Lecture Notes on Machine Learning: Frank-Wolfe for Minimum Enclosing Balls

Technical Report · February 2019

2 authors:

Christian Bauckhage
University of Bonn

Tiansi Dong
Fraunhofer Institute for Intelligent Analysis and Information Systems IAIS

Lecture Notes on Machine Learning
Frank-Wolfe for Minimum Enclosing Balls

Christian Bauckhage and Tiansi Dong
B-IT, University of Bonn

In this note, we show how the Frank-Wolfe algorithm applies to the problem of computing the minimum enclosing ball of a set of data.

The Minimum Enclosing Ball Problem

Previously[1], we studied the minimum enclosing ball problem which asks for the smallest Euclidean m-ball that contains all the column vectors of a given data matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{m \times n}$.

We saw that the center $\boldsymbol{c}^* \in \mathbb{R}^m$ and radius $r^* \in \mathbb{R}$ of this ball result from solving a constrained quadratic minimization problem. Analyzing its Lagrangian and the KKT conditions, we found that the corresponding dual maximization problem amounts to

$$\begin{aligned} \boldsymbol{\mu}^* = \underset{\boldsymbol{\mu}}{\operatorname{argmax}} \;\; & \boldsymbol{\mu}^\intercal \boldsymbol{z} - \boldsymbol{\mu}^\intercal \boldsymbol{X}^\intercal \boldsymbol{X} \boldsymbol{\mu} \\ \text{s.t.} \;\; & \boldsymbol{\mu}^\intercal \boldsymbol{1} = 1 \\ & \boldsymbol{\mu} \succeq \boldsymbol{0} \end{aligned} \tag{1}$$

where $\boldsymbol{\mu} \in \mathbb{R}^n$ is a vector of Lagrange multipliers, $\boldsymbol{0}$ and $\boldsymbol{1}$ denote the vectors of all zeros and ones, and the entries of $\boldsymbol{z} \in \mathbb{R}^n$ are given by $z_j = \boldsymbol{x}_j^\intercal \boldsymbol{x}_j$.

Finally, we saw that, upon solving (1), the center and radius of the sought after minimum enclosing ball can be computed as

$$\boldsymbol{c}^* = \boldsymbol{X} \boldsymbol{\mu}^* \tag{2}$$

$$r^* = \sqrt{\boldsymbol{\mu}^{*\intercal} \boldsymbol{z} - \boldsymbol{\mu}^{*\intercal} \boldsymbol{X}^\intercal \boldsymbol{X} \boldsymbol{\mu}^*}. \tag{3}$$

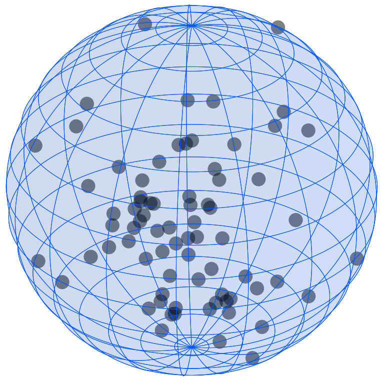

Figure 1: Minimum enclosing ball of a set of data points $\boldsymbol{x}_j \in \mathbb{R}^3$.

[1] C. Bauckhage and T. Dong. Lecture Notes on Machine Learning: Minimum Enclosing Balls. B-IT, University of Bonn, 2019

However, we did not yet discuss how to actually solve the problem in (1) but merely claimed there is a simple algorithm for this purpose. Indeed, the nature of problem (1) suggests to use the the Frank-Wolfe algorithm[2] and we next discuss the details as to why and how.

In order for our discussion to be more or less self-contained, we first recall general characteristics of the Frank-Wolfe algorithm before we discuss and demonstrate how it applies to our current problem.

