ECE599 / AI539 Convex Optimization

Woonki Kim
kimwoon@oregonstate.edu

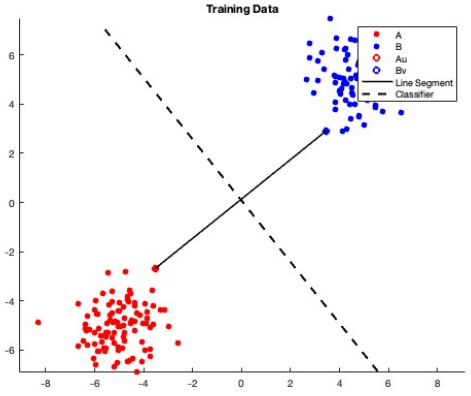
Q1. Consider the classification problem in Homework 2. Download the datasets train separable.mat and test separable.mat from the course website. Download CVX and learn how to use it. Implement the following using CVX.

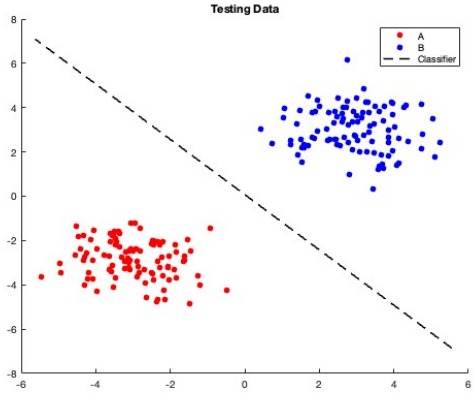
(a) Apply the C-Hull formulation to train a classifer, i.e.,

$$\begin{array}{ll} \text{minimize}_{\mathbf{u},\mathbf{v}} & \|A\mathbf{u} - B\mathbf{v}\|_2^2 \\ \text{subject to} & \mathbf{1}^\top \mathbf{u} = 1, \quad \mathbf{u} \succeq 0, \\ & \mathbf{1}^\top \mathbf{v} = 1, \quad \mathbf{v} \succeq 0. \end{array}$$

Visualize the training data together with the classifier. Also visualize the testing data and the classifier in another figure, and report the classification error on the testing data using the true labels provided in test separable.mat .

Plot





- Training:
  - Au: [-3.526076, -2.678110]

- Bv: [3.454110, 2.916204]
- Objective value: 40.009677

- Testing:
  - Classification Error on Testing Data: 0.000%

(b)Repeat the above for train overlap.mat and test overlap.mat using the reduced C-Hull, i.e.,

$$\begin{aligned} \text{minimize}_{\mathbf{u},\mathbf{v}} \quad & \|A\mathbf{u} - B\mathbf{v}\|_2^2 \\ \text{subject to} \quad & \mathbf{1}^\top \mathbf{u} = 1, \quad \mathbf{d} \succeq \mathbf{u} \succeq 0, \\ & \mathbf{1}^\top \mathbf{v} = 1, \quad \mathbf{d} \succeq \mathbf{v} \succeq 0. \end{aligned}$$

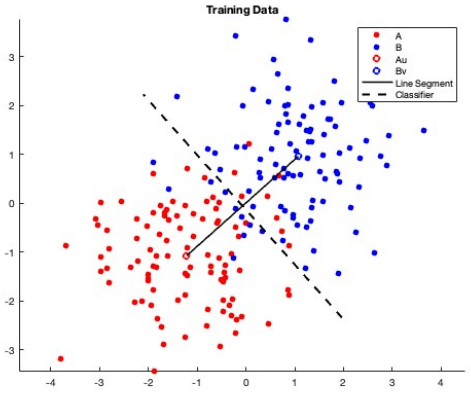
Report the classification error on the testing data using an appropriate d.

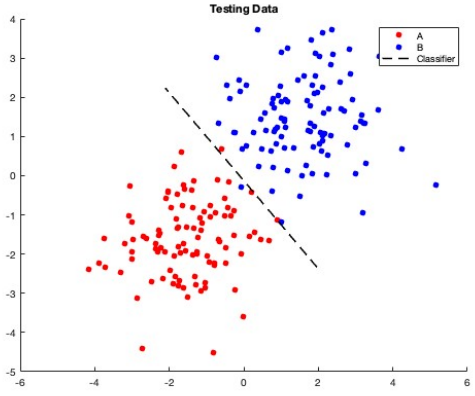
Choosing d.

When making convex hull we typically choose $D = \frac{1}{K}$ where $K \geq 1$ and $K \leq m$, where m is number of points in each convex hulls [1].

