ISE 5406 - Optimization II
Project - Part 1

Due on Tuesday (03/30) by 8 am EST

Team Members' Name:
1. Andrew Hartley (Contribution 33.33%)
2. Ruochen Wang (Contribution 33.33%)
3. Tai-Jung Chen (Contribution 33.33%)

Total Points: 100 Points (50 points each)

The objective of this project is to give you an experience of utilizing theory and algorithms discussed in this course for solving problems in the eld of Finance (in particular, Portfolio Optimization), Economics, and Machine Learning (ML).

Problem 1:

- Identify different problems in portfolio optimization that can be formulated as nonlinear programming problems, in particular Convex/Concave or Quadratic Optimization Problem. Clearly describe them and discuss the significance of these problems.
- Discuss the significance of Quasiconcave functions in the area of Economics.

Problem 2:

- Identify different problems in ML that can be formulated as nonlinear programming problems, in particular Convex/Concave or Quadratic Optimization Problem.
- Clearly describe them and discuss the significance of these problems in the area of ML.

Evaluation Criteria.

- **Quality of Report**. Each team (with at most 3 members) is required to turn in a well-written typed report (Font: 12 pt Times New Roman; Line spacing: 1.5 or double; One-Inch Left, Right, Top, and Bottom Margin). The report should have this page as the cover page, including team members' names, signatures, and peer evaluation. It is very important to cite references and acknowledge any help used to prepare this report.
- **Peer Evaluation.** On the cover sheet, you are required to provide contribution of each team member in this part of the project; for example, in case all members of a team of three students have contributed equally, write 33.33% in front of each member's name. This evaluation will be used while assigning scores for each team member.

Portfolio Optimization

(i) Markowitz' theory

Markowitz' theory of mean-variance optimization (MVO) provides a mechanism for the selection of portfolios of securities (or asset classes) in a manner that trades off the expected returns and the risk of potential portfolios.

Consider assets $S_1, S_2, \ldots, S_n (n >= 2)$ with random returns. Let μ_i and δ_i denote the expected return and the standard deviation of the return of assets . For $i \neq j$, ρ_{ij} denotes the correlation coefficient of the returns of assets S_i and S_j. Let $\mu = [\mu_1, \ldots, \mu_n]^T$, and $\Sigma = (\sigma_{ij})$ be the n × n symmetric covariance matrix with $\sigma_{ii} = (\sigma_{i)^2}$ and $\sigma_{ij} = \rho_{ij}\sigma_i\sigma_j$ for $i \neq j$

Denoting by x_i the proportion of the total funds invested in security i, one can represent the expected return and the variance of the resulting portfolio $x = (x_1, \ldots, x_n)$ as follows:

$$E[x] = x_1\mu_1 + \ldots + x_n\mu_n = \mu^T x$$

and

$$Var[x] = \sum_{i,j} \rho_{ij}\sigma_i\sigma_j x_i x_j = x^T \Sigma x$$

where $\rho_{ii} \equiv 1$.

Since variance is always nonnegative, it follows that $x^T \Sigma x \geq 0$ for any x, i.e., Σ is positive semidefinite. It is usually assumed that it is in fact positive definite, which is essentially equivalent to assuming that there are no redundant assets in our collection $S_1, \ldots, S_n$. We can further assume that the set of admissible portfolios is a nonempty polyhedral set and represent it as $X := \{x: Ax = b, Cx >= d\}$, where A is an m × n matrix, b is an m-dimensional vector, C is a p × n matrix and d is a p-dimensional vector. In particular, one of the constraints in the set X is

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

A feasible portfolio x is called efficient if it has the maximal expected return among all portfolios with the same variance, or alternatively, if it has the minimum variance among all portfolios that have at least a certain expected return. The collection of efficient portfolios forms the efficient frontier of the portfolio universe. The efficient frontier is often represented as a curve in a two-dimensional graph where the coordinates of a plotted point correspond to the standard deviation and the expected return of an efficient portfolio.

When we assume that Σ is positive definite, the variance is a strictly convex function of the portfolio variables and there exists a unique portfolio in X that has the minimum variance. Let us denote this portfolio with $x_{min}$and its return $\mu^T x_{min}$ with R_{min}. Note that x_{min} is an efficient portfolio. Let R_{max} denote the maximum return for an admissible portfolio. Markowitz' mean-variance optimization (MVO) problem can be formulated in three different but equivalent ways.

