Name: Aryan Ritwajeet Jha
ID: 011807182

Project 12 Report: Smallest Enclosing Ellipsoid

Problem Description

Given a point cloud of m R^n points, find the smallest (in volume) ellipsoid which can completely enclose it.

The ellipsoid can be defined by this inequality:

$$E = \{x \in \mathbb{R}^n \mid (x - x_0)^T U^T D U (x - x_0) \leq 1\}$$

Where x_0 is its centre, **D is a diagonal matrix which contains the square of the inverse of axes lengths** and U is an orthogonal matrix of describing the 'rotation' of the ellipsoid.

Size specifications of the Original Problem Statement:

Unknowns:

- x_0: A Dense vector defined by n elements.
- D: A Diagonal Matrix defined by n elements.
- U: A Dense Matrix defined by $n \times n$ elements.

Inputs:

- $\{x_1, x_2, \dots x_m\}$: A set of m vectors defined by n values each.

Example:

An $\mathbb{R}^2$ ellipse defined by $\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 \leq 1$ whose axes lengths are a factor of a, b respectively, will have a volume which is a factor of the product of the axes' lengths i.e. $V = k * ab$. The elements of D will be $\{\frac{1}{a^2}, \frac{1}{b^2}\}$.

Approach

Defining the objective function

In general, D's elements are of the form $\{\frac{1}{d_1^2}, \frac{1}{d_2^2}, \dots, \frac{1}{d_n^2}\}$ and we wish to minimize the Volume $V(d) = d_1 * d_2 * \dots d_n$.

We may equivalently minimize the square of the Volume, i.e. $V^2(d) = d_1^2 * d_2^2 * \dots d_n^2$ which should not change the optimal solution at all.

An important thing to note (the motivation for which will be shortly explained later) here is that the expression for $V^2(d)$ is equivalent to the product of the inverse of the eigen values of D or equivalently, the inverse of the determinant of D (property of a pd diagonal matrix).

Thus, we can formulate the expression for $V^2(d)$ as:

Equation 1 Volume Expression using Eigen Values

$$V^2(d) = \frac{1}{\prod_{i=1}^{n} \lambda_i}$$

Where λ_i is the i-th eigenvalue of D.

Name: Aryan Ritwajeet Jha
ID: 011807182

Alternatively, based on the determinant expression:

Equation 2 Volume Expression using Matrix Determinant

$$V^2(d) = \frac{1}{|D|}$$

Equivalent Objective Function after some remodelling

Instead of trying to solve for $U^T DU$ separately, we can first attempt to solve for their product M i.e. we wish to first solve for the elements of $M = U^T DU$. M is also, like U and D, an $n \times n$ matrix. Once M is solved for, i.e. its $\frac{n(n+1)}{2}$ elements are computed (note that M will be a symmetric matrix), we can readily *diagonalize* M into $\Lambda^T D\Lambda$, here, Λ would just be the same as U.

Both M and D will have the same product of eigen values and determinant. Thus, we may now modify the objective function as:

Equation 3 Optimization Problem post modification

$$Min.\ V^2 = \frac{1}{|M|} \text{ where } M = U^T DU$$

s.t:

$$1 - (x_i - x_0)^T M(x_i - x_0) \geq 0 \ \forall\ i = 1{:}m$$

Thus, we obtain a constrained nonlinear problem with many inequality constraints. If we wish to use Derivative Based Optimization Methods, Augmented Lagrangian Method is way of solving this problem. We'll convert the inequalities into equalities using slack variables y, to obtain:

Task 1

Equation 4 Augmented Lagrangian Problem

$$Min.\ f(M) = \frac{1}{|M|} \text{ where } M = U^T DU$$

s.t:

$$c_i(x, M, y) = 1 - (x_i - x_0)^T M(x_i - x_0) - y_i^2 = 0 \ \forall\ i = 1{:}m$$

Thus, our Augmented Lagrangian Problem (switching V^2 with f for convention), or simply ALP becomes a problem with $n + \frac{n(n+1)}{2} + m = \frac{n^2}{2} + \frac{3n}{2} + m$ decision variables and m equality constraints.

Task 2

The objective function **is continuous**, as the value of $|M|$ will always be strictly positive which means that its reciprocal in f cannot have any discontinuities. ✓

The objective function **is coercive** (think in terms of $f = (d_1 * d_2 *..d_n)$^2 from the original problem). As a sequence of $\{d_i\}$ approaches infinity, the value of f will also approach. Here, d_i(s) can only lie in the constrained domain of positive values only, as they represent the axis lengths. (I hope that does not violate the definition of coerciveness) OK. we have 'coercivity' on the feasible domain.

With these two conditions being true, we already can guarantee a global minimum.

The objective function is **asymptotically lower bounded at zero**, as $|M| \in (0, \infty)$ $f \in (0, \infty)$

✓

Name: Aryan Ritwajeet Jha
ID: 011807182

Derivatives: Let's say there are three 'classes' of decision variables here, namely x_0, y, M (for now, just taking M as a matrix, though in an optimizer we'd convert it into a vector of length $\frac{n(n+1)}{2}$)

We'll analyze how the derivatives of the objective function and of the equality constraints look wrt the three classes of decision variables. If all six types of derivatives are continuous, then we can say that the optimization problem is differentiable.

$$\nabla_{x_0} f = 0, \qquad \nabla_y f = 0$$

As for $\nabla_M f = \nabla_M \left(\frac{1}{|M|}\right)$, I've referred to the Jacobi's method (which I've not derived, and am only using at face value) for computing the derivative of the determinant of a matrix wrt the matrix [1, 2] and then used the chain rule to obtain (not completely sure if I did it right):

$$\nabla_M f = -\frac{1}{|M|}(M^{-1})^T$$

I think it is correct.

