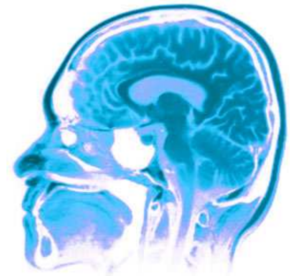
CPSC540



Constrained Optimization

Nando de Freitas
2011

KