Constrained Optimization

Optimization methods in machine learning

Aleksandr Beznosikov

Innopolis University

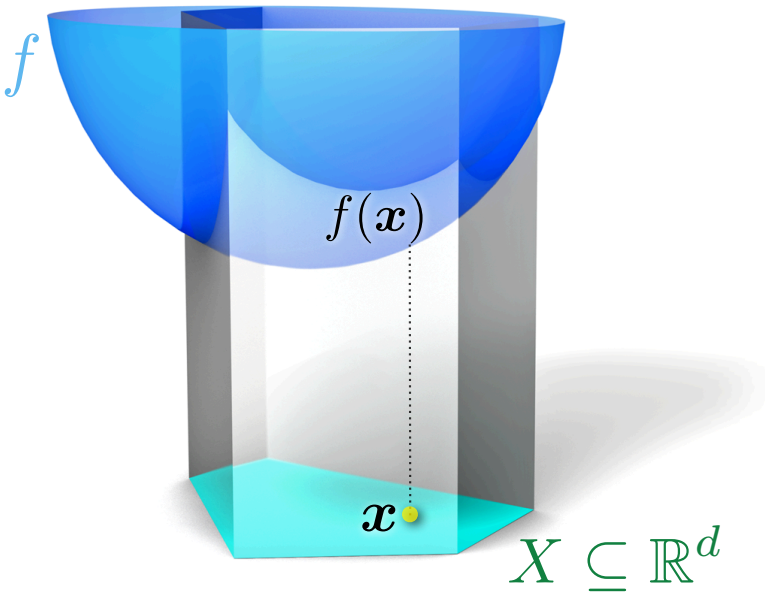
5 September 2023

INNOPOLIS UNIVERSITY

Constrained Optimization

Constrained Optimization Problem

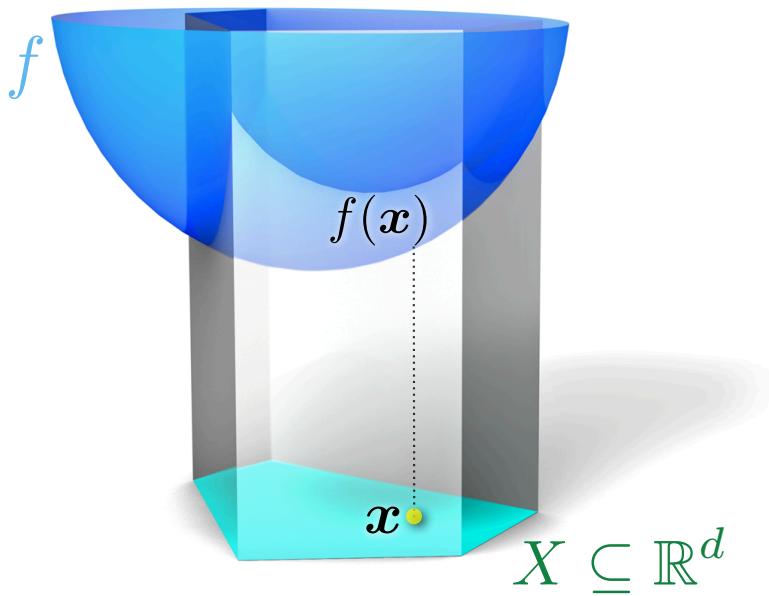
$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & x \in X \end{aligned}$$



Constrained Optimization

Constrained Optimization Problem

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & x \in X \end{aligned}$$

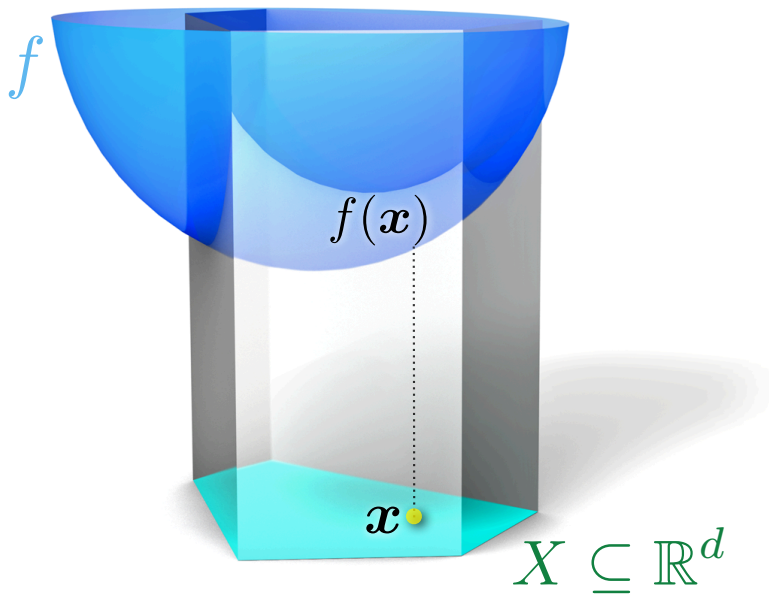


- Before $X = \mathbb{R}^d$ (unconstrained optimization)

Constrained Optimization

Constrained Optimization Problem

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & x \in X \end{aligned}$$

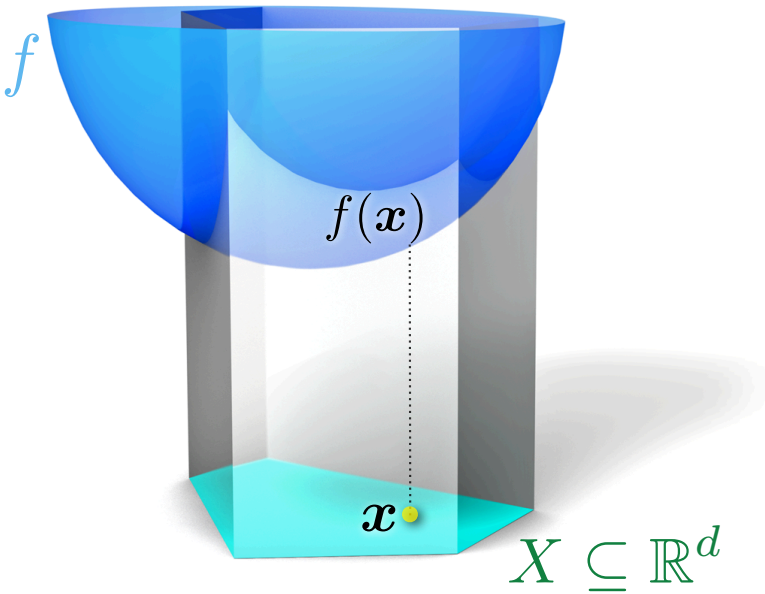


- Before $X = \mathbb{R}^d$ (unconstrained optimization)
- This lecture: $X \subsetneq \mathbb{R}^d$ (convex set)

Constrained Optimization

Constrained Optimization Problem

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & x \in X \end{aligned}$$



- Before $X = \mathbb{R}^d$ (unconstrained optimization)
- This lecture: $X \subsetneq \mathbb{R}^d$ (convex set)

Question: Why do we need constrains?