[2] C. Bauckhage. Lecture Notes on Machine Learning: The Frank-Wolfe Algorithm. B-IT, University of Bonn, 2019b

The Frank-Wolfe Algorithm

The ***Frank-Wolfe algorithm*** in Fig. 2 constitutes an efficient and easy to implement procedure for solving constrained convex optimization problems of the form

$$\boldsymbol{x}^* = \underset{\boldsymbol{x} \in \mathcal{S}}{\operatorname{argmin}} \; f(\boldsymbol{x}) \tag{4}$$

where the objective function $f : \mathbb{R}^m \to \mathbb{R}$ is convex and differentiable and the feasible set $\mathcal{S} \subset \mathbb{R}^m$ is compact and convex.

guess a feasible point

$$\boldsymbol{x}_0^* \in \mathcal{S} \tag{5}$$

for $t = 0, \ldots, t_{\max}$ **do**

determine the step direction

$$\boldsymbol{s}_t = \underset{\boldsymbol{s} \in \mathcal{S}}{\operatorname{argmin}} \;\; \boldsymbol{s}^\intercal \nabla f(\boldsymbol{x}_t^*) \tag{6}$$

update the step size

$$\eta_t = \frac{2}{t+2} \tag{7}$$

update the current estimate

$$\boldsymbol{x}_{t+1}^* = \boldsymbol{x}_t^* + \eta_t \cdot [\boldsymbol{s}_t - \boldsymbol{x}_t^*] \tag{8}$$

Figure 2: The ***Frank-Wolfe algorithm*** iteratively solves convexity constrained convex optimization problems such as in (4).

Given an initial feasible guess $\boldsymbol{x}_0^*$ as to the solution, each iteration of the algorithm determines which point $\boldsymbol{s}_t \in \mathcal{S}$ minimizes the inner product $\boldsymbol{s}^\intercal \nabla f(\boldsymbol{x}_t^*)$ and then updates $\boldsymbol{x}_{t+1}^* = \boldsymbol{x}_t^* + \eta_t \cdot [\boldsymbol{s}_t - \boldsymbol{x}_t^*]$ where the step size $0 \leq \eta_t \leq 1$ decreases over time.

This conditional gradient scheme guarantees that updates never leave the feasible set and the efficiency of the algorithm stems from the fact that it turns the original problem into a series of simple linear optimization problems.

We also recall that the estimate $\boldsymbol{x}_t^*$ in iteration t is $O(1/t)$ away from the optimal solution.[3] This provides a convenient criterion for choosing the number $t_{\max}$ of iterations to be performed. For instance, if we had reason to believe that a precision of 0.01 is good enough for whatever problem we are dealing with, it would be sufficient to run $t_{\max} \in O(100)$ iterations.

[3] K.L. Clarkson. Coresets, Sparse Greedy Approximation, and the Frank-Wolfe Algorithm. *ACM Trans. on Algorithms*, 6(4), 2010

Frank-Wolfe for Minimum Enclosing Balls

In order to see, why and how the Frank-Wolfe algorithm applies to the minimum enclosing ball optimization problem in (1), we make two crucial observations:

recall that the Lagrangian dual of the minimum enclosing ball problem is

$$\mathcal{D}(\boldsymbol{\mu}) = \boldsymbol{\mu}^\intercal \boldsymbol{z} - \boldsymbol{\mu}^\intercal \boldsymbol{X}^\intercal \boldsymbol{X}\, \boldsymbol{\mu}$$

1. The Lagrangian dual $\mathcal{D}(\boldsymbol{\mu}) = \boldsymbol{\mu}^\intercal \boldsymbol{z} - \boldsymbol{\mu}^\intercal \boldsymbol{X}^\intercal \boldsymbol{X}\, \boldsymbol{\mu}$ is a concave function which is to say that $-\mathcal{D}(\boldsymbol{\mu})$ is convex. Accordingly, our current ***concave maximization problem***

$$\begin{aligned} \boldsymbol{\mu}^* = \underset{\boldsymbol{\mu}}{\operatorname{argmax}} \;\; & \mathcal{D}(\boldsymbol{\mu}) \\ \text{s.t.} \;\; & \boldsymbol{\mu}^\intercal \mathbf{1} = 1 \\ & \boldsymbol{\mu} \succeq \mathbf{0} \end{aligned} \tag{9}$$

can equivalently be expressed as a ***convex minimization problem***

$$\begin{aligned} \boldsymbol{\mu}^* = \underset{\boldsymbol{\mu}}{\operatorname{argmin}} \;\; & -\mathcal{D}(\boldsymbol{\mu}) \\ \text{s.t.} \;\; & \boldsymbol{\mu}^\intercal \mathbf{1} = 1 \\ & \boldsymbol{\mu} \succeq \mathbf{0}. \end{aligned} \tag{10}$$

