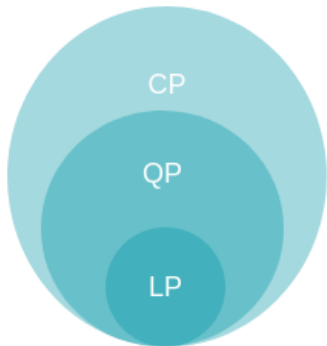


Optimization in Machine Learning

Optimization Problems

Constrained problems



Learning goals

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

CONSTRAINED OPTIMIZATION PROBLEM

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

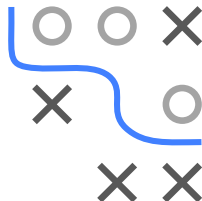
- **Constrained**, if domain \mathcal{S} is restricted: $\mathcal{S} \subsetneq \mathbb{R}^d$.
- **Convex** if f convex function and \mathcal{S} convex set
- Typically \mathcal{S} is defined via functions called **constraints**

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

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

Equivalent formulation:

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



CONVEX PROGRAM (CP)

- f convex, convex inequality constraints, linear equality constraints.

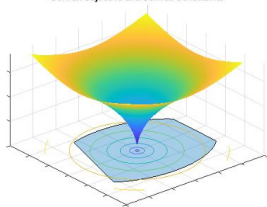
Standard form:

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^d} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \leq 0, i = 1, \dots, k \\ & \mathbf{Ax} = \mathbf{b} \end{aligned}$$

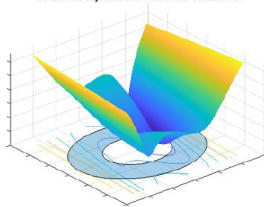
for $\mathbf{A} \in \mathbb{R}^{l \times d}$ and $\mathbf{b} \in \mathbb{R}^l$.



Convex Objective and Convex Constraints

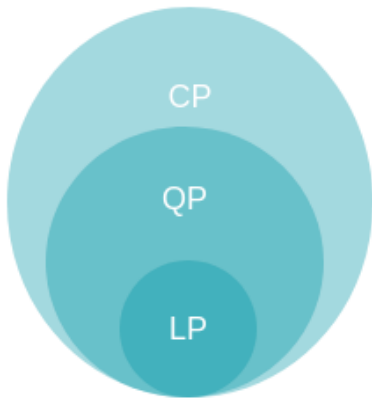


Nonconvex Objective and Nonconvex Constraints



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

FURTHER TYPES

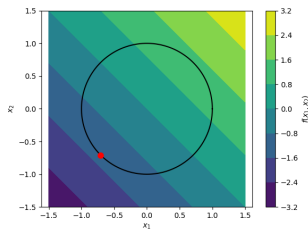


Quadratically constrained linear program (QCLP) and quadratically constrained quadratic program (QCQP).



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.

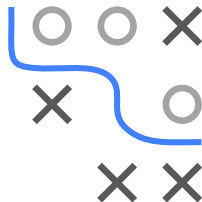
The probability to realize the outcome $\mathbf{x} = (x_1, \dots, x_k)$, x_j being the number of balls drawn in color j , is:

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

The parameters p_j are subject to the following constraints:

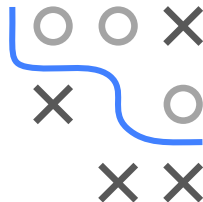
$$0 \leq p_j \leq 1 \quad \text{for all } i$$

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



EXAMPLE 2: MAXIMUM LIKELIHOOD / 2

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



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

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.

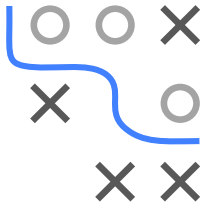
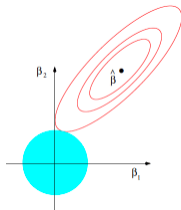
EXAMPLE 3: RIDGE REGRESSION

Ridge regression can be formulated as regularized ERM:

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

Equivalently it can be written as constrained optimization problem:

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



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

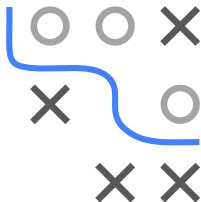
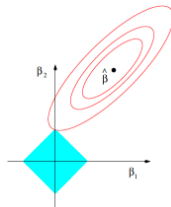
EXAMPLE 4: LASSO REGRESSION

Lasso regression can be formulated as regularized ERM:

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

Equivalently it can be written as constrained optimization problem:

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



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

EXAMPLE 5: SUPPORT VECTOR MACHINES

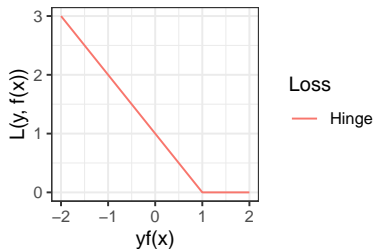
The SVM problem can be formulated in 3 equivalent ways: two primal, and one dual one (we will see later what "dual" means).