Example: Master's Admission

- Innopolis university is admitting top students to its MSc program, in a competitive application process.
- Applicants are submitting various documents (GPA, current salary in the IT company,...)
- Admission committee would like to compute a (rough) forecast of the applicant's performance in the program, based on the submitted documents.
- Data on the actual performance of students admitted in the past is available.
- In the following (made-up toy) example: consider GPA and current salary only...
- ...as predictors for Perfomance (final score of success obtained in the program).

Example: Master's Admission

- $0.0 \leq$ GPA ≤ 4.0 (from F to A)
- $0 \leq$ Salary
- $1.0 \leq$ Perfomance ≤ 6.0 (final score of secess)
- Historical data:

GPA	Salary	Perfomance
3.52	100	3.92
3.66	109	4.34
3.76	113	4.80
3.74	100	4.67
3.93	100	5.52
3.88	115	5.44
3.77	115	5.04
3.66	107	4.73
3.87	106	5.03
3.84	107	5.06

Master's Admission: Linear model

Hypothesis:

$$\text{Perfomance} \approx w_0 + w_1 \cdot \text{GPA} + w_2 \cdot \text{Salary}$$

for weights w_0, w_1, w_2 to be learned.
Approach: Find w_0, w_1, w_2 by minimizing least squares error over the historical data.
Question: what we need to do with data before solving something?

Master's Admission: Linear model

Hypothesis:

$$\text{Perfomance} \approx w_0 + w_1 \cdot \text{GPA} + w_2 \cdot \text{Salary}$$

for weights w_0, w_1, w_2 to be learned.
Approach: Find w_0, w_1, w_2 by minimizing least squares error over the historical data.
Question: what we need to do with data before solving something?

- Relevant GPA scores span a range of 0.5 (take only top students).
- Relevant Salary scores span a range of 20 (from 100 to 120 - others go to jobs, not to master).
- $\Rightarrow$ normalize first so that w_1, w_2 can be compared

General setting

n inputs $x_1, \ldots, x_n$, $x_i \in \mathbb{R}^d$ for all i
d input variables $1, 2, \ldots, d$

- 10 (GPA, Salary) pairs, two input variables

n outputs $y_1, \ldots, y_n \in \mathbb{R}$

- 10 Perfomance scores

(x_i, y_i): an observation

- $((3.93, 100), 5.52)$, observation (of a student doing very well)

With weights $w_0, w = (w_1, \ldots, w_d) \in \mathbb{R}^d$, we plan to minimize the least squares objective

$$f(w_0, w) = \sum_{i=1}^{n} (w_0 + w^T x_i - y_i)^2.$$

General setting: centering

Want to assume that

$$\frac{1}{n}\sum_{i=1}^{n} x_i = 0, \quad \frac{1}{n}\sum_{i=1}^{n} y_i = 0.$$

Can be achieved by

- subtracting the mean $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ from every input
- subtracting the mean $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ from every output.

Question: after centering what we can assume?

General setting: centering

Want to assume that

$$\frac{1}{n}\sum_{i=1}^{n} x_i = 0, \quad \frac{1}{n}\sum_{i=1}^{n} y_i = 0.$$

Can be achieved by

- subtracting the mean $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ from every input
- subtracting the mean $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ from every output.

Question: after centering what we can assume?

After centering: $w_0^* = 0, w^*$ is unaffected

$\Rightarrow$ From now on consider function

$$f(w) = \sum_{i=1}^{n} (w^T x_i - y_i)^2.$$

General setting: normalization

Want to assume that for all j, the n input values $x_{1j}, \dots x_{nj}$ are on the same scale:

$$\frac{1}{n}\sum_{i=1}^{n} x_{ij}^2 = 1, \quad j = 1, \dots, d.$$

Can be achieved by

- multiplying x_{ij} by $s(j) = \sqrt{n/\sum_{i=1}^{n} x_{ij}^2}$ for all i, j
- in w*, this just multiplies w_j^* by $1/s(j)$

Master's Admission: Centered and normalized data

x_{i1} (GPA)	x_{i2} (Salary)	y_i (Perfomance)
-2.04	-1.28	-0.94
-0.88	0.32	-0.52
-0.05	1.03	-0.05
-0.16	-1.28	-0.18
1.42	-1.28	0.67
1.02	1.39	0.59
0.06	1.39	0.19
-0.88	-0.04	-0.12
0.89	-0.21	0.17
0.62	-0.04	0.21

Least-squares objective:

$$f(w_1, w_2) = \sum_{i=1}^{10} (w_1 x_{i1} + w_2 x_{i2} - y_i)^2.$$

Master's Admission: Results

Optimal solution: $\mathrm{w}^* = (w_1^*, w_2^*) \approx (0.43, 0.097)$

Master's Admission: Results

Optimal solution: $\mathrm{w}^* = (w_1^*, w_2^*) \approx (0.43, 0.097)$

Under hypothesis (linear model), we expect $y_i \approx y_i^* = 0.43x_{i1} + 0.097x_{i2}$

x_{i1}	x_{i2}	y_i	y_i^*
-2.04	-1.28	-0.94	-1.00
-0.88	0.32	-0.52	-0.35
-0.05	1.03	-0.05	0.08
-0.16	-1.28	-0.18	-0.19
1.42	-1.28	0.67	0.49
1.02	1.39	0.59	0.57
0.06	1.39	0.19	0.16
-0.88	-0.04	-0.12	-0.38
0.62	-0.04	0.21	0.26

Questiob: what we can say about results? Salary has only very small influence ($w_2^* = 0.097$)

Predicting Perfomance in the future

Problems:

- least squares solution is optimized for the training data, not for the future ("overfitting")
- "unimportant" variables should have weight 0, but they typically don't

Predicting Perfomance in the future

Problems:

- least squares solution is optimized for the training data, not for the future ("overfitting")
- "unimportant" variables should have weight 0, but they typically don't

Subset selection heuristics: drop variables with seemingly "small" contribution

Predicting Perfomance in the future

Problems:

- least squares solution is optimized for the training data, not for the future ("overfitting")
- "unimportant" variables should have weight 0, but they typically don't

Subset selection heuristics: drop variables with seemingly "small" contribution (various methods to decide what "small" means, and how many to drop)

Best subset selection: solve least squares subject to an additional constraint that there are at most k nonzero weights. **Easy of not?**

Predicting Perfomance in the future

Problems:

- least squares solution is optimized for the training data, not for the future ("overfitting")
- "unimportant" variables should have weight 0, but they typically don't

Subset selection heuristics: drop variables with seemingly "small" contribution (various methods to decide what "small" means, and how many to drop)

Best subset selection: solve least squares subject to an additional constraint that there are at most k nonzero weights. **Easy of not?** Non-convex or NP-hard – various k might have to be tried.

