Paper Review Presentation on

Optimization Problems for Machine Learning: A survey

European Journal of Operational Research (2020)

Claudio Gambellaa,* , Bissan Ghaddar b , Joe Naoum-Sawaya b

By
Jir Et Katharpi
IIT Kharagpur

Under the Mentorship of
Professor Prasanna R
IIT Delhi

Contents

Introduction

Application of Machine Learning

Wildly explored in many domain

- Data Driven Decision Making
- Automation and Efficiency
- Personalization
- Predictive Analysis
- Anomaly Detection

Challenges

Ml faces various types of challenges

- Data Quality and Quantity
- Model Quality and Quantity
- Overfitting and Generalisation
- Bias and Fairness
- Scalability and Efficiency

Application of Optimisation

To solve complex design problems in order to improve cost, reliability, and performance in a wide range of applications

- Supply Chain Optimisation
- Transportation and Logistics Optimisation
- Inventory Management
- Production Planning and Scheduling
- Network Design and Telecommunication Optimisation

Challenges

To solve complex design problems in order to improve cost, reliability, and performance in a wide range of applications

- Complexity and Scalability
- Non Convexity
- Uncertainty and Variability
- Interdisciplinary Nature

Machine Learning

- It is about teaching computers to predict or find patterns.

- **There are two main types:**
  - Supervised Learning
  - Unsupervised Learning

Types of Machine Learning:

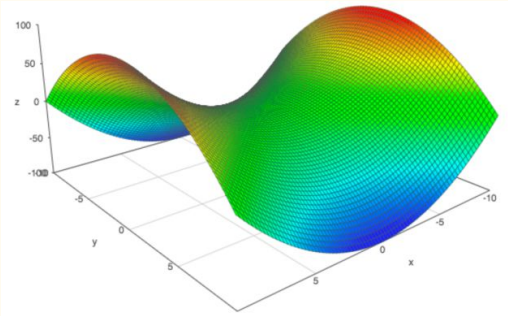
- **Supervised Learning**
  In supervised learning, we create models to predict outputs accurately. We use a "loss function" to measure accuracy and training data to build our model.

- **Unsupervised Learning**
- We don't have output data. It's more about exploring and finding patterns in the input data.

Introduction to Machine Learning for Optimisation

- Machine learning for Optimisation relies on optimization models for solving complex problems
- Optimization plays a crucial role in regression, classification, clustering, deep learning, and adversarial learning within machine learning.
- The advancement of numerical optimization techniques has greatly impacted machine learning algorithms.
- This paper surveys machine learning literature and offers an optimization framework for common machine learning approaches.

Optimisation Framework for Machine Learning



Utilizing mathematical optimization models in tasks like:

1. Regression,
2. Classification,
3. Clustering,
4. Deep Learning, and
5. Adversarial Learning.

It provides effective solutions for machine learning challenges such as:

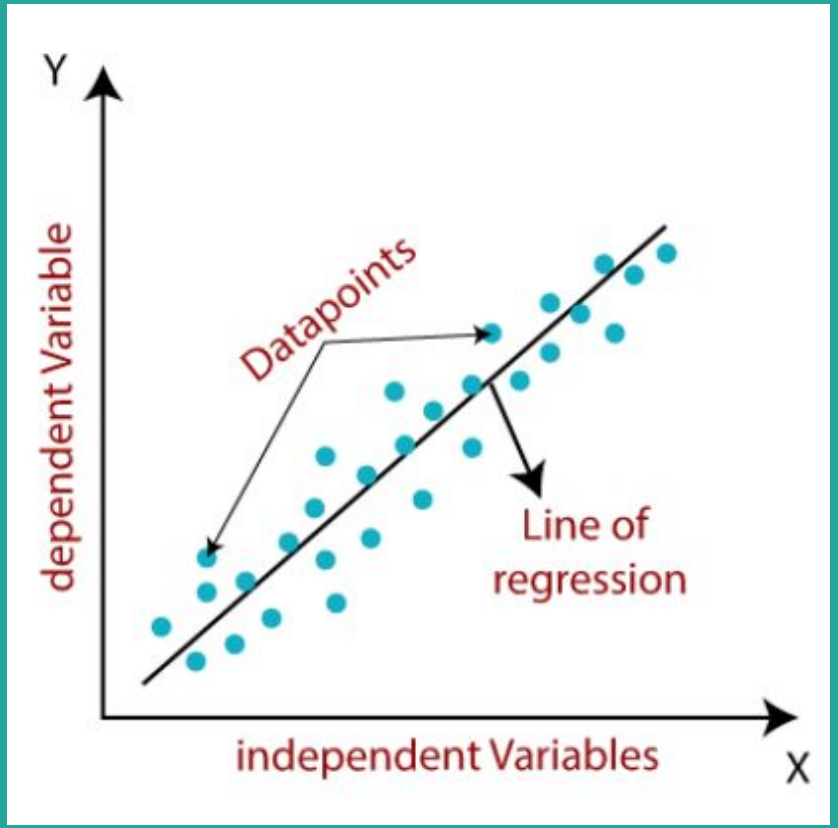
1. overfitting,
2. data uncertainty, and
3. data poisoning.

1. Regression Models

a. **Linear Regression**
b. Shrinkage Method
c. Regression models beyond linearity

$$RSS(\beta) = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 .$$

The most commonly used loss function for regression, *least Squared estimate*



linear regression is a statistical technique used to model and quantify the relationship between one or more independent variables and a dependent variable by fitting a linear equation to the data.