- Since A and B both has 100 points, $0.01 \leq D \leq 1$.
- Iterating by testing with $d = 0.01$ to 1 incrementing by 0.01 and pick d with least classification errors in testing data.
- While iterating through candidates for d, exclude d that yields $||Au - Bv||_2^2 \leq \epsilon$, since two reduced convex hulls must not overlap.

Plot





- Training:
  - Optimal d: 0.010000

- Au: [-1.212795, -1.069193]
- Bv: [1.074449, 0.966863]
- Objective value: 4.688504

- Testing:
  - Classification Error on Testing Data: 1.500%

Q2. Under the same setting of Q1, do the following:

(a) Implement C-Hull and Reduced C-Hull using projected gradient. Do the same visualization as in Q1.

Objective function

$$f(u,v) = \frac{1}{2}||Au - Bv||_2^2$$

Constraints

- For separable case:

$$\begin{gathered}\text{subject to} \\ 1^T u = 1, u > 0 \\ 1^T v = 1, v > 0\end{gathered}$$

- For overlapping case:

$$\begin{gathered}\text{subject to} \\ 1^T u = 1, d1 \geq u \geq 0 \\ 1^T v = 1, d1 \geq v \geq 0\end{gathered}$$

Step Size

The Hessian of $f(u,v)$ with respect to u and v is constant and the largest eigenvalue can be calculated directly, since they are quadratic form.

So we can use constant step size:

$$\begin{gathered}0 < \alpha^{(k)} \leq \frac{1}{L} \\ \text{where, } \nabla^2 f(u,v) \leq LI\end{gathered}$$

Step size for u:

$$\alpha_u = \frac{1}{\lambda_{max}(A^T A)}$$

Step size for v:

$$\alpha_v = \frac{1}{\lambda_{max}(B^T B)}$$

Update rule:

$$u^{(r+1)} = Proj_u(u^{(r)} - \alpha_u \nabla(f(u^{(r)}))$$
$$v^{(r+1)} = Proj_v(v^{(r)} - \alpha_v \nabla(f(v^{(r)}))$$

where gradient of $f(u, v)$ with respect to u is:

$$\nabla_u f(u,v) = \nabla_u(\frac{1}{2}(Au - Bv)^T(Au - Bv))$$
$$= \nabla_u \frac{1}{2}\left[(Au)^T(Au) - 2(Au)^T(Bv) + (Bv)^T(Bv)\right].$$
$$= A^T Au - A^T Bv.$$
$$= A^T(Au - Bv)$$

Similarly gradient of $f(u, v)$ with respect to v is:

$$-B^T(Au - Bv)$$

Stopping criteria for projected gradient:

$$u^{(r+1)} = u^{(r)} \text{ and } v^{(r+1)} = v^{(r)} \text{ or } \text{maxIter } < r$$

Projection onto standard simplex:

Objective function:

$$\text{minimize}_p \frac{1}{2}||x - z||_2^2$$
$$\text{subject to} \sum_i^n x_i = 1, x_i \geq 0$$

Lagrangian:

$$\mathcal{L}(\mathbf{x}, \lambda) = \frac{1}{2}\|\mathbf{x} - \mathbf{z}\|_2^2 + \lambda \left(\sum i = 1^n x_i - 1\right)$$

Take gradient and set it to 0:

$$\frac{\partial \mathcal{L}}{\partial x_i} = x_i - z_i + \lambda = 0$$
$$x_i = z_i - \lambda.$$

To make sure $1^T x = 1$ holds:

$$\sum_{i=1}^{n} x_i = 1$$

Substitute in $x_i = z_i - \lambda$:

$$\sum_{i=1}^{n}(z_i - \lambda) = \sum_{i=1}^{n} z_i - n\lambda = 1$$

$$\lambda = \frac{\sum_{i=1}^{n} z_i - 1}{n}.$$

To make sure $x_i \geq 0$ holds:

$$x_i = \max(z_i - \lambda, 0).$$

Projection onto upper bounded simplex($1^T u = 1, d1 \geq u \geq 0$)

If vector is not empty, there exists λ where:

$$\sum_{i=1}^{n} \text{clip}_{[0,u_i]}(q_i - \lambda) - 1 = 0.$$

When λ^* is found, clipping it in between 0 and d is equal to p^*[2]:

$$p_i = \text{clip}_{[l_i,u_i]}(q_i - \lambda^*)$$

Pseudo Code

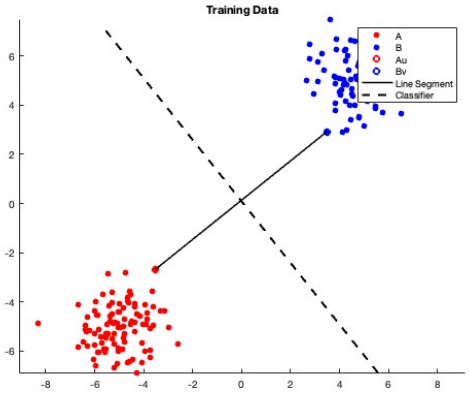
Algorithm 1: Projected Gradient

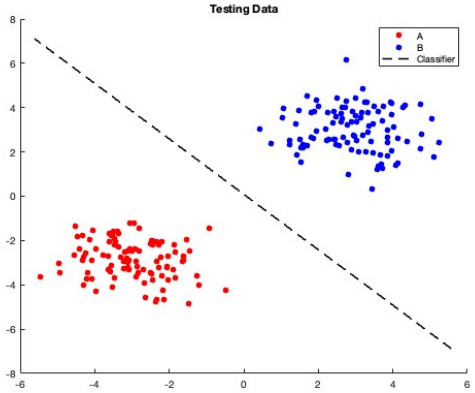
Input: A, B
Output: u, v

$\alpha_u = \frac{1}{\lambda_{max}(A^T A)}$ *and* $\alpha_v = \frac{1}{\lambda_{max}(B^T B)}$;
$r \leftarrow 0$;
while $r < maxIter$ **do**
  $u^{(r+1)} \leftarrow Proj_u\big(u^{(r)} - \alpha_u \nabla_u f(u^{(r)}, v^{(r)})\big)$;
  $v^{(r+1)} \leftarrow Proj_v\big(v^{(r)} - \alpha_v \nabla_v f(u^{(r)}, v^{(r)})\big)$;
  if $u^{(r+1)} = u^{(r)}$ ***and*** $v^{(r+1)} = v^{(r)}$ *break*;
  $r \leftarrow r + 1$;
end

Plot

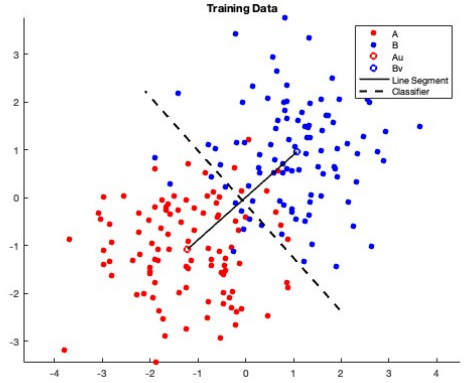
Separable case:

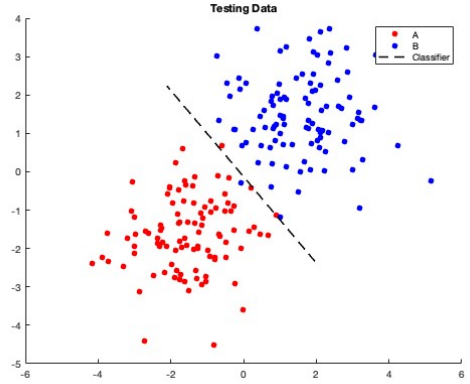




- Training:
  - Au: [-3.526076, -2.678110]
  - Bv: [3.474065, 2.916251]
  - Objective value: 40.149424
- Testing:
  - Classification Error on Testing Data: 0.000%

Overlapping case:





- Training:
  - Optimal d: 0.010000
  - Au: [-1.212795, -1.069193]
  - Bv: [1.074449, 0.966863]
  - Objective value: 4.688504
- Testing:

- Classification Error on Testing Data:
  1.500%

(b) Repeat the above using Nesterov's acceleration.

Nesterov's acceleration:

$$a_0 = 0, a_r = 1/2\left(1+\sqrt{4a_{r-1}^2+1)}\right)$$
$$A_r = \sum_{i=0}^{r} a_i = a_r^2 \text{ and } a_r^2 - a_r = a_{r-1}^2$$
$$\text{at iteration r:}$$
$$t^r = (a_r - 1)/a_{(r+1)}$$
$$y^{(r+1)} = (1+t^r)x^{(r)} - t^r x^{(r-1)}$$
$$x^{(r+1)=y^{(r+1)}-\frac{1}{L}\nabla f(y^{(r+1)})}$$

Update Rule of Projected Gradient using Nesterov's acceleration:

- Compute a_r:

$$a_r = (1+\sqrt{(1+4a_{r-1}^2)})/2$$

- Compute t_r:

$$t_r = (a_r - 1)/a_{r+1}$$

- Update extrapolated point y:

$$y_u^{(r)} = u^{(r)} + t_r(u^{(r)} - u^{(r-1)})$$
$$y_v^{(r)} = v^{(r)} + t_r(v^{(r)} - v^{(r-1)})$$