Question: if we have 100 features, how many different subsets (of features) can we have?

Predicting Perfomance in the future

Problems:

- least squares solution is optimized for the training data, not for the future ("overfitting")
- "unimportant" variables should have weight 0, but they typically don't

Subset selection heuristics: drop variables with seemingly "small" contribution (various methods to decide what "small" means, and how many to drop)

Best subset selection: solve least squares subject to an additional constraint that there are at most k nonzero weights. **Easy of not?** Non-convex or NP-hard – various k might have to be tried.

Question: if we have 100 features, how many different subsets (of features) can we have? $2^{100} \approx 1.26 \cdot 10^{30}$.

LASSO: popular approach with some favorable statistical properties

The LASSO: a constrained optimization problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{n} \|\mathrm{w}^\top \mathrm{x}_i - y_i\|^2 \\ \text{subject to} & \|\mathrm{w}\|_1 \leq R, \end{array} \tag{1}$$

where $R \in \mathbb{R}_+$ is some parameter.

The LASSO: a constrained optimization problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{n} \|\mathrm{w}^\top \mathrm{x}_i - y_i\|^2 \\ \text{subject to} & \|\mathrm{w}\|_1 \leq R, \end{array} \tag{1}$$

where $R \in \mathbb{R}_+$ is some parameter.
$\|\mathrm{w}\|_1 = \sum_{i=1}^{d} |w_j|$ is the 1-norm.

The LASSO: a constrained optimization problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{n} \|\mathrm{w}^\top \mathrm{x}_i - y_i\|^2 \\ \text{subject to} & \|\mathrm{w}\|_1 \leq R, \end{array} \tag{1}$$

where $R \in \mathbb{R}_+$ is some parameter.
$\|\mathrm{w}\|_1 = \sum_{i=1}^{d} |w_j|$ is the 1-norm.
In our case:
$R = 0.2 \Rightarrow \mathrm{w}^* = (w_1^*, w_2^*) = (0.2, 0)$:

The LASSO: a constrained optimization problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{n} \|\mathrm{w}^\top \mathrm{x}_i - y_i\|^2 \\ \text{subject to} & \|\mathrm{w}\|_1 \leq R, \end{array} \tag{1}$$

where $R \in \mathbb{R}_+$ is some parameter.
$\|\mathrm{w}\|_1 = \sum_{i=1}^{d} |w_j|$ is the 1-norm.
In our case:
$R = 0.2 \Rightarrow \mathrm{w}^* = (w_1^*, w_2^*) = (0.2, 0)$: Salary is gone!

The LASSO: a constrained optimization problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{n} \|\mathrm{w}^\top \mathrm{x}_i - y_i\|^2 \\ \text{subject to} & \|\mathrm{w}\|_1 \leq R, \end{array} \tag{1}$$

where $R \in \mathbb{R}_+$ is some parameter.
$\|\mathrm{w}\|_1 = \sum_{i=1}^{d} |w_j|$ is the 1-norm.
In our case:
$R = 0.2 \Rightarrow \mathrm{w}^* = (w_1^*, w_2^*) = (0.2, 0)$: Salary is gone!
$R = 0.3 \Rightarrow \mathrm{w}^* = (w_1^*, w_2^*) = (0.3, 0)$

The LASSO: a constrained optimization problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{n} \|\mathrm{w}^\top \mathrm{x}_i - y_i\|^2 \\ \text{subject to} & \|\mathrm{w}\|_1 \leq R, \end{array} \tag{1}$$