1. Regression Models

a. Linear Regression
b. Shrinkage Method
c. Regression models beyond linearity

The aim is to obtain a more interpretable model with less relevant features:

i) Ridge Regression

$$\mathcal{L}_{ridge}(\beta_0, \beta) = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2,$$

ii) Lasso Regression

$$\mathcal{L}_{lasso}(\beta_0, \beta) = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|.$$

Shrinkage Method

Sparse regression can be formulated as the best subset selection problem

$$\min \frac{1}{2}\|y - \beta_0 - X\beta\|_2^2$$
$$\text{s.t } \|\beta\|_0 \le k,$$
$$\beta_0 \in \mathbb{R}, \beta \in \mathbb{R}^p,$$

The sparse regression problem can be transformed into the MIQP formulation.
Specifically, by introducing the binary variables
s ∈ {0, 1}p,

$$\min \frac{1}{2}\|y - \beta_0 - X\beta\|_2^2$$
$$\text{s.t } -Ms_j \le \beta_j \le Ms_j \quad \forall j = 1, \ldots, p,$$
$$\sum_{j=1}^{p} s_j \le k,$$
$$\beta_0 \in \mathbb{R}, \beta \in \mathbb{R}^p,$$
$$s \in \{0, 1\}^p,$$

1. Regression Models

a. Linear Regression
b. Shrinkage Method
c. **Regression models beyond linearity**

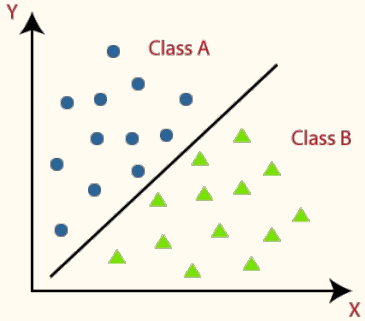
Consider the non-linear terms, that captures the complex relationships between predictors and regressors

$$y = \beta_0 + \sum_{j=1}^{p} f_j(X_j).$$

The relationship between each feature and the response y is expressed using nonlinear functions fj(Xj) such as

2. Classification

The process of teaching a computer to categorize data into predefined classes or groups based on patterns and features learned from training examples



a. Logistic regression
b. Linear discriminant analysis
c. Decision trees
d. Support vector machines
    i. Hard margin SVM
    ii. Soft margin SVM
    iii. Sparse SVM
    iv. The dual problem and kernel tricks
    v. Support vector regression
    vi. Support vector ordinal regression

Logistic Regression

Logistic regression calculates the class membership probability for one of the two categories in the dataset as above given

$$P(y = 1|x, \beta_0, \beta) = h(x, \beta_0, \beta) = \frac{1}{1 + e^{-(\beta_0 + \beta^\top x)}},$$
$$P(y = 0|x, \beta_0, \beta) = 1 - h(x, \beta_0, \beta).$$

The parameters β0 and β are usually obtained by maximum-likelihood estimation.
The problem above is convex and differentiable such as
i)**gradient descent** first order methods
ii)**Newton's method** second order methods
can be applied to find a global optimal solution.

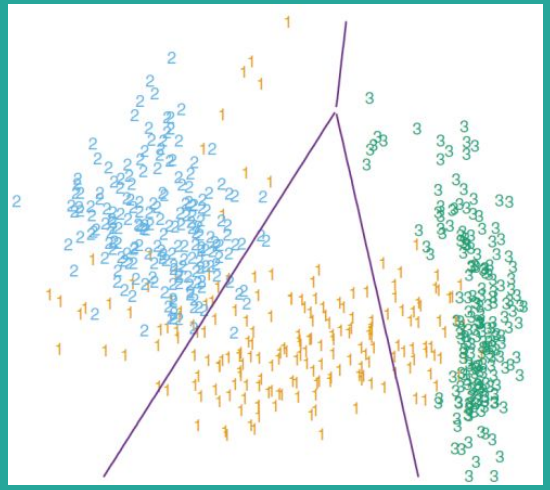
$$\min - \sum_{i=1}^{n} (y_i \log h(x_i, \beta_0, \beta) + (1 - y_i) \log(1 - h(x_i, \beta_0, \beta))).$$

2. Classification:

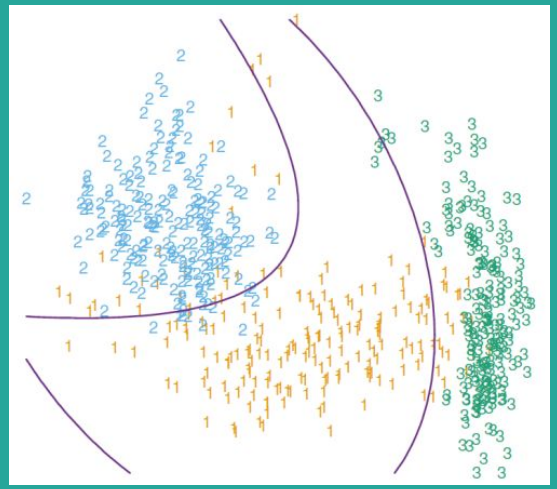
b. Linear discriminant analysis

i. Linear decision boundaries found by LDA
ii. Quadratic decision boundaries using LDA

i.

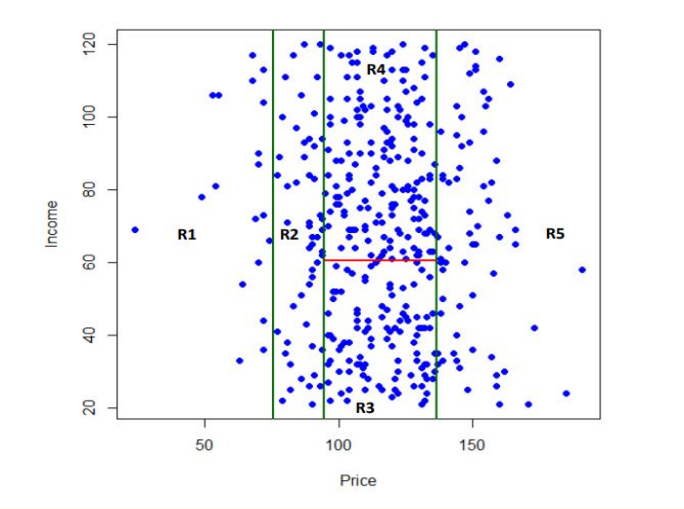


ii.



2. Classification:

Decision trees are classical models for making a decision or classification using splitting rules organized into a tree data structure.