2. The non-negative ($\boldsymbol{\mu} \succeq \mathbf{0}$) and sum-to-one ($\boldsymbol{\mu}^\intercal \mathbf{1} = 1$) constraints for this problem imply that any feasible solution must reside in the standard simplex

$$\Delta^{n-1} = \left\{ \boldsymbol{x} \in \mathbb{R}^n \;\middle|\; \boldsymbol{x} \succeq \mathbf{0},\; \boldsymbol{x}^\intercal \mathbf{1} = 1 \right\}. \tag{11}$$

Accordingly, we can write the problem in (10) in a very condensed form, namely

$$\boldsymbol{\mu}^* = \underset{\boldsymbol{\mu} \in \Delta^{n-1}}{\operatorname{argmin}} \; -\mathcal{D}(\boldsymbol{\mu}). \tag{12}$$

the minimum enclosing ball problem is a problem that can be solved using the Frank-Wolfe algorithm

Looking at the expression in (12) and noting that the standard simplex $\Delta^{n-1} \subset \mathbb{R}^n$ is a compact convex set, we now recognize that the dual of the minimum enclosing ball problem is an instance of the kind of problems we may solve via Frank-Wolfe optimization.

To be specific, Tab. 1 summarizes how the general constrained convex optimization problem in (4) specializes to the one in (12).

Next, we show how to specialize the general form of the Frank-Wolfe computations in (5), (6), and (8) to the minimum enclosing ball problem in (12).

Table 1: Comparison of the constituent parts of the general problem in (4) and our specific problem in (12).

general problem	specific problem
$\boldsymbol{x}$	$\boldsymbol{\mu}$
$f(\boldsymbol{x})$	$-\mathcal{D}(\boldsymbol{\mu})$
$\mathcal{S}$	Δ^{n-1}

With respect to adapting the initialization in (5) to the minimum enclosing ball problem, we note that the midpoint of Δ^{n-1} is contained in Δ^{n-1} and therefore feasible. Hence, we simply let

$$\boldsymbol{\mu}_0^* = \tfrac{1}{n}\mathbf{1}. \tag{13}$$

Finding the step direction in (6) involves the gradient of the objective function at the current estimate. In our specific case, this gradient is given by

$$\nabla\Big(-\mathcal{D}(\boldsymbol{\mu}_t^*)\Big) = -\nabla\mathcal{D}(\boldsymbol{\mu}_t^*) = 2\,\boldsymbol{X}^\intercal\boldsymbol{X}\,\boldsymbol{\mu}_t^* - \boldsymbol{z} \tag{14}$$

X=(2,1024)
u=(1024,)

(1024,2)*(2,1024) = 1024

so that we have to solve

$$\boldsymbol{s}_t = \underset{\boldsymbol{s}\in\Delta^{n-1}}{\operatorname{argmin}}\ \boldsymbol{s}^\intercal\big[2\,\boldsymbol{X}^\intercal\boldsymbol{X}\,\boldsymbol{\mu}_t^* - \boldsymbol{z}\big]. \tag{15}$$

Note that the expression $\boldsymbol{s}^\intercal\big[2\,\boldsymbol{X}^\intercal\boldsymbol{X}\,\boldsymbol{\mu}_t^* - \boldsymbol{z}\big]$ on the right hand side of (15) is linear in $\boldsymbol{s}$ and needs to be minimized over Δ^{n-1} which is a compact convex set. Also recall that a minimum of a linear function over a convex set is necessarily attained at a vertex of said set. As the vertices of Δ^{n-1} correspond to the standard basis vectors $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n$ of $\mathbb{R}^n$, we therefore only have to determine which of these minimizes the above inner product. Hence, the problem in (15) simplifies to

recall that minima of linear functions over convex sets are attained at vertices and that the vertices of Δ^{n-1} are the standard basis vectors $\boldsymbol{e}_j \in \mathbb{R}^n$