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

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

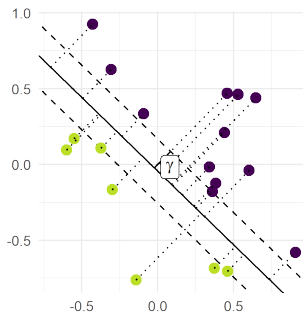


Unconstrained, convex problem
with non-smooth objective

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**

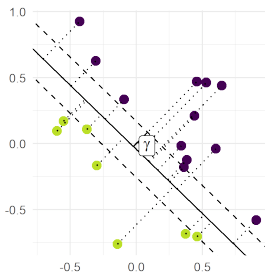


EXAMPLE 5: SUPPORT VECTOR MACHINES

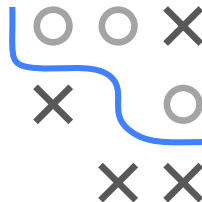
Formulation 2 (primal): Geometric formulation

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

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



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

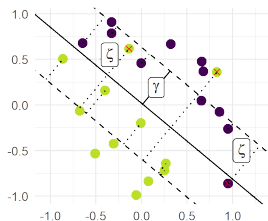


The problem is a **QP**: Quadratic objective with linear constraints.

EXAMPLE 5: SUPPORT VECTOR MACHINES

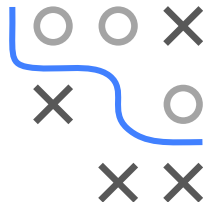
Formulation 2 (primal): Geometric formulation (soft constraints)

$$\begin{aligned} \min_{\theta, \theta_0, \zeta^{(i)}} \quad & \frac{1}{2} \|\theta\|^2 + c \sum_{i=1}^n \zeta^{(i)} \\ \text{s.t.} \quad & y^{(i)} \left(\langle \theta, \mathbf{x}^{(i)} \rangle + \theta_0 \right) \geq 1 - \zeta^{(i)} \quad \forall i \in \{1, \dots, n\}, \\ \text{and} \quad & \zeta^{(i)} \geq 0 \quad \forall i \in \{1, \dots, n\}. \end{aligned}$$



Maximize safety margin γ .

Margin violations are allowed,
but are minimized.



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

EXAMPLE 5: SUPPORT VECTOR MACHINES

Formulation 3 (dual): Dualizing the primal formulation

$$\max_{\alpha \in \mathbb{R}^n} \quad \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y^{(i)} y^{(j)} \langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \rangle$$

$$\text{s.t.} \quad 0 \leq \alpha_i \leq C \quad \forall i \in \{1, \dots, n\}, \quad \sum_{i=1}^n \alpha_i y^{(i)} = 0$$

Matrix notation:

$$\begin{aligned} \max_{\alpha \in \mathbb{R}^n} \quad & \alpha^\top \mathbf{1} - \frac{1}{2} \alpha^\top \text{diag}(\mathbf{y}) \mathbf{X}^\top \mathbf{X} \text{diag}(\mathbf{y}) \alpha \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C \quad \forall i \in \{1, \dots, n\}, \quad \alpha^\top \mathbf{y} = 0 \end{aligned}$$

Kernelization: Replace dot product between \mathbf{x} 's with $\mathbf{K}_{ij} = k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$, where $k(\cdot, \cdot)$ is a positive definite kernel function ($\Rightarrow \mathbf{K}$ positive semi-definite).

$$\begin{aligned} \max_{\alpha \in \mathbb{R}^n} \quad & \alpha^\top \mathbf{1} - \frac{1}{2} \alpha^\top \text{diag}(\mathbf{y}) \mathbf{K} \text{diag}(\mathbf{y}) \alpha \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C \quad \forall i \in \{1, \dots, n\}, \quad \alpha^\top \mathbf{y} = 0 \end{aligned}$$

This is QP with a single affine equality constraint and n box constraints.