Find the minimum variance portfolio of the securities 1 to n that yields at least a target value of expected return (say b). Mathematically, this formulation produces a quadratic programming problem:

$$min_x \ \frac{1}{2} x^T \Sigma x$$
$$s.t. \mu^T x >= R$$
$$Ax = b$$
$$Cx >= d$$

The first constraint indicates that the expected return is no less than the target value R. Solving this problem for values of R ranging betweenR_{min} and R_{max} one obtains all efficient portfolios. As we discussed above, the objective function corresponds to one half the total variance of the portfolio. The constant $\frac{1}{2}$ is added for convenience in the optimality conditions–it obviously does not affect the optimal solution.

This is a convex quadratic programming problem for which the first order conditions are both necessary and sufficient for optimality.

The two other variations of the MVO problem are the following:

(2)

$$max_x \ \mu^T x$$
$$s.t. x\Sigma x <= \sigma^2$$
$$Ax = b$$
$$Cx >= d$$

(3)

$$max_x\ \mu^T x - \frac{\delta}{2} x^T \Sigma x$$

$$s.t. Ax = b$$

$$Cx >= d$$

In (2), σ^2 is a given upper limit on the variance of the portfolio. In (3), the objective function is a risk-adjusted return function where the constant δ serves as a risk-aversion constant. While (3) is another quadratic programming problem, (2) has a convex quadratic constraint and therefore is not a QP. This problem can be solved using the general nonlinear programming solution techniques. And a reformulation of (2) can be a second-order cone program. This opens the possibility of using specialized and efficient second-order cone programming methods for its solution.

(ii) Transaction Costs

We can add a portfolio turnover constraint to ensure that the change between the current holdings x^0 and the desired portfolio x is bounded by h. This constraint is essential when solving large mean-variance models since the covariance matrix is almost singular in most practical applications and hence the optimal decision can change significantly with small changes in the problem data. To avoid big changes when reoptimizing the portfolio, turnover constraints are imposed. Let y_i be the amount of asset i bought and z_i the amount sold. We write

$$x_i - x_i^0 <= y_i, y_i >= 0$$

$$x_i^0 - x_i <= z_i, z_i >= 0$$

$$\sum_{i=1}^{n} (y_i + z_i) <= h$$

Instead of a turnover constraint, we can introduce transaction costs directly into the model. Suppose that there is a transaction cost t_i proportional to the amount of asset i bought, and a transaction cost t_i' proportional to the amount of asset i sold. Suppose that the portfolio is reoptimized once per period. As above, let x^0 denote the current portfolio. Then a reoptimized portfolio is obtained by solving

$$min \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j$$

$$s.t$$

$$\sum_{i=1}^{n} (\mu_i x_i - t_i y_i - t_i' z_i) >= R$$

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

$$x_i - x_i^0 <= y_i \; for \; i = 1, \ldots, n$$

$$x_i^0 - x_i <= z_i \; for \; i = 1, \ldots, n$$

$$y_i >= 0 \; for \; i = 1, \ldots, n$$

$$z_i >= 0 \; for \; i = 1, \ldots, n$$

$$x_i \; unrestricted \; for \; i = 1, \ldots, n$$

(iii) The Black-Litterman Model

Black and Litterman recommend to combine the investor's view with the market equilibrium, as follows.

The expected return vector μ is assumed to have a probability distribution that is the product of two multivariate normal distributions. The first distribution represents the returns at market equilibrium, with mean π and covariance matrix $\tau\Sigma$, where τ is a small constant and $\Sigma = (\sigma_{ij})$ denotes the covariance matrix of asset returns (Note that the factor τ should be small since the variance $\tau\sigma_i^2$ of the random variable μ_i is typically much smaller than the variance σ_i^2 of the underlying asset returns). The second distribution represents the investor's view about the μ_i's. These views are expressed as

$$P\mu = q + \epsilon$$

where P is a k × n matrix and q is a k-dimensional vector that are provided by the investor and ϵ is a normally distributed random vector with mean 0 and diagonal covariance matrix Ω (the stronger the investor's view, the smaller the corresponding $\omega_i = \Omega_{ii}$).

The resulting distribution for μ is a multivariate normal distribution with mean

$$\mu_{mean} = [(\tau\Sigma)^{-1} + P^T\Omega^{-1}P]^{-1}[(\tau\Sigma)^{-1}\pi + P^T\Omega^{-1}q]$$

Black and Litterman use μ_{mean} as the vector of expected returns in the Markowitz model.