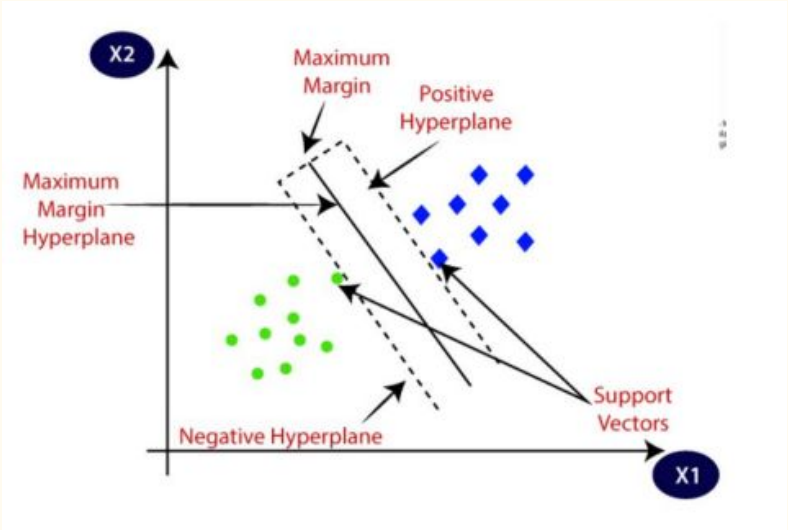
a. Logistic regression
b. Linear discriminant analysis
c. Decision trees
d. Support vector machines
    i. Hard margin SVM
    ii. Soft margin SVM
    iii. Sparse SVM
    iv. The dual problem and kernel tricks
    v. Support vector regression
    vi. Support vector ordinal regression

Decision Tree

The mixed integer programming formulation is

$$
\begin{aligned}
\min\ & \frac{1}{\hat{L}} \sum_{t \in T_L} L_t + \alpha \sum_{t \in T_B} d_t \\
\text{s.t.}\ & L_t \geq N_t - N_{kt} - n(1 - c_{kt}), \quad \forall k = 1, \ldots, K,\ t \in T_L, \\
& 0 \leq L_t \leq N_t - N_{kt} + n c_{kt} \quad \forall k = 1, \ldots, K,\ t \in T_L, \\
& N_{kt} = \frac{1}{2} \sum_{i=1}^{n} (1 + Y_{ik}) z_{it}, \quad \forall k = 1, \ldots, K,\ t \in T_L, \\
& N_t = \sum_{i=1}^{n} z_{it} \quad \forall t \in T_L, \\
& \sum_{k=1}^{K} c_{kt} = l_t \quad \forall t \in T_L, \\
& \sum_{t \in T_L} z_{it} = 1 \quad \forall i = 1, \ldots, n, \\
& z_{it} \leq l_t \quad \forall i = 1, \ldots, n,\ t \in T_L, \\
& \sum_{i=1}^{n} z_{it} \geq N_{\min} l_t \quad \forall t \in T_L, \\
& a_m^\top (x_i + \epsilon) \leq b_m + (1 + \epsilon_{\max})(1 - z_{it}) \\
& \qquad \forall i = 1, \ldots, n,\ t \in T_L,\ m \in A_L(t), \\
& a_m^\top x_i \geq b_m - (1 - z_{it}) \quad \forall i = 1, \ldots, n,\ t \in T_L,\ \forall m \in A_R(t), \\
& \sum_{j=1}^{p} a_{jt} = d_t \quad \forall t \in T_B, \\
& 0 \leq b_t \leq d_t \quad \forall t \in T_B, \\
& d_t \leq d_{p(t)} \quad \forall t \in T_B \setminus \{1\}, \\
& z_{it},\ l_t \in \{0, 1\} \quad \forall i = 1, \ldots, n,\ \forall t \in T_L, \\
& c_{kt} \in \{0, 1\} \quad \forall k = 1, \ldots, K,\ t \in T_L, \\
& a_{jt},\ d_t \in \{0, 1\} \quad \forall j = 1, \ldots, p,\ t \in T_B.
\end{aligned}
$$

2. Classification:

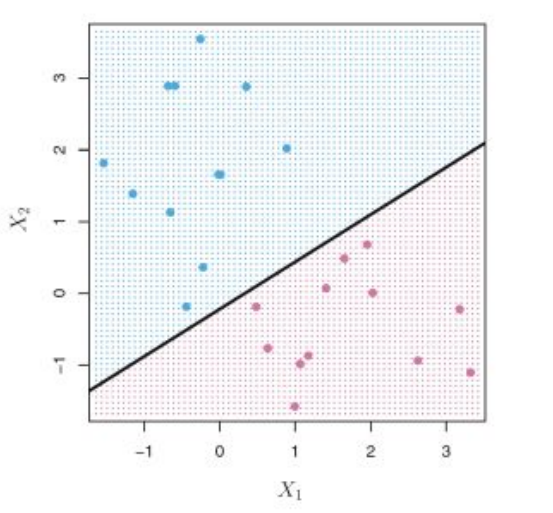


a. Logistic regression
b. Linear discriminant analysis
c. Decision trees
d. Support vector machines
    i. Hard margin SVM
    ii. Soft margin SVM
    iii. Sparse SVM
    iv. The dual problem and kernel tricks
    v. Support vector regression
    vi. Support vector ordinal regression

2. Classification:

d. Support Vector Machine

i. Hard Margin SVM



$$\min \|w\|_2^2$$

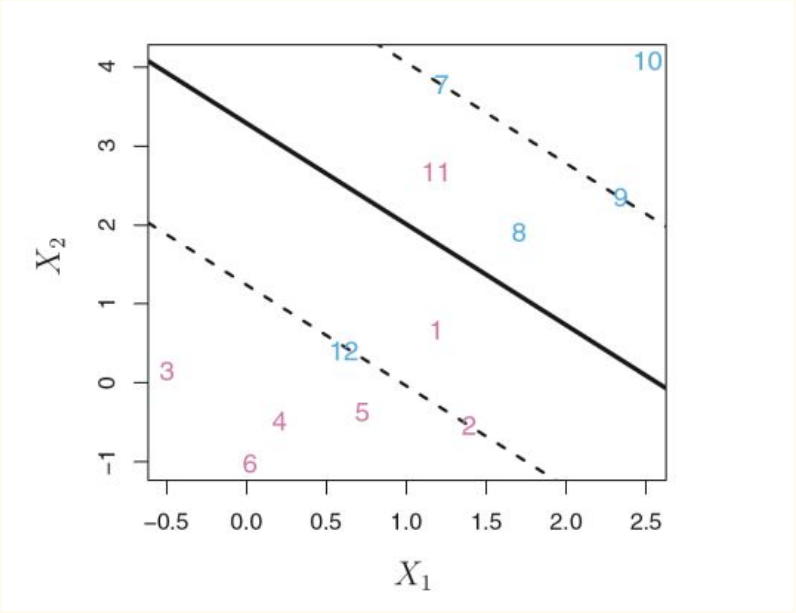
$$\text{s.t. } y_i(w^\top x_i + \gamma) \geq 1 \quad \forall i = 1, \ldots, n,$$

$$w \in \mathbb{R}^p, \gamma \in \mathbb{R},$$

The optimization problem for finding the separating hyperplane is then (Convex quadratic problem)

