Distributionally Robust Optimization and Application in Machine Learning

Report submitted as the part of
B.Tech project

by

Devyani Tushar Maladkar
Siddharth Singh Solanki

Under the guidance of

Dr. Divya Padmanabhan

SCHOOL OF MATHEMATICS AND COMPUTER SCIENCE

INDIAN INSTITUTE OF TECHNOLOGY GOA

ACKNOWLEDGEMENTS

We would like to thank Dr. Divya Padmanabhan for her guidance and constant support during the project. Her faith in us and our ability to work on this topic was our biggest motivation.

Devyani Tushar Maladkar
Siddharth Singh Solanki

Contents

List of Figures

Chapter 1

Understanding hardness and tractability of optimization problems involving Random Variables

1.1 Optimization with Independent Random Variables

1.1.1 Introduction

In this portion of the study, we worked on understanding the complexity and formulations for quantities relating to the following optimization problem.

$$\begin{aligned} Z(c) = max \;\; & c'x \\ & s.t \\ & x \in X \end{aligned} \tag{1.1}$$

where X is the feasible region and Z(c) is the value of the optimal objective as a function of the vector c. We study the computation of various quantities, treating c as a random variable and hence Z(c) is studied as a random linear program.

CDF-SUM

INSTANCE: A random vector $\tilde{c} = (\tilde{c}_1, ..., \tilde{c}_n)$ with n mutually independent, discrete random variables where the marginal probabilities are given by the rational numbers

$p_i(c_i) = P(\tilde{c}_i = c_i)\ \forall c_i \in C_i, i \in [n]$, with each set C_i given by a finite set of rational numbers and a rational number d.

OUTPUT: Probability that the sum of random variables is less than or equal to d, given by

$$P(\sum_{i=1}^{n} \tilde{c}_i \leq d)$$

ESF-SUM

INSTANCE: A random vector $\tilde{c} = (\tilde{c}_1, ..., \tilde{c}_n)$ with n mutually independent, discrete random variables where the marginal probabilities are given by the rational numbers $p_i(c_i) = P(\tilde{c}_i = c_i)\ \forall c_i \in C_i, i \in [n]$, with each set C_i given by a finite set of rational numbers and a rational number d.

OUTPUT: Expected surplus of the sum of random variables exceeding a number d, given by

$$E[\sum_{i=1}^{n} \tilde{c}_i - d]^+$$

Both the above problems are shown to be #P complete. We provide the proof sketch for the ESF-SUM below.

Theorem 1 *ESF-SUM is P-hard, even when each random variable is restricted to take only two possible values with equal probabilities.*

Proof Sketch We know that CDF-SUM is P-hard, we thus need to only show a reduction from CDF-SUM to ESF-SUM.

$$\begin{aligned} P(\sum_{i=1}^{n} \tilde{c}_i \leq d) &= 1 - P(\sum_{i=1}^{n} \tilde{c}_i \geq d+1) \\ &= 1 - \sum_{t=d+1}^{\infty} P(\sum_{i=1}^{n} \tilde{c}_i \leq t) + \sum_{t=d+2}^{\infty} P(\sum_{i=1}^{n} \tilde{c}_i \leq t) \end{aligned} \tag{1.2}$$

Now for any non-negative integer random vector $\tilde{c}$

$$E[\tilde{c}] = \sum_{t=1}^{\infty} P(\tilde{c} \geq t)$$

Hence we have,

$$P(\sum_{i=1}^{n} \tilde{c}_i \leq d) = 1 - E[\sum_{i=1}^{n} \tilde{c}_i - d]^+ + E[\sum_{i=1}^{n} \tilde{c}_i - (d+1)]^+$$

Thus we can solve CDF-SUM with a polynomial number of calls to ESF-SUM and hence ESF-SUM is P-hard.

In the special case of random variables as Bernoulli, with support restricted on {0,1}, the calculation for both of the above quantities is tractable and can be solved in polynomial time.
On the other hand similar-looking quantities CDF-MAX $P(\max_{i\in[n]} \tilde{c}_i \leq d)$ and ESF-MAX $E[\max_{i\in[n]} \tilde{c}_i - d]^+$ are solvable in polynomial time without any support restrictions.
The results obtained for the above computations give us an idea of the role the setting (independence, size of support, nature of objective) has to play with regards to the tractability of the computation. We can in some instances reduce the support by considering the Bernoulli setting and obtain polynomial solvable instances. In another case, we can replace the summation of RVs by the maximum of RVs, thereby reducing the amount of information we need about the support of the random variable to solve the linear program.

1.1.2 Counting problems in graph

In this section, we summarize the learning from the complexity analysis of related network optimization problems that were studied. The problems studies were 1-0 PERMANENT computation, BIPARTITE VERTEX COVER, BIPARTITE PERFECT MATCHING, BIPARTITE INDEPENDENT SET, MIN-SIZE s-t CUTS.
Extending from the graph counting problems, we also studied the evaluation of network reliability under randomness. The s-t connectedness reliability problem is formulated as below.

s-t CONNECTEDNESS RELIABILITY
INSTANCE: A directed graph G = (V, E) with two specified nodes $s, t \in V$ where the arcs are operational with independent probabilities given by rational numbers $p_{ij} \in [0, 1]$ for $(i, j) \in E$ (we allow for directed multi-graphs).
OUTPUT: Probability there is a directed path of operational edges from node s to node t in G.

The above problem is #P-hard even when we assume that the failure probabilities for all the arcs are identical. Similar to the CDF-SUM and ESF-SUM there are formulations such as the CDF-MAX FLOW, MEAN-MAX FLOW, CDF-PERT, MEAN-PERT. These optimization problems are also #P-hard, even in the scenario when the flow on each arc is restricted to {0,1}. Drawing from the intuition of the problems we saw earlier, here relaxing of the support set does not give significant benefit. This can be because there is an additional contribution from the network structure itself, that affects the optimization problem which was absent in the previous problems.

Understanding Polynomial-Time Solvable Instances

The complexity arising in the network optimization problems is influenced by both the structure of the graph and the support set. The graphs, when in series-parallel form, we can obtain polynomial-time solvable formulations when we apply the support constraints.

1.1.3 Two Stage Stochastic Optimization

Let us suppose that we are dealing with problems that need to take two sequential decisions and the latter depends on the value of the first, then we are in the domain of two stage stochastic problems.
For modeling such problems SO-SUM is used as given below,

$$min\{(1-\eta)x + E[\sum_{i=1}^{n} \tilde{c}_i - x] \mid x \in \mathbb{R}\}$$

In this formulation, the first stage loss calculation is given by $(1-\eta)x$. The second stage cost, involving expectation, can also be formulated as a linear program. The linear program's decision variable in turn depends on the value of first stage realization of x.
Now using the #P-hardness results from previous sections we can easily show that SO-SUM is also #P hard, even for independent RVs with two point distributions.
This hardness again arises because the support of the second stage cost is exponentially large, even if marginal distributions of the random vector take only two values.

But if we restrict marginal supports then we can derive some polynomial time instances. Network reliability problem can also be formulated as a two stage stochastic optimization problem known as SO-REL

$$min\{-cx + E[Q(x, \tilde{u})] \mid 1 \geq x \geq 0\}$$

Here the first stage revenue is -cx and the second stage revenue is the optimal value of a network optimization problem which we will not delve into. As seen previously, SO REL can be proven to be #P hard using the results from previous sections, but here the hardness arises from the network structure of the graph.

1.2 Optimization with Random Variables with fixed univariate marginals

1.2.1 Introduction

In this section, we understand and study the optimization problem in Section 1.1, taking the joint distributions of the random vector c with fixed univariate marginal distributions. In particular, we study the formulation for the Expected value of the maximization problem introduced in section 1.1 and identification of the instances where the tightest upper bound is efficiently computable. The characterisation of the random variables that are used for the section is given by the assumption below.

Assumption Each random variable $\tilde{c}_i$ is discrete with probabilities given by rational numbers $p_i(c_i) = P(\tilde{c}_i = c_i)$ for $c_i \in C_i$, where C_i is a finite set of rational numbers. Let $p_i = (p_i(c_i); c_i \in C_i)$ for $i \in [n]$. The marginal probabilities satisfy the conditions $p_i(c_i) \geq 0$ for all $c_i \in C_i$ and $c_i \in C_i p_i(c_i) = 1$ for each $i \in [n]$.

The upper bound of the expected value, where the random vector c has the univariate marginals given by p_i and joint distribution θ, is given by Z^*,

$$\begin{aligned} Z^* &= \max_{\theta \in \Theta(p_1, p_2, \ldots, p_n)} \mathbb{E}_\theta[Z(\tilde{(c)})] \\ &= \max_{\theta \in \Theta(p_1, p_2, \ldots, p_n)} \mathbb{E}_\theta[\max\{\tilde{c}' x \mid x \in X\}] \end{aligned} \tag{1.3}$$

Z_* is the optimal value of the finite dimensional linear program :

$$\begin{aligned} Z^* = & \max_{\theta(c);c\in C} \sum_c \theta(c) Z(c) \\ & s.t \sum_{c\in C;c_i=\tilde{c}_i} \theta(c) = p_i(\tilde{c}_i), \qquad \forall \tilde{c}_i \in C_i, \forall i \in [n], \\ & \sum_{c\in C} \theta(c) = 1 \\ & \theta(c) \geq 0, \qquad \forall c \in C \end{aligned} \tag{1.4}$$

Here the decision variables are c and θ the corresponding joint distribution for the random vector c $\in C = C_1 \times C_2 \times ... \times C_n$. The support set is C_i for i^{th} component of random vector c. Hence, the possible decision variables are exponential in the input $|C| \leq \prod_{i=1}^n |Ci| \leq (max(Ci))^n$, where $c \in C$. Further, the optimization problem $Z(c)$ itself may be NP-hard to compute for a given vector c. We study the following theorems which help us understand the above formulations.

Theorem 2 *a) Suppose $X \subseteq \{0,1\}_n$ and the marginals satisfy the Assumption. Define :*

$$Z_u^* = \min_d (Z(d) + \sum_{i\in[n]}) \mathbb{E}[\tilde{c}_i - d_i]^+$$

where $d = (d_1, d_2 ... d_n)$. Then $Z^* = Z_u^*$.
b) Suppose the deterministic optimization is solvable in polynomial time for each $c \in conv(C)$. Then Z^* is computable in polynomial time.

We give a proof sketch below.
a) Proof Sketch
To recall

$$Z^* = \max_{\theta\in\Theta(p_1,p_2,...,p_n)} \mathbb{E}_\theta[Z(\tilde{c})]$$

where Θ is set of all distributions with fixed marginals for each c_i, θ is one such distribution for $\tilde{c}$ the random vector.
To prove $Z^* = Z_u^*$, we do so in 2 steps.

Step 1) We prove $Z^* \leq Z_u^*$

For any feasible solution $x \in X \subseteq \{0,1\}_n$ and vector $d \in \mathbb{R}_n$. We can write

$$\begin{aligned} c' &= d'x + (c-d)'x \\ &\leq \max_{x \in X} d'x + \sum_{i \in [n]} [c_i - d_i]^+ \end{aligned}$$

Taking max for $d'x$ and component-wise

maximum for the second term. Since RHS is general in x

$$Z(c) \leq Z(d) + \sum_{i \in [n]} [c_i - d_i]^+$$

We can consider Expectation on both sides since RHS is separable in c_i. We also take min over d.

$$\mathbb{E}_\theta[Z(\tilde{c})] \leq \min_d (Z(d) + \sum_{i \in [n]} \mathbb{E}[c_i - d_i]^+)$$

where $\theta \in \Theta(p_1, p_2, ..., p_n)$.
Since this holds for any θ in particular it should hold for Z^* (the optimal). Hence $Z^* \leq Z_u^*$.

Step 2) Now to show $Z^* \geq Z_u^*$.
We can write Z_u^* as an LP. (It is a dual form).

$$\begin{aligned} Z_{u,d}^* = \min_{t,d,y} \quad & t + \sum_{i \in [n]} \sum_{c_i \in C_i} y_i(c_i) \\ s.t \quad & t \geq d'x && \forall x \in X \subset \{0,1\}_n \\ & y_i(c_i) \geq p_i(c_i)(c_i - d_i) && \forall c_i \in C_i, \forall i \in [n] \\ & y_i(c_i) \geq 0, && \forall c_i \in C_i, \forall i \in [n] \end{aligned} \tag{1.5}$$

The primal form is as below.

$$\begin{aligned} Z_{u,p}^* = \max_{\lambda,\gamma} \quad & \sum_{i \in [n]} \sum_{c_i \in C_i} p_i(c_i) c_i \gamma_i(c_i) \\ s.t \quad & \sum_x \lambda(x) = 1 \\ & \lambda(x) \geq 0 && \forall x \in X \subset \{0,1\}_n \\ & 0 \leq \gamma_i(c_i) \leq 1 && \forall c_i \in C_i, \forall i \in [n] \\ & \sum_{x \in X; x_i = 1} \lambda(x) = \sum_{c_i \in C_i} p_i(c_i) \gamma_i(c_i), && \forall i \in [n] \end{aligned} \tag{1.6}$$

At optimality, γ^*, λ^* are optimal values of the primal and t^*, d^*, y^* are the optimal values for the dual. By strong duality we will have :

$$\begin{aligned} Z^* &= t^* + \sum_{i \in [n]} \sum_{c_i i} y_i^*(c_i) \\ &= Z_{u,p}^* \\ &= \sum_i \sum_{c_i} p_i(c_i) c_i \gamma_i^*(c_i) \end{aligned} \tag{1.7}$$

We use the optimality condition and values of decision variables to create a distribution θ^*. If we observe λ^*, γ^* they have the nature of probability distribution like constraints.

We then use constructed θ^* and compute Expectation, which must be less than Z^*.

$$\mathbb{E}_{\theta^*} \leq Z^*$$

The expectation, by virtue of construction of θ^* evaluates to Z_u^*.

$$Z_u^* \leq Z^*$$

and from Step 1 and Step 2 $Z_u^* = Z^*$.

b) Proof Sketch We assume Z(c) to be polynomial time computable. Using the equivalence of separation and optimization we can show Z^* is computable in polynomial time.

Conclusion

The problems analyzed helped us in understanding optimization involving random variables. The common #P-hard results involving independent random variables and computation of quantities under different settings indicate the evolution of tractability under different assumptions. Overall, it provided us with an understanding of stochastic programming in different settings of probability distributions. The proofs and theory presented were obtained from [3].

Chapter 2

Distributionally Robust Optimization

2.1 Introduction

In machine learning, particularly in regression and classification problems, we often come across the below formulation of stochastic programming:

$$\min_{x \in X} \mathbb{E}[L(x, \xi)]$$

The probability is associated with the random vector ξ and L is often the loss function, the problem formulation then becomes an Empirical Risk Minimisation formulation. In the above formulation, the loss function can often be a piece-wise affine function such as the hinge-loss function L(x)=max(0,1-x). The surplus function shown below is also of a similar form, where c is a random vector with n mutually independent discrete random variables, where the marginal probabilities are given by rational numbers, and d is a threshold.

$$[\sum_{i=1}^{n} c_i - d]_+$$

The problem of computing Expectation of the surplus function, over the space of random vectors c is the well-known problem of ESF-SUM which is known to be #P-hard. Thus in some instances of the above stochastic program, computing the objective value is itself intractable. This is one of the shortcomings of the above formulation known as the curse of dimensionality.
The second shortcoming, known as the optimizer's curse, arises from the Probability

Distribution of ξ which is required to perform the computation. Typically, we do not have the true probability distribution and must infer it from data. The data itself may have biases which may lead to poor out of sample performance. In machine learning, our goal is to improve the out-of-sample performance given only training data obtained from the true probability distribution. More details can be found in [1].

Distributionally Robust Optimization (DRO) is one approach to obtain a decision x, using the training data, such that we can provide good out-of-sample performance guarantees. This approach involves optimizing over a set of distributions for ξ. This set is the ambiguity set and consists of possible distributions from which the data can come.

Wasserstein metric

The Wasserstein distance between two distributions Q and Q' supported on the same space of random variables is defined as,

$$W(\mathbb{Q}, \mathbb{Q}') := \inf_{\Pi} \left\{ \int_{\Xi^2} d(\xi, \xi')\Pi(d\xi, d\xi') \leq \epsilon \;\middle|\; \begin{array}{l} \Pi\text{is a joint distribution of } \xi \text{ and } \xi' \\ \text{with marginals } \mathbb{Q} \text{ and } \mathbb{Q}', \text{ respectively} \end{array} \right\}$$

where d is a transportation metric on Ξ.

The Wasserstein metric is the optimal solution to the transportation problem. It corresponds to the minimal cost of moving the distribution from Q to Q', where the cost of moving one unit from ξ to ξ' is given by the transportation metric $d(\xi, \xi')$. This can be easily seen from the Kantorovich-Rubinstein Theorem which gives the Wasserstein metric as the LP dual of the above formulation. The value of d takes different forms one such is norm induced distance.