For computing the derivatives of the constraints, I'm first opening up the compact form of $c_i(x_0, M, y_i)$ and then using the fact that $x_i^T M x_0 = x_0^T M x_i$ (both are scalars):

$$c_i = x_i^T M x_i - x_i^T M x_0 - x_0^T M x_i + x_0^T M x_0$$

$$\text{or } c_i = x_i^T M x_i - 2x_i^T M x_0 + x_0^T M x_0 \text{ (symmetric } M\text{, scalar function)}$$

$$\nabla_{x_0} c_i = -2Mx_i + 2Mx_0$$

$$\nabla_M c_i = (x - x_0)^T (x - x_0)$$

$$\nabla_y c_i = -2y_i$$

I don't see any discontinuities in any derivative, i.e. **all derivatives seem to be continuous only**. Thus, I'd say that the optimization problem **is differentiable**. ✓

Note: All M related derivatives need to be handled appropriately (elements may need to be converted-into/converted-back-from an equivalent vector representation to suit the optimization solver's syntax)

Task 3

Well I know one relationship and that is that **the number of points in the cloud m increase the number of decision variables (via slack variables y) in the ALP by the same amount**. But this may not hold for other kinds of problem formulation?

ok

Other than that I'm not aware of any other relationships.

Task 4

From derivative based approach, **we may use Augmented Lagrangian Programming (ALP) method**. We already have the model and functions in Task 1, and the appropriate function derivatives in Task 2 which is all we require to solve the problem.
Now, if for a high-valued n (in fact, the number of the decision variables increases quadratically and function/gradient computations increase cubically (such as computation of matrix determinant or eigen-values), this approach may be unsustainable.

Name: Aryan Ritwajeet Jha
ID: 011807182

One 'patch' to this problem may be to use approximate derivatives, which may at least alleviate the complexity of the gradient computation. ok

Or, we may use a derivative free approach like Nelder Mead, by modelling all equality constraints as soft penalizing constraints to the objective function. ok

Once M is solved for, we can then diagonalize it to obtain U and D. [3]

Task 5

Which points are 'outlier' points? If we wish to pre-filter the outlier points, some sort of clustering method may be used.

But if we wish to only let the solver decide it as part of the optimization algorithm, **we may append the objective function with some sort of regularization term**, say:

$$f_{distance}(x, x_0) = \lambda_{fixed}\Sigma(\mathrm{x}_0 - x_i)^2$$

distance is given by the ellipsoid equation

This soft constraint should discourage the ellipsoid to try to fit any x_i too distant from x_0. Because we're using a soft constraint, we have to be careful with the value of λ_{fixed}, as too low a value (in comparison to value of the volume of the ellipsoid) may do nothing, and too high a value may exclude most points as outliers.

As far as Task 2 analysis of this new appending function is concerned, **this L2 regularization term should not affect the conclusions of the analysis of the original problem in any way** (continuity, differentiability), apart from shifting the theoretical lower bound upwards (making it more positive from an already nonnegative lower bound).

Because soft constraints are a bit dicey, **I can think of a somewhat 'deterministic' method of excluding 'outlier' points** which will not be dependent on any regularization parameter. Basically, we solve for the original problem with all points. Once, x_0 is computed, we take some sort of distance norm for every x_i from it. We can then choose to exclude the furthest m_{exlude} points from the set, and resolve the problem. ✓ but may not result in optimal ellipsoid!

In fact, we can do an iterative approach, where we exclude a certain number of points (perhaps even just one, if m is not too many) after every time a new ellipsoid is solved for, and once we see that the volume of the ellipsoid is not decreasing enough in subsequent iterations (actually this condition is highly dependent on the point cloud, and I'm assuming that some points are clearly 'outliers', i.e. significantly away from a general, dense point cloud), we may stop solving for the ellipsoid.

References

1. Contributors to Wikimedia projects. (2023, May 04). Jacobi's formula - Wikipedia. Retrieved from https://en.wikipedia.org/w/index.php?title=Jacobi%27s_formula&oldid=1153212981
2. Proof for the derivative of the determinant of a matrix. (2024, May 01). Retrieved from https://mathoverflow.net/questions/214908/proof-for-the-derivative-of-the-determinant-of-a-matrix
3. Contributors to Wikimedia projects. (2024, April 12). Diagonalizable matrix - Wikipedia. Retrieved from https://en.wikipedia.org/w/index.php?title=Diagonalizable_matrix&oldid=1218564824
4. Nocedal, J., & Wright, S. J. . Numerical Optimization. Springer. Retrieved from https://link.springer.com/book/10.1007/978-0-387-40065-5
5. Prof. Tom Asaki's course notes as part of MATH 565 Nonsmooth Analysis and Optimization taught at Washington State University, Spring 2024. Problem Statement retrieved from

Name: Aryan Ritwajeet Jha
ID: 011807182

Project Descriptions: 2024-Spr-MATH-565-PULLM-1-01-03816-Nonsmooth Analy & Optimization. (2024, April 10). Retrieved from https://wsu.instructure.com/courses/1687163