Quasiconcave Functions in Economics

Economists model markets and try to predict and optimize decision making so that "utility" is maximized. Utility is a unitless metric that attempts to quantify how "happy" a person or firm is with a decision. Since utility is unitless, the actual value related to utility is meaningless, but economists use utility scores as a way to compare different combinations of choices. Of course, economists are concerned with utility, and concepts similar to utility, because there are limited resources so people, and firms, must make decisions on how to best allocate these limited resources.

Allocating limited resources is a natural place for optimization to occur. Since utility functions model preferences and there are limited resources, economists set up optimization problems to maximize utility. Many of the problems are nonlinear since some preferences may show some dependence. For instance, a factory with all machines and no laborers is unlikely to be very productive, and similarly a factory with all laborers and no machinery is also unlikely to be very productive but when the "right" (optimal) combination of labor and capital is achieved the factory can work most efficiently. This is the intuition behind the Cobb-Douglas function, one of the most prominent ideas in microeconomic theory.

The Cobb-Douglas production function is as follows: $z = Cx^{\alpha}y^{\beta}$. In this equation C is a constant, z is the production level, x is the amount of labor and y is the amount of capital while α, β are in [0,1] and are elasticity measures, meaning they determine the impact of adding (or subtracting) an input. Since x and y are the variables we control, we can show this equation is quasiconcave by showing that the function has a strictly negative second derivative (is concave):

$$\frac{dz}{dx} = C\alpha x^{\alpha-1}y^{\beta} > 0$$

$$\frac{dz}{dy} = C\beta x^{\alpha}y^{\beta-1} > 0$$

$$\frac{dz^2}{d^2x} = C\alpha(\alpha-1)x^{\alpha-2}y^{\beta} < 0$$

$$\frac{dz^2}{d^2y} = C\beta(\beta-1)x^{\alpha}y^{\beta-2} < 0$$

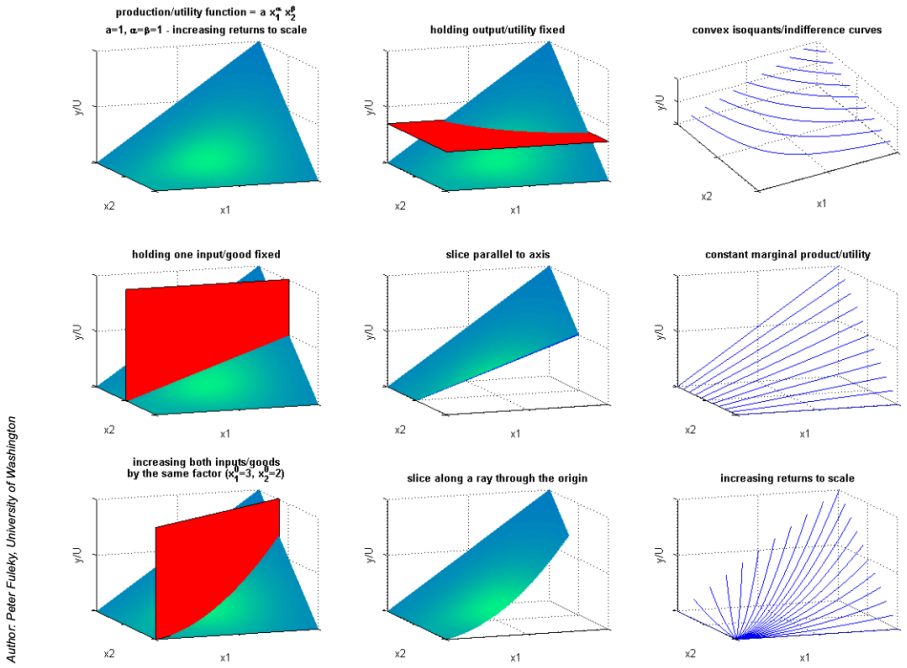
$$\frac{dz^2}{d\ xy} = C\alpha\beta x^{\alpha-1}y^{\beta-1} > 0$$

So, $\frac{dz^2}{d^2x}\left(\frac{dz}{d\ y}\right)^2 - 2\left(\frac{dz^2}{dxy}\right)\left(\frac{dz}{dx}\right)\left(\frac{dz}{dy}\right) + \frac{dz^2}{d^2y}\left(\frac{dz}{dx}\right)^2 < 0$

This shows that the Cobb-Douglas production function is strictly concave, thus it is quasiconcave.