- Compute gradients:

$$grad_u = A^T(Ay_u^{(r)} - By_v^{(r)})$$
$$grad_v = -B^T(Ay_u^{(r)} - By_v^{(r)})$$

- Update variables with projection:

$$u_{temp} = y_u^{(r)} - (1/L_u) * grad_u$$
$$v_{temp} = y_v^{(r)} - (1/L_v) * grad_v$$
$$u^{(r+1)} = proj(u_{temp})$$
$$v^{(r+1)} = proj(v_{temp})$$

Stopping criteria

Nesterov acceleration only ends when iteration meets max iteration.

$$\text{maxIter} < r$$

Pseudo Code

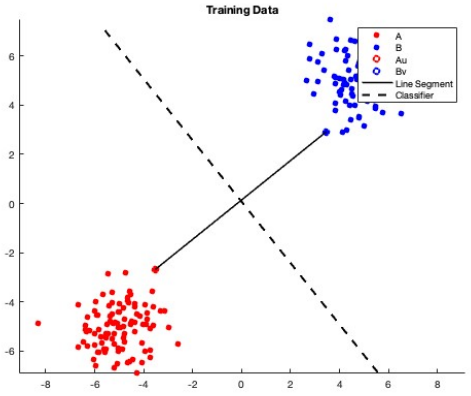
Algorithm 2: Projected Gradient with Nesterov's Acceleration

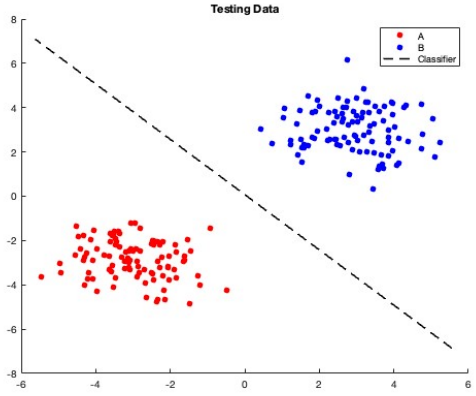
Input: A, B
Output: u, v

$\alpha_u = \frac{1}{\lambda_{max}(A^TA)}$ and $\alpha_v = \frac{1}{\lambda_{max}(B^TB)}$;
$a_0 = 0$;
$r \leftarrow 0$;
while $r <$ *maxIter* **do**
  $a_r \leftarrow \frac{1+\sqrt{1+4a_{r-1}^2}}{2}$;
  $t_r \leftarrow \frac{a_{r-1}-1}{a_r}$;
  $y_u^{(r)} \leftarrow u^{(r)} + t_r(u^{(r)} - u^{(r-1)})$;
  $y_v^{(r)} \leftarrow v^{(r)} + t_r(v^{(r)} - v^{(r-1)})$;
  $u^{(r+1)} \leftarrow Proj_u\left(y_u^{(r)} - \alpha_u \nabla_u f\right)$;
  $v^{(r+1)} \leftarrow Proj_v\left(y_v^{(r)} - \alpha_v \nabla_v f\right)$;
  $r \leftarrow r + 1$;
end

Plot

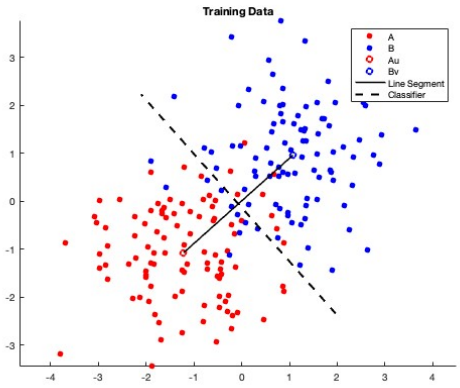
Separable case:

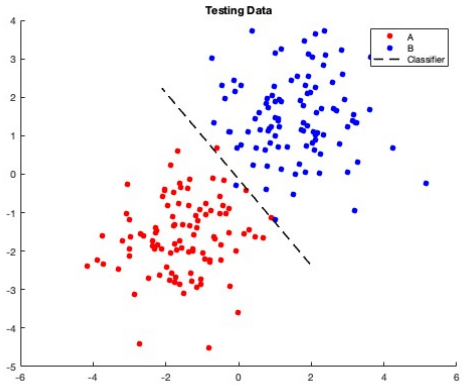




- Training:
  - Au: [-3.526076, -2.678110]
  - Bv: [3.454110, 2.916204]
  - Objective value: 40.009677
- Testing:
  - Classification Error on Testing Data: 0.000%

Overlapping case:





- Training:
  - Optimal d: 0.010000
  - Au: [-1.212795, -1.069193]
  - Bv: [1.074449, 0.966863]
  - Objective value: 4.688504
- Testing:
  - Classification Error on Testing Data: 1.500%

Q3. Under the same setting of Q1, implement C-Hull and Reduced C-Hull using ADMM. Do the same visualization as in Q1.