2. Classification:

d. Support Vector Machine
ii. Soft SVM



$$\min \|w\|_2^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{s.t. } y_i(w^\top x_i + \gamma) \geq 1 - \xi_i \quad \forall i = 1, \ldots, n,$$

$$w \in \mathbb{R}^p, \gamma \in \mathbb{R},$$

$$\xi_i \geq 0 \quad \forall i = 1, \ldots, n.$$

The soft-margin SVM optimization problem

2. Classification:

d. Support Vector Machine
iii. Sparse SVM

$$
\begin{aligned}
\min\ & \|w\|_2^2 + C\sum_{i=1}^{n}\xi_i \\
\text{s.t.}\ & y_i(w^\top x_i + \gamma) \geq 1 - \xi_i \quad \forall i = 1, \ldots, n, \\
& -Mz_j \leq w_j \leq Mz_j \quad \forall j = 1, \ldots, p, \\
& \sum_{j=1}^{p} z_j \leq r, \\
& w \in \mathbb{R}^p, \gamma \in \mathbb{R}, \\
& z_j \in \{0, 1\} \quad \forall j = 1, \ldots, p, \\
& \xi_i \geq 0 \quad \forall i = 1, \ldots, n.
\end{aligned}
$$

2. Classification:

d. Support Vector Machine

iv. The dual Problem and Kernel Tricks

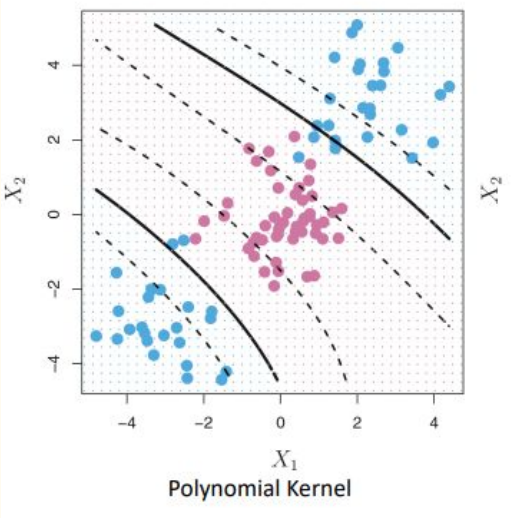
$$\min \ \|w\|_2^2 + C\sum_{i=1}^{n}\xi_i \tag{53}$$

$$\text{s.t.} \quad y_i(w^\top \phi(x_i) + \gamma) \geq 1 - \xi_i \quad \forall i = 1, \ldots, n, \tag{54}$$

$$w \in \mathbb{R}^p, \gamma \in \mathbb{R}, \tag{55}$$

$$\xi_i \geq 0 \quad \forall i = 1, \ldots, n. \tag{56}$$

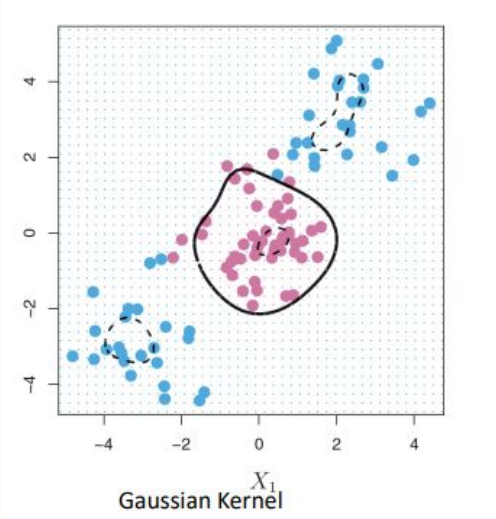
To solve problem (53)–(56), the following dual problem is first obtained



2. Classification:

d. Support Vector Machine

iv. The dual Problem and Kernel Tricks



Gaussian Kernel

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \phi(x_i)^\top \phi(x_j)$$

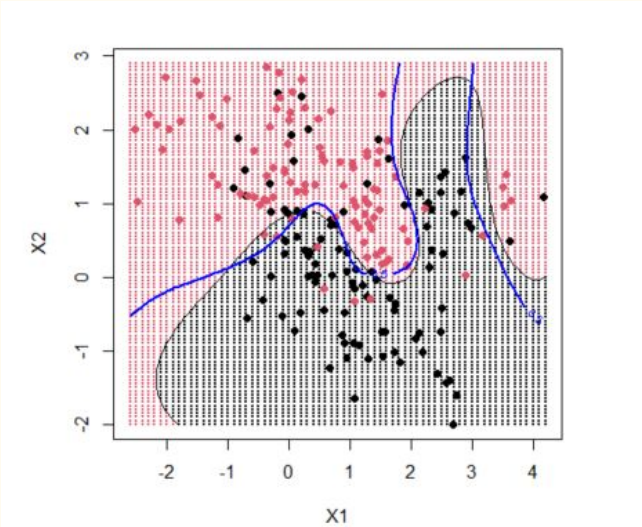
$$\text{s.t.} \sum_{i=1}^{n} \alpha_i y_i = 0 \quad \forall i = 1, \ldots, n,$$

$$0 \le \alpha_i \le C \quad \forall i = 1, \ldots, n,$$

2. Classification:

d. Support Vector Machine

iv. The dual Problem and Kernel Tricks



The dual problem is:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

$$\text{s.t.} \sum_{i=1}^{n} \alpha_i y_i = 0, \quad \forall i = 1, \ldots, n,$$

$$0 \leq \alpha_i \leq C, \quad \forall i = 1, \ldots, n,$$

This is a convex quadratic optimization problem

Bayesian boundary

Decision boundary using SVM

2. Classification:

d. Support Vector Machine

i. Support Vector Regression

$$\min \ \|w\|_2^2 + C\sum_{i=1}^{n}(\xi_i^+ + \xi_i^-)$$

$$\text{s.t.} \quad y_i - w^\top x_i - \gamma \le \epsilon + \xi^+ \quad \forall i = 1, \ldots, n,$$

$$w^\top x_i + \gamma - y_i \le \epsilon + \xi^- \quad \forall i = 1, \ldots, n,$$