Ambiguity Sets

The ambiguity set is defined as a Wasserstein ball. It is a ball consisting of probability distributions centred around a particular distribution at a distance less than epsilon. The empirical distribution below is considered the centre of the ball, from which distances are taken.

$$\hat{P}_N = \frac{1}{N}\sum_{i=1}^{N} \delta_{\hat{\xi}_i}(\xi)$$

where, $\delta_{\hat{\xi}_i}(\xi)$ is the dirac-delta function. The ambiguity sets have specific desired properties that make them a good fit:

- **Finite Sample Guarantees**: For a given confidence requirement, we can obtain the Wasserstein radius such that the true distribution lies in the Wasserstein ball with this confidence.
- **Asymptotic Guarantees**: Convergence to optimal values occurs when the sample size is taken to infinity.
- **Tractability**: We can get tractable formulations such as convex optimization problems and linear/conic programs.

The proof for the above requirements comes from statistics and probability theory which were not delved in. We only implicitly make use of these. For more information on the definitions and desired properties [1] and [2] is the reference.

2.2 Tractable Formulation

In this section, we provide the derivation for the tractable formulation for DRO. The formulation is applicable for classification problems with a piece-wise affine function $L(z) = max_{j\in J}(a_j z + b_j)$. The labels y_i belong to the set $\{+1, -1\}$. The formulation and its forms are studied from [1],[2],[4]. The formulation is as given below.

$$\begin{aligned}
&\inf_{w,\lambda,s_i,p_{ij}^+,p_{ij}^-} \sup_{Q\in B_\epsilon(\hat{P_n})} \mathbb{E}^Q[\ell(\langle w, x\rangle, y)] \\
&= \inf_{w,\lambda,s_i,p_{ij}^+,p_{ij}^-} \lambda\epsilon + \frac{1}{N}\sum_{i=1}^{N} s_i \\
&s.t \\
&\quad S_x(a_j\hat{y}_i w - p_{ij}^+) + b_j + \langle p_{ij}^+, \hat{x}_i\rangle \le s_i, && i\in[N], j\in[J] \\
&\quad S_x(-a_j\hat{y}_i w - p_{ij}^-) + b_j + \langle p_{ij}^-, \hat{x}_i\rangle - \kappa\lambda \le s_i, && i\in[N], j\in[J] \\
&\quad \|p_{ij}^+\|_* \le \lambda, \|p_{ij}^-\|_* \le \lambda && i\in[N], j\in[J] \\
&\quad s_i \ge 0, && i\in[N]
\end{aligned} \tag{2.1}$$

Here $B_\epsilon[\hat{P}_n]$ is the Wasserstein ball centered around the empirical distribution and radius ϵ.
$\hat{P}_n = \frac{1}{N}\sum_{i=1}^{N} \delta_{\hat{\xi}_i}(\xi)$: Empirical Distribution
$\|.\|_*$: Dual Norm
$(\hat{x}_i, \hat{y}_i)$: Data Point $\in \Xi = X \times Y$
κ : Cost of label switching.

Proof
For simplicity let us begin by considering the inner problem.

$$\begin{aligned} & \sup_{Q \in B_\epsilon(\hat{P_n})} \quad \mathbb{E}^Q[\ell(\langle w, x\rangle, y)] \\ = & \sup_{\Pi} \int_{\Xi^2} \ell(\xi)\Pi(d\xi, d\xi') \\ s.t \quad & \int_{\Xi^2} d(\xi, \xi')\Pi(d\xi, d\xi') \leq \epsilon \end{aligned}$$

Π is a joint distribution of ξ and ξ' with marginals Q and $\hat{P}_N$.
Here we used the definition of the Wasserstein metric and ball, while opening the Expectation in terms of the loss and probability.
Since we know that ξ, has marginal probability defined using the empirical distribution, in the next step we make use of it.

$$\begin{aligned} \Pi(d\xi, d\xi') &= P(d\xi)P(d\xi \mid d\xi') \\ &= \frac{1}{N}\sum_{i=1}^{N} \delta_{\hat{\xi}_i(d\xi)} Q^i(d\xi) \end{aligned}$$

$Q^i(d\xi)$ is the conditional probability for $d\xi$ given $\xi' = \hat{\xi}_i$. From above we have.

$$\begin{aligned} & \sup_{Q^i} \frac{1}{N}\sum_{i=1}^{N} \int_{\Xi} \ell(\xi) Q^i(d\xi) \\ s.t \quad & \frac{1}{N}\sum_{i=1}^{N} \int_{\Xi} d(\xi, \xi') Q^i(d\xi) \leq \epsilon \\ & \int_{\Xi} Q^i(d\xi) = 1, \qquad\qquad i \in [N] \end{aligned}$$

Strong duality can be shown to hold for $\epsilon 0$. Hence we have the dual as.

$$\begin{aligned}
&\sup_{Q \in B_\epsilon(\hat{P_n})} && \mathbb{E}^Q[\ell(\langle w, x\rangle, y)] \\
&= && \inf_{\lambda, s_i} \lambda\epsilon + \frac{1}{N}\sum_{i=1}^{N} s_i \\
&s.t && \sup_{\xi \in \Xi} \ell(\xi) - \lambda d(\xi, \xi') \leq s_i, && i \in [N] \\
& && \lambda \geq 0
\end{aligned}$$

We define the transportation metric $d((x, y), (x', y'))$ as

$$d((x, y), (x', y')) = \|x - x'\| + \frac{\kappa}{2}|y - y'|$$

where, $\|.\|$ is a norm in the input space and κ is the cost of switching a label. Henceforth, we will substitute $\xi = (x, y)$.

$$\begin{aligned}
&= && \inf_{\lambda, s_i} \lambda\epsilon + \frac{1}{N}\sum_{i=1}^{N} s_i \\
s.t \quad &\sup_{(x,y) \in \Xi^2} && \ell(\langle w, x\rangle, y) - \lambda\|x - \hat{x}_i\| - \frac{\kappa\lambda}{2}|y - \hat{y}_i| \leq s_i, && i \in [N] \\
& && \lambda \geq 0
\end{aligned}$$

Since we are dealing with a classification problem we have labels $y \in \{+1, -1\}$. We consider the two scenarios possible $y = \hat{y}_i$ and $y = -\hat{y}_i$, also $\ell(\langle w, x\rangle, y) = L(y\langle w, x\rangle)$ is used. Hence we can write the new constraints as

$$\begin{aligned}
&= && \inf_{\lambda, s_i} \lambda\epsilon + \frac{1}{N}\sum_{i=1}^{N} s_i \\
&s.t \\
& && \sup_{x \in X} L(\hat{y}_i \langle w, x\rangle) - \lambda\|x - \hat{x}_i\| \leq s_i, && i \in [N] \\
& && \sup_{x \in X} L(-\hat{y}_i \langle w, x\rangle) - \lambda\|x - \hat{x}_i\| - \kappa\lambda \leq s_i, && i \in [N] \\
& && \lambda \geq 0
\end{aligned}$$

Using the definition of the loss $L(z) = max_{j \in J}(a_j z + b_j)$,

$$
\begin{aligned}
= \quad & \inf_{\lambda, s_i} \lambda\epsilon + \frac{1}{N}\sum_{i=1}^{N} s_i \\
s.t \quad & \\
& \sup_{x \in X} \; a_j \hat{y}_i \langle w, x \rangle + b_j - \lambda \|x - \hat{x}_i\| \leq s_i, \qquad i \in [N], j \in [J] \\
& \sup_{x \in X} \; -a_j \hat{y}_i \langle w, x \rangle + b_j - \lambda \|x - \hat{x}_i\| - \kappa\lambda \leq s_i, i \in [N], j \in [J] \\
& \lambda \geq 0
\end{aligned}
$$

Using the relation of norm and its dual, we introduce p_{ij}^{+} and p_{ij}^{-}.

$$\lambda \|x - \hat{x}_i\| = \inf_{\|p_{ij}^{+}\|_* \leq \lambda} \left\langle p_{ij}^{+}, x - \hat{x}_i \right\rangle$$

Let us consider this substitution for the first constraint, it follows for the second constraint as well,

$$\sup_{x \in X} \; a_j \hat{y}_i \langle w, x \rangle + b_j - \inf_{\|p_{ij}^{+}\|_* \leq \lambda} \left\langle p_{ij}^{+}, x - \hat{x}_i \right\rangle \leq s_i$$

We can move the inf before the sup,

$$\inf_{\|p_{ij}^{+}\|_* \leq \lambda} \sup_{x \in X} \; a_j \hat{y}_i \langle w, x \rangle + b_j - \left\langle p_{ij}^{+}, x - \hat{x}_i \right\rangle \leq s_i$$

Rearranging the terms,

$$\inf_{\|p_{ij}^{+}\| \leq \lambda} \sup_{x \in X} \; \left\langle a_j \hat{y}_i w - p_{ij}^{+}, x \right\rangle + b_j + \left\langle p_{ij}^{+}, \hat{x}_i \right\rangle \leq s_i$$

$$\inf_{\|p_{ij}^{+}\|_* \leq \lambda} \sup_{x \in \mathbb{R}^d} \; \left\langle a_j \hat{y}_i w - p_{ij}^{+}, x \right\rangle - \delta_X(x) + b_j + \left\langle p_{ij}^{+}, \hat{x}_i \right\rangle \leq s_i$$

where, $\delta_X(x)$ is the indicator function for the set X defined by,

$$\delta_X(x) = \begin{cases} 0 & x \in X \\ \infty & x \notin X \end{cases}$$

We introduce the support function on the set X given by,

$$S_X(t) = \inf_{x \in \mathbb{R}^d} \langle t, x \rangle - \delta_X(x)$$

The constraint then becomes,

$$\inf_{\|p_{ij}^+\|_* \leq \lambda} \quad S_x(a_j \hat{y}_i w + p_{ij}^+) + b_j + \left\langle p_{ij}^+, \hat{x}_i \right\rangle \leq s_i, \qquad i \in [N], j \in [J]$$

The second constraint also follows,

$$\inf_{\|p_{ij}^+\|_* \leq \lambda} \quad S_x(-a_j \hat{y}_i w - p_{ij}^-) + b_j + \left\langle p_{ij}^-, \hat{x}_i \right\rangle - \kappa\lambda \leq s_i, \qquad i \in [N], j \in [J]$$

Now, we consider the inf_w constraint in the original Expectation formulation and add the decision variable w to the inf formulation we have obtained. The constraints itself have a inf construct. We can argue that if $\exists p_{ij}^+$ such that $\|p_{ij}^+\|_* \leq \lambda$ and the constraint is satisfied, then it is easy to see that the constraint above will hold for the inf case. Hence, we can remove the inf and consider p_{ij}^+ as a decision variable of the outer, main optimization problem and add another constraint for the dual-norm of the variable. The same follows for p_{ij}^-.
Final formulation is,

$$\begin{aligned}
&\inf_{w,\lambda,s_i,p_{ij}^+,p_{ij}^-} \sup_{Q \in B_\epsilon(\hat{P_n})} \quad \mathbb{E}^Q[\ell(\langle w, x \rangle, y)] \\
&= \quad \inf_{w,\lambda,s_i,p_{ij}^+,p_{ij}^-} \lambda\epsilon + \frac{1}{N}\sum_{i=1}^{N} s_i \\
&s.t \\
&\quad S_x(a_j \hat{y}_i w - p_{ij}^+) + b_j + \left\langle p_{ij}^+, \hat{x}_i \right\rangle \leq s_i, \qquad i \in [N], j \in [J] \\
&\quad S_x(-a_j \hat{y}_i w - p_{ij}^-) + b_j + \left\langle p_{ij}^-, \hat{x}_i \right\rangle - \kappa\lambda \leq s_i, i \in [N], j \in [J] \\
&\quad \|p_{ij}^+\|_* \leq \lambda, \|p_{ij}^-\|_* \leq \lambda \qquad i \in [N], j \in [J] \\
&\quad s_i \geq 0, \qquad i \in [N]
\end{aligned}$$

Hence proved. □

2.3 Application in Machine Learning: Support Vector Machine

In this section, we further obtain formulations for Support Vector Machines considering the hinge loss. We then implement the Linear Program in Gurobi and perform an analysis of the results obtained on different datasets and for different values of the Wasserstein radius, kappa and the dual norm.

2.3.1 Formulation

The Support Vector Machine Problem formulation uses the hinge loss function. The loss function is piece-wise linear, hence we can obtain an LP formulation for the same. The resulting formulation is :

$$\inf_{w,\lambda,s_i,p_i^+,p_i^-} \quad \lambda\epsilon + \frac{1}{N}\sum_{i=1}^{N} s_i$$

$$s.t$$

$$\begin{aligned}
& 1 - \hat{y}_i \langle w, \hat{x}_i \rangle + \left\langle p_i^+, d - C\hat{x}_i \right\rangle \leq s_i, && i \in [N] \qquad (2.2)\\
& 1 + \hat{y}_i \langle w, \hat{x}_i \rangle + \left\langle p_i^+, d - C\hat{x}_i \right\rangle - \kappa\lambda \leq s_i, && i \in [N]\\
& \|C^T p_i^+ + \hat{y}_i w\|_* \leq \lambda, \|C^T p_i^- - \hat{y}_i w\|_* \leq \lambda && i \in [N]\\
& s_i \geq 0, p_i^+, p_i^- \in \mathscr{C} && i \in [N]
\end{aligned}$$

where C,d are obtained from conic representation of the data. The loss function is the Hinge Loss function $L(z)$:

$$L(z) = \begin{cases} 1 - z, & \text{if } z \leq 1 \\ 0, & \text{if } z > 1 \end{cases} \qquad (2.3)$$

The formulation is obtained from the formulation 2.1 by making use of the Hinge Loss. We provide a brief proof below.

Proof :

For SVM setting we consider the input space X admits a connive representation,

$$X = \{x \in \mathbb{R}^n \mid Cx \leq_{\mathscr{C}} d\}$$

where C is some matrix, d a vector and proper convex cone $\mathscr{C}$ of appropriate dimensions. We also assume that X admits a slater point $x_s \in \mathbb{R}^n$ with $Cx_s <_{\mathscr{C}} d$. Using the conic duality the support function X can be expressed as,

$$\begin{aligned} S_X(z) &= \sup_x \{\langle z, x\rangle : Cx \leq_{\mathscr{C}} d\} \\ &= \inf_{q \in \mathscr{C}^*} \{\langle q, d\rangle : C^T q = z\} \end{aligned}$$

where $\mathscr{C}^*$ is the dual cone. Strong duality (thus the last equality) holds because X admits the a slater point.
Consider the hinge loss and coefficients a_j and b_j for the two pieces,

$$L(z) = \begin{cases} 1 - z, & \text{if } z \leq 1 \quad \Rightarrow a_j = -1, b_j = 1, (j = 1) \\ 0, & \text{if } z > 1 \quad \Rightarrow a_j = 0, b_j = 0, (j = 0) \end{cases}$$

Now using the constraints of 2.1 and substituting the values of a_j and b_j, For j=0 the constraints do not add any information. The constraints are applicable for j=1 only, so we drop the notation of j.

$$S_x(-\hat{y}_i w - p_i^+) + 1 + \left\langle p_i^+, \hat{x}_i \right\rangle \leq s_i$$

$$S_x(\hat{y}_i w - p_i^-) + 1 + \left\langle p_i^-, \hat{x}_i \right\rangle - \kappa\lambda \leq s_i$$

We consider only the first constraint here and provide simplification, the same follows for the second constraint. We make use of the support function and conic duality relation from earlier,

$$S_x(-\hat{y}_i w - p_i^+) = \inf_{q_i^+ \in \mathscr{C}^*} \{\left\langle q_i^+, d\right\rangle : C^T q_i^+ = -\hat{y}_i w - p_i^+\}$$

Hence the constraint becomes,

$$\inf_{q_i^+ \in \mathscr{C}^*} \left\langle q_i^+, d\right\rangle + 1 + \left\langle p_i^+, \hat{x}_i\right\rangle$$

where, $C^T q_i^+ = -\hat{y}_i w - p_i^+ \Rightarrow p_i^+ = C^T q_i^+ - \hat{y}_i w$

$$\inf_{q_i^+ \in \mathscr{C}^*} \left\langle q_i^+, d\right\rangle + 1 + \left\langle C^T q_i^+ - \hat{y}_i w, \hat{x}_i\right\rangle$$

If the above inequality holds for some q_i^+ then it must hold for the inf. Therefore,

we add the variable q_i^+ to the decision variables of the outer, main formulation and add the constraints accordingly. By rearranging we have,

$$1 - \hat{y}_i \langle w, \hat{x}_i \rangle + \left\langle p_i^+, d - C\hat{x}_i \right\rangle \leq s_i$$