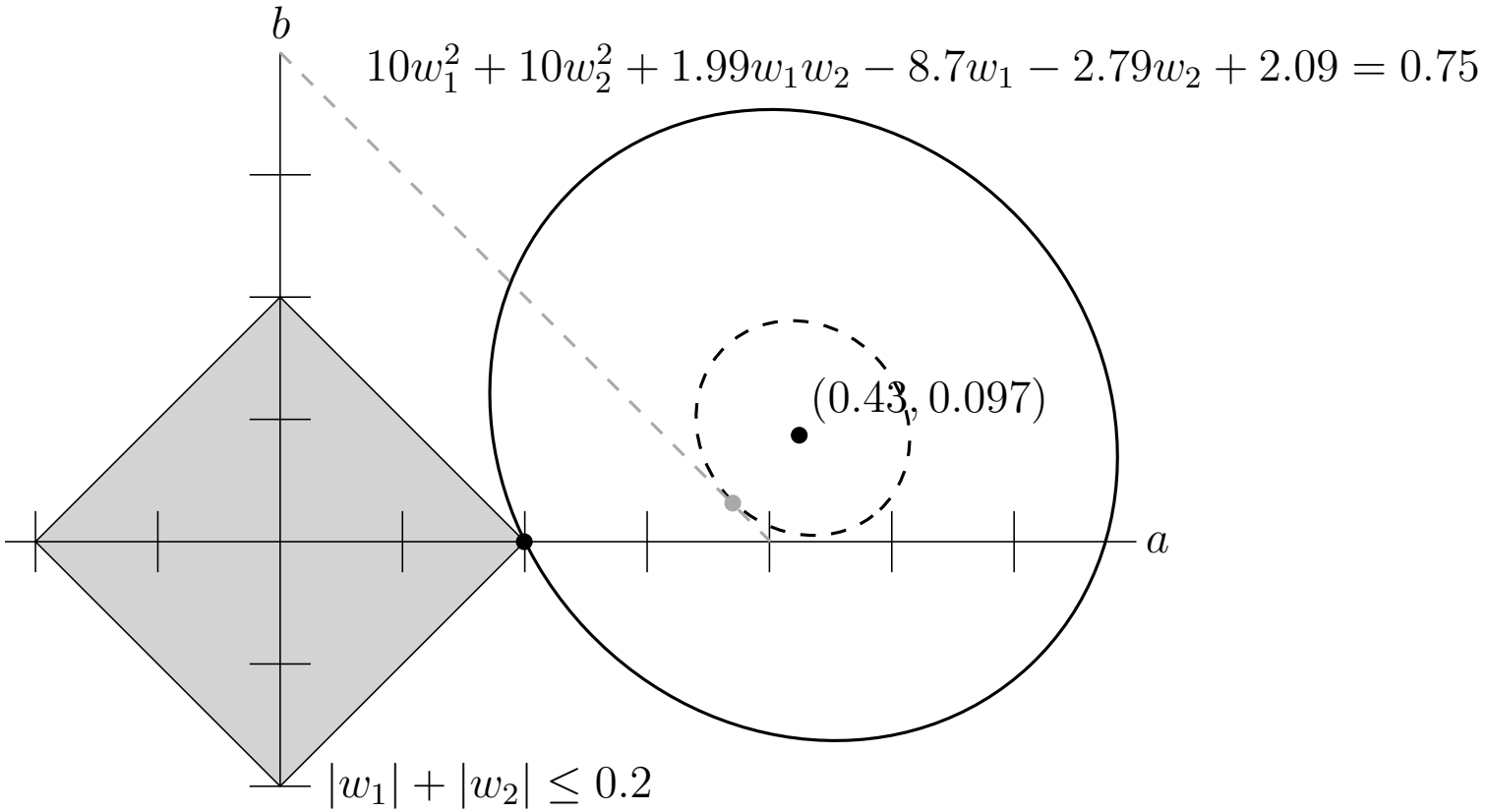
where $R \in \mathbb{R}_+$ is some parameter.
$\|\mathrm{w}\|_1 = \sum_{i=1}^{d} |w_j|$ is the 1-norm.
In our case:
$R = 0.2 \Rightarrow \mathrm{w}^* = (w_1^*, w_2^*) = (0.2, 0)$: Salary is gone!
$R = 0.3 \Rightarrow \mathrm{w}^* = (w_1^*, w_2^*) = (0.3, 0)$
$R = 0.4 \Rightarrow \mathrm{w}^* = (w_1^*, w_2^*) = (0.36, 0.036)$

The LASSO: a constrained optimization problem

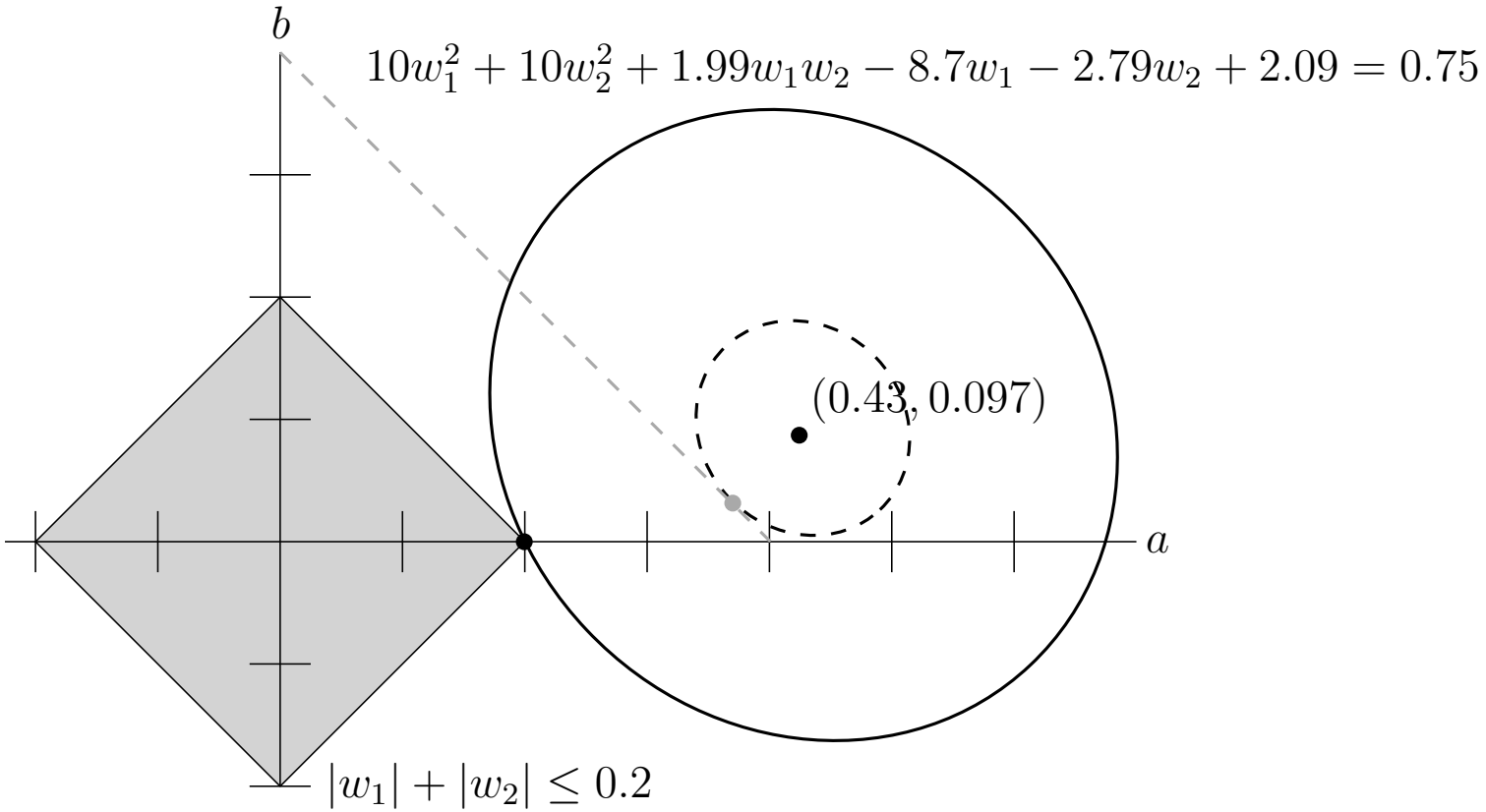
$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{n} \|\mathrm{w}^\top \mathrm{x}_i - y_i\|^2 \\ \text{subject to} & \|\mathrm{w}\|_1 \leq R, \end{array} \tag{1}$$

where $R \in \mathbb{R}_+$ is some parameter.
$\|\mathrm{w}\|_1 = \sum_{i=1}^{d} |w_j|$ is the 1-norm.
In our case:
$R = 0.2 \Rightarrow \mathrm{w}^* = (w_1^*, w_2^*) = (0.2, 0)$: Salary is gone!
$R = 0.3 \Rightarrow \mathrm{w}^* = (w_1^*, w_2^*) = (0.3, 0)$
$R = 0.4 \Rightarrow \mathrm{w}^* = (w_1^*, w_2^*) = (0.36, 0.036)$
$R \geq 0.6 \Rightarrow \mathrm{w}^* = (w_1^*, w_2^*) = (0.43, 0.097)$