$$w \in \mathbb{R}^p, \gamma \in \mathbb{R},$$

$$\xi_i^+, \ \xi_i^- \ge 0 \quad \forall i = 1, \ldots, n.$$

Corresponding optimization problem

2. Classification:

d. Support Vector Machine

i.Support Vector Ordinal Regression

$$
\begin{aligned}
\min \quad & \|w\|_2^2 + C \sum_{j=1}^{r-1} \left(\sum_{k=1}^{j} \sum_{i=1}^{n_k} \xi_{i,kj}^{+} + \sum_{k=j+1}^{r} \sum_{i=1}^{n_k} \xi_{i,kj}^{-} \right) \\
\text{s.t.} \quad & w^\top x_{i,k} - \beta_j \le -1 + \xi_{i,kj}^{+} \\
& \quad \forall k = 1, \ldots, j, \ j = 1, \ldots, r-1, \ i = 1, \ldots, n_k, \\
& w^\top x_{i,k} - \beta_j \ge 1 - \xi_{i,kj}^{-} \\
& \quad \forall k = j+1, \ldots, r, \ j = 1, \ldots, r-1, \ i = 1, \ldots, n_k, \\
& w \in \mathbb{R}^p, \ \beta_j \in \mathbb{R} \quad \forall j = 1, \ldots, r-1, \\
& \xi_{i,kj}^{+} \ge 0 \quad \forall k = 1, \ldots, j, \ j = 1, \ldots, r-1, \ i = 1, \ldots, n_k, \\
& \xi_{i,kj}^{-} \ge 0 \quad \forall k = j+1, \ldots, r, \ j = 1, \ldots, r-1, \ i = 1, \ldots, n_k.
\end{aligned}
$$

SVM Formulation

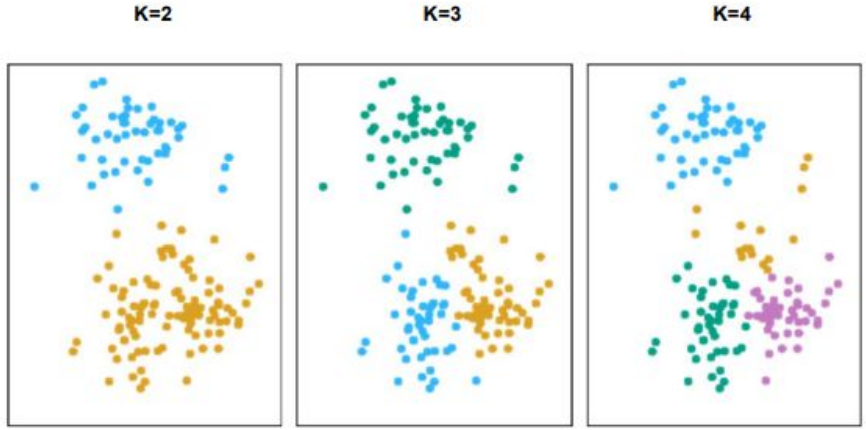
3. Clustering

d. Support Vector Machine

i. Minimum sum-of-squares clustering(K-means Clustering)

ii. Capacitated Clustering

iii. K-hyperplane Clustering



$$\min \sum_{i=1}^{n}\sum_{j=1}^{K} u_{ij}\|x_i - \mu_j\|_2^2$$

$$\text{s.t.} \quad \sum_{j=1}^{K} u_{ij} = 1 \quad \forall i = 1, \ldots, n,$$

$$\mu_j \in \mathbb{R}^p \quad \forall j = 1, \ldots, K,$$

$$u_{ij} \in \{0, 1\} \quad \forall i = 1, \ldots, n, \; j = 1, \ldots, K.$$

Mixed Integer nonlinear programming

Many common solution approaches for K-means clustering are based on heuristics

3. Clustering

d. Support Vector Machine

i. Minimum sum-of-squares clustering(K-means Clustering)

ii. Capacitated Clustering

iii. K-hyperplane Clustering

$$\min \sum_{i=1}^{n}\sum_{j=1}^{K} u_{ij}\|x_i - \mu_j\|_2^2$$

$$\text{s.t.} \quad \sum_{j=1}^{K} u_{ij} = 1 \quad \forall i = 1, \ldots, n,$$

$$\mu_j \in \mathbb{R}^p \quad \forall j = 1, \ldots, K,$$

$$u_{ij} \in \{0, 1\} \quad \forall i = 1, \ldots, n, \; j = 1, \ldots, K.$$

It deals with finding a set of clusters with a capacity limitation and homogeneity expressed by the similarity to the cluster centre

3. Clustering

d. Support Vector Machine

i. Minimum sum-of-squares clustering(K-means Clustering)

ii. Capacitated Clustering

iii. K-hyperplane Clustering

$$\min \sum_{i=1}^{n} \delta_i^2$$

$$\text{s.t} \quad \sum_{j=1}^{K} u_{ij} = 1 \quad \forall i = 1, \ldots, n,$$

$$\delta_i \geq (w_j^T x_i - \gamma_j) - M(1 - u_{ij}) \quad \forall i = 1, \ldots, n, \ j = 1, \ldots, K, \tag{73}$$

$$\delta_i \geq (-w_j^T x_i + \gamma_j) - M(1 - u_{ij}) \quad \forall i = 1, \ldots, n, \ j = 1, \ldots, K, \tag{74}$$

$$\|w_j\|_2 \geq 1 \quad \forall j = 1, \ldots, K, \tag{75}$$

$$\delta_i \geq 0 \quad \forall i = 1, \ldots, n, \tag{76}$$

$$w_j \in \mathbb{R}^p, \gamma_j \in \mathbb{R} \quad \forall j = 1, \ldots, K, \tag{77}$$

$$u_{ij} \in \{0, 1\} \quad \forall i \in 1, \ldots, n, \ j = 1, \ldots, K. \tag{78}$$

A hyperplane, instead of a center, is associated with each cluster

4. Linear dimension reduction

a. **Principal components Analysis**
b. Partial least squares

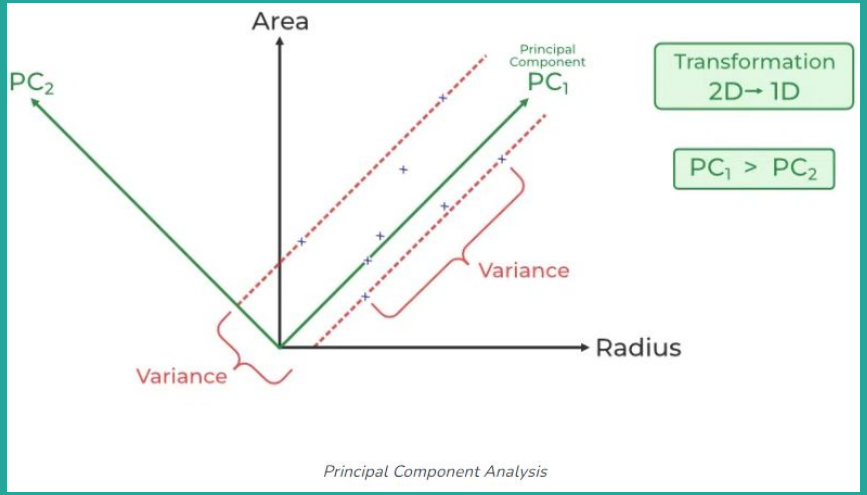
PCA is a dimensionality reduction technique used to transform high-dimensional data into a lower-dimensional space while preserving as much variance as possible.

