matrices. We follow the pseudo-code given in **Algorithm 3** in the lecture slides as already stated. The most interesting part of the algorithm is implementing the update $\mathrm{x}_{k+1} \leftarrow \arg\min_{\mathrm{x}\in\mathcal{X}} \langle \mathrm{x}, \mathrm{s}_{k+1}\rangle + \frac{\beta_{k+1}}{2} \|\mathrm{x}-\mathrm{x}_0\|^2$ for problems 6,7 and 8. The following lemma shows us that this is equivalent to the projection of $x_0 - \frac{1}{\beta_{k+1}}\mathrm{s}_{k+1}$ onto the convex set $\mathcal{X}$.

Lemma 4.1. *Solving* $\operatorname{argmin}_{\mathrm{x}\in\mathcal{X}} \langle \mathrm{x}, \mathrm{s}_{k+1}\rangle + \frac{\beta_{k+1}}{2} \|\mathrm{x}-\mathrm{x}_0\|^2$ *is equivalent to solving* $\operatorname{argmin}_{\mathrm{x}\in\mathcal{X}} \left\|\mathrm{x} - (\mathrm{x}_0 - \frac{1}{\beta_{k+1}}\mathrm{s}_{k+1})\right\|^2$

Proof.

$$\begin{aligned}
\arg\min_{\mathrm{x}\in\mathcal{X}} \langle \mathrm{x}, \mathrm{s}_{k+1}\rangle + \frac{\beta_{k+1}}{2}\left\|\mathrm{x}-\mathrm{x}_0\right\|^2 = \\
\arg\min_{\mathrm{x}\in\mathcal{X}} \langle \mathrm{x}, \mathrm{s}_{k+1}-\beta_{k+1}\mathrm{x}_0\rangle + \frac{\beta_{k+1}}{2}\left\|\mathrm{x}\right\|^2 = \\
\arg\min_{\mathrm{x}\in\mathcal{X}} \langle \mathrm{x}, \mathrm{s}_{k+1}-\beta_{k+1}\mathrm{x}_0\rangle + \frac{\beta_{k+1}}{2}\left\|\mathrm{x}\right\|^2 + \frac{1}{2\beta_{k+1}}\left\|\mathrm{s}_{k+1}-\beta_{k+1}\mathrm{x}_0\right\|^2 = \\
\arg\min_{\mathrm{x}\in\mathcal{X}} \frac{\beta_{k+1}}{2}\left\|\mathrm{x}-(\mathrm{x}_0-\frac{1}{\beta_{k+1}}\mathrm{s}_{k+1})\right\|^2 = \\
\arg\min_{\mathrm{x}\in\mathcal{X}} \left\|\mathrm{x}-(\mathrm{x}_0-\frac{1}{\beta_{k+1}}\mathrm{s}_{k+1})\right\|^2
\end{aligned} \tag{13}$$

□

For the set $\mathcal{X} := \{\mathrm{x}_{i,j} \geq 0\}$ it is easy to verify that the solution to $\arg\min_{\mathrm{x}\in\mathcal{X}} \left\|\mathrm{x}-(\mathrm{x}_0-\frac{1}{\beta_{k+1}}\mathrm{s}_{k+1})\right\|^2$ is exactly given by the operator $\mathcal{P}(\mathrm{x}) = \tilde{\mathrm{x}}$ where

$$\mathcal{P}(\mathrm{x})_{i,j} = \begin{cases} \mathrm{x}_{i,j} & \mathrm{x}_{i,j} \geq 0 \\ 0 & \text{otherwise} \end{cases}.$$

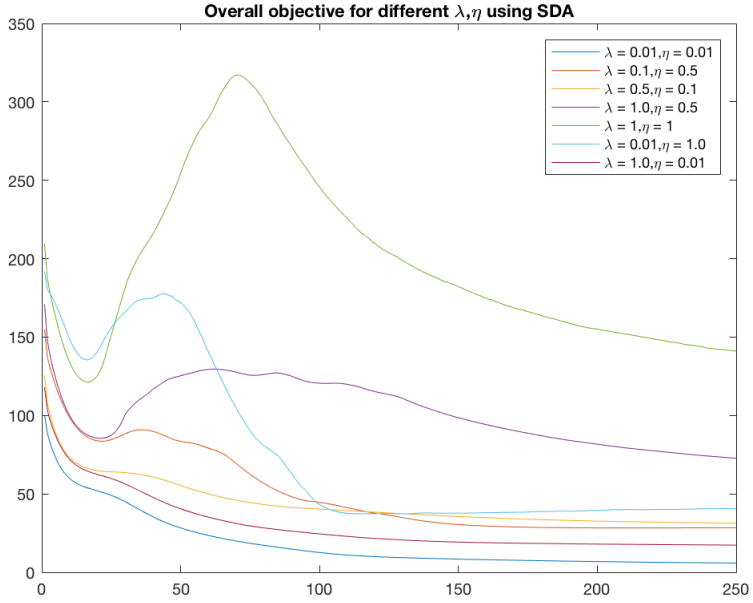
A simple way to see this is to form the Lagrangian $\mathcal{L}(\mathrm{x},\mathrm{y})$ for $\min_{\mathrm{x}\in\mathcal{X}} \frac{1}{2}\left\|\mathrm{x}-(\mathrm{x}_0-\frac{1}{\beta_{k+1}}\mathrm{s}_{k+1})\right\|^2$ and notice that the pair $(\mathcal{P}(\mathrm{x}_0-\frac{1}{\beta_{k+1}}\mathrm{s}_{k+1}), \mathrm{y}^*)$ satisfies KKT conditions. Here y^* is given by

$$\mathrm{y}^*_{i,j} = \begin{cases} 0 & (\mathrm{x}_0-\frac{1}{\beta_{k+1}}\mathrm{s}_{k+1})_{i,j} \geq 0 \\ -(\mathrm{x}_0-\frac{1}{\beta_{k+1}}\mathrm{s}_{k+1})_{i,j} & \text{otherwise} \end{cases}$$

All other steps of the algorithm are as given in the lecture slides and the β_k's are chosen according to Theorem 1.15 in lecture slides 4.

4.1.1 SDA experiment results

Even though we are given a reliable stopping criteria for this algorithm we are not able to implement it as it requires computation of the conjugate function of the objective which we do not know how to do except for problem 8. One can argue that an iterative method for approximately computing the conjugate functions and therefore the stopping criteria might give sufficient results, however, the overall run-time to obtain a solution to optimization problem 5 is long enough without having to solve an additional optimization problem at each step of the sub-problem optimization routines. In the light of these concerns we choose to stop each SDA routine either after a fixed number of iterations or when consecutive iterates become ϵ close to each other. From Theorem 1.17 in lecture slides 4 we know that to get ϵ-suboptimality for the optimization problem we would need the order of $\frac{1}{\epsilon^2}$ iterations, where the constant depends on an upper bound on the gradients and the radius of the ball centered at an optimal solution containing the initialization. From our experiments the norm of the gradients are bounded by roughly ~ 100 and the norm of the difference between initializations and the best solution we manage to recover is roughly ~ 600. The number of iterations for which we run our algorithms are 200, 1000 and 5000 with additional 500 iterations for optimization problem 8. The initializations are the same as for the experiments in projected gradient descent. As before we choose to run our λ and η experiments for only 200 iterations per alternating minimization step for problems 6 and 7 and for 500 iterations for problem 8 in the interest of time. As before we run the alternating minimization algorithm 2 for 200 iterations.



We see that the smallest objective is obtained at $\eta = 0.01, \lambda = 0.01$ – this is no surprise as already discussed since the original matrix X was not obtained as a product of matrices which are sparse or smooth. We see that the smoothness constraint is what seems to hurt the overall objective the most and might be the reason why we see a peak of the objective around the 100 iteration of the alternating minimization. We also see that the sparsity constraint does not hurt the objective too much. To compare the rank of Θ we recover for $\lambda = 1, \eta = 0.01$ is 32 while when $\lambda = 0.01, \eta = 0.01$ it is 37. To remind the reader the rank of X is 18. The next figures show how the objective decreases at each alternating minimization step of 200/500 iterations for different values of λ, η. In each figure we plot how the objective in 5 changes for a complete run of the SDA procedure to solve problem 6,7 or 8. We do this for the first 25 and the last 25 calls to the SDA algorithm. The objective value is only computed every 10 iterations.

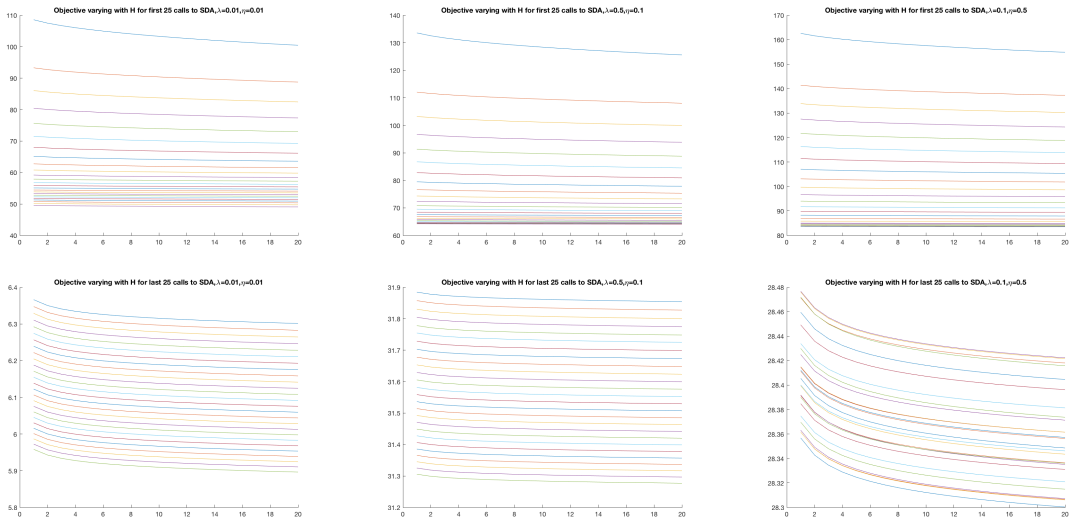


Figure 10: Plots of the overall objective as problem 8 is being solved by SDA

In figure 10 we show the objective as H is being optimized over. As it is evident from the plots the objective does not change too much with H for later calls to SDA. This is consistent with the $\mathcal{O}(\frac{1}{\epsilon^2})$ convergence rate as the constant (by constant we refer to the quantity $D + \frac{k_g^2}{2}$ in the proof of Theorem 1.17 in lecture slides 4) depends on the norm of the gradient of problem 8 (as already discussed) which in this case is around 900.

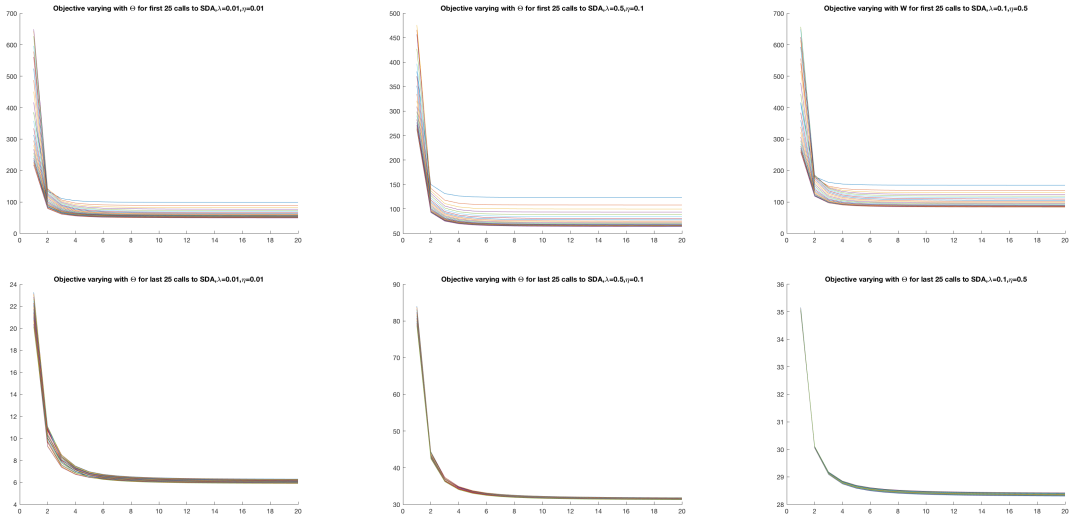


Figure 11: Plots of the overall objective as problem 7 is being solved by SDA

Figure 11 contains similar experiments for Θ, however, we observe something interesting – unlike for H we see that at each consecutive call to SDA in the alternating minimization algorithm the objective for Θ goes

up. This as a whole should not be a surprise as the objective in 5 is not jointly convex in Θ, H and W so we do not expect that an alternating minimization approach is always going to decrease the objective at each call. The more interesting observation here is the rate at which the objective decreases in the SDA algorithm – it looks like it almost decreases at a linear rate which is unexpected as the theory suggest it should have sublinear behavior. One can argue that this might be related to the constant in the $\mathcal{O}(\frac{1}{\epsilon^2})$ being small for 7 – from our experiments it is around 120.

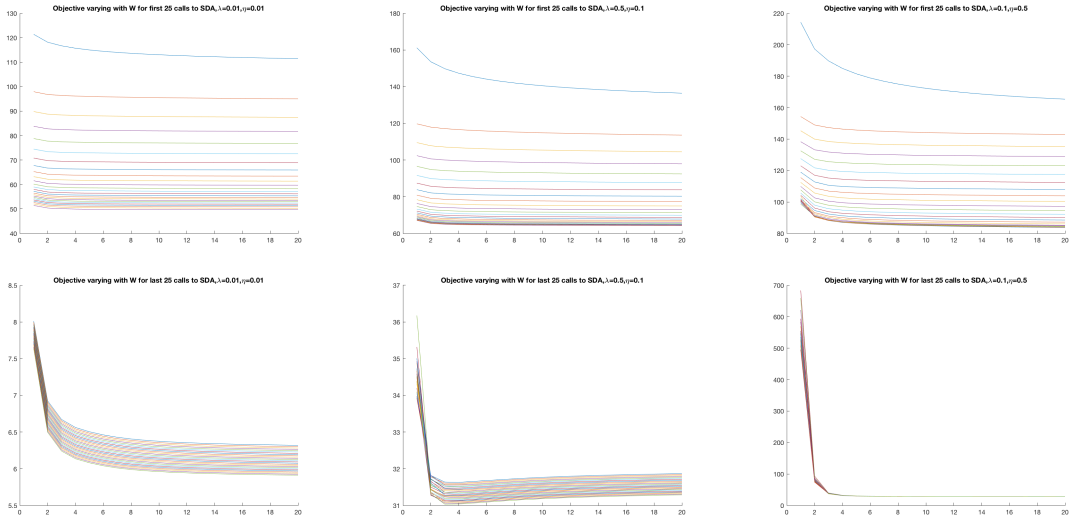


Figure 12: Plots of the overall objective as problem 6 is being solved by SDA

Finally in figure 12 we show how the objective changes when solving 6. For early runs of SDA we see similar behavior to what is happening when solving 8 with slightly faster convergence. This is within our expectations as the constant hidden in $\mathcal{O}(\frac{1}{\epsilon^2})$ is around 600. For later applications (the last 25) we see behavior similar to 7. Our conjecture is that the norm of the gradients becomes much smaller as W_t approaches the overall optimal W^* and thus the constant $D + \frac{k_g^2}{2}$ since D naturally decreases as W_t converges to W^*.

Even though we manage to get good performance in terms of convergence (compared to our best overall result) we still only use a small amount of iterations when solving each of the problems 6,7,8 so we have no guarantees to be close to the optimal solution. To address this problem we run another two experiments in which the the number of iterations each SDA is ran for is 1000 and 5000. The number of iterations for the outer-most loop of the alternating optimization algorithm is still 250. We only run these experiments for $\eta = 0.01$ and $\lambda = 0.01$ as they are very time-consuming.

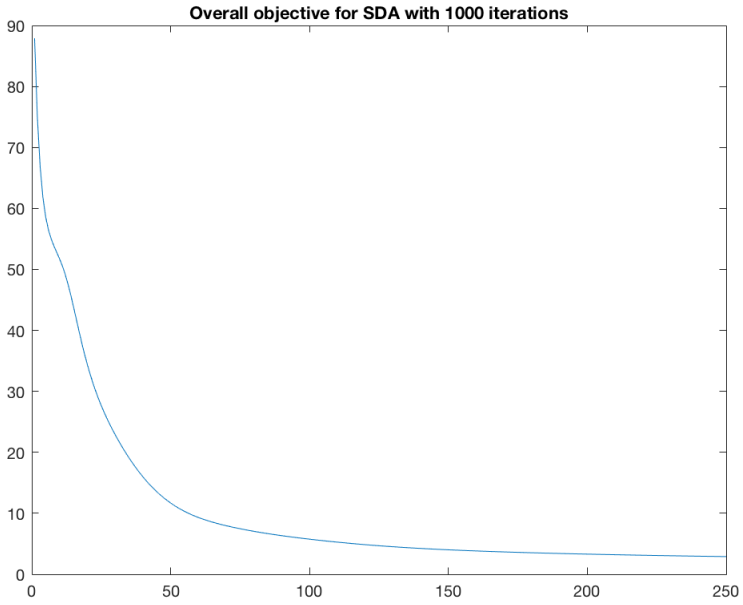


Figure 13: Overall objective when SDA is ran for 1000 iterations.

The overall objective when SDA is ran for 1000 iterations at each step is plotted in figure 13. The best value achieved is 2.9022. Comparing this to the best value achieved when SDA was ran for only 200 iterations – 5.8968, it is better. In figure 14 the individual runs are plotted for W, Θ and H.

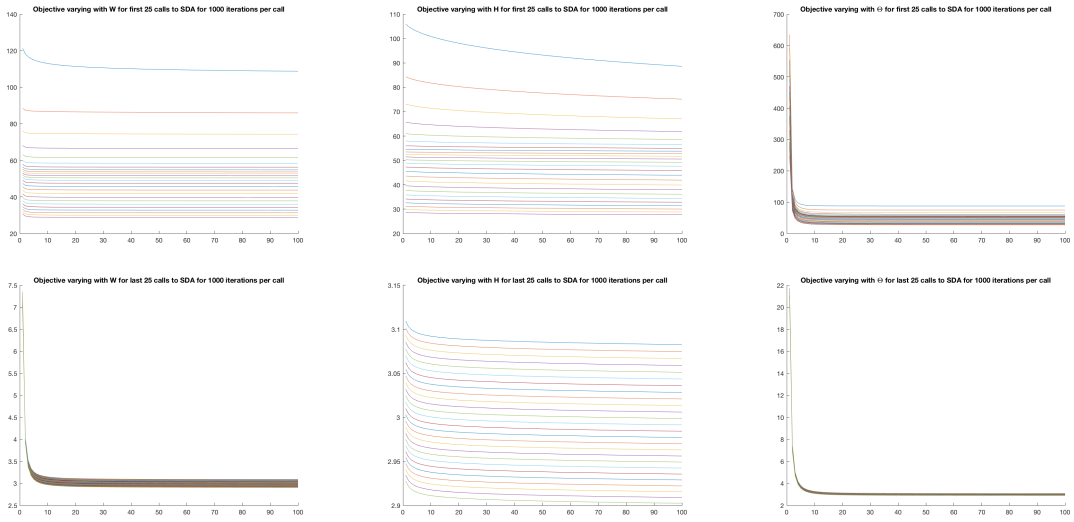


Figure 14: Plots of objective as W, Θ and H are being optimized over with SDA for 1000 iterations.

All the plots look very similar to the ones where SDA is only ran for 200 iterations in terms of how quickly the objective converges. The main difference seems to stem from the fact that after each individual call to SDA with 1000 iterations the obtained solution has a slightly lower objective value compared to the SDA with 200 iterations. These differences seem to diminish over the course of the 250 iterations of the outer most loop and we conjecture that if we ran the outer loop for more than 250 we would get about the same results. Comparing the run-time of both SDA with 200 iterations and SDA with 1000 iterations the gain is almost insignificant. We do not expect running SDA for 5000 iterations per optimization problem is going to perform much better. The plots containing the experiments in which SDA is ran for 5000 iterations can be seen in figures 15 and 16. The best value we get for the overall objective is 2.0425.

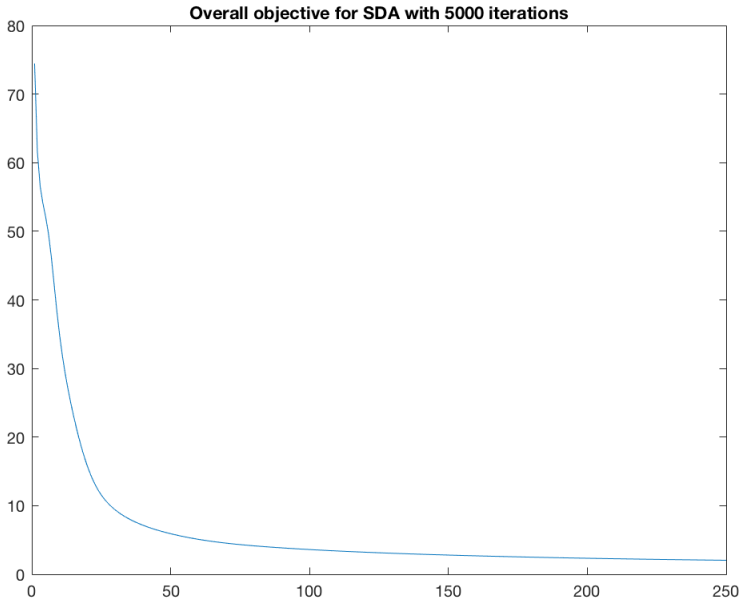


Figure 15: Overall objective when SDA is ran for 5000 iterations.

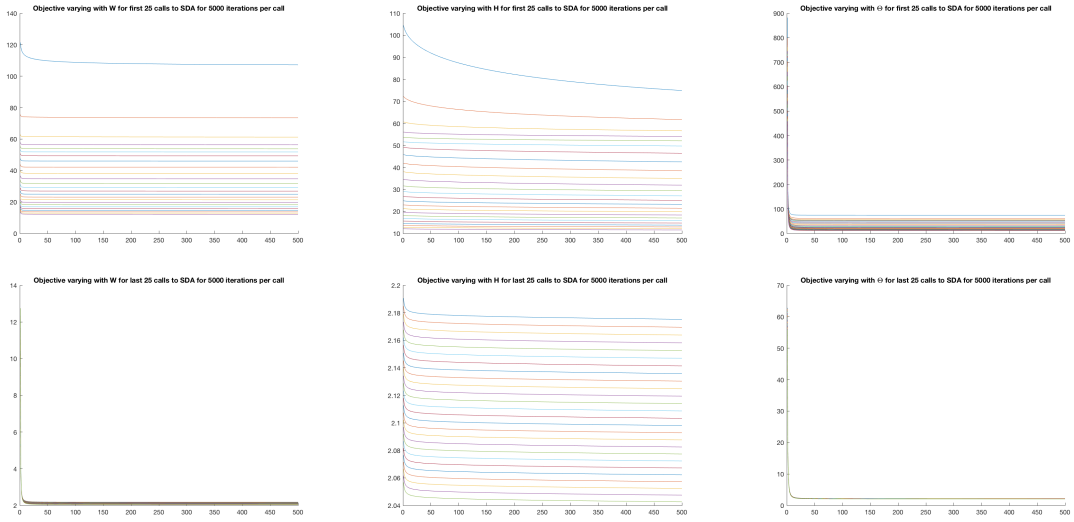


Figure 16: Plots of objective as W, Θ and H are being optimized over with SDA for 5000 iterations.

For a final experiment on stopping criteria we ran SDA stopping the algorithm if the distance between two consecutive iterates became less than 10^{-4} (or since at early stages of the overall optimization procedure it took too many iterations we stopped it after the first 5000 iterations). We noticed that after the first 50 iterations of the overall alternating minimization procedure the SDA algorithm was stopping before the 5000 iterations expired. The result for the overall objective at the end was 2.8784 which is slightly better than just stopping SDA after 1000 iterations and worse than stopping SDA after 5000 iterations. This implies that stopping when iterates do not change too much between iterations is not enough to guarantee better overall performance. The only plot we provide 17 is for the overall objective as the plots for the separate problems are almost identical to the ones for SDA with 1000 iterations.

To conclude this section – we did not observe anything unexpected considering theory of convergence of SDA. The most surprising results were in the convergence speed of Θ but this can still be explained by the smaller distance between initialization and optimal Θ and smaller norm of the gradients for the objective. We also observed that for Θ and at later iterations of the alternating minimization procedure for W that a lot of computational work is wasted as each optimization subproblem starts almost at the same objective value as the previous.

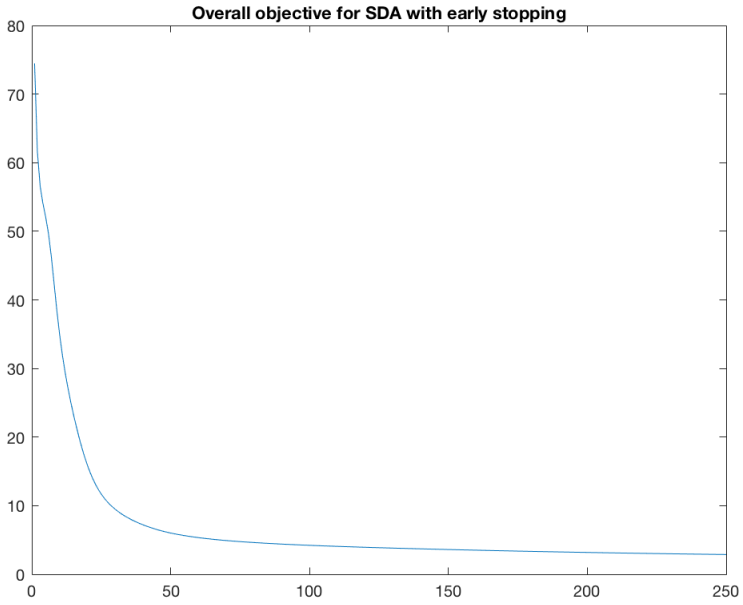


Figure 17: Overall objective when SDA is stopped after small difference in iterates

5 Augmented Lagrangian

The final method we consider is augmented Lagrangian. First, we take the Lagrangian of our objective

$$\mathcal{L}(W,\Theta,H) = ||X - W\Theta H||_F^2 + \lambda||\Theta||_1 + \eta\sum_{i,j}(\mathrm{W}_{i,j} - \mathrm{W}_{i+1,j})^2 - \sum_{i=1}^{m}\sum_{j=1}^{m}\mu_{ij}W_{ij} - \sum_{i=1}^{m}\sum_{j=1}^{n}\nu_{ij}H_{ij} - \sum_{i=1}^{n}\xi\Theta_{ii}. \tag{14}$$

The augmented Lagrangian, in contrast, considers the following objective

$$\begin{aligned}\mathcal{L}_{aug}(W,\Theta,H) = ||X - W\Theta H||_F^2 + \lambda||\Theta||_1 + \eta\sum_{i,j}(\mathrm{W}_{i,j} - \mathrm{W}_{i+1,j})^2 - \sum_{i=1}^{m}\sum_{j=1}^{m}\max\left(0, -W_{ij} + \frac{\mu_{ij}}{\rho}\right)^2 \\ - \sum_{i=1}^{m}\sum_{j=1}^{n}\max\left(0, -H_{ij} + \frac{\nu_{ij}}{\rho}\right)^2 - \sum_{i=1}^{n}\max\left(0, \Theta_{ii} - \frac{\xi}{\rho}\right)^2.\end{aligned} \tag{15}$$

We then proceed to optimize $\mathcal{L}_{aug}$ with respect to each of the parameters W, Θ, H separately. We note that the objective is not jointly convex in all three, but convex in each individual subproblem. We employ the algorithm discussed in Chapter 4 of Birgin (2014) [**?**].

Algorithm 2 Alternating minimization Augmented Lagrangian algorithm for problem 5

Input: $\mathrm{X}, \mathrm{W}_0, \mathrm{H}_0, \Theta_0, \epsilon$
Output: $\mathrm{W}_T, \mathrm{H}_T, \Theta_T$
  $k \leftarrow 0$
  while $\|\mathrm{W}_{t-1}\mathrm{H}_{t-1}\Theta_{t-1} - \mathrm{W}_t\mathrm{H}_t\Theta_t\|_F^2 > \epsilon$ **do**
    for $k = 1, \ldots, K$ **do**
      $W_{t+1} := \underset{W \in \mathbb{R}^{d\times d}}{\operatorname{argmin}} \frac{1}{n} \|\mathrm{X} - \mathrm{W}\Theta_t\mathrm{H}_t\|_F^2 + \eta \sum_{i,j} \left(\mathrm{W}_{i,j} - \mathrm{W}_{i+1,j}\right)^2 + \sum_{i,j} \max\left(0, -W_{ij} + \frac{\mu_{ij}^k}{\rho^k}\right)^2$
      $\mu_{ij}^{k+1} \leftarrow \left[\mu_{ij}^k - \rho_k W_{ij}\right]_+, \forall i, j$
      $V_{ij}^{\mu^k} = \min\left\{W_{ij}, \mu_{ij}^k/\rho^k\right\}, \forall i, j$
      $\mu_{ij}^{k+1} = \Pi_{\mu_{ij}^k}\left([0, \mu_{\max}]\right), \forall i, j$
    end for
    $W_{t+1} = \mathbf{Proj}_{\geq 0}(W_{t+1})$
    for $k = 1, \ldots, K$ **do**
      $H_{t+1} := \underset{H \in \mathbb{R}^{d\times n}}{\operatorname{argmin}} \frac{1}{n} \|\mathrm{X} - \mathrm{W}_{t+1}\Theta_t\mathrm{H}\|_F^2 + \sum_{i,j} \max\left(0, -H_{ij} + \frac{\eta_{ij}}{\rho^k}\right)^2$
      $\eta_{ij}^{k+1} \leftarrow \left[\eta_{ij}^k - \rho_k H_{ij}\right]_+, \forall i, j$
      $V_{ij}^{\nu^k} = \min\left\{H_{ij}, \nu_{ij}^k/\rho^k\right\}, \forall i, j$
      $V_i^{\xi^k} = \min\left\{\Theta_{ii}, \xi_{ii}^k/\rho^k\right\}, \forall i, j$
    end for
    $H_{t+1} = \mathbf{Proj}_{\geq 0}(H_{t+1})$
    for $k = 1, \ldots, K$ **do**
      $\Theta_{t+1} := \underset{\Theta \in \mathbb{R}^{d\times d}}{\operatorname{argmin}} \frac{1}{n} \|\mathrm{X} - \mathrm{W}_{t+1}\Theta\mathrm{H}_{t+1}\|_F^2 + \lambda||\Theta||_1 + \sum_i \max\left(0, -\Theta_{ii} + \frac{\xi_i^k}{\rho^k}\right)^2$
      $\xi_i^{k+1} \leftarrow \left[xi_i^k - \rho_k \Theta_{ii}\right]_+, \forall i$
      $\nu_{ij}^{k+1} = \Pi_{\nu_{ij}^k}\left([0, \nu_{\max}]\right), \forall i, j$
      $\xi_i^{k+1} = \Pi_{\xi_i^k}\left([0, \xi_{\max}]\right), \forall i$
    end for
    $\Theta_{t+1} = \mathbf{Proj}_{\geq 0}(\Theta_{t+1})$
  end while

We perform the unconstrained optimization in Algorithm 2 with L-BFGS [**?**], using the standard SciPy implementation. We set $\mu_{\max}, \nu_{\max}, \xi_{\max}$ to 100. We note that the function $\max(0, \cdot)^2$ is differentiable in x. To handle the L_1 penalty of Θ, we use the projected L-BFGS method described in Schmidt 2009 [**?**]. We chose L-BFGS because it was best available blackbox (assuming to access to a gradient) in SciPy. Each internal iteration of L-BFGS was run to convergence, which is described in great detail at `https://docs.scipy.org/doc/scipy-0.19.0/reference/generated/scipy.optimize.fmin_l_bfgs_b.html`.

5.1 Theory

Our augmented Lagrangian method differs from the one presented in class due to the presence of the inequality constraints. However, the general idea is the same. By Theorem 4.1 in Birgin (2014) [**?**], we are guaranteed that the algorithm generates as a convergence sequence, assuming a compact feasible set. Convergence rates for augmented Lagrangian were not discussed in either the class slides or the book chapter we applied.

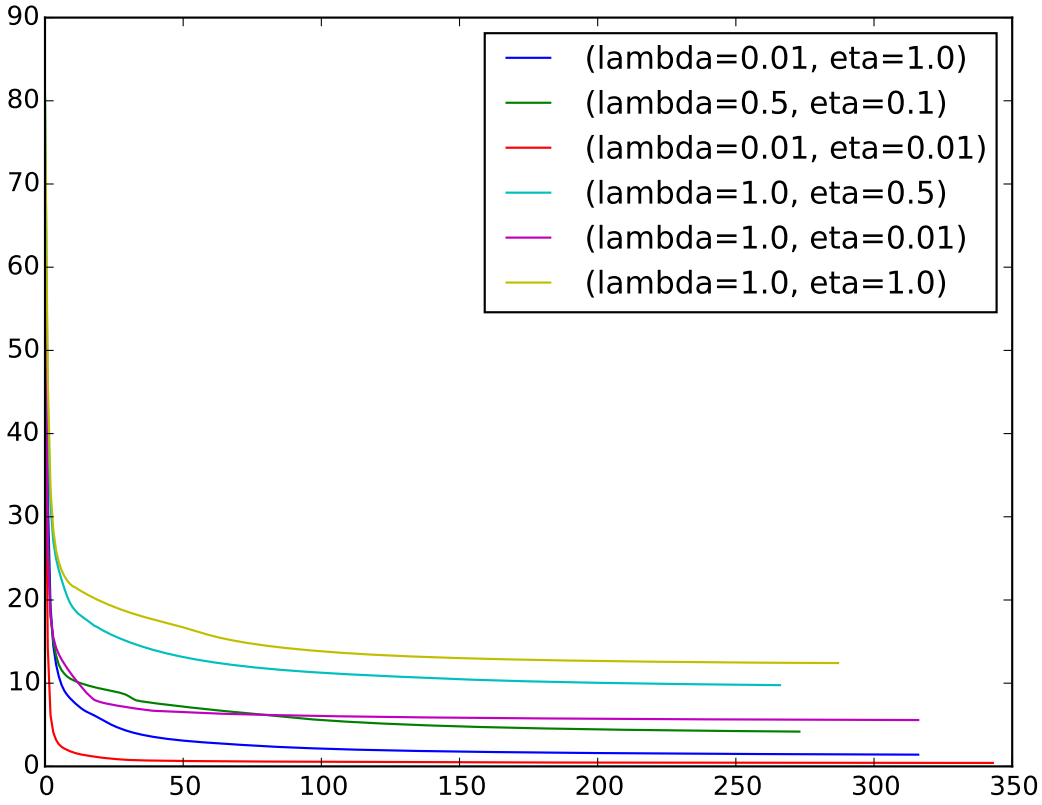
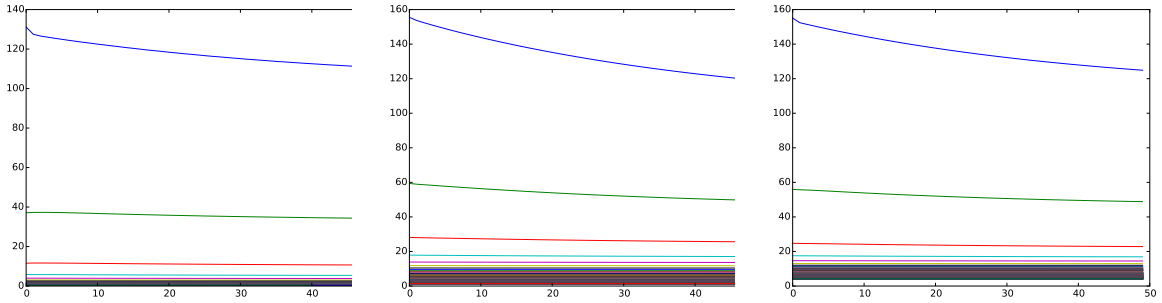


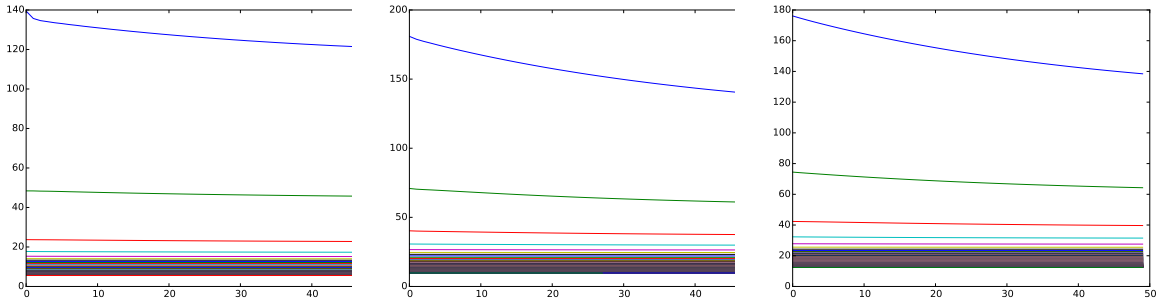
Figure 18: Our objective optimized with Augmented Lagrangian varying with different λ and η values.

5.2 Experiments

We run the augmented Lagrangian block coordinate descent for 300 iterations since convergence is apparent. This algorithm achieves the loss values on the objective (≈ 0.41). We attribute this algorithm's speed and success to the internal calls to the second-order optimization algorithm L-BFGS to solve the subproblems with each augmented Lagrangian run. The Figures 19, 20, 21 show that each individual augmented Lagrangian problem is solved relatively quickly. For speed, we run each algorithm for a maximum of 50 iterations, and we observe that for each problem, we appear to converge well before then. As a bonus, augmented Lagrangian also has very few tunable parameters, e.g., a learning rate, so it requires less human intervention to achieve good performance.



(a) $\lambda = 0.01, \eta = 0.01$ (b) $\lambda = 0.01, \eta = 1.0$ (c) $\lambda = 0.5, \eta = 0.1$



(d) $\lambda = 1.0, \eta = 0.01$ (e) $\lambda = 1.0, \eta = 0.5$ (f) $\lambda = 1.0, \eta = 1.0$

Figure 19: The convex objective (in W) at each iteration. The x-axis shows each iteration of the an augmented Lagrangian run (internal iteration) and each horizontal line an external iteration. We have not labeled the external iterations to avoid cluster: they are ordered from lowest to highest in town down.

6 OTHER STUFF MOVE THIS

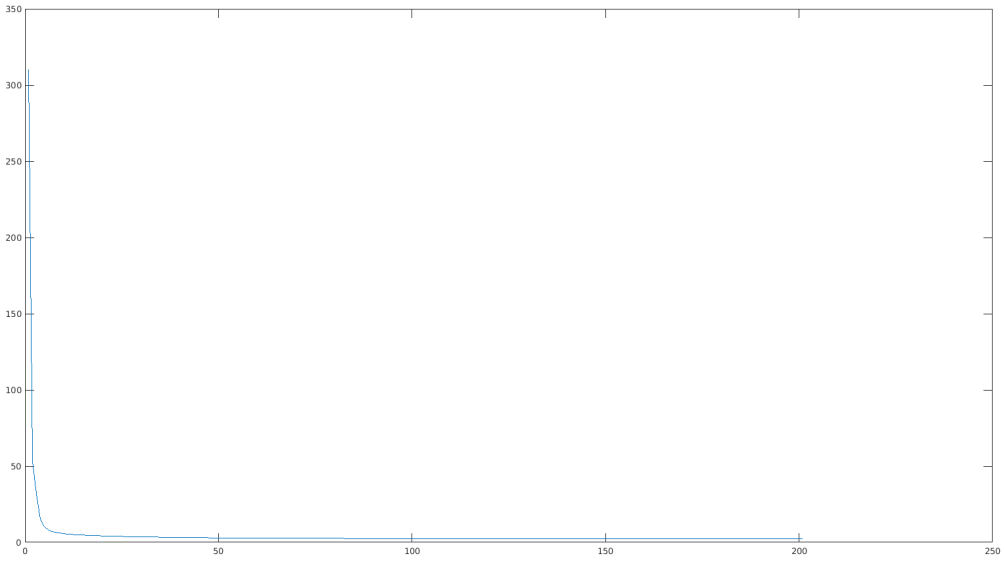
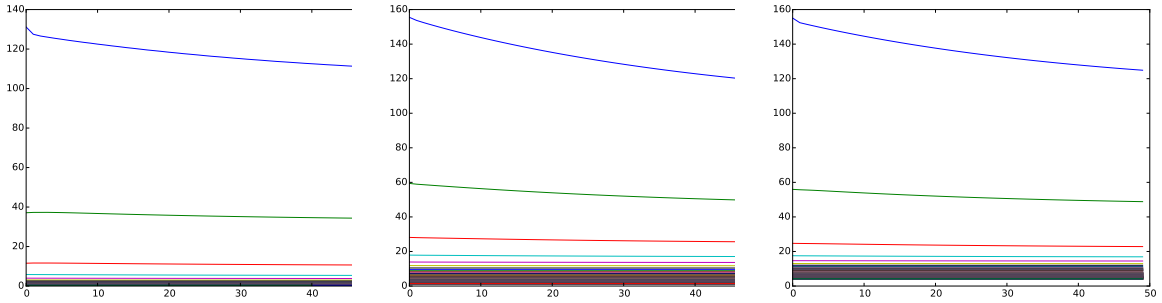
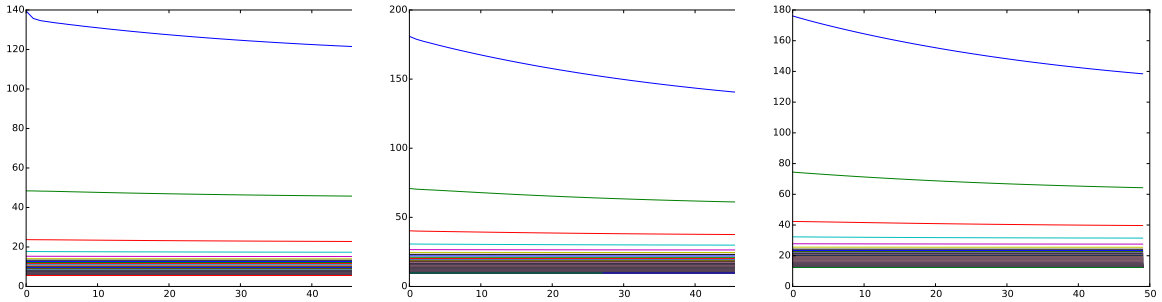


Figure 22: Overall objective for changing the alpha between different suboptimization problems



(a) $\lambda = 0.01, \eta = 0.01$ (b) $\lambda = 0.01, \eta = 1.0$ (c) $\lambda = 0.5, \eta = 0.1$



(d) $\lambda = 1.0, \eta = 0.01$ (e) $\lambda = 1.0, \eta = 0.5$ (f) $\lambda = 1.0, \eta = 1.0$

Figure 20: The convex objective (in H) at each iteration. The x-axis shows each iteration of the an augmented Lagrangian run (internal iteration) and each horizontal line an external iteration. We have not labeled the external iterations to avoid cluster: they are ordered from lowest to highest in town down.

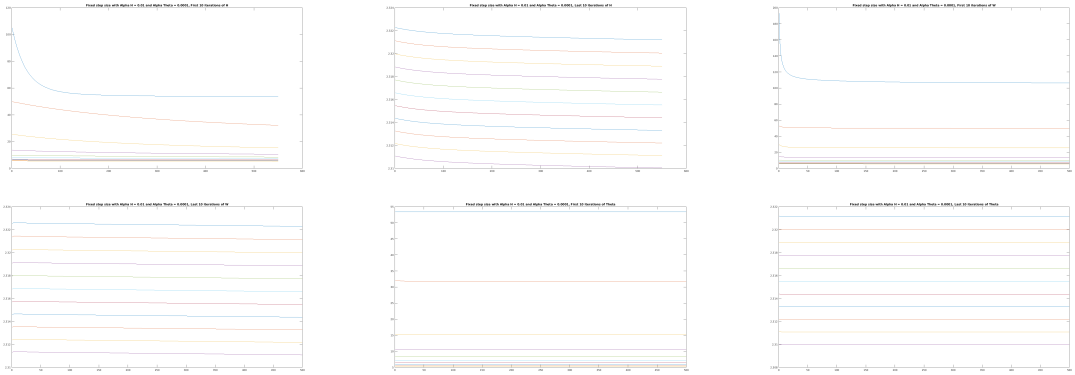
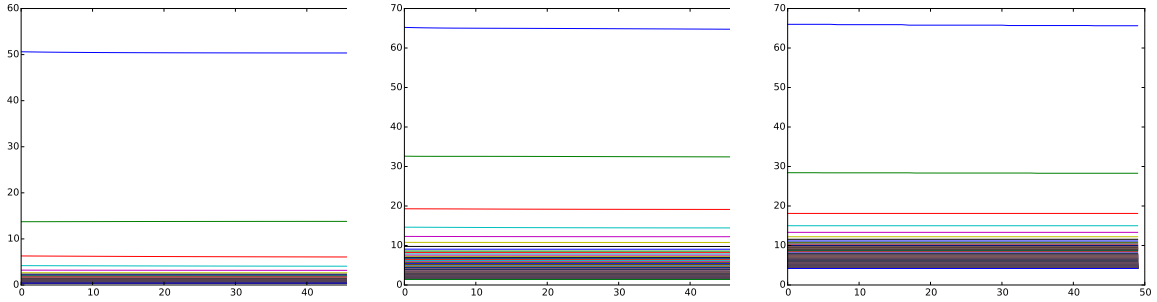
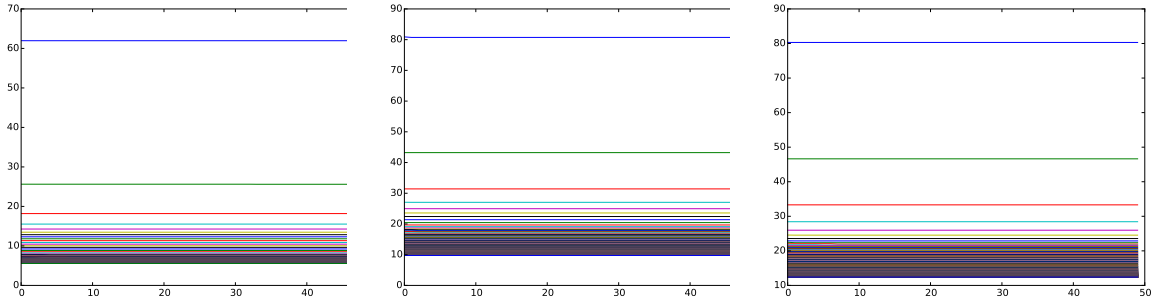


Figure 23: Plots of W, H, Θ optimized on the first and last 10 iterations of projected gradient decent with different values of α for each sub problem



(a) $\lambda = 0.01, \eta = 0.01$ (b) $\lambda = 0.01, \eta = 1.0$ (c) $\lambda = 0.5, \eta = 0.1$



(d) $\lambda = 1.0, \eta = 0.01$ (e) $\lambda = 1.0, \eta = 0.5$ (f) $\lambda = 1.0, \eta = 1.0$

Figure 21: The convex objective (in Θ) at each iteration. The x-axis shows each iteration of the an augmented Lagrangian run (internal iteration) and each horizontal line an external iteration. We have not labeled the external iterations to avoid cluster: they are ordered from lowest to highest in town down.