Geometry of the LASSO



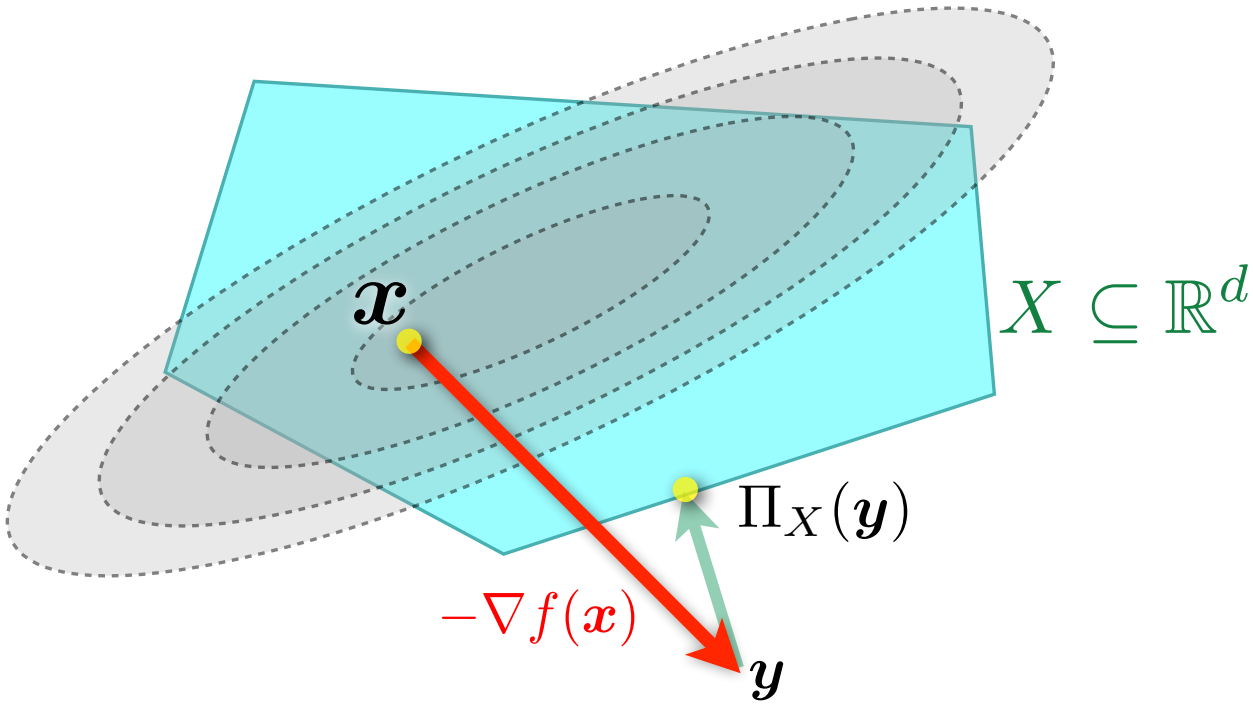
Geometry of the LASSO



Question: Can we somehow modify gradient method to work with constraints?

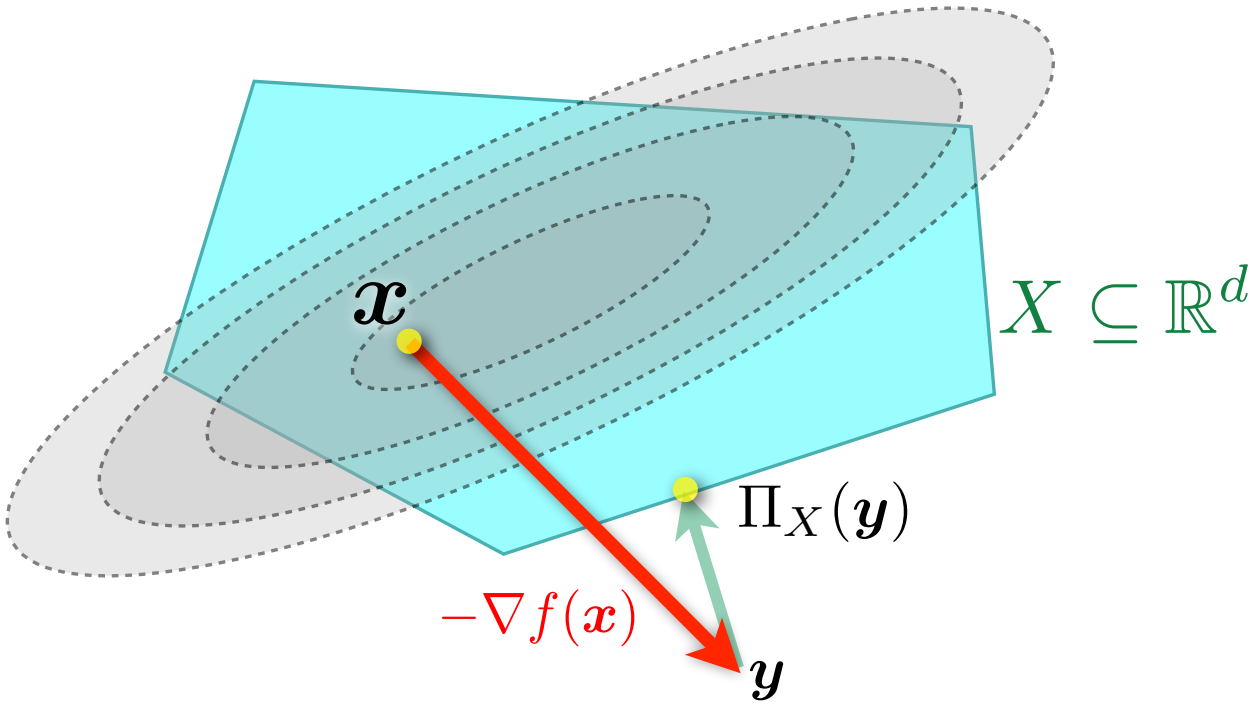
Projected Gradient Descent

Idea: project onto X after every step: $\Pi_X(y) := \operatorname{argmin}_{x \in X} \|x - y\|^2$



Projected Gradient Descent

Idea: project onto X after every step: $\Pi_X(y) := \operatorname{argmin}_{x \in X} \|x - y\|^2$



Projected gradient descent: $x_{k+1} := \Pi_X\left[x_k - \gamma \nabla f(x_k)\right]$

The Algorithm

Projected gradient descent:

The Algorithm

Projected gradient descent: choose $x_0 \in \mathbb{R}^d$.

