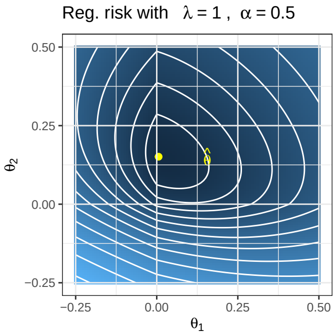
Optimization in Machine Learning

Optimization Problems
Unconstrained problems



Learning goals

- Definition
- Max. likelihood
- Linear regression
- Regularized risk minimization
- SVM
- Neural network

UNCONSTRAINED OPTIMIZATION PROBLEM

$$\min_{\mathbf{x} \in \mathcal{S}} f(\mathbf{x})$$

with objective function

$$f : \mathcal{S} \to \mathbb{R}.$$

The problem is called

- **unconstrained**, if the domain $\mathcal{S}$ is not restricted:

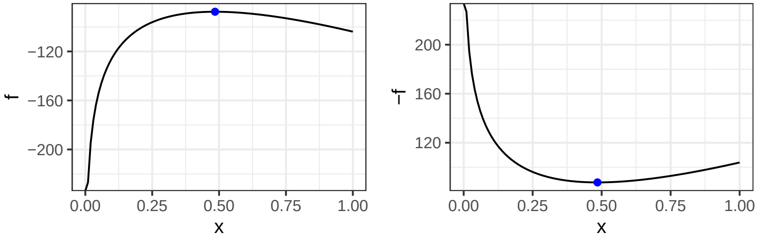
$$\mathcal{S} = \mathbb{R}^d$$

- **smooth** if f is at least $\in \mathcal{C}^1$
- **univariate** if $d = 1$, and **multivariate** if $d > 1$.
- **convex** if f convex function and $\mathcal{S}$ convex set

NOTE: A CONVENTION IN OPTIMIZATION

We always **minimize** functions f.

Maximization results from minimizing $-f$.

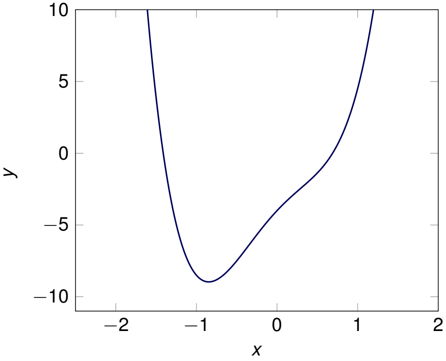


The solution to maximizing f (left) is equivalent to the solution to minimizing f (right).

EXAMPLE 1: UNIVARIATE CONVEX FUNCTION

$$\min_{x \in \mathbb{R}} f(x)$$

$$f(x) = 5 \cdot x^4 + \frac{1}{2} \cdot (x-2)^3$$



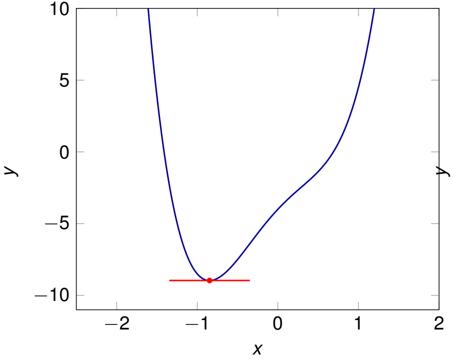
EXAMPLE 1: UNIVARIATE CONVEX FUNCTION

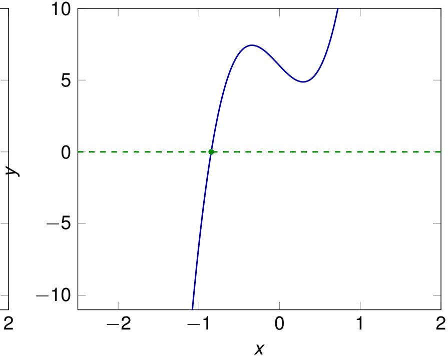
Extrema:

$$f(x) = 5 \cdot x^4 + \frac{1}{2} \cdot (x-2)^3$$

Condition:

$$f'(x) = 20 \cdot x^3 + \frac{3}{2} \cdot (x-2)^2 = 0$$





EXAMPLE 1: UNIVARIATE CONVEX FUNCTION

Condition:

$$f'(x) = 20a \cdot x^3 + \frac{3}{2} \cdot (x-2)^2 = 0$$

$$ax^3 + bx^2 + cx + d = 0$$

Discriminant:

$$\Delta = 18abcd - 4b^3d + b^2c^2 - 4ac^3 - 27a^2d^2 = -390960$$

If $\Delta < 0$, there is only one real root.

EXAMPLE 1: UNIVARIATE CONVEX FUNCTION / 2

Substitution with $y = x + \frac{b}{3a}$ results in

$$y^3 + py + q = 0$$

with

$$p = \frac{3ac - b^2}{3a^2}$$

$$q = \frac{2b^3}{27a^3} - \frac{bc}{3a^2} + \frac{d}{a}$$

Root:

$$y_1 = \sqrt[3]{-\frac{q}{2} + \sqrt{\left(\frac{q}{2}\right)^2 + \left(\frac{p}{3}\right)^3}} + \sqrt[3]{-\frac{q}{2} - \sqrt{\left(\frac{q}{2}\right)^2 + \left(\frac{p}{3}\right)^3}}$$

$$x_1 = y_1 - \frac{b}{3a} = -0.847151$$

WHAT'S THE MATTER?

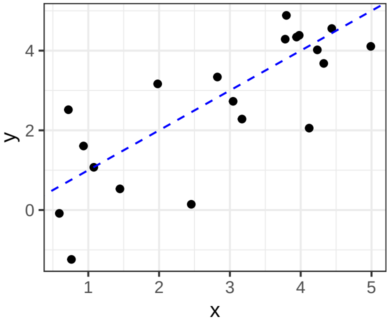
Can we always just calculate the optimal solution from the tangent equation?

- Solving $f'(x) = 0$ can be challenging in univariant cases
- Only applicable for continuous and differentiable cases
- In multivariant cases one could solve $det(\mathbf{J}(\mathbf{x})) = 0$ for $\mathbf{x}$
- Determinant is nonlinear in $\mathbf{x}$ and therefore the equation can be hard to solve analytically for $\mathbf{x}$ with dimension higher than 3
- In practice, often more convenient to just use iterative algorithms from the start

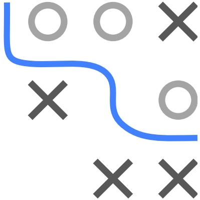
EXAMPLE 2: NORMAL REGRESSION

Assume (multivariate) data $\mathcal{D} = \left(\left(x^{(1)}, y^{(1)}\right), \ldots, \left(x^{(n)}, y^{(n)}\right)\right)$ and we want to fit a linear function to it

$$y = f(x) = \boldsymbol{\theta}^\top \mathbf{x} = \theta_1 + \theta_0 \cdot x, \qquad \mathbf{x} = \begin{pmatrix} x \\ 1 \end{pmatrix}, \boldsymbol{\theta} = \begin{pmatrix} \theta_0 \\ \theta_1 \end{pmatrix}$$

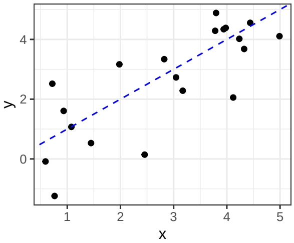


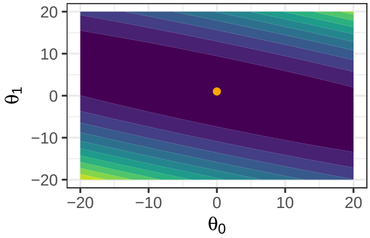
EXAMPLE 2: LEAST SQUARES LINEAR REGR.



Find param vector $\boldsymbol{\theta}$ that minimizes sum of square errors (SSE) / risk with L2 loss

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \sum_{i=1}^{n} \left(\boldsymbol{\theta}^\top \mathbf{x}^{(i)} - y^{(i)} \right)^2$$





- **Smooth**, **multivariate**, **unconstrained**, **convex** problem
- Analytic solution: $\boldsymbol{\theta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \boldsymbol{y}$, where $\mathbf{X}$ is design matrix

EXAMPLE 2: LEAST SQUARES LINEAR REGR. / 2

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \sum_{i=1}^{n} \left(\boldsymbol{\theta}^\top \mathbf{x}^{(i)} - y^{(i)}\right)^2$$

$$\mathbf{X} = \begin{pmatrix}\mathbf{x}^{(1)} & \dots & \mathbf{x}^{(n)}\end{pmatrix}^\top, \qquad \mathbf{y} = \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{pmatrix}$$

$$\sum_{i=1}^{n} \left(\boldsymbol{\theta}^\top \mathbf{x}^{(i)} - y^{(i)}\right)^2 = (\mathbf{X}\boldsymbol{\theta} - \mathbf{y})^\top (\mathbf{X}\boldsymbol{\theta} - \mathbf{y})$$

Ableiten nach $\boldsymbol{\theta}$:

$$2\mathbf{X}^\top (\mathbf{X}\boldsymbol{\theta} - \mathbf{y})^\top = \mathbf{0}$$

$$\mathbf{X}^\top\mathbf{X}\boldsymbol{\theta} - \mathbf{X}^\top\mathbf{y} = \mathbf{0}$$

$$\boldsymbol{\theta} = \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{y}$$

RISK MINIMIZATION IN ML

In the above example, if we exchange

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \sum_{i=1}^{n} \left(\boldsymbol{\theta}^\top \mathbf{x}^{(i)} - y^{(i)} \right)^2$$

- the linear model $\boldsymbol{\theta}^\top \mathbf{x}$ by an arbitrary model $f(\mathbf{x} \mid \boldsymbol{\theta})$
- the L2-loss $(f(\mathbf{x} \mid \boldsymbol{\theta}) - y)^2$ by any loss $L\left(y, f(\mathbf{x})\right)$

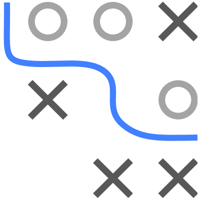
we arrive at general **empirical risk minimization** (ERM)

$$\mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) = \sum_{i=1}^{n} L\left(y^{(i)}, f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)\right) = \min!$$

Usually, we add a regularizer to counteract overfitting:

$$\mathcal{R}_{\text{reg}}(\boldsymbol{\theta}) = \sum_{i=1}^{n} L\left(y^{(i)}, f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)\right) + \lambda J(\boldsymbol{\theta}) = \min!$$

RISK MINIMIZATION IN ML / 2



ML models usually consist of the following components:

$$\textbf{ML} = \underbrace{\textbf{Hypothesis Space + Risk + Regularization}}_{\text{Formulating the optimization problem}} + \underbrace{\textbf{Optimization}}_{\text{Solving it}}$$

- **Hypothesis Space:** Parametrized function space
- **Risk:** Measure prediction errors on data with loss L
- **Regularization:** Penalize model complexity
- **Optimization:** Practically minimize risk over parameter space

EXAMPLE 3: REGULARIZED LINEAR MODEL

ERM with L2 loss, Linear Model (LM), and L2 regularization term:

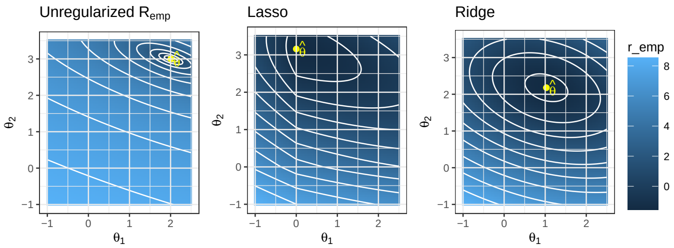
$$\mathcal{R}_{\text{reg}}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \left(\boldsymbol{\theta}^\top \mathbf{x}^{(i)} - y^{(i)}\right)^2 + \lambda \cdot \|\boldsymbol{\theta}\|_2^2 \quad \text{(Ridge regr.)}$$

Problem **multivariate**, **unconstrained**, **smooth**, **convex** and has analytical solution $\boldsymbol{\theta} = (\mathbf{X}^\top \mathbf{X} + \lambda \boldsymbol{I})^{-1} \mathbf{X}^\top \boldsymbol{y}$.

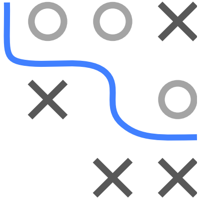
ERM with L2-loss, LM, and L1 regularization:

$$\mathcal{R}_{\text{reg}}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \left(\boldsymbol{\theta}^\top \mathbf{x}^{(i)} - y^{(i)}\right)^2 + \lambda \cdot \|\boldsymbol{\theta}\|_1 \quad \text{(Lasso regression)}$$

The problem is still **multivariate**, **unconstrained**, **convex**, but **not smooth**.



EXAMPLE 3: REGULARIZED LINEAR MODEL / 2

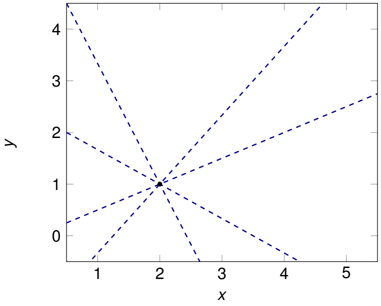


Why regularization?

- If number of variables θ exceeds number of data-points, the linear function cannot be fit properly
- There are infinite solutions
- The problem is ill-posed
- Machine-Learning-model can suffer from poor generalization

Regularization:

- Add a constraint to limit the solution-space
- Regularization can introduce a-priori knowledge about the problem (e.g. correlation of certain parameters)

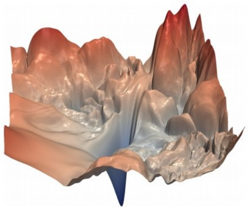


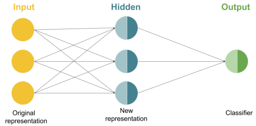
EXAMPLE 4: NEURAL NETWORK

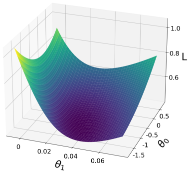
Normal loss, but complex f defined as computational feed-forward graph. Complexity of optimization problem

$$\arg\min_{\boldsymbol{\theta}} \mathcal{R}_{\text{reg}}(\boldsymbol{\theta}),$$

so smoothness (maybe) or convexity (usually no) is influenced by loss, neuron function, depth, regularization, etc.





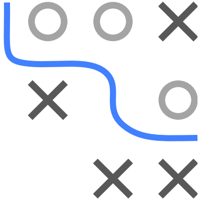


Loss landscapes of ML problems.

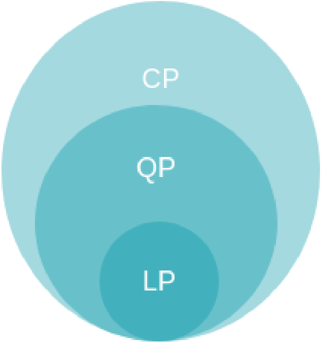
Left: Deep learning model ResNet-56, right: Logistic regression with cross-entropy loss

Source: `https://arxiv.org/pdf/1712.09913.pdf`

Optimization in Machine Learning



Optimization Problems
Constrained problems



Learning goals

- Definition
- LP, QP, CP
- Ridge and Lasso
- Soft-margin SVM

CONSTRAINED OPTIMIZATION PROBLEM

$$\min_{\mathbf{x} \in \mathcal{S}} f(\mathbf{x}), \text{ with } f: \ \mathcal{S} \to \mathbb{R}.$$

- **Constrained**, if domain $\mathcal{S}$ is restricted: $\mathcal{S} \subsetneq \mathbb{R}^d$.
- **Convex** if f convex function and $\mathcal{S}$ convex set
- Typically $\mathcal{S}$ is defined via functions called **constraints**

$$\mathcal{S} := \{\mathbf{x} \in \mathbb{R}^d \mid g_i(\mathbf{x}) \leq 0, h_j(\mathbf{x}) = 0 \ \forall \ i, j\}, \text{ where}$$

  - $g_i : \mathbb{R}^d \to \mathbb{R}, i = 1, ..., k$ are called inequality constraints,
  - $h_j : \mathbb{R}^d \to \mathbb{R}, j = 1, ..., l$ are called equality constraints.

Equivalent formulation:

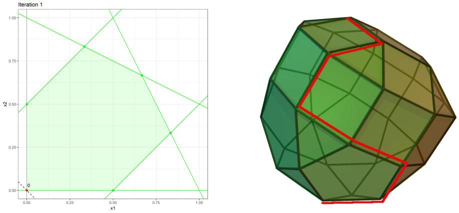
$$\begin{aligned} \min \quad & f(\mathbf{x}) & \\ \text{such that} \quad & g_i(\mathbf{x}) \leq 0 & \text{for } i = 1, \ldots, k \\ & h_j(\mathbf{x}) = 0 & \text{for } j = 1, \ldots, l. \end{aligned}$$

LINEAR PROGRAM (LP)

- f linear such that linear constraints. Standard form:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^d} \quad & \boldsymbol{c}^\top \mathbf{x} \\ \text{such that} \quad & \boldsymbol{A}\mathbf{x} \geq \boldsymbol{b} \\ & \mathbf{x} \geq 0 \end{aligned}$$

for $\boldsymbol{c} \in \mathbb{R}^d$, $\boldsymbol{A} \in \mathbb{R}^{k\times d}$ and $\boldsymbol{b} \in \mathbb{R}^k$.



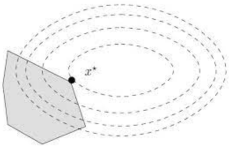
Visualization of constraints of 2D and 3D linear program (Source right figure: Wikipedia).

QUADRATIC PROGRAM (QP)

- f quadratic form such that linear constraints. Standard form:

$$
\begin{aligned}
\min_{\mathbf{x}\in\mathbb{R}^d} \quad & \frac{1}{2}\mathbf{x}^\top \boldsymbol{A}\mathbf{x} + \boldsymbol{b}^\top\mathbf{x} + c \\
\text{such that} \quad & \boldsymbol{E}\mathbf{x} \leq \boldsymbol{f} \\
& \boldsymbol{G}\mathbf{x} = \boldsymbol{h}
\end{aligned}
$$

$$\boldsymbol{A} \in \mathbb{R}^{d\times d}, \boldsymbol{b} \in \mathbb{R}^d, c \in \mathbb{R}, \boldsymbol{E} \in \mathbb{R}^{k\times d}, \boldsymbol{f} \in \mathbb{R}^k, \boldsymbol{G} \in \mathbb{R}^{l\times d}, \boldsymbol{f} \in \mathbb{R}^l.$$



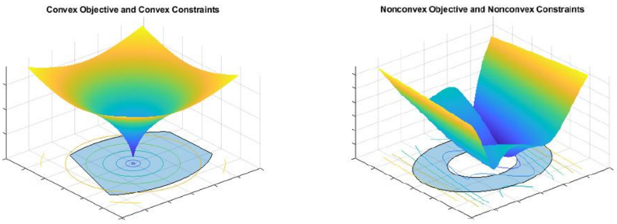
Visualization of quadratic objective (dashed) over linear constraints (grey). Source: Ma, Signal Processing Optimization Techniques, 2015.

CONVEX PROGRAM (CP)

- f convex, convex inequality constraints, linear equality constraints. Standard form:

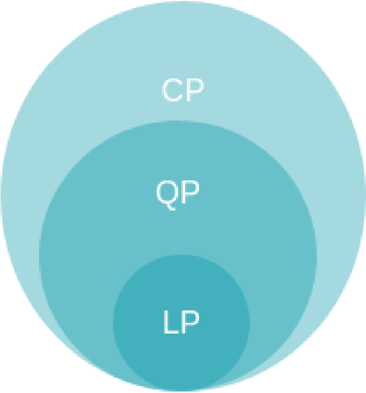
$$
\begin{aligned}
\min_{\mathbf{x} \in \mathbb{R}^d} \quad & f(\mathbf{x}) \\
\text{such that} \quad & g_i(\mathbf{x}) \leq 0, i = 1, \dots, k \\
& \boldsymbol{A}\mathbf{x} = \boldsymbol{b}
\end{aligned}
$$

for $\boldsymbol{A} \in \mathbb{R}^{l \times d}$ and $\boldsymbol{b} \in \mathbb{R}^{l}$.



Convex program (left) vs. nonconvex program (right). Source: Mathworks.

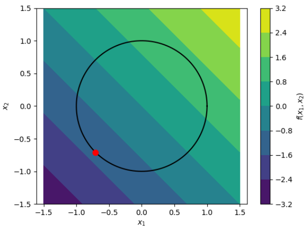
FURTHER TYPES



Quadratically constrained linear program (QCLP) and quadratically constrained quadratic program (QCQP).

EXAMPLE 1: UNIT CIRCLE

$$
\begin{array}{ll}
\min & f(x_1, x_2) = x_1 + x_2 \\
\text{s.t.} & h(x_1, x_2) = x_1^2 + x_2^2 - 1 = 0
\end{array}
$$



f, h smooth. Problem **not convex** ($\mathcal{S}$ is not a convex set).

Note: If the constraint is replaced by $g(x_1, x_2) = x_1^2 + x_2^2 - 1 \leq 0$, the problem is a convex program, even a quadratically constrained linear program (QCLP).

EXAMPLE 2: MAXIMUM LIKELIHOOD

Experiment: Draw m balls from a bag with balls of k different colors. Color j has a probability of p_j of being drawn.