Economists use graphs called isoquants which are curves where the value of z (production/utility) is fixed and a curve is plotted on the xy plane. Isoquants are basically contour curves for production or utility functions. Of course, economists want to maximize production or utility so they try to attain a solution of x and y that puts them at the isoquant with the highest z value, but this is subject to a constraining cost function as resources are generally limited.

Increasing returns to scale with linear marginal product/utility (Quasiconcave y/U)



This image also illustrates the quasiconcavity of the Cobb-Douglas production function and the relationship between isoquants and the Cobb-Douglas production function. As mentioned previously, there are many similarities between the Cobb-Douglas production function and more generalized utility functions. Just like the Cobb-Douglas production function, utility functions help economists determine how much a person or firm should "invest" in each object in order to maximize utility subject to a cost constraint.

One other important use of quasiconcave functions in economics is in game theory. Game theory studies how players in a game, an environment with a given set of rules, impact each other through their decision making. Economists are interested in finding equilibriums, decisions such that no player can make another decision and improve their results based on the decisions of

the other players. A common economic example of game theory involves two ice cream stands at a beach on a hot summer day. The ice cream vendors must choose where to set up their stand and the placement of one of the vendors stands impacts the other as people are most likely to go to the closest ice cream stand to their location on the beach. In equilibrium, both of the ice cream stands set up in the center of the beach so that they are the same distance away from all people. The “prisoner’s dilemma” is another famous example of game theory. We can find these equilibriums mathematically by modelling the “strategies" each player can choose from and their respective payoffs and then optimizing each player’s outcomes relative to the decisions of the other players. John von Neumann and others found that when these strategy functions are quasiconcave/quasiconvex they can be solved to find an equilibrium. Because quasiconcavity/quasiconvexity allows economists to solve for an equilibrium they often construct their models such that the equilibrium function is quasiconcave/quasiconvex. This is just an overview of the applications of quasiconcave functions in economics. Quasiconcave functions are a powerful tool in economics as they allow economists to model many different types of problems while also allowing them to find optimal solutions.

Nonlinear Programming in Machine Learning

1. Support Vector Machine (SVM):

SVM is a supervised learning machine learning method. It is also a classical nonlinear programming model. The basic idea of SVM is to find a hyperplane to separate and classify the data. However, this hyperplane has the constraint of maximizing the margin to the nearest data point on both sides of the hyperplane. In this section, we will discuss three types of SVM (Hard Margin SVM, Soft Margin SVM, and Kernel SVM) and one variant form (Lagrangian Dual Form) of nonlinear optimization problems which is formulated by the Karush-Kuhn and Tucker (KKT) condition.

1-1. Linearly decision boundary - Perfectly classified (hard margin)

Hard Margin SVM is a type of SVM which generates a linear decision boundary perfectly classifying the data points (figure 1). To formulate the problem into a nonlinear programming problem, we need to define a cost function first. The cost function adopted by SVM is shown as the green lines in figure 2. Compared to the Logistic Regression Classifier, which is another machine learning model to classify data points, SVM wants to separate the data points harder. We want SVM to predict $\hat{y} = \theta^T x$ to be 1 when the true label y is 1; we want SVM to predict $\hat{y} = \theta^T x$ to be -1 (not just 0) when the true label y is 0. In this way, we are able to find the hyperplane having a maximum margin to the nearest data point. This kind of cost function is the so-called "Hinge loss". The objective makes the cost function of SVM to be designed as follow (formula (1)):

$$J(\theta) = \sum_{i=1}^{m} \quad y^{(i)} cost_1(\theta^T x^{(i)}) + (1 - y^{(i)}) cost_0(\theta^T x^{(i)})$$

$$where\ cost_1 = max(0, 1 - \theta^T x), if\ y = 1; cost_0 = max(0, 1 + \theta^T x), if\ y = 0 \text{ ..... (1)}$$

By obtaining the cost function of SVM, we introduce another term, $\frac{1}{2}\sum_{j=1}^{n} \quad {\theta_j}^2$, to serve as the regularization purpose. We now are able to write the objective function of the SVM into the below form:

$$min_\theta \{C[\sum_{i=1}^{m} \quad y^{(i)} cost_1(\theta^T x^{(i)}) + (1 - y^{(i)}) cost_0(\theta^T x^{(i)})] + \frac{1}{2}\sum_{j=1}^{n} \quad {\theta_j}^2\} \text{..... (2)}$$