The Algorithm

Projected gradient descent: choose $x_0 \in \mathbb{R}^d$.

$$y_{k+1} := x_k - \gamma \nabla f(x_k),$$

The Algorithm

Projected gradient descent: choose $x_0 \in \mathbb{R}^d$.

$$\begin{aligned} y_{k+1} &:= x_k - \gamma \nabla f(x_k), \\ x_{k+1} &:= \Pi_X(y_{k+1}) := \operatorname*{argmin}_{x \in X} \|x - y_{k+1}\|^2 \end{aligned}$$

The Algorithm

Projected gradient descent: choose $x_0 \in \mathbb{R}^d$.

$$\begin{aligned} y_{k+1} &:= x_k - \gamma \nabla f(x_k), \\ x_{k+1} &:= \Pi_X(y_{k+1}) := \operatorname*{argmin}_{x \in X} \|x - y_{k+1}\|^2 \end{aligned}$$

for times $k = 0, 1, \ldots,$ and stepsize $\gamma \geq 0$.

Property of Projection

Fact

Let $X \subseteq \mathbb{R}^d$ be closed and convex, $x \in X, y \in \mathbb{R}^d$. Then

$$\|x - \Pi_X(y)\| \leq \|x - y\|.$$

Property of Projection: proof

- We define a line L_λ with points x and $\Pi_X(y)$:

$$L_\lambda = \{\Pi_X(y) + \lambda(x - \Pi_X(y)) \text{ for } \lambda \in \mathbb{R}\}$$

- Consider optimization problem:

$$\min \operatorname{dist}(y, L_\lambda)$$

Property of Projection: proof

- We define a line L_λ with points x and $\Pi_X(y)$:

$$L_\lambda = \{\Pi_X(y) + \lambda(x - \Pi_X(y)) \text{ for } \lambda \in \mathbb{R}\}$$

- Consider optimization problem:

$$\min \text{dist}(y, L_\lambda)$$

or

$$\min_{\lambda \in \mathbb{R}} \|y - \Pi_X(y) + \lambda (\Pi_X(y) - x)\|^2$$
$$= \|y - \Pi_X(y)\|^2 + \lambda^2 \|\Pi_X(y) - x\|^2 + 2\lambda \langle y - \Pi_X(y), \Pi_X(y) - x \rangle$$

- **Question:** what is optimal λ?

Property of Projection: proof

- We define a line L_λ with points x and $\Pi_X(y)$:

$$L_\lambda = \{\Pi_X(y) + \lambda(x - \Pi_X(y)) \text{ for } \lambda \in \mathbb{R}\}$$

- Consider optimization problem:

$$\min \text{dist}(y, L_\lambda)$$

or

$$\min_{\lambda \in \mathbb{R}} \|y - \Pi_X(y) + \lambda\left(\Pi_X(y) - x\right)\|^2$$

$$= \|y - \Pi_X(y)\|^2 + \lambda^2 \|\Pi_X(y) - x\|^2 + 2\lambda\langle y - \Pi_X(y), \Pi_X(y) - x\rangle$$

- **Question:** what is optimal λ? $\lambda_{opt} = -\frac{\langle y - \Pi_X(y), \Pi_X(y) - x\rangle}{\|\Pi_X(y) - x\|^2}$.

Property of Projection: proof

Cases for λ_{opt}:

- $\lambda_{opt} < 0$

Property of Projection: proof

Cases for λ_{opt}:

- $\lambda_{opt} < 0 \Rightarrow \ \langle y - \Pi_X(y), \Pi_X(y) - x \rangle < 0.$

Property of Projection: proof

Cases for λ_{opt}:

- $\lambda_{opt} < 0 \Rightarrow \ \langle y - \Pi_X(y), \Pi_X(y) - x \rangle < 0.$
- $\lambda_{opt} > 1$

Property of Projection: proof

Cases for λ_{opt}:

- $\lambda_{opt} < 0 \Rightarrow \ \langle y - \Pi_X(y), \Pi_X(y) - x \rangle < 0.$
- $\lambda_{opt} > 1 \Rightarrow \ \text{dist}(y, x) = \text{dist}(y, L_1) \leq \text{dist}(y, L_0) = \text{dist}(y, \Pi_X(y))$
  $\Rightarrow \ \Pi_X(y) = x$

Property of Projection: proof

Cases for λ_{opt}:

- $\lambda_{opt} < 0 \Rightarrow \ \langle y - \Pi_X(y), \Pi_X(y) - x \rangle < 0.$
- $\lambda_{opt} > 1 \Rightarrow \ \text{dist}(y, x) = \text{dist}(y, L_1) \leq \text{dist}(y, L_0) = \text{dist}(y, \Pi_X(y))$
  $\Rightarrow \ \Pi_X(y) = x$
- $\lambda_{opt} \in [0; 1]$

Property of Projection: proof

Cases for λ_{opt}:

- $\lambda_{opt} < 0 \Rightarrow \ \langle y - \Pi_X(y), \Pi_X(y) - x \rangle < 0.$
- $\lambda_{opt} > 1 \Rightarrow \ \text{dist}(y, x) = \text{dist}(y, L_1) \leq \text{dist}(y, L_0) = \text{dist}(y, \Pi_X(y))$
  $\Rightarrow \ \Pi_X(y) = x$
- $\lambda_{opt} \in [0; 1] \Rightarrow \ \text{dist}(y, L_{\lambda_{opt}}) \leq \text{dist}(y, L_0) = \text{dist}(y, \Pi_X(y))$
  $\Rightarrow \ \Pi_X(y) = x$