One challenge of PCA is interpretability, which can be addressed by promoting sparsity in the projected components using techniques like Mixed Integer Nonlinear Programming (MINLP) or bi-objective optimization.

$$\max_{\phi^1 \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_j^1 x_{ij} \right)^2$$

$$\text{s.t.} \sum_{j=1}^{p} (\phi_j^1)^2 = 1.$$

Traditional formulation of PCA and can be solved via Lagrange multipliers methods.



4. Linear dimension reduction

a. Principal components Analysis
b. Partial least squares

$$\max_{\phi^h \in \mathbb{R}^p} \mathrm{Corr}(y, X\phi^h)^2 \times \mathrm{Var}(X\phi^h)$$

$$\text{s.t.} \quad \sum_{j=1}^{p} (\phi_j^h)^2 = 1,$$

$$\phi^{h\top} S \phi^l = 0 \quad \forall l = 1, \ldots, h-1,$$

It is especially useful when dealing with datasets where the number of features is much larger than the number of observations.

Difference

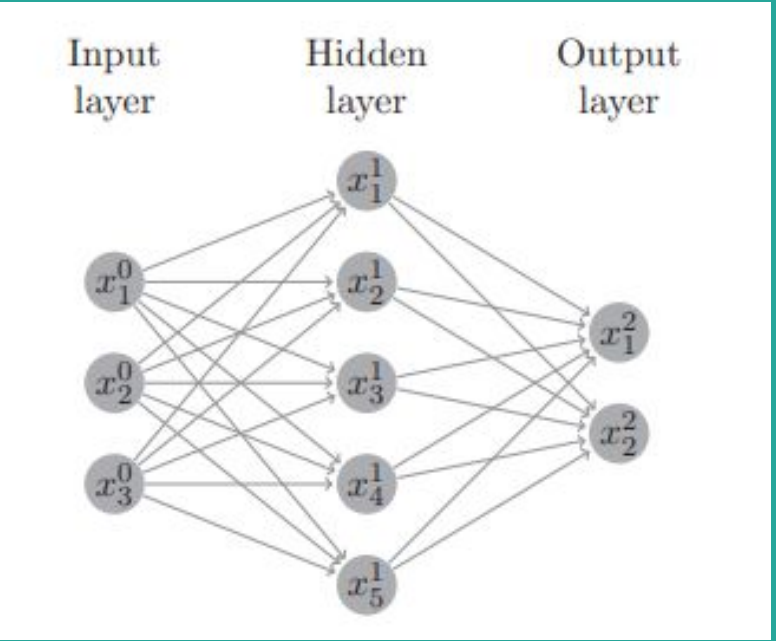
PCR	PLS
PCR focuses on maximizing variance alone.	PLS seeks directions that balance high variance and high correlation with the response variable.

Table 1
Notation for DNN architectures.

$\{0, \ldots, L\}$	*layers* indices.
n^l	number of *units*, or *neurons*, in layer l.
σ	element-wise activation function.
$U(j, l)$	jth unit of layer l.
$W^l \in \mathbb{R}^{n^l \times n^{l+1}}$	weight matrix for layer $l < L$.
$b^l \in \mathbb{R}^{n^l}$	bias vector for layer $l > 0$.
(X, y)	*training* dataset, with observations x_i and responses $y_i, i = 1, \ldots, n$.
x^l	output vector of layer l ($l = 0$ indicates *input feature* vector, $l > 0$ indicates *derived feature* vector).

5. Deep Learning

Purpose of optimization is to minimize training error



5. Deep Learning

a. **Mixed integer programming for DNN architectures**
b. Activation ensembles

$$\min \sum_{l=0}^{L}\sum_{j=1}^{n_l} c_j^l x_j^l + \sum_{l=1}^{L}\sum_{j=1}^{n_l} \gamma_j^l z_j^l \quad (89)$$

$$\text{s.t.} \sum_{i=1}^{n_{l-1}} w_{ij}^{l-1} x_i^{l-1} + b_j^{l-1} = x_j^l - s_j^l \qquad \forall l = 1,\dots,L,\ j = 1,\dots,n_l, \quad (90)$$

$$x_j^l \le (1 - z_j^l) M_x^{j,l} \qquad \forall l = 1,\dots,L,\ j = 1,\dots,n_l, \quad (91)$$

$$s_j^l \ge z_j^l M_s^{j,l} \qquad \forall l = 1,\dots,L,\ j = 1,\dots,n_l, \quad (92)$$

$$0 \le x_j^l \le ub_j^l \qquad \forall l = 1,\dots,L,\ j = 1,\dots,n_l, \quad (93)$$

$$0 \le s_j^l \le \overline{ub}_j^l \qquad \forall l = 1,\dots,L,\ j = 1,\dots,n_l, \quad (94)$$

The following mixed integer linear problem is proposed

Applications of MIP-based DNN modeling,

- pooling operations,
- maximizing unit activations,
- crafting adversarial examples, and
- training DNNs.

5. Deep Learning

a. Mixed integer programming for DNN architectures

Limitations in MIP-based DNN modeling:

- Computational Complexity
- Weak Continuous Relaxation
- Choice of Constants
- Verification Limitations
- Efficiency for Training
- Numerical Conditioning
- Limited Contributions

Approaches aim to Enhance the ability to analyze and verify the correctness and safety of neural networks

- Satisfiability-based Verification
- Simplex Method Extensions

5. Deep Learning

a. Mixed integer programming for DNN architectures

b. Activation ensembles

$$\min \sum_{j=1}^{m} \frac{1}{2}(\alpha^j - \hat{\alpha}^j)^2$$

$$\text{s.t.} \quad \sum_{j=1}^{m} \alpha^j = 1,$$

$$\alpha^j \geq 0 \quad \forall j = 1, \ldots, m,$$

Activation ensembles introduce an optimization process that allows neural networks to dynamically select and weight activation functions, enhancing their ability to adapt to complex data patterns and improving overall model performance.