We use the same steps for the other constraint and add additional constraints for p_i^+ to include it as a decision variable. We substitute $p_i^+ = C^T q_i^+ - \hat{y}_i w$ as well. Hence the final formulation, after relabelling q_i^+ as p_i^+, is as follows,

$$\begin{aligned}
&\inf_{w,\lambda,s_i,p_i^+,p_i^-} \quad \lambda\epsilon + \frac{1}{N}\sum_{i=1}^{N} s_i \\
s.t& \\
&1 - \hat{y}_i \langle w, \hat{x}_i \rangle + \left\langle p_i^+, d - C\hat{x}_i \right\rangle \leq s_i, && i \in [N] \qquad (2.4)\\
&1 + \hat{y}_i \langle w, \hat{x}_i \rangle + \left\langle p_i^+, d - C\hat{x}_i \right\rangle - \kappa\lambda \leq s_i, && i \in [N] \\
&\|C^T p_i^+ + \hat{y}_i w\|_* \leq \lambda, \|C^T p_i^- - \hat{y}_i w\|_* \leq \lambda && i \in [N] \\
&s_i \geq 0, p_i^+, p_i^- \in \mathscr{C}^* && i \in [N]
\end{aligned}$$

Hence proved. □

2.3.2 Implementation

The formulation from the above section with slight modification is implemented in Gurobi. The final formulations used for implementation are included in **Appendix-A**. We perform analysis, for different values of Wasserstein radius (ϵ), dual norm and cost of switching a label (κ). We implement the formulation using the IRIS dataset and the MNIST digit dataset. The problem formulation is a 2 class classification problem where the data is linearly separable. We perform a 5 Fold Cross-Validation with different parameters and obtain the parameter values which perform well. These parameters are then tested on unseen data and the metrics are reported. Since this is a classification problem, we use the AUC-ROC score to decide on the best performance.

IRIS Dataset

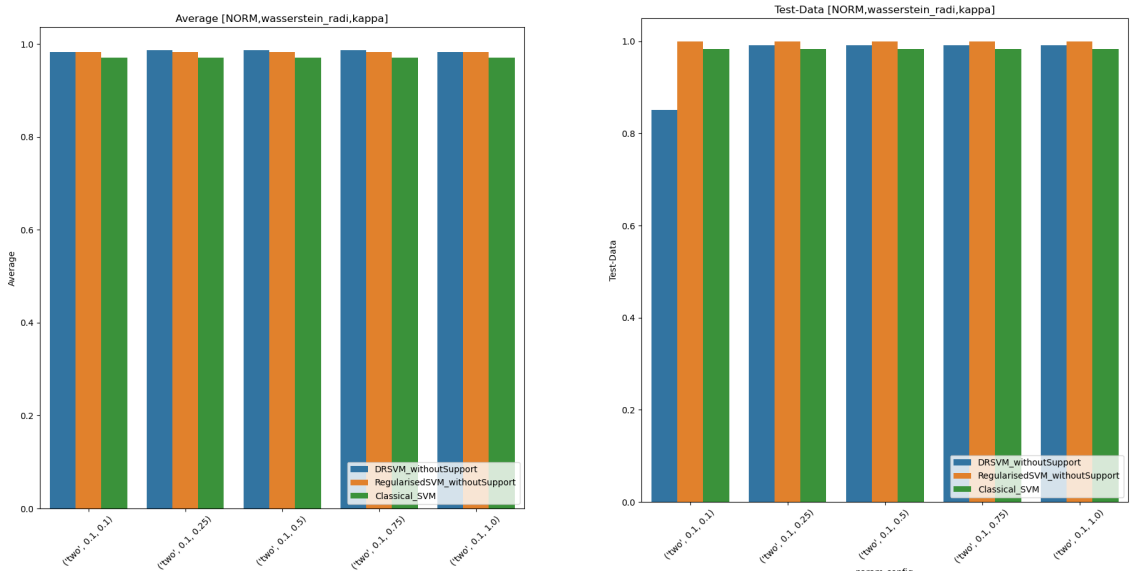
The iris dataset used consists of a total of 100 data points.
Features : Sepal Width, Sepal Length
Training Data : 67 Samples = 54 (Train) + 13 (Test) for 5CV Split
Unseen Test Data : 33 Samples
Labels : {+1,-1}

The optimization parameter gives comparable results with the builtin implementations for the following setting :
Wasserstein Radius = [,0.1]
Dual NORM = two norm
Kappa = [0.1,0.25,0.5,0.75]
[We test it for more values of Wasserstein radius but obtain comparable results for this value]

Results

	Average	Split_1	Split_2	Split_3	Split_4	Split_5	Test-Data	wasserstein_radii	NORM
DRSVM_withoutSupport	0.9833	1	1.0000	1	0.9167	1	1	0.1	two
RegularisedSVM_withoutSupport	0.9833	1	1.0000	1	0.9167	1	1	0.1	two
Classical_SVM	0.9708	1	0.9375	1	0.9167	1	1	0.1	two
DRSVM_withoutSupport	0.9875	1	0.9375	1	1.0000	1	1	0.1	two
RegularisedSVM_withoutSupport	0.9833	1	1.0000	1	0.9167	1	1	0.1	two
Classical_SVM	0.9708	1	0.9375	1	0.9167	1	1	0.1	two
DRSVM_withoutSupport	0.9875	1	0.9375	1	1.0000	1	1	0.1	two
RegularisedSVM_withoutSupport	0.9833	1	1.0000	1	0.9167	1	1	0.1	two
Classical_SVM	0.9708	1	0.9375	1	0.9167	1	1	0.1	two

The performance on the test data is perfect because the dataset is easily separable and the Distributionally Robust setting is able to learn the separating hyperplane easily while giving a tight upper bound for the expected loss.



(a) Average (b) Test

Figure 2.1: **(a)** The figure on the left indicates the AUC-ROC score averaged across all the 5 cross-validation splits.**(b)** The figure on the right shows the performance on the unseen test data. The y-axis for both the graphs is the AUC-ROC score, while the x-axis is the configuration i.e. the wasserstein_radii and kappa values indicated as a tuple (NORM, Wasserstein radii, kappa).

Visualisation

The plots (taken on a random sample of the data) provide a visualisation of how the separating hyperplane is affected for different values of kappa. Kappa is the cost of switching a label, hence as we increase the kappa value it tries to learn a hyperplane that makes less errors on the label while still trying to better the worst case.

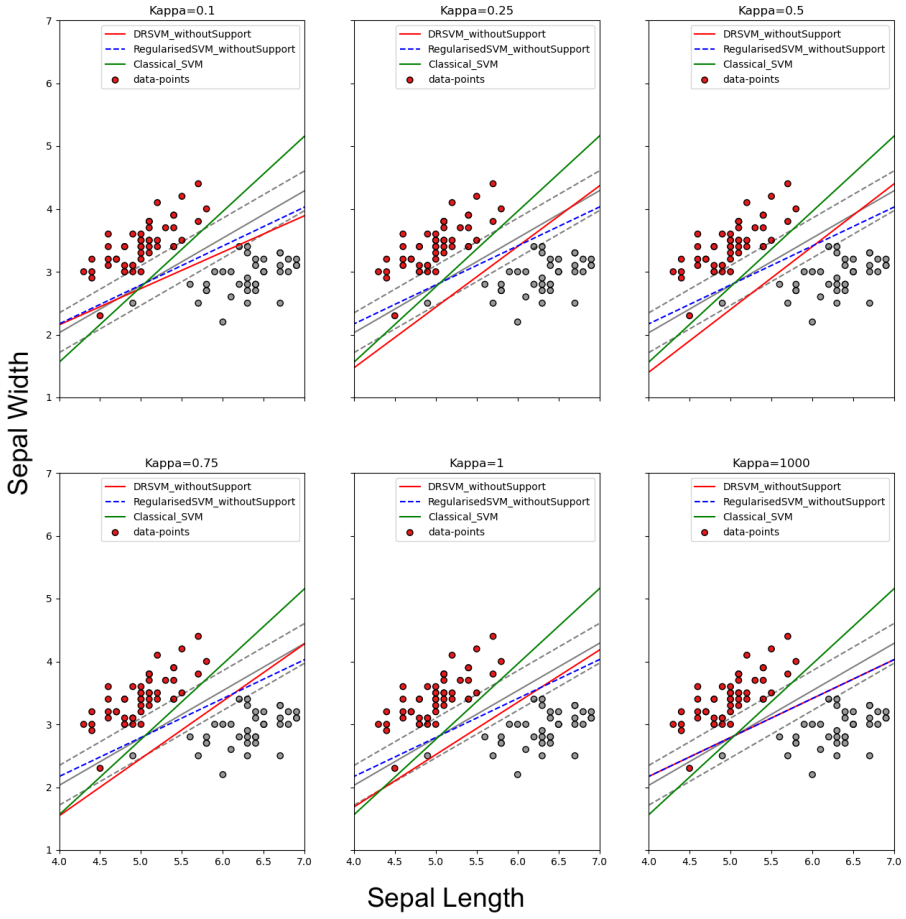


Figure 2.2: The figure shows the plot of the separating hyperplane for the different algorithms. The data points along with the labels are plotted.

Observations

The RegularisedSVM_withoutSupport is dependent only on the Wasserstein radius while classical SVM does not depend on any of the parameters, but both the decision boundaries are plotted for observing purposes.

- When the kappa value is 0.1 it switches the labels of close to three points but as the kappa value increases it makes lesser label switches.
- When the kappa value is large the RegularisedSVM_ withoutSupport and DRSVM_withoutSupport coincide. This matches theoretical understanding because as the kappa approaches infinity the DRSVM constraints reduce to the RegularisedSVM.

MNIST dataset

The MNIST dataset used consists of a total of 241 data points.

Features : 64 pixels (for 8x8 black and white image)

Training Data : 241 Samples = 198 (Train) + 43 (Test) for 5CV Split.

Unseen Test Data : 80 Samples.

Labels : 1 vs 7 (two digits that can be confused).

The optimization parameter gives comparable results with the builtin implementations for the following setting :

Wasserstein Rad = [0.1]

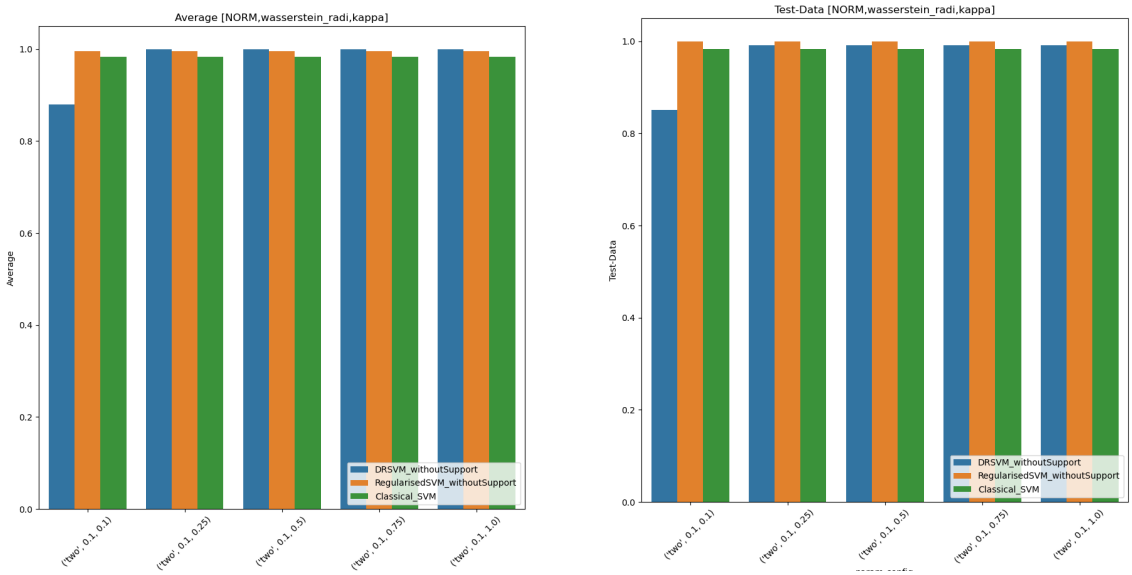
Dual NORM = two norm

Kappa = [0.1,0.25,0.5,0.75]]

[We test it for more values of wasserstein radius but obtain comparable results for this value]

Results

	Average	Split_1	Split_2	Split_3	Split_4	Split_5	Test-Data	wasserstein_radii	NORM
DRSVM_withoutSupport	0.8793	0.8758	0.9375	0.7708	0.8750	0.9375	0.8508	0.1000	two
RegularisedSVM_withoutSupport	0.9958	1.0000	1.0000	1.0000	1.0000	0.9792	1.0000	0.1000	two
Classical_SVM	0.9833	0.9792	0.9583	1.0000	1.0000	0.9792	0.9833	0.1000	two
DRSVM_withoutSupport	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9915	0.1000	two
RegularisedSVM_withoutSupport	0.9958	1.0000	1.0000	1.0000	1.0000	0.9792	1.0000	0.1000	two
Classical_SVM	0.9833	0.9792	0.9583	1.0000	1.0000	0.9792	0.9833	0.1000	two
DRSVM_withoutSupport	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9915	0.1000	two
RegularisedSVM_withoutSupport	0.9958	1.0000	1.0000	1.0000	1.0000	0.9792	1.0000	0.1000	two
Classical_SVM	0.9833	0.9792	0.9583	1.0000	1.0000	0.9792	0.9833	0.1000	two
DRSVM_withoutSupport	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9915	0.1000	two
RegularisedSVM_withoutSupport	0.9958	1.0000	1.0000	1.0000	1.0000	0.9792	1.0000	0.1000	two
Classical_SVM	0.9833	0.9792	0.9583	1.0000	1.0000	0.9792	0.9833	0.1000	two
DRSVM_withoutSupport	0.7098	0.6117	0.8542	0.8542	0.6250	0.6042	0.5242	0.2500	two
RegularisedSVM_withoutSupport	0.9958	1.0000	1.0000	1.0000	1.0000	0.9792	1.0000	0.2500	two
Classical_SVM	0.9833	0.9792	0.9583	1.0000	1.0000	0.9792	0.9833	0.2500	two
DRSVM_withoutSupport	0.9583	1.0000	0.8958	0.9583	0.9792	0.9583	0.9746	0.2500	two
RegularisedSVM_withoutSupport	0.9958	1.0000	1.0000	1.0000	1.0000	0.9792	1.0000	0.2500	two
Classical_SVM	0.9833	0.9792	0.9583	1.0000	1.0000	0.9792	0.9833	0.2500	two
DRSVM_withoutSupport	0.9917	1.0000	1.0000	0.9792	1.0000	0.9792	0.9915	0.2500	two
RegularisedSVM_withoutSupport	0.9958	1.0000	1.0000	1.0000	1.0000	0.9792	1.0000	0.2500	two
Classical_SVM	0.9833	0.9792	0.9583	1.0000	1.0000	0.9792	0.9833	0.2500	two
DRSVM_withoutSupport	0.9958	1.0000	1.0000	0.9792	1.0000	1.0000	0.9915	0.2500	two
RegularisedSVM_withoutSupport	0.9958	1.0000	1.0000	1.0000	1.0000	0.9792	1.0000	0.2500	two
Classical_SVM	0.9833	0.9792	0.9583	1.0000	1.0000	0.9792	0.9833	0.2500	two
DRSVM_withoutSupport	0.9958	1.0000	1.0000	0.9792	1.0000	1.0000	0.9915	0.2500	two
RegularisedSVM_withoutSupport	0.9958	1.0000	1.0000	1.0000	1.0000	0.9792	1.0000	0.2500	two
Classical_SVM	0.9833	0.9792	0.9583	1.0000	1.0000	0.9792	0.9833	0.2500	two



(a) Average (b) Test

Figure 2.3: **(a)** The figure on the left indicates the AUC-ROC score averaged across all the 5 cross-validation splits.**(b)** The figure on the right shows the performance on the unseen test data. The y-axis for both the graphs is the AUC-ROC score, while the x-axis is the configuration i.e. the wasserstein_radii and kappa values indicated as a tuple (NORM, Wasserstein radii, kappa).

Observations

The results obtained showed the best performance when the Wasserstein radius is set to 0.1 and the 2-Norm is used along with kappa values of 0.25,0.5 and 1. The results obtained are better than the classical SVM during 5CV and on the unseen test data. As the kappa value increased from 0.1 to 0.25 the results, on the unseen test data, become comparable to RegularisedSVM_withoutSupport.

2.4 Application in Machine Learning: Logistic Regression

In this section, we further obtain formulations for Logistic Regression considering the log loss. We then implement the Linear Programme in Gurobi and perform an analysis of the results obtained on different datasets and for different values of the Wasserstein radius, kappa and the dual norm.