The probability to realize the outcome $\mathbf{x} = (x_1, ..., x_k)$, x_j being the number of balls drawn in color j, is:

$$f(\mathbf{x}, m, \boldsymbol{p}) = \begin{cases} \frac{m!}{x_1! \cdots x_k!} \cdot p_1^{x_1} \cdots p_k^{x_k} & \text{if } \sum_{i=1}^{k} x_i = m \\ 0 & \text{otherwise} \end{cases}$$

The parameters p_j are subject to the following constraints:

$$0 \leq p_j \leq 1 \qquad \text{for all } i$$

$$\sum_{j=1}^{m} p_j = 1.$$

EXAMPLE 2: MAXIMUM LIKELIHOOD / 2

For a fixed m and a sample $\mathcal{D} = \left(\mathbf{x}^{(1)}, ..., \mathbf{x}^{(n)}\right)$, where $\sum_{j=1}^{k} \mathbf{x}_j^{(i)} = m$ for all $i = 1, ..., n$, the negative log-likelihood is:

$$
\begin{aligned}
-\ell(\boldsymbol{p}) &= -\log\left(\prod_{i=1}^{n} \frac{m!}{\mathbf{x}_1^{(i)}! \cdots \mathbf{x}_k^{(i)}!} \cdot p_1^{\mathbf{x}_1^{(i)}} \cdots p_k^{\mathbf{x}_k^{(i)}}\right) \\
&= \sum_{i=1}^{n} \left[-\log(m!) + \sum_{j=1}^{k} \log(\mathbf{x}_j^{(i)}!) - \sum_{j=1}^{k} \mathbf{x}_j^{(i)} \log(p_j)\right] \\
&\propto -\sum_{i=1}^{n} \sum_{j=1}^{k} \mathbf{x}_j^{(i)} \log(p_j)
\end{aligned}
$$

f, g, h are smooth.
Convex program: convex$^{(*)}$ objective + box/linear constraints).

$(*)$: log is concave, $-\log$ is convex, and the sum of convex functions is convex.

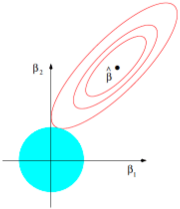
EXAMPLE 3: RIDGE REGRESSION

Ridge regression can be formulated as regularized ERM:

$$\hat{\theta}_{\text{Ridge}} = \arg\min_{\boldsymbol{\theta}} \left\{ \sum_{i=1}^{n} \left(y^{(i)} - \boldsymbol{\theta}^\top \mathbf{x} \right)^2 + \lambda ||\boldsymbol{\theta}||_2^2 \right\}$$

Equivalently it can be written as constrained optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \sum_{i=1}^{n} \left(\boldsymbol{\theta}^\top \mathbf{x}^{(i)} - y^{(i)} \right)^2 \\ \text{such that} \quad & \|\boldsymbol{\theta}\|_2 \leq t \end{aligned}$$



f, g smooth. **Convex program** (convex objective, quadratic constraint).

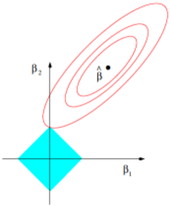
EXAMPLE 4: LASSO REGRESSION

Lasso regression can be formulated as regularized ERM:

$$\hat{\theta}_{\text{Lasso}} = \arg\min_{\boldsymbol{\theta}} \left\{ \sum_{i=1}^{n} \left(y^{(i)} - \boldsymbol{\theta}^\top \mathbf{x} \right)^2 + \lambda ||\boldsymbol{\theta}||_1 \right\}$$

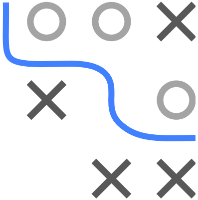
Equivalently it can be written as constrained optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \sum_{i=1}^{n} \left(\boldsymbol{\theta}^\top \mathbf{x}^{(i)} - y^{(i)} \right)^2 \\ \text{such that} \quad & \|\boldsymbol{\theta}\|_1 \leq t \end{aligned}$$



f smooth, g **not smooth**. Still **convex program**.

EXAMPLE 5: SUPPORT VECTOR MACHINES

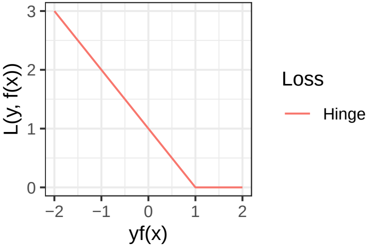


The SVM problem can be formulated in 3 equivalent ways: two primal, and one dual one (we will see later what "dual" means).

Here, we only discuss the nature of the optimization problems. A more thorough statistical derivation of SVMs is given in “Supervised learning”.