The C parameter is used to tune how hard we want to train our model. If C is large, we are going to focus more on the cost function of SVM (formula (1)); on the other hand, if C is small, we tend to penalize more on the regularization term so that we don't overtrain the model. In this fashion, we finally can turn the SVM problem into a nonlinear programming problem:

$$Min_\theta\{\sum_{j=1}^{n} \frac{1}{2}\theta_j{}^2\}$$

$$subject\ to\ \theta^T x^{(i)} >= 1, if\ y^{(i)} = 1$$

$$\theta^T x^{(i)} <= -1, if\ y^{(i)} = 0 .....(3)$$

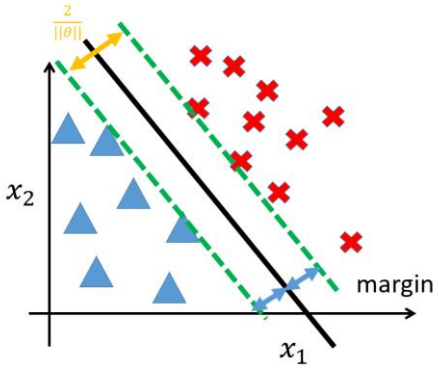


Figure 1. Graphical view of SVM

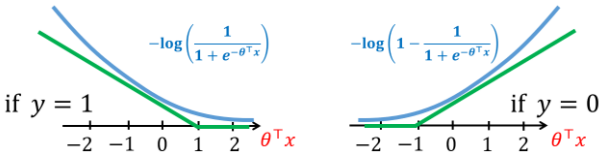


Figure 2. Hinge Loss

Another easier way to see how SVM could be formulated as nonlinear programming is simply to start from the graph. Since the purpose of SVM is to maximize the margin of the hyperplane which separates the data points, we could directly model it into formula (4). Simply modify the maximizing objective to a minimizing objective, we get formula (5).

$$Max_\theta\{\frac{2}{||\theta||}\}$$

$$subject\ to\ \ \theta^T x^{(i)} >= 1, if\ y^{(i)} = 1$$

$$\theta^T x^{(i)} <= -1, if\ y^{(i)} = 0 .....(4)$$

$$Min \quad_{\theta} \{\frac{||\theta||}{2} = \frac{\theta^T \theta}{2}\}$$

$$subject\ to \quad \theta^T x^{(i)} \quad >= \ 1, if\ y^{(i)} = 1$$

$$\theta^T x^{(i)} \quad <= \ -1, if\ y^{(i)} = 0 \ \ldots\ldots(5)$$

1-2. Linearly decision boundary - Not perfectly classified (soft margin)

With the barebone of section 1-1, let us discuss a more practical scenario, which is what if the linear decision boundary is not able to perform a perfectly classifying job (figure 3). If we want to deal with this problem, we could let r denotes the number of misclassification and put a penalty weight C to it (formula (6))

$$Min_{\theta}\{\frac{1}{2}\sum_{j=1}^{n} \quad \theta_j^{\ 2} \ + \ Cr\}$$

$$subject\ to \quad \theta^T x^{(i)} \quad >= \ 1, if\ y^{(i)} = 1$$

$$\theta^T x^{(i)} \quad <= \ -1, if\ y^{(i)} = 0 \ \ldots\ldots(6)$$

However, this problem is NP-hard. We then apply a convex relaxation to this problem by introducing a slack variable $\xi^{(i)}$ which denotes the tolerance of each $x^{(i)}$ to its correct margin (shown by the orange line in figure 3). The new formulation is as follow:

$$Min_{\theta}\{\frac{1}{2}\sum_{j=1}^{n} \quad \theta_j^{\ 2} \ + \ C\sum_{i} \quad \xi^{(i)} \ \}$$