Finally, we get

$$\langle y - \Pi_X(y), \Pi_X(y) - x \rangle \leq 0$$

Property of Projection: proof

- From the previous slide

$$\langle y - \Pi_X(y), \Pi_X(y) - x \rangle \leq 0$$

- Then

$$\langle y - \Pi_X(y) - x + \Pi_X(x), \Pi_X(y) - x \rangle \leq 0$$

- Next

$$\|\Pi_X(y) - \Pi_X(x)\|^2 + \langle y - x, \Pi_X(y) - x \rangle \leq 0$$

- And

$$\|\Pi_X(y) - \Pi_X(x)\|^2 \leq \langle x - y, \Pi_X(y) - x \rangle \leq \|\Pi_X(y) - x\| \cdot \|x - y\|$$

- Finally,

$$\|\Pi_X(y) - x\| \leq \|x - y\|.$$

Proof of convergence

- We use the property to change (a bit) proofs of gradient descent convergence:

$$\|x^* - \Pi_X(y)\|^2 \leq \|x^* - y\|$$

- Changes in vanilla analysis:

$$\begin{aligned}\|x_{k+1} - x^*\|^2 =& \|\Pi_X(x_k - \gamma \nabla f(x_k)) - x^*\|^2 \\ \leq& \|x_k - x^* - \gamma \nabla f(x_k)\|^2\end{aligned}$$

- It means that results in Lipschitz convex case remains the same!
- It remains the same results also in smooth case.

Proof of convergence

- We use the property to change (a bit) proofs of gradient descent convergence:

$$\|x^* - \Pi_X(y)\|^2 \leq \|x^* - y\|$$

- Changes in vanilla analysis:

$$\begin{aligned}\|x_{k+1} - x^*\|^2 =& \|\Pi_X(x_k - \gamma \nabla f(x_k)) - x^*\|^2 \\ \leq& \|x_k - x^* - \gamma \nabla f(x_k)\|^2\end{aligned}$$

- It means that results in Lipschitz convex case remains the same!
- It remains the same results also in smooth case.
- But there are bad news...

Projecting is additional problem

- The projection has analytical solution for ℓ_2-ball with radius $R > 0$:

$$X = B_2(R) = \left\{ x \in \mathbb{R}^d \mid \|x\|_2^2 = \sum_{i=1}^{d} x_i^2 \leq R \right\}$$

$$\Pi_X(x) = \max\left\{1, \frac{R}{\|x\|_2}\right\} x$$

- For other quite simple sets we need to run additional optimization algorithms. For example, for $B_1(R)$ $\left(\left\{ x \in \mathbb{R}^d \mid \|x\|_1 = \sum_{i=1}^{d} |x_i| \leq R \right\}\right)$ the projection can be computed in time $\mathcal{O}(d \log d)$.

Projecting is additional problem

- The projection has analytical solution for ℓ_2-ball with radius $R > 0$:

$$X = B_2(R) = \left\{ x \in \mathbb{R}^d \mid \|x\|_2^2 = \sum_{i=1}^{d} x_i^2 \leq R \right\}$$

$$\Pi_X(x) = \max\left\{1, \frac{R}{\|x\|_2}\right\} x$$

- For other quite simple sets we need to run additional optimization algorithms. For example, for $B_1(R)$ $\left(\left\{ x \in \mathbb{R}^d \mid \|x\|_1 = \sum_{i=1}^{d} |x_i| \leq R \right\}\right)$ the projection can be computed in time $\mathcal{O}(d \log d)$.
- The next approach can help.

Linear minimization

Quadratic problem (projection) is hard... **Question:** what we can try instead?

Linear minimization

Quadratic problem (projection) is hard... **Question:** what we can try instead? Linear problems:

$$\min_{s \in X} \langle s, g \rangle$$

Question: Is it easy to solve?

- ℓ_1-ball: $B_1(1) = \left\{ x \in \mathbb{R}^d \mid \|x\|_1 = \sum_{i=1}^{d} |x_i| \leq 1 \right\}$

Linear minimization

Quadratic problem (projection) is hard... **Question:** what we can try instead? Linear problems:

$$\min_{s \in X} \langle s, g \rangle$$

Question: Is it easy to solve?

- ℓ_1-ball: $B_1(1) = \left\{ x \in \mathbb{R}^d \mid \|x\|_1 = \sum_{i=1}^{d} |x_i| \leq 1 \right\}$
- probability simplex: $\triangle = \left\{ x \in \mathbb{R}^d \mid x_i \geq 0, \ \sum_{i=1}^{d} x_i = 1 \right\}$

Linear minimization

Quadratic problem (projection) is hard... **Question:** what we can try instead? Linear problems:

$$\min_{s \in X} \langle s, g \rangle$$

Question: Is it easy to solve?

- ℓ_1-ball: $B_1(1) = \left\{ x \in \mathbb{R}^d \mid \|x\|_1 = \sum\limits_{i=1}^{d} |x_i| \leq 1 \right\}$
- probability simplex: $\triangle = \left\{ x \in \mathbb{R}^d \mid x_i \geq 0, \sum\limits_{i=1}^{d} x_i = 1 \right\}$
- ℓ_∞-ball: $B_\infty(1) = \left\{ x \in \mathbb{R}^d \mid \|x\|_\infty = \max_{i=1,\dots d} |x_i| \leq 1 \right\}$

Linear minimization

Quadratic problem (projection) is hard... **Question:** what we can try instead? Linear problems:

$$\min_{s \in X} \langle s, g \rangle$$

Question: Is it easy to solve?

- ℓ_1-ball: $B_1(1) = \left\{ x \in \mathbb{R}^d \mid \|x\|_1 = \sum\limits_{i=1}^{d} |x_i| \leq 1 \right\}$
- probability simplex: $\triangle = \left\{ x \in \mathbb{R}^d \mid x_i \geq 0, \sum\limits_{i=1}^{d} x_i = 1 \right\}$
- ℓ_∞-ball: $B_\infty(1) = \left\{ x \in \mathbb{R}^d \mid \|x\|_\infty = \max_{i=1,\ldots d} |x_i| \leq 1 \right\}$