Formulation 1 (primal): ERM with Hinge loss

$$\sum_{i=1}^{n} \max\left(1 - y^{(i)} f^{(i)}, 0\right) + \lambda \|\boldsymbol{\theta}\|_2^2, \quad f^{(i)} := \boldsymbol{\theta}^\top \mathbf{x}^{(i)}$$

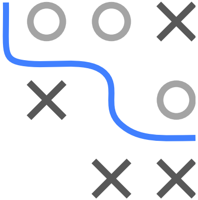


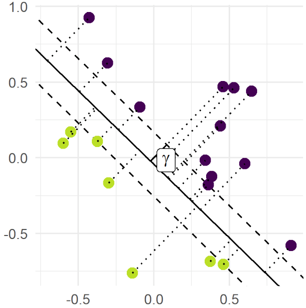
Unconstrained, convex problem with non-smooth objective

EXAMPLE 5: SUPPORT VECTOR MACHINES / 2

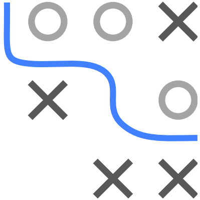
Formulation 2 (primal): Geometric formulation

- Find decision boundary which separates classes with **maximum** safety distance
- Distance to points closest to decision boundary ("safety margin γ") should be **maximized**





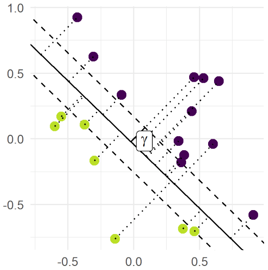
EXAMPLE 5: SUPPORT VECTOR MACHINES



Formulation 2 (primal): Geometric formulation

$$\min_{\boldsymbol{\theta}, \theta_0} \quad \frac{1}{2}\|\boldsymbol{\theta}\|^2$$

$$\text{such that} \quad y^{(i)}\left(\left\langle \boldsymbol{\theta}, \mathbf{x}^{(i)} \right\rangle + \theta_0\right) \geq 1 \quad \forall i \in \{1, \ldots, n\}$$

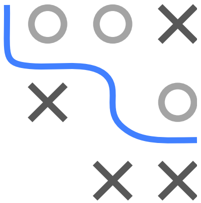


Maximize safety margin γ. No point is allowed to violate safety margin constraint.

The problem is a **QP**: Quadratic objective with linear constraints.

EXAMPLE 5: SUPPORT VECTOR MACHINES

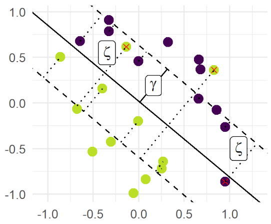
Formulation 2 (primal): Geometric formulation (soft constraints)



$$\min_{\boldsymbol{\theta},\boldsymbol{\theta}_0,\zeta^{(i)}} \quad \frac{1}{2}\|\boldsymbol{\theta}\|^2 + C\sum_{i=1}^{n}\zeta^{(i)}$$

$$\text{s.t.} \quad y^{(i)}\left(\left\langle \boldsymbol{\theta}, \mathbf{x}^{(i)} \right\rangle + \theta_0\right) \geq 1 - \zeta^{(i)} \quad \forall i \in \{1,\dots,n\},$$

$$\text{and} \quad \zeta^{(i)} \geq 0 \quad \forall i \in \{1,\dots,n\}.$$

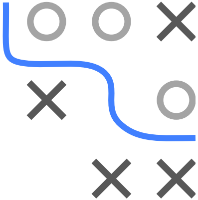


Maximize safety margin γ.

Margin violations are allowed, but are minimized.

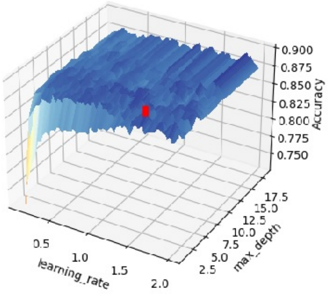
The problem is a **QP**: Quadratic objective with linear constraints.

Optimization in Machine Learning



Optimization Problems

Other optimization problems

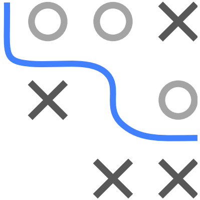


(a) vehicle

Learning goals

- Discrete / feature selection
- Black-box / hyperparameter optimization
- Noisy
- Multi-objective

OTHER CLASSES OF OPTIMIZATION PROBLEMS



So far: “nice” (un)constrained problems:

- Problem defined on continuous domain $\mathcal{S}$
- Analytical objectives (and constraints)

Other characteristics:

- Discrete domain $\mathcal{S}$
- f **black-box**: Objective not available in analytical form; computer program to evaluate
- f **noisy**: Objective can be queried but evaluations are noisy $f(\mathbf{x}) = f_{\text{true}}(\mathbf{x}) + \epsilon, \quad \epsilon \sim F$
- f **expensive**: Single query takes time / resources
- f multi-objective: $f(\mathbf{x}) : \mathcal{S} \to \mathbb{R}^m$, $f(\mathbf{x}) = (f_1(\mathbf{x}), ..., f_m(\mathbf{x}))$

These make the problem typically much harder to solve!