$$subject\ to \quad \theta^T x^{(i)} \quad >= \ 1 - \xi^{(i)}, if\ y^{(i)} = 1$$

$$\theta^T x^{(i)} \quad <= \ -1 + \xi^{(i)}, if\ y^{(i)} = 0$$

$$\xi^{(i)} >= 0, \forall i \ \ldots\ldots(7)$$

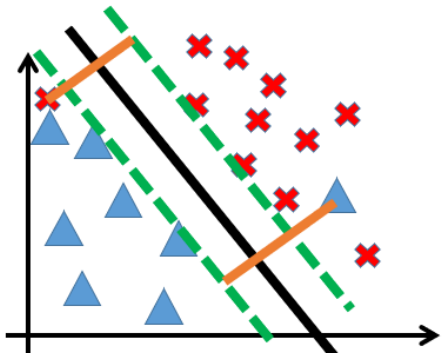


Figure 3. Not perfectly classified SVM

1-3. Nonlinearly decision boundary - Kernel SVM

After discussing the SVM that generates linear decision boundaries, let's talk about how SVM generates nonlinear decision boundaries. In figure 4, it is shown that we couldn't classify these data points by a simple linear decision boundary. However, if we map the data points to a higher dimension, we could successfully use a simple linear boundary to separate these data points. The applied mapping procedure is called the kernel function. A kernel takes two vectors in and outputs a scalar. There are many types of kernel to choose from when using kernel SVM. Table 1 consists of some commonly used kernels.

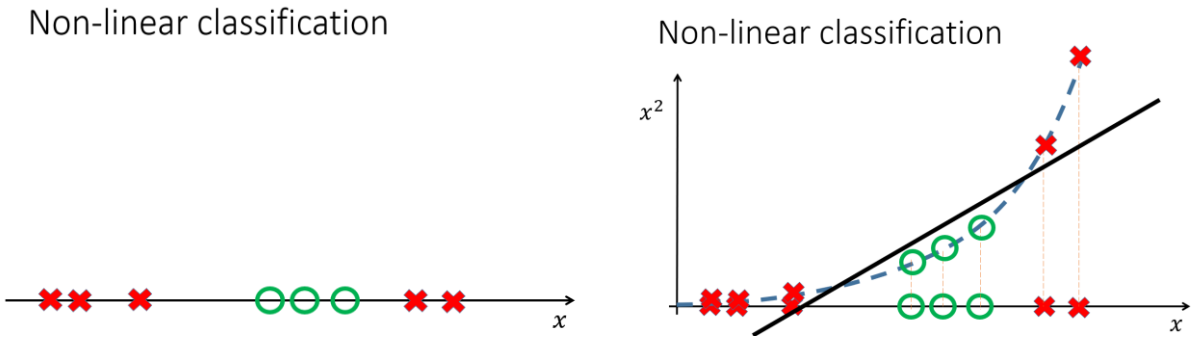


Figure 4a. Unable to separate data points with a linear decision boundary

Figure 4b. Able to separate data points with a linear decision boundary after mapping into a higher dimension

Table 1. Commonly used kernel for SVM	
Linear kernel	$K(x,z) = x^T z$
Gaussian (Radial Basis Function) kernel	$K(x,z) = exp[-\frac{1}{2}(x-z)^T \Sigma^{-1}(x-z)]$
Sigmoid kernel	$K(x,z) = tanh(ax^T z + b)$

With the help of the kernels, we are able to use SVM to construct nonlinear decision boundaries. The way to achieve this is to assign each data point $x^{(i)}$to be a landmark $l^{(i)}$. We then apply Gaussian kernel to get an output f_i. Gaussian kernel is really the similarity of the two input vectors (formula (8)). Finally, we could get a nonlinear decision boundary by substituting

$x^{(i)}$to f_i(figure 5) . Additionally, the nonlinear programming formulation is shown in formula (9).

$$f_i = similarity(x, l^{(i)}) = exp(-\frac{||x-l^{(i)}||^2}{2\sigma^2}) \ldots\ldots(8)$$

$$min_\theta C[\sum_{i=1}^{m} y^{(i)} cost_1(\theta^T f^{(i)}) + (1-y^{(i)}) cost_0(\theta^T f^{(i)})] + \frac{1}{2}\sum_{j=1}^{n} {\theta_j}^2$$

$$subject\ to\ \ \theta^T f_i >= 1, if\ y^{(i)} = 1$$

$$\theta^T f_i <= -1, if\ y^{(i)} = 0 \ldots\ldots(9)$$

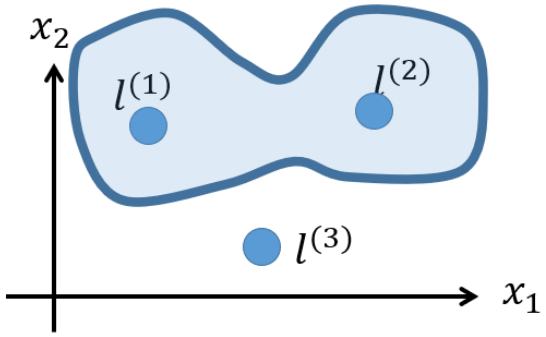


Figure 5. Kernel SVM

1-4. Lagrangian Dual Form

The most interesting part of SVM is that by applying the KKT condition, and Lagrangian dual function to formulate the Lagrangian dual form of SVM in formula (10) and (11) (here we take soft margin SVM as an example).