2.4.1 Formulation

The Logistic Regression Problem formulation uses the log loss function. The resulting formulation is :

$$\begin{aligned} \min_{w,\lambda,s_i} \quad & \lambda\epsilon + \frac{1}{N}\sum_{i=1}^{N} s_i \\ \text{s.t.} \quad & log(1+exp(-\hat{y_i}\langle w,\hat{x_i}\rangle)) \leq s_i, i \in [N], \\ & log(1+exp(\hat{y_i}\langle w,\hat{x_i}\rangle)) - \kappa\lambda \leq s_i, i \in [N], \\ & \|w\|_* \leq \lambda\,, i \in [N] \end{aligned} \tag{2.5}$$

where the loss function is the Log Loss function $L(z)$:

$$L(z) = log(1+e^{-z}) \tag{2.6}$$

Proof :

The log loss function is convex and has lipschitz modulus 1.

If $\mathbb{X}=\mathbb{R}^n$ and L is lipschitz continous then the tractable formulation derived in Section 2.2 is equivalent to

$$\begin{aligned} \min_{w,\lambda,s_i} \quad & \lambda\epsilon + \frac{1}{N}\sum_{i=1}^{N} s_i \\ \text{s.t.} \quad & L(\hat{-y_i}\langle w,\hat{x_i}\rangle) \leq s_i, i \in [N], \\ & L(\hat{y_i}\langle w,\hat{x_i}\rangle) - \kappa\lambda \leq s_i, i \in [N], \\ & lip(L)\|w\|_* \leq \lambda\,, i \in [N] \end{aligned} \tag{2.7}$$

Making suitable substitution for the log loss function gives us the formulation for logistic regression.

2.4.2 Implementation

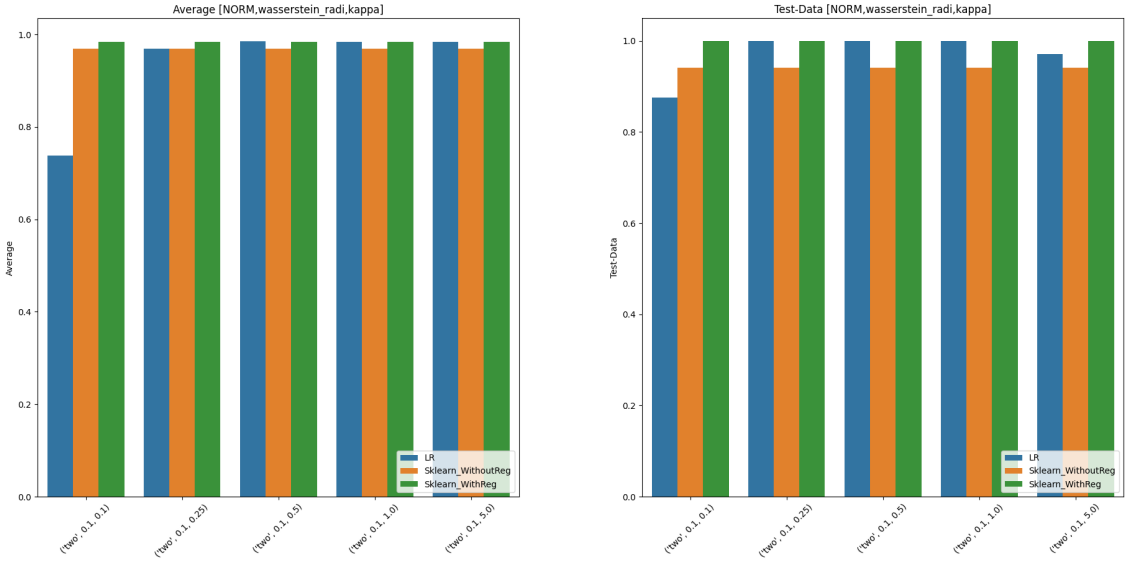
The formulation from the above section is implemented. 5 Fold cross validation is performed here as well to find the optimal parameter values on MNIST and Iris dataset. The parameters have been tested as well and AUC-ROC scores were used again to evaluate the performance.

IRIS Dataset

Dataset details are same as given in the SVM implementation. The optimization parameter gives comparable results with the builtin implementations for the following setting :
Wasserstein Rad = [,0.1]
Dual NORM = two norm
Kappa = [0.1,0.25,0.5,1,5]
[We test it for more values of wasserstein radius but obtain comparable results for this value]

Results

	Average	Split_1	Split_2	Split_3	Split_4	Split_5	Test-Data	wasserstein_radii	NORM
LR	0.7381	0.9286	0.4286	0.8571	0.7738	0.7024	0.8750	0.1	two
Sklearn_WithoutReg	0.9690	1.0000	0.9286	0.9167	1.0000	1.0000	0.9412	0.1	two
Sklearn_WithReg	0.9833	1.0000	1.0000	0.9167	1.0000	1.0000	1.0000	0.1	two
LR	0.9690	1.0000	0.9286	0.9167	1.0000	1.0000	1.0000	0.1	two
Sklearn_WithoutReg	0.9690	1.0000	0.9286	0.9167	1.0000	1.0000	0.9412	0.1	two
Sklearn_WithReg	0.9833	1.0000	1.0000	0.9167	1.0000	1.0000	1.0000	0.1	two
LR	0.9857	1.0000	0.9286	1.0000	1.0000	1.0000	1.0000	0.1	two
Sklearn_WithoutReg	0.9690	1.0000	0.9286	0.9167	1.0000	1.0000	0.9412	0.1	two
Sklearn_WithReg	0.9833	1.0000	1.0000	0.9167	1.0000	1.0000	1.0000	0.1	two



(a) Average (b) Test

Figure 2.4: **(a)** The figure on the left indicates the AUC-ROC score averaged across all the 5 cross-validation splits.**(b)** The figure on the right shows the performance on the unseen test data. The y-axis for both the graphs is the AUC-ROC score, while the x-axis is the configuration i.e. the wasserstein_radii and kappa values indicated as a tuple (NORM, Wasserstein radii, kappa).

Visualisation

The plots (taken on a random sample of the data) provide a visualisation of how the separating hyperplane is affected for different values of kappa. Kappa is the cost of switching a label, hence as we increase the kappa value it tries to learn a hyperplane that makes less errors on the label while still trying to better the worst case.

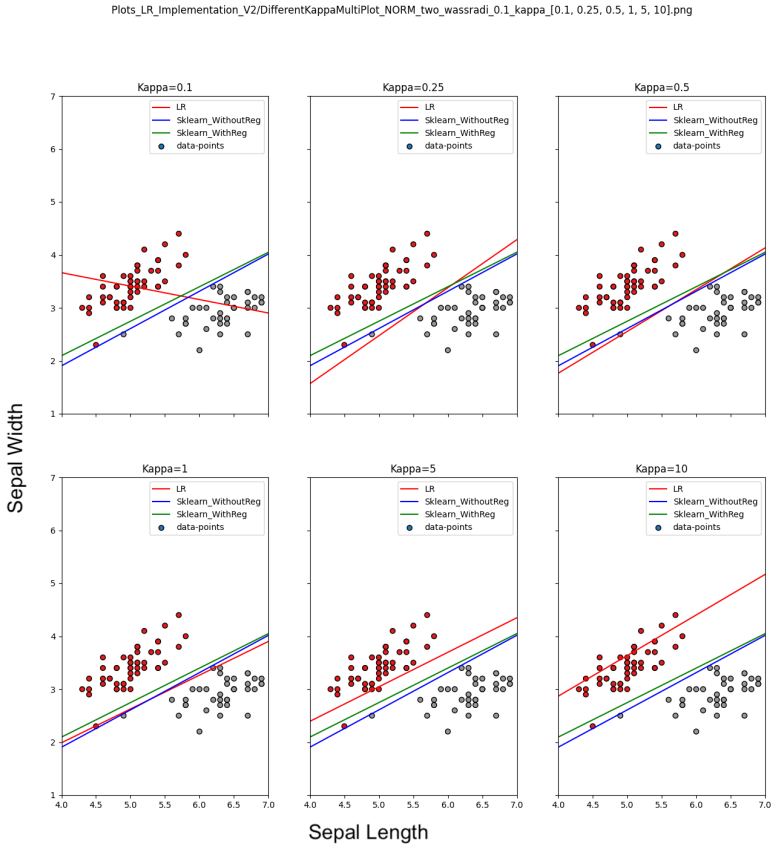


Figure 2.5: The figure shows the plot of the separating hyperplane for the different algorithms. The data points along with the labels are plotted.