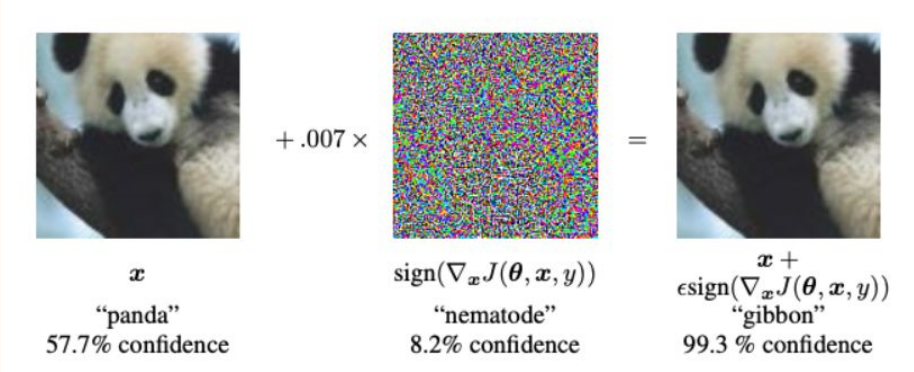
Activation ensembles explore the concept of using multiple activation functions within network layers to enhance the model's expressiveness.

6. Adversarial learning

- Adversarial learning in machine learning involves creating deceptive data to trick models into making errors, often with small, imperceptible changes.

- Attackers can exploit both known and unknown model details.

- Researchers study this to make models more robust and secure against such attacks.



a. Targeted attacks b .Untargeted attacks

c. adversarial robustness d.Data Poisoning

6. Adversarial learning

a. **Targeted attacks**
b. Untargeted attacks
c. adversarial robustness
d. Data Poisoning

It involves finding a perturbation to an input that misclassifies it as a specific target class while ensuring the perturbed input remains within the feasible input space.

This optimization problem is computationally challenging, especially for complex neural network classifiers

6. Adversarial learning

a. **Targeted attacks**
b. Untargeted attacks
c. adversarial robustness
d. Data Poisoning

Approaches have been proposed to address this challenge and find adversarial perturbations effectively:

- **Box-Constraint Approximation**
- **Alternative Approximation**

$$\min_{r\in\mathbb{R}^p} \|r\|_2 \quad \text{s.t.} \quad f(x+r)=y', \quad x+r\in\psi.$$

$$\min_{r\in\mathbb{R}^p} c|r| + \mathcal{L}(x+r,y') \quad \text{s.t. } x+r\in[0,1]^p,$$

$$\min_{r\in\mathbb{R}^p} \|r\|_l + \Lambda\mathcal{F}(x+r) \quad \text{s.t. } x+r\in\psi,$$

6. Adversarial learning

a. Targeted attacks
b. Untargeted attacks
c. adversarial robustness
d. Data Poisoning

$$\min_{r\in\mathbb{R}^p, z\in\Upsilon} d(r)$$

$$\text{s.t.} \quad z - y \le -\epsilon + Ms,$$

$$z - y \ge \epsilon - (1-s)M,$$

$$z \in \arg\max_{y'\in\Upsilon}\{f_{y'}(x+r)\},$$

$$x + r \in \psi,$$

$$s \in \{0, 1\},$$

Bilevel optimization problem: involves an inner-level optimization to determine the classified label.

Alternative Formulation for Untargeted Attacks:

- Introduction of ReLU Functions
- Complexity

6. Adversarial learning

a. Targeted attacks
b. Untargeted attacks
c. adversarial robustness
d. Data Poisoning

$$\min_{\theta} \mathbb{E}_{(x,y)} \left[\max_{r \in \mathcal{S}} \mathcal{L}(\theta; x + r, y) \right],$$

$$\min_{\theta} \sum_{x \in X} \max_{r \in \mathcal{S}_x} \mathcal{L}(\theta; x + r, y).$$

Optimizing with Respect to Worst-Case Data: An alternating ascent and descent steps procedure is often used to find solutions to this optimization problem

This formulation is essential for formalizing the process of adversarial training and quantifying the network's robustness.

Limitation:

Adversarial training, while effective, does not provide guarantees against all types of adversarial attacks.

6. Adversarial learning

a. Targeted attacks
b. Untargeted attacks
c. adversarial robustness
d. Data Poisoning

$$\max_{S \in \Gamma_T'} g(w_T)$$

$$\text{s.t. } |\{S \setminus \Gamma\}| \leq U,$$

$$w_t = w_0 - \sum_{\tau=0}^{t-1} \eta_\tau (\nabla \mathcal{L}(\omega_\tau, S_\tau) + \nabla \mathcal{L}(w_\tau)), 1 \leq t \leq T.$$

Data poisoning attacks are a class of attacks that aim to decrease the training accuracy of machine learning classifiers. Data poisoning attacks,which typically involve two steps:

- Data Cleaning
- Margin Based loss minimization

Poisoning attacks can be performed in different ways, including

- semi-online and
- fully-online scenarios

Emerging Applications in Machine Learning Within Optimisation

- **Machine Teaching,**
- Empirical Model learning,
- Bayesian network Structure learning

Emerging Applications in Machine Learning within Optimisation

1. **Machine Teaching,**
2. Empirical Model learning,
3. Bayesian network Structure learning.

$$\min_{\Gamma,\theta} \mathrm{TC}(\Gamma) \quad \text{s.t.} \quad \mathrm{TR}(\theta) \le \epsilon, \quad \theta = \mathrm{L}(\Gamma),$$

$$\min_{\Gamma,\theta} \mathrm{TR}(\theta) \quad \text{s.t.} \quad \mathrm{TC}(\Gamma) \le B, \quad \theta = \mathrm{L}(\Gamma).$$

Bilevel Optimization Problem

Machine Teaching Framework::

- Teaching Risk (TR)
- Teaching Cost (TC).
- Learner (L)

Emerging Applications in Machine Learning within Optimisation

1. Machine Teaching
2. **Empirical Model learning**
3. Bayesian network Structure learning

$$\min f(\eta, z)$$

$$\text{s.t.} \quad g_j(\eta, z) \quad \forall j \in J,$$

$$z = h(\eta),$$

$$\eta_i \in D_i \quad \forall i = 1, \ldots, n.$$

Embedding techniques utilize optimization approaches such as:

- mixed-integer nonlinear programming,
- constraint programming,
- SAT Modulo Theories, and
- local search.

while traditional "what-if" approaches primarily use predictive models to estimate parameters for separate optimization models, Empirical Model Learning takes a more integrated and dynamic approach by directly incorporating machine learning models into the decision-making process.