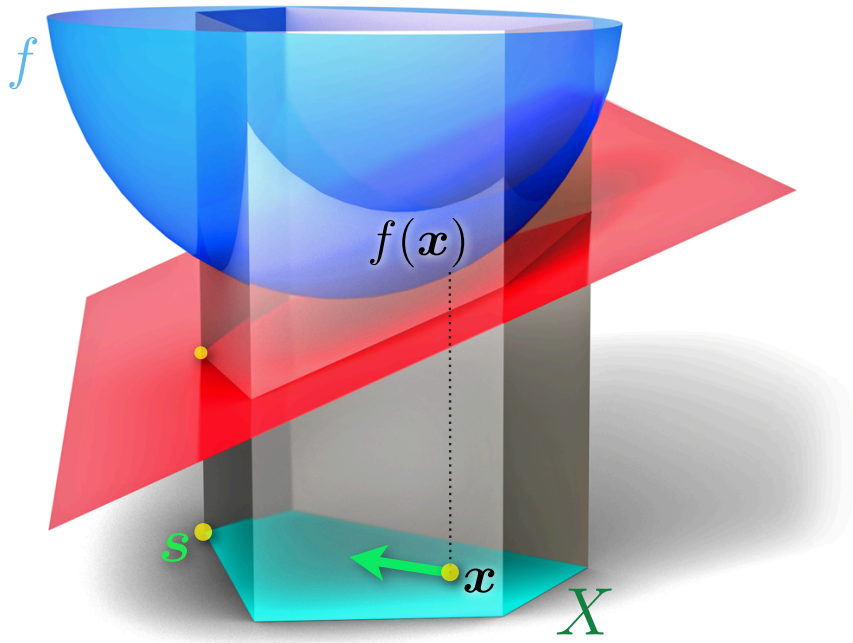
Examples	Solution of LM
ℓ_1-ball	$-\text{sign}(g_i)e_i$ with $\text{argmax}_i \lvert g_i \rvert$
Simplex	e_i with $\text{argmin}_i g_i$
ℓ_∞-ball	$-\sum_{i=1}^{d} \text{sign}(g_i)e_i$

Frank-Wolfe Algorithm

Frank-Wolfe Algorithm:

$$\begin{aligned} s &:= \mathrm{LMO}(\nabla f(x_k)), \\ x_{k+1} &:= (1-\gamma)x_k + \gamma s, \end{aligned}$$

for timesteps $k = 0, 1, \ldots,$ and stepsize $\gamma := \frac{2}{k+2}$.



Linear Minimization Oracle:

$$\mathrm{LMO}(g) := \operatorname*{argmin}_{s \in X} \langle s, g \rangle$$

Properties

- Aways feasible: $x_0, x_1, \ldots, x_k \in X$.
  x_{k+1} is on line segment $[s, x_k]$, for $\gamma \in [0, 1]$.
- Reduces non-linear to linear optimization
- Projection-free
- Sparse iterates (in terms of corners s used)

Invented and analyzed 1956 by Marguerite Frank and Philip Wolfe.
Check **Marguerite Frank - Honorary Discussion Panel @ NeurIPS 2013 workshop**
`https://www.youtube.com/watch?v=24e08AX9Eww`

Convergence in $\mathcal{O}(1/\varepsilon)$ steps

Theorem

Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and smooth with parameter L, and $x_0 \in X$. Then choosing any of the above stepsizes, the Frank-Wolfe algorithm yields

$$f(x_K) - f(x^*) \leq \frac{\max\{2L\,\mathrm{diam}(X)^2, f(x_0) - f(x^*)\}}{K+1}$$

Where $\mathrm{diam}(X) := \max_{x,y \in X} \|x - y\|$ is the diameter of X.

Proof of Convergence in $\mathcal{O}(1/\varepsilon)$ steps

- L -smoothness gives

$$\begin{aligned} f(x_{k+1}) =& f(x_k + \gamma_k(s - x_k)) \\ \leq& f(x_k) + \gamma_k \langle s - x_k, \nabla f(x_k) \rangle + \frac{\gamma_k^2}{2} L \|s - x_k\|^2 \\ \leq& f(x_k) + \gamma_k \langle s - x_k, \nabla f(x_k) \rangle + \frac{\gamma_k^2}{2} L \operatorname{diam}(X)^2. \end{aligned}$$

- By LMO property and convexity gives

$$\begin{aligned} f(x_{k+1}) - f(x^*) \leq& f(x_k) - f(x^*) + \gamma_k \langle x^* - x_k, \nabla f(x_k) \rangle + \frac{\gamma_k^2}{2} L \operatorname{diam}(X)^2 \\ =& f(x_k) - f(x^*) - \gamma_k (f(x_k) - f(x^*)) + \frac{\gamma_k^2}{2} L \operatorname{diam}(X)^2 \\ =& (1 - \gamma_k)(f(x_k) - f(x^*)) + \gamma_k^2 C. \end{aligned}$$

Here $C = \frac{L \operatorname{diam}(X)^2}{2}$.

The induction step

We will now use induction over k to prove our claimed bound, i.e.

$$f(x_k) - f(x^*) \leq \frac{\max\{4C; f(x_0) - f(x^*)\}}{k+2} \quad k = 0, 1, \ldots$$

The induction step

We will now use induction over k to prove our claimed bound, i.e.

$$f(x_k) - f(x^*) \leq \frac{\max\{4C; f(x_0) - f(x^*)\}}{k+2} \quad k = 0, 1, \ldots$$

The base-case $k = 0$ follows automatically.
Now considering $k \geq 1$,

$$\begin{aligned} f(x_{k+1}) - f(x^*) \leq & \quad (1 - \gamma_k)(f(x_k) - f(x^*)) + {\gamma_t}^2 C \\ = & \quad \left(1 - \frac{2}{k+2}\right)(f(x_k) - f(x^*)) + \left(\frac{2}{k+2}\right)^2 C \\ \leq & \quad \left(1 - \frac{2}{k+2}\right) \frac{\max\{4C; f(x_0) - f(x^*)\}}{k+2} + \left(\frac{2}{k+2}\right)^2 C \ , \end{aligned}$$

where in the last inequality we have used the induction hypothesis of induction.

The induction step

Simply rearranging the terms gives

$$
\begin{aligned}
f(x_{k+1}) - f(x^*) &\leq \frac{\max\{4C; f(x_0) - f(x^*)\}}{k+2}\left(1 - \tfrac{2}{k+2} + \tfrac{1}{k+2}\right) \\
&= \frac{\max\{4C; f(x_0) - f(x^*)\}}{k+2}\left(1 - \tfrac{1}{k+2}\right) \\
&= \frac{\max\{4C; f(x_0) - f(x^*)\}}{k+2}\,\frac{k+2-1}{k+2} \\
&\leq \frac{\max\{4C; f(x_0) - f(x^*)\}}{k+2}\,\frac{k+2}{k+3} \\
&= \frac{\max\{4C; f(x_0) - f(x^*)\}}{k+3},
\end{aligned}
$$

which is our claimed bound for $k \geq 1$.