C. Bauckhage. Lecture Notes on Machine Learning: Convex Functions. B-IT, University of Bonn, 2019a

$$\boldsymbol{s}_t = \underset{\boldsymbol{e}_j\in\mathbb{R}^n}{\operatorname{argmin}}\ \boldsymbol{e}_j^\intercal\big[2\,\boldsymbol{X}^\intercal\boldsymbol{X}\,\boldsymbol{\mu}_t^* - \boldsymbol{z}\big] \tag{16}$$

which, in order to explicate that the resulting $\boldsymbol{s}_t$ will be a standard basis vector $\boldsymbol{e}_i$, can also be written as

$$\boldsymbol{e}_i = \underset{\boldsymbol{e}_j\in\mathbb{R}^n}{\operatorname{argmin}}\ \boldsymbol{e}_j^\intercal\big[2\,\boldsymbol{X}^\intercal\boldsymbol{X}\,\boldsymbol{\mu}_t^* - \boldsymbol{z}\big]. \tag{17}$$

Next, we note that the inner product between a standard basis vector $\boldsymbol{e}_j$ and an arbitrary vector $\boldsymbol{v}$ amounts to $\boldsymbol{e}_j^\intercal\boldsymbol{v} = v_j$ where v_j is the j-th entry of $\boldsymbol{v}$. This is to say that (17) can be solved without computing inner products: we just need to determine the index of the smallest component of the gradient, namely

$$i = \underset{j\in\{1,\ldots,n\}}{\operatorname{argmin}}\ \big(2\,\boldsymbol{X}^\intercal\boldsymbol{X}\,\boldsymbol{\mu}_t^* - \boldsymbol{z}\big)_j\,. \tag{18}$$

Given these considerations, the general update step in (8) thus specializes to

$$\boldsymbol{\mu}_{t+1}^* = \boldsymbol{\mu}_t^* + \frac{2}{t+2}\cdot\big[\boldsymbol{e}_i - \boldsymbol{\mu}_t^*\big] \tag{19}$$

where we used (7) to compute the step size.

All in all, the rather simple algorithm in Fig. 3 will therefore solve the minimum enclosing ball problem in (1).

$\boldsymbol{\mu}_0^* = \frac{1}{n}\mathbf{1}$

for $t = 0, \ldots, t_{\max}$ **do**

$$i = \underset{j\in\{1,\ldots,n\}}{\operatorname{argmin}}\ \big(2\,\boldsymbol{X}^\intercal\boldsymbol{X}\,\boldsymbol{\mu}_t^* - \boldsymbol{z}\big)_j$$

$$\boldsymbol{\mu}_{t+1}^* = \boldsymbol{\mu}_t^* + \frac{2}{t+2}\cdot\big[\boldsymbol{e}_i - \boldsymbol{\mu}_t^*\big]$$

Figure 3: *Frank-Wolfe algorithm for the minimum enclosing ball problem.*

Summary

In this note, we discussed how to apply the Frank-Wolfe algorithm for constrained convex optimization in order to solve the dual of the minimum enclosing ball problem.

Exploiting particular characteristics of linear functions over the standard simplex, we found that the corresponding procedure can be condensed into just four lines of pseudo-code.

Running the algorithm in Fig. 3 yields the solution to the problem in (1) which can then be used to compute center and radius of the sought after ball according to (2) and (3), respectively.

Acknowledgments

This material was prepared within project P3ML which is funded by the Ministry of Education and Research of Germany (BMBF) under grant number 01/S17064. The authors gratefully acknowledge this support.

References

C. Bauckhage. Lecture Notes on Machine Learning: Convex Functions. B-IT, University of Bonn, 2019a.

C. Bauckhage. Lecture Notes on Machine Learning: The Frank-Wolfe Algorithm. B-IT, University of Bonn, 2019b.

C. Bauckhage and T. Dong. Lecture Notes on Machine Learning: Minimum Enclosing Balls. B-IT, University of Bonn, 2019.

K.L. Clarkson. Coresets, Sparse Greedy Approximation, and the Frank-Wolfe Algorithm. *ACM Trans. on Algorithms*, 6(4), 2010.