Observations

The results obtained showed the best performance when the Wasserstein radius is set to 0.1 and the 2-Norm is used along with kappa values of 0.25,0.5 and 1. Setting the radius to 0.5 and keeping kappa as 5 or 10 also gives promising results which have not been included here. The performance deteriorates quickly if we increase kappa beyond 5 keeping epsilon fixed at 0.1, which is erratic and was not expected. Another specific observation is that the robust implementation and the inbuilt regularised one don't underperform on the test results generally in all the cases.

MNIST dataset

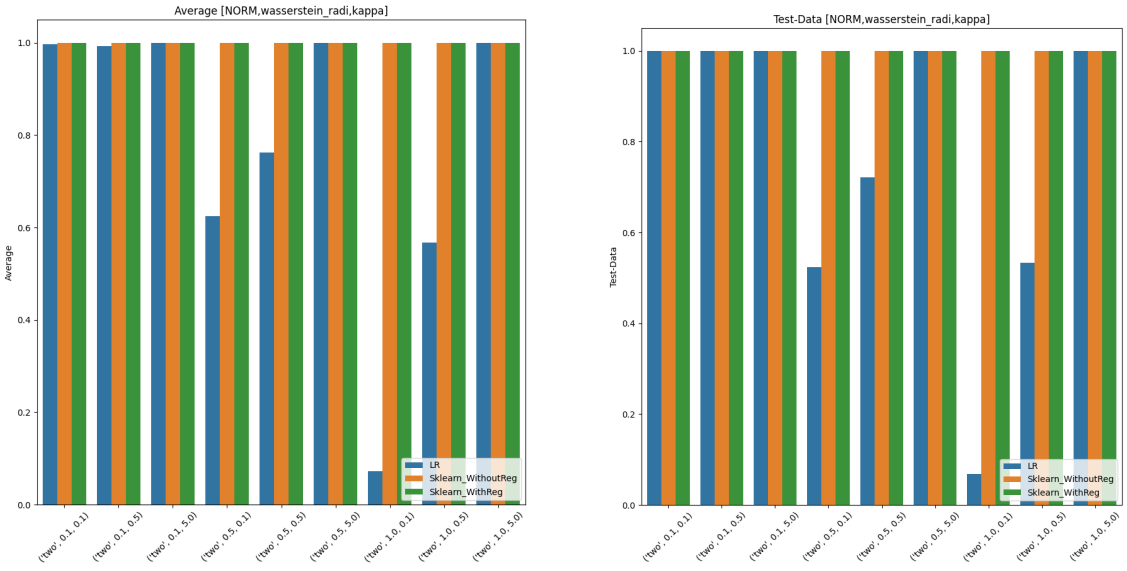
Dataset details are same as given in the SVM implementation. The optimization parameter gives comparable results with the builtin implementations for the following settings :

Wasserstein Radius	Kappa	Norm
0.1	>0.1	two
0.5	>1	two
1	>5	two

[We test it for more values of Wasserstein radius and also obtain comparable results for those values but only some are presented here.]

Results

	Average	Split_1	Split_2	Split_3	Split_4	Split_5	Test-Data	wasserstein_radii	NORM	kappa
LR	0.9962	1.0000	1.0000	1.0000	0.9808	1.0000	1.0000	0.1	two	0.10
Sklearn_WithoutReg	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.1	two	0.10
Sklearn_WithReg	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.1	two	0.10
LR	0.9922	1.0000	0.9800	1.0000	0.9808	1.0000	1.0000	0.1	two	0.50
Sklearn_WithoutReg	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.1	two	0.50
Sklearn_WithReg	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.1	two	0.50
LR	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.1	two	5.00
Sklearn_WithoutReg	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.1	two	5.00
Sklearn_WithReg	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.1	two	5.00
LR	0.6242	0.6722	0.4574	0.6556	0.7517	0.5839	0.5243	0.5	two	0.10
Sklearn_WithoutReg	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.5	two	0.10
Sklearn_WithReg	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.5	two	0.10
LR	0.7626	1.0000	0.6661	0.5087	0.9196	0.7185	0.7214	0.5	two	0.50
Sklearn_WithoutReg	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.5	two	0.50
Sklearn_WithReg	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.5	two	0.50
LR	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.5	two	5.00
Sklearn_WithoutReg	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.5	two	5.00
Sklearn_WithReg	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.5	two	5.00
LR	0.0719	0.0217	0.1087	0.1486	0.0804	0.0000	0.0686	1.0	two	0.10
Sklearn_WithoutReg	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0	two	0.10
Sklearn_WithReg	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0	two	0.10
LR	0.5676	0.6095	0.4191	0.7290	0.6451	0.4353	0.5329	1.0	two	0.50
Sklearn_WithoutReg	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0	two	0.50
Sklearn_WithReg	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0	two	0.50
LR	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0	two	5.00
Sklearn_WithoutReg	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0	two	5.00
Sklearn_WithReg	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0	two	5.00



(a) Average (b) Test

Figure 2.6: **(a)** The figure on the left indicates the AUC-ROC score averaged across all the 5 cross-validation splits.**(b)** The figure on the right shows the performance on the unseen test data. The y-axis for both the graphs is the AUC-ROC score, while the x-axis is the configuration i.e. the wasserstein_radii and kappa values indicated as a tuple (NORM, Wasserstein radii, kappa).

Observations

The results obtained showed best performance with multiple values of epsilon and kappa. A general trend of good performance on higher kappa values is observed when the epsilon is increased. The epsilon kappa correlation is interesting to observe and can be interpreted as follows - As wasserstein radius increases there are more chances of a probability distribution being there in the ball affects our decision badly but is not realisable in real life. Now when we increase kappa the performance gets better because it reinforces greater faith in the current data points and thus prevents the model from giving an output based on non-relevant probability distributions.

Conclusions

- Distributionally Robust Optimization provides a regularised approach in the optimisation problems discussed.
- We see that using kappa (our trust in the labels) and Wasserstein radius (search space for worst-case distribution) provides handles for training the algorithms such that they anticipate more scenarios from which the data can come and provide better out of sample performance by considering the worst case.
- Further using more domain knowledge, ambiguity sets can be specialised.
- The DRO approach provides tractable formulations for many use cases, which allows it to be useful for problems where independence/dependence assumptions cause tractability issues.

Extensions

During the course of the project we were able to think about the following extensions on which more work could be done.

- Extending the classification for Multiclass problems and obtaining formulations for the piecewise affine function.
- Application of DRO to other optimization problems such as Non-Negative Matrix Factorisation. The problem of NMF involves finding two matrices W and H such that X=WH and all the entries of W and H should be positive.

Appendix A

Implemented Formulations - SVM

For the purpose of implementation and use of the dataset we do not consider any conic representation for the data points. The change in the formulation is that the conic related terms do not appear.

1. Distributionally Robust SVM without Support (DRSVM_without Support)

$$\begin{aligned} \min_{w,\lambda,s_i,p_i^+,p_i^-} \quad & \lambda\epsilon + \frac{1}{N}\sum_{i=1}^{N} s_i \\ \text{s.t.} \quad & 1 - \hat{y}_i(\langle w, \hat{x}_i\rangle) \leq s_i, i \in [N], \\ & 1 + \hat{y}_i(\langle w, \hat{x}_i\rangle) - \kappa\lambda \leq s_i, i \in [N], \\ & s_i \geq 0, \\ & \|w\|_* \leq \lambda \end{aligned}$$

2. Regularised SVM without Support (Regularised SVM_withoutSupport)

$$\begin{aligned} \min_{w,\lambda} \quad & \lambda\epsilon + \frac{1}{N}\sum_{i=1}^{N} s_i \\ \text{s.t.} \quad & 1 - y_i w^T x_i \leq s_i, i \in [N], \\ & s_i \geq 0, \\ & \|w\|_* \leq \lambda \end{aligned}$$

NOTE:
When K tends to ∞ in DRSVM the $-\kappa\lambda \to -\infty \therefore (\to -\infty) \leq s_i$ constraint is trivially true and thus dropped, which results in the exact formulation of the Regularised SVM (non-DR).

Bibliography

[1] Daniel Kuhn. Lecture: Data-driven and distributionally robust optimization and applications on modern optimization in energy systems., 2018.

[2] P. Mohajerin Esfahani and D. Kuhn. Data-driven distributionally robust optimisation using the wasserstein metric: performance guarantees and tractable reformulations. *Mathematical Programming*, 171:115–166, 2018.

[3] Karthik Natarajan. *Marginals and Moments in Optimization under Uncertainty*. 2020.

[4] Soroosh Shafieezadeh Abadeh. *Wasserstein Distributionally Robust Learning*. PhD thesis, EPFL, 2020.