Emerging Applications in Machine Learning within Optimisation

1. Machine Teaching
2. Empirical Model learning
3. **Bayesian network Structure learning**

$$\max \sum_{i \in N} \sum_{t=1}^{r_i} p_{it} s_i(P_{it})$$

$$\text{s.t.} \quad \sum_{j \in N} y_{ij} \leq w, \quad \forall i \in N,$$

$$(|N|+1) y_{ij} \leq |N| + z_j - z_i \quad \forall i, j \in N,$$

$$y_{ij} + y_{ik} - y_{jk} - y_{kj} \leq 1 \quad \forall i, j, k \in N,$$

$$\sum_{t=1}^{r_i} p_{it} = 1 \quad \forall i \in N,$$

$$(|N|+1) p_{it} \leq |N| + v_j - v_i \quad \forall i \in N, \ \forall t = 1, \ldots, r_i, \ \forall j \in P_{it},$$

$$p_{it} \leq y_{ij} + y_{ji} \quad \forall i \in N, \ \forall t = 1, \ldots, r_i, \ \forall j \in P_{it},$$

$$p_{it} \leq y_{jk} + y_{kj} \quad \forall i \in N, \ \forall t = 1, \ldots, r_i, \ \forall j, k \in P_{it},$$

$$z_i \in [0, |N|], \ v_i \in [0, |N|], \ y_{ij} \in \{0, 1\}, \ p_{it} \in \{0, 1\} \quad \forall i, j \in N, \quad \forall t = 1, \ldots, r_i.$$

- Bayesian network structure learning, an emerging application involving the learning of probabilistic graphical model structures from data.

Emerging Applications in Machine Learning within Optimisation

1. Machine Teaching
2. Empirical Model learning
3. **Bayesian network Structure learning**

Bounded tree-width restricts the complexity of Bayesian network structures for computational tractability.

Computational Challenges and Heuristics:

- Computational Complexity: Both formulations become computationally demanding with increasing features or tree-width.
- Solution Approaches: Several search heuristics are proposed to find approximate solutions efficiently.

Strength and Shortcomings of Optimisation Model

Strengths:

Optimization models offer a structured framework for tackling a wide range of machine learning challenges, including regression, classification, clustering, deep learning, and adversarial learning.
They provide the flexibility to incorporate diverse constraints and objectives, making it possible to address complex machine learning problems effectively.
Optimization techniques have already demonstrated their value in various machine learning scenarios, capitalizing on advancements in numerical optimization methods.

Strengths:

Optimization models offer a structured framework for tackling a wide range of machine learning challenges, including regression, classification, clustering, deep learning, and adversarial learning.
They provide the flexibility to incorporate diverse constraints and objectives, making it possible to address complex machine learning problems effectively.
Optimization techniques have already demonstrated their value in various machine learning scenarios, capitalizing on advancements in numerical optimization methods.

Strengths:

Optimization models offer a structured framework for tackling a wide range of machine learning challenges, including regression, classification, clustering, deep learning, and adversarial learning.
They provide the flexibility to incorporate diverse constraints and objectives, making it possible to address complex machine learning problems effectively.
Optimization techniques have already demonstrated their value in various machine learning scenarios, capitalizing on advancements in numerical optimization methods.

Shortcomings:

The paper doesn't explicitly outline the shortcomings of optimization models in machine learning, it hints at the existence of challenges and open problems that warrant further exploration.

Practical Implications

Practical Implications:

1. The paper's optimization models apply to various machine learning tasks, including regression, classification, clustering, deep learning, and emerging areas like machine teaching and Bayesian network structure learning.
2. Numerical optimization advancements can improve the performance and efficiency of these machine learning models.
3. Commercial optimization software can solve large-scale machine learning optimization problems for optimal solutions or efficient heuristics.

Practical Implications:

1. The paper's optimization models apply to various machine learning tasks, including regression, classification, clustering, deep learning, and emerging areas like machine teaching and Bayesian network structure learning.
2. Numerical optimization advancements can improve the performance and efficiency of these machine learning models.
3. Commercial optimization software can solve large-scale machine learning optimization problems for optimal solutions or efficient heuristics.

Practical Implications:

1. The paper's optimization models apply to various machine learning tasks, including regression, classification, clustering, deep learning, and emerging areas like machine teaching and Bayesian network structure learning.
2. Numerical optimization advancements can improve the performance and efficiency of these machine learning models.
3. Commercial optimization software can solve large-scale machine learning optimization problems for optimal solutions or efficient heuristics.

Conclusions

Conclusion:

1. The paper introduces optimization models for various machine learning tasks, including regression, classification, clustering, deep learning, and adversarial learning.

2. It emphasizes the significance of mathematical programming in machine learning and discusses both the strengths and limitations of these models, along with research directions and open problems.

3. The paper highlights the impact of numerical optimization techniques and commercial software in large-scale machine learning while exploring new applications like machine teaching and empirical model learning.

Conclusion:

1. The paper introduces optimization models for various machine learning tasks, including regression, classification, clustering, deep learning, and adversarial learning.

2. It emphasizes the significance of mathematical programming in machine learning and discusses both the strengths and limitations of these models, along with research directions and open problems.

3. The paper highlights the impact of numerical optimization techniques and commercial software in large-scale machine learning while exploring new applications like machine teaching and empirical model learning.

Conclusion:

1. The paper introduces optimization models for various machine learning tasks, including regression, classification, clustering, deep learning, and adversarial learning.

2. It emphasizes the significance of mathematical programming in machine learning and discusses both the strengths and limitations of these models, along with research directions and open problems.

3. The paper highlights the impact of numerical optimization techniques and commercial software in large-scale machine learning while exploring new applications like machine teaching and empirical model learning.

Thank You