$$Primal: Min_\theta\{\frac{1}{2}\sum_{j=1}^{n} {\theta_j}^2 + C\sum_{i} \xi^{(i)}\}$$

$$subject\ to\ \ \theta^T x^{(i)} >= 1, if\ y^{(i)} = 1$$

$$\theta^T x^{(i)} <= -1, if\ y^{(i)} = 0 \ldots\ldots(10)$$

$$Dual: Min_{\alpha}\{\frac{1}{2}\sum_{i}\sum_{j} y^{(i)}y^{(j)}\alpha^{(i)}\alpha^{(j)}x^{(i)^T}x^{(j)} - \sum_{i}\alpha^{(i)}\}$$

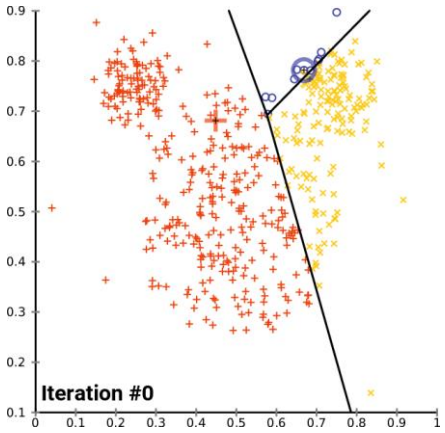
$$subject\ to\ 0 <= a^{(i)} <= C_i$$

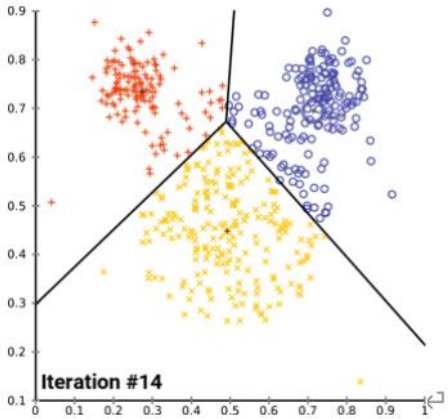
$$\sum_i y^{(i)}\alpha^{(i)} = 0 \ldots\ldots (11)$$

As observed, SVM is really a Quadratic Programming problem! Furthermore, we could use some delicate optimization solvers to solve the Quadratic Programming problem. These approaches would be further discussed in part 2 of the project.

2. K-means Clustering

Compared to SVM, which is a supervised learning method, another category of machine learning is the unsupervised learning method, here we introduce a vanilla unsupervised learning method, K-means clustering. K-means clustering is a machine learning technique that attempts to assign data points to one of K centroids, where K is the number of centroids. This is useful as it helps us determine which data points are most "similar" (have the smallest euclidean distance) from each other so that we can perform comparisons between each of the groupings or make predictions based off of the groupings. There are two main parts to the K-means clustering algorithm. The first is solving for the centroids and the second is assigning the data points to the centroids. This is an iterative process in which we must begin with some initial centroids and then the centroids evolve over each iteration until the overall distance is minimized between centroids and data points. Each of these steps individually is convex, but the process as a whole is not necessarily convex. This algorithm is related to the least squares regression as it uses the concept of minimizing euclidean distance to help us interpret data.





The following is the form of the nonlinear (quadratic) programming problem being solved in each iteration of the algorithm:

$$argmin_{c^{(1)},\ldots,c^{(m)},\mu_1 \ \ldots,\mu_k} \{J(c^{(1)},\ldots,c^{(m)},\mu_1 \ \ldots,\mu_k)\} = \{\frac{1}{m}\sum_{i=1}^{m} ||x^{(i)} - \mu_{c^{(i)}}||^2\}$$

Where $c^{(i)}$: index of which cluster example $x^{(i)}$ was assigned

μ_k: cluster k's centroid

$\mu_{c^{(i)}}$: cluster centroid which example $x^{(i)}$ was assigned

And finally, the algorithm is as follows:

- Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_k \in R^n$
- Repeat{

for i = 1 to m $<J(c^{(1)},\ldots,c^{(m)},\mu_1 \ \ldots,\mu_k) = \frac{1}{m}\sum_{i=1}^{m} ||x^{(i)} - \mu_{c^{(i)}}||^2>$