EXAMPLE 1: BEST SUBSET SELECTION

Let $\mathcal{D} = \left(\left(\mathbf{x}^{(i)}, y^{(i)}\right)\right)_{1 \leq i \leq n}$, $\mathbf{x}^{(i)} \in \mathbb{R}^p$. Fit LM based on best feature subset.

$$\min_{\boldsymbol{\theta} \in \Theta} \left(y^{(i)} - \boldsymbol{\theta}^\top \mathbf{x}^{(i)}\right)^2, ||\boldsymbol{\theta}||_0 \leq k$$

Problem characteristics:

- White-box: Objective available in analytical form
- Discrete: $\mathcal{S}$ is mixed continuous and discrete
- Constrained

The problem is even **NP-hard** (Bin et al., 1997, The Minimum Feature Subset Selection Problem)!

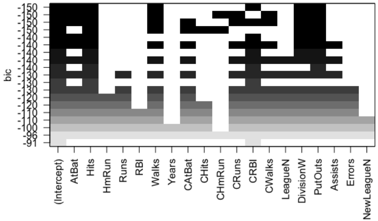


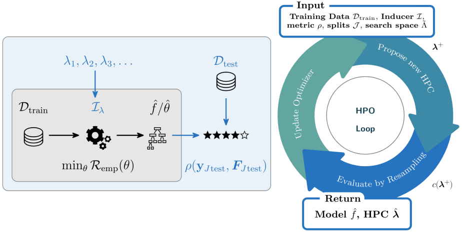
Figure: Source: RPubs, Subset Selection Methods

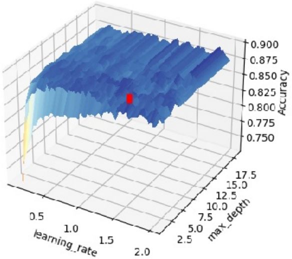
EXAMPLE 2: HYPERPARAMETER OPTIMIZATION

- Learner $\mathcal{I}$ usually configurable by hyperparameters $\boldsymbol{\lambda} \in \Lambda$.
- Find best HP configuration $\boldsymbol{\lambda}^*$

$$\boldsymbol{\lambda}^* \in \arg\min_{\boldsymbol{\lambda}\in\Lambda} c(\boldsymbol{\lambda}) = \arg\min \widehat{\mathrm{GE}}(\mathcal{I}, \mathcal{J}, \rho, \boldsymbol{\lambda})$$

$\widehat{\mathrm{GE}}$ general. err. with metric ρ and estim. with resampling splits $\mathcal{J}$





(a) vehicle

XGBoost HP landscape; source:

`ceur-ws.org/Vol-2846/paper22.pdf`

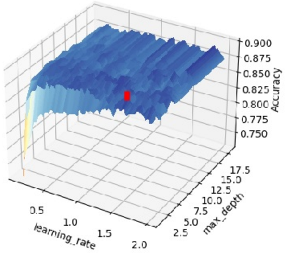
EXAMPLE 2: HYPERPARAMETER OPTIMIZATION / 2

Solving

$$\boldsymbol{\lambda}^* \in \arg\min_{\boldsymbol{\lambda} \in \Lambda} c(\boldsymbol{\lambda})$$

is very challenging:

- *c* black box
  eval by progrmm
- expensive
  1 eval: 1 or multiple ERM(s)!
- noisy
  uses data / resampling
- the search space Λ might be mixed
  continuous, integer, categ. or hierarchical



(a) vehicle

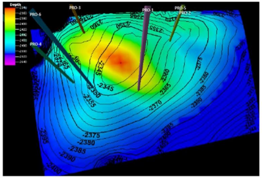
XGBoost HP landscape; source:

`ceur-ws.org/Vol-2846/paper22.pdf`

MORE BLACK-BOX PROBLEMS

Black-box problems from engineering: **oil well placement**

- The goal is to determine the optimal locations and operation parameters for wells in oil reservoirs
- Basic premise: achieving maximum revenue from oil while minimizing operating costs
- In addition, the objective function is subject to complex combinations of geological, economical, petrophysical and fluiddynamical constraints
- Each function evaluation requires several computationally expensive reservoir simulations while taking uncertainty in the reservoir description into account



Oil saturation at various depths with possible location of wells.

Source: https://doi.org/10.1007/s13202-019-0710-1