Original problem:

$$\begin{aligned}&\text{minimize}_{u,v}\ \frac{1}{2}||Au-Bv||_2^2\\&\text{subject to } 1^Tu=1, u\geq 0\\&\text{subject to } 1^Tv=1, v\geq 0\end{aligned}$$

Reformulating problem to match standard ADMM form:

Standard ADMM form:

$$\begin{aligned}&\text{minimize}_{x,z}\ f(x)+g(z)\\&\text{subject to } Ax+Bz=c\end{aligned}$$

Let, $x=[u,v]$

Now we can reformulate out constraint $1^T u = 1, 1^T v = 1$ to:

$$\begin{bmatrix} 1^T & 0^T \\ 0^T & 1^T \end{bmatrix} x = [1, 1]^T$$

Now let $F = [A, -B]$

Our formula becomes:

$$\begin{gathered} \text{minimize}_{x,z} ||Fx||_2^2 + I(z) \\ \text{subject to } x = z \end{gathered}$$

where $I(z)$ is a indicator function of checking whether z is non-negative.

Now, we need to make constraint to $Cx + Dz = E$ form.

Since $x = z$, $Cx + Dz = E$ holds when $C = I, D = -I, E = 0$.

While $\begin{bmatrix} 1^T & 0^T \\ 0^T & 1^T \end{bmatrix} x = [1, 1]^T$ holds when $C = \begin{bmatrix} 1^T & 0^T \\ 0^T & 1^T \end{bmatrix}, D = 0, E = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

Combining these two we get,

$$C = \begin{bmatrix} I \\ \begin{bmatrix} 1^T & 0^T \\ 0^T & 1^T \end{bmatrix} \end{bmatrix}, D = \begin{bmatrix} -I \\ 0 \end{bmatrix}, E = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}.$$

Rewriting formula into Standard ADMM form:

$$\begin{gathered} \text{minimize}_{x,z} f(x) + g(z) \\ \text{subject to } Cx + Dz = E \end{gathered}$$

$$\begin{gathered} \text{Where, } f(x) = ||Fx||_2^2,\ g(z) = I_c(z) \\ F = [A, -B],\ I_c(z) : \text{Positiveness Indicator function}, \\ C = \begin{bmatrix} I \\ \begin{bmatrix} 1^T & 0^T \\ 0^T & 1^T \end{bmatrix} \end{bmatrix}, D = \begin{bmatrix} -I \\ 0 \end{bmatrix}, E = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}. \end{gathered}$$

Augmented Lagrangian:

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(Cx + Dz - E) + \frac{\rho}{2}\|Cx + Dz - E\|_2^2$$

Let $u = y/\rho$:

$$L_\rho(x,z,u) = f(x) + g(z) + u^T(Cx + Dz - E) + \frac{\rho}{2}\|Cx + Dz - E\|_2^2.$$
$$= f(x) + g(z) + u^T(Cx + Dz - E) + \frac{\rho}{2}(Cx + Dz - E)^T(Cx + Dz - E).$$
$$= f(x) + g(z) + \frac{\rho}{2}\|Cx + Dz - E + u\|_2^2 - \frac{\rho}{2}\|u\|_2^2.$$
$$= f(x) + g(z) + \frac{\rho}{2}(\|Cx + Dz - E + u\|_2^2 - \|u\|_2^2)$$

Update x

$$x^{r+1} = \arg\min_x \left(f(x) + \frac{\rho}{2}\|Cx + Dz^r - E + y^r/\rho\|_2^2\right)$$

Substituting $f(x) = \|Fx\|_2^2$, we get:

$$x^{r+1} = \arg\min_x \|Fx\|_2^2 + \frac{\rho}{2}\|Cx + Dz^r - E + y^r/\rho\|_2^2$$
$$x^{r+1} = \arg\min_x \; x^T(F^TF + \frac{\rho}{2}C^TC)x + \rho x^TC^T(Dz^r - E + \frac{y^r}{\rho})$$

Taking gradient with respect to x and set it 0:

$$2F^TFx^{r+1} + \rho C^T(Cx^{r+1} + Dz^k - E + u^r) = 0$$
$$\left(2F^TF + \rho C^TC\right)x^{r+1} = -\rho C^T\left(Dz^r - E + \frac{y^r}{\rho}\right)$$
$$x^{r+1} = \left(2F^TF + \rho C^TC\right)^{-1}\left(-\rho C^T\left(Dz^r - E + \frac{y^r}{\rho}\right)\right)$$

Update z

$$z^{r+1} = \arg\min_z \left(g(z) + \frac{\rho}{2}\|Cx^{r+1} + Dz - E + y^r/\rho\|_2^2\right)$$

Substituting in $C = \begin{bmatrix} I \\ \begin{bmatrix} 1^T & 0^T \\ 0^T & 1^T \end{bmatrix} \end{bmatrix}, D = \begin{bmatrix} -I \\ 0 \end{bmatrix}, E = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$.

We get:

$$Cx^{r+1} + Dz - E + y^r/\rho = \begin{bmatrix} I \\ \begin{bmatrix} 1^T & 0^T \\ 0^T & 1^T \end{bmatrix} \end{bmatrix} x^{r+1} + \begin{bmatrix} -I \\ 0 \end{bmatrix} z - \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} y^{(1)r}/\rho \\ y^{(2)r}/\rho \end{bmatrix}$$

Let $L = \begin{bmatrix} 1^T & 0^T \\ 0^T & 1^T \end{bmatrix}$, $M = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

- First Block

  $Ix^{r+1} - Iz + 0 + y^{(1)r}/\rho = x^{r+1} - z + y^{(1)r}/\rho$
- Second Block

  $Lx^{r+1} + 0z - M + y^{(2)r}/\rho = Lx^{r+1} - M + y^{(2)r}/\rho$
- Combining

  $$Cx^{r+1} + Dz - E + y^r = \begin{bmatrix} x^{r+1} - z + y^{(1)r}/\rho \\ Lx^{r+1} - b + y^{(2)r}/\rho \end{bmatrix}$$

Since there is no z in second block and we are minimizing function with the respect to z, consider it as constant

Rewriting updating z:

$$z^{r+1} = \arg\min_z \quad \frac{\rho}{2}\|x^{r+1} - z + y^{(1)r}/\rho\|_2^2$$

Objective function:

$$h(z) = \|x^{r+1} + y^{(1)r}/\rho - z\|_2^2$$

Gradient with respect to z and set it to 0:

$$\nabla_z h(z) = -2(x^{r+1} + y^{(1)r}/\rho - z) = 0$$
$$\implies z = x^{r+1} + y^{(1)r}/\rho$$

However, since we have used only first block now we need to consider the constraints.

$$z^{r+1} = g(z) + (x^{r+1} + y^{(1)r}/\rho, 0)$$
$$\implies z^{r+1} = \max\left(x^{r+1} + y^{(1)r}/\rho, 0\right)$$

$$Proj_u(z^{(1)(r+1)}, d)$$
$$Proj_v(z^{(2)(r+1)}, d)$$
$$\text{where, d=1 for separable case}$$

Update y

$$u^{r+1} = u^r + (Cx^{r+1} + Dz^{r+1} - E)$$
$$\implies y^{r+1} = y^r + \rho(Cx^{r+1} + Dz^{r+1} - E)$$

Stopping criteria

Admm stops when both L2-norm of primal and dual problem is less than tolerance.

$$||Cx + Dz - E|| \leq \epsilon$$
$$||\rho C^T(D(z^{(r+1)} - z^{(r)}))|| \leq \epsilon$$

Choosing penalty parameter

Iterating through $\rho = 10^{-3}$ to 10^3 to finds classifier with least iteration on training set.

- Separable: $\rho = 215$
- Overlapping: $\rho = 1000$

Pseudo Code

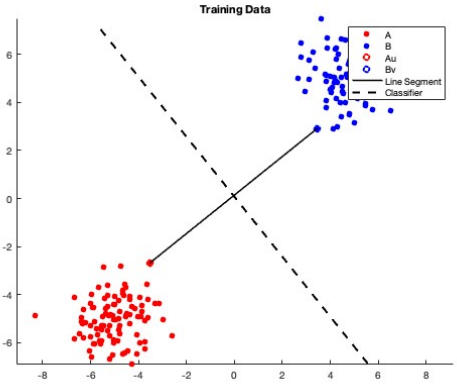
Algorithm 4: ADMM Algorithm

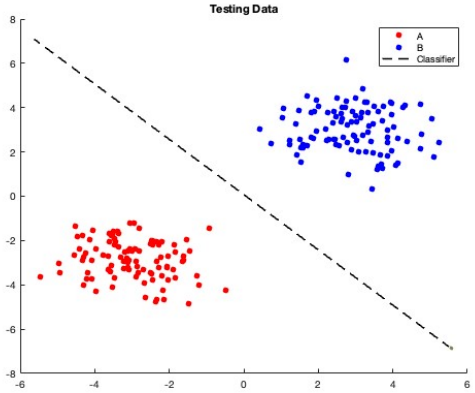
Input: A, B, ρ
Output: x, z

Set $\alpha_x = \frac{1}{\lambda_{\max}(A^TA)}$ and $\alpha_z = \frac{1}{\lambda_{\max}(B^TB)}$;
$C = \begin{bmatrix} I \\ \begin{bmatrix} 1^T & 0^T \\ 0^T & 1^T \end{bmatrix} \end{bmatrix}, D = \begin{bmatrix} -I \\ 0 \end{bmatrix}, E = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$ $F = [A, -B]$;
$x = z$;
$r \leftarrow 0$;
while $r < maxIter$ **do**
  $x^{r+1} \leftarrow \left(2F^TF + \rho C^TC\right)^{-1}\left(-\rho C^T(Dz^r - E + \frac{y^r}{\rho})\right)$;
  $z^{r+1} \leftarrow \max(x^{r+1} + y^{(1)r}/\rho, 0)$;
  $z^{(1)r+1} \leftarrow \text{Proj}_z\left(z^{(1)r+1}\right)$;
  $z^{(2)r+1} \leftarrow \text{Proj}_z\left(z^{(2)r+1}\right)$;
  $y^{r+1} \leftarrow y^r + \rho\left(Cx^{r+1} + Dz^{r+1} - E\right)$;
  if $\|Cx^{r+1} + Dz^{r+1} - E\|_2 \leq \epsilon$ *and* $\|\rho C^T(D(z^{(r+1)} - z^{(r)}))\|_2 \leq \epsilon$ **then**
    break;
  end
  $r \leftarrow r + 1$;
end

Plot

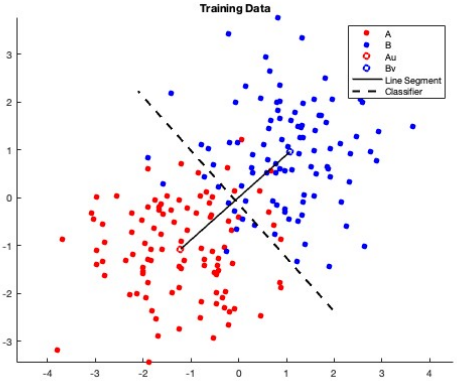
Separable case:

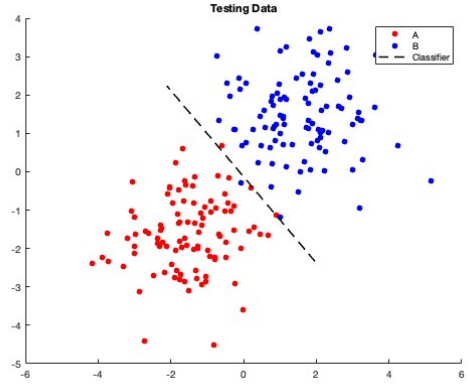




- Training:
  - Au: [-3.526081, -2.678114]
  - Bv: [3.454115, 2.916208]
  - Objective value: 40.009787
- Testing:
  - Classification Error on Testing Data: 0.000%

Overlapping case:



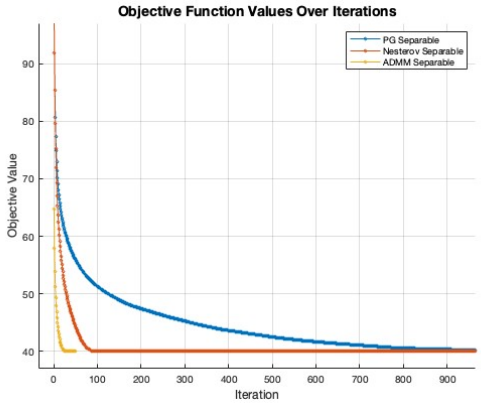


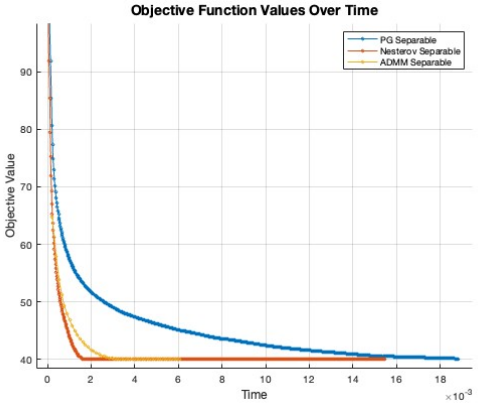
- Training:
  - Optimal d: 0.010000
  - Au: [-1.212791, -1.069190]
  - Bv: [1.074444, 0.966858]
  - Objective value: 4.688468
- Testing:

- Classification Error on Testing Data:
  1.500%

Q4.Plot an "iteration v.s. objective value" figure for the training process. Compare all the algorithms that you implemented in this figure. In addition, plot a "time v.s. objective value" figure using all the algorithms

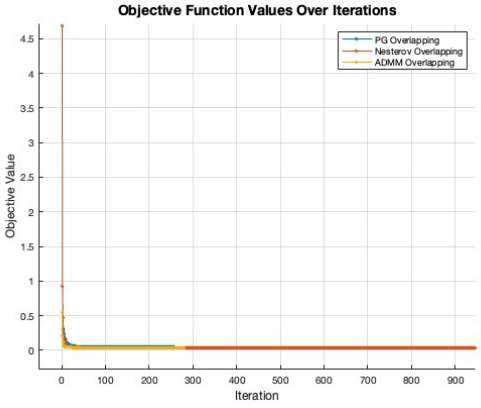
Separable case:

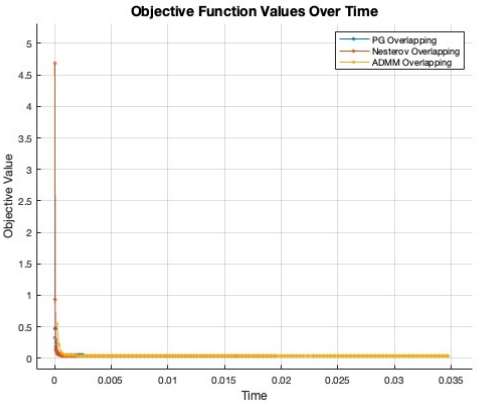




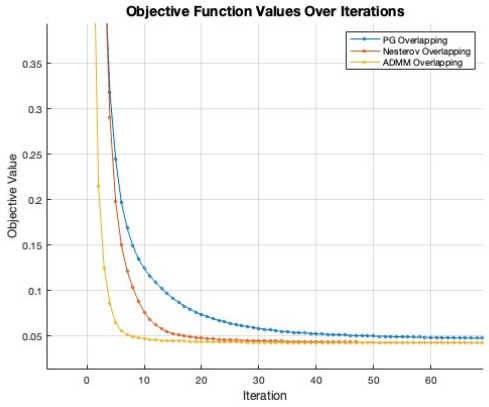
In separable case Admm method definitely outperforms Nesterov and projected gradient both in terms of iteration and time. Admm converges within less than 100 iteration showing its efficiency. While Nesterov's accelerated gradient descent converges very fast to the optimal point proving theoretical convergence rate $O(1/r^2)$, but since it does not have convergence checking condition it iterates until the maximum iteration set in the program.

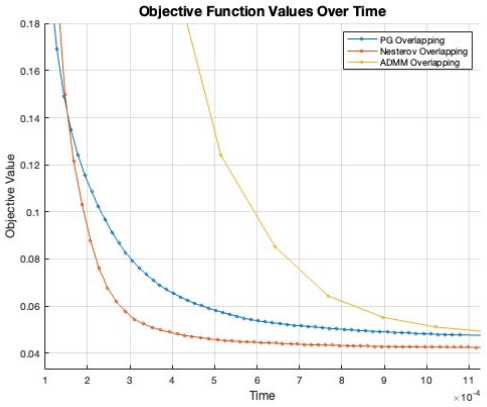
Overlapping case:





Overlapping case zoom in version:





With d set to 0.01, it was hard to analyze the difference between the algorithms since it converged with in 1 or 2 iteration. So to give more variance I have set another candidate d which is 0.04 with classifier error 2.00%.

In overlapping case, still Admm method outperforms in terms of iteration, while taking more time than the other two methods. This result shows that Admm's single calculation takes more time than Nesterov's acceleration or Projected gradient, suggesting that if the projection is efficient enough and constraints are not complicated, we could consider using Nesterov's for this optimization problem. Additionally, as well as in separable case Nesterov's algorithm shows faster convergence compared to projected gradient, showing the theoretical convergence rate of $O(1/r^2)$. In conclusion, if computational resource is limited and your goals is to minimize the iteration of the model, use Admm. On the other hand, if you aim to get efficient algorithm in terms of time, use Nesterov's accelerated gradient for this dataset.

Reference

[1] G. Fung and O. L. Mangasarian, "Duality and Geometry in SVM Classifiers," in *Proceedings of the 17th International Conference on Machine Learning (ICML 2001)*, 2001, pp. 194–201. Available: https://www.researchgate.net/publication/2801700_Duality_and_Geometry_in_SVM_Classifiers

[2] D. Needell, N. Srebro, and R. Ward, "Stochastic Gradient Descent, Weighted Sampling, and the Randomized Kaczmarz Algorithm," *arXiv preprint arXiv:1905.03488*, 2019. [Online]. Available: https://arxiv.org/pdf/1905.03488