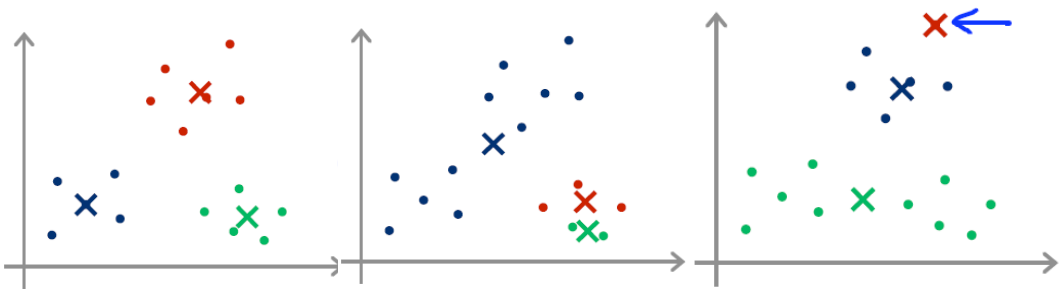
$c^{(i)}$ = index (from 1 to K) of cluster centroid closest to $x^{(i)}$

for k = 1 to K $<J(c^{(1)},\ldots,c^{(m)},\mu_1 \ \ldots,\mu_k) = \frac{1}{m}\sum_{i=1}^{m} ||x^{(i)} - \mu_{c^{(i)}}||^2>$

μ_k = average (mean) of points assigned to cluster k

}

Note that this algorithm is influenced by the initial starting centroids so it may give bad results if the centroid initialization is bad. It may be difficult to determine this beforehand so it is important to run the algorithm multiple times with different starting centroids to compare the final centroids. This is shown in the figure below.



Machine Learning is largely based on applications of different optimization techniques to make determinations about past and present data. These are just a few of the many applications of nonlinear programming in Machine Learning.

References

Quasiconcave Functions in Economics:

"EC202- PS1 Solutions." - https://www.economicsnetwork.ac.uk/drupal/sites/default/files/Ashley/Microeconomics%20Problem%20Set%20Solutions.pdf

Fuleky, P. (2006). Anatomy of a Cobb-Douglas Type Production/Utility Function in Three Dimensions. Mimeo, University of Washington, September 2006. - http://www2.hawaii.edu/~fuleky/anatomy/anatomy.html

Guerraggio, Angelo, and Elena Molho. "The Origins of Quasi-Concavity: a Development between Mathematics and Economics." *Historia Mathematica*, vol. 31, no. 1, 2004, pp. 62–75., doi:10.1016/j.hm.2003.07.001.

Wolitzky, Alexander. "Lectures 1—2: Choice, Preference, and Utility." Microeconomic Theory I - Fall 2015. Mar. 2021. - http://web.mit.edu/14.102/www/notes/lecturenotes1007.pdf

Portfolio optimization:

1. Umit Saglam. "Advanced Optimization and Statistical Methods in Portfolio Optimization and Supply Chain Management". https://core.ac.uk/download/pdf/190335403.pdf

2.Gerard Cornuejols, Reha Tutuncu. Book "Optimization Methods in Finance".

Machine Learning:

1. towards data science: Loss Function(Part III): Support Vector Machine

https://towardsdatascience.com/optimization-loss-function-under-the-hood-part-iii-5dff33fa015d

2. COMS 4721: Machine Learning for Data Science, lecture notes, Department of Electrical Engineering & Data Science Institute Columbia University

http://www.columbia.edu/~jwp2128/Teaching/W4721/Spring2017/slides/lecture_3-21-17.pdf

3. stack overflow: Why doesn’t k-means give the global minimum

https://stackoverflow.com/questions/14577329/why-doesnt-k-means-give-the-global-minima#:~:text=The%20k%2Dmeans%20problem%20is,minimum%2C%20but%20in%20exponential%20runtime.

4. Medium: K-means clustering

https://medium.com/dataregressed/k-means-clustering-the-premier-league-2592d1870dc5#:~:text=By%20minimising%20the%20cost%20function,into%20how%20the%20algorithm%20works%3A&text=Subplot%20c

5. Slides from ECE 5424/ CS 5824 Advanced Machine Learning Instructed by Jia-Bin Huang, Virginia Tech, The Bradley Department of Electrical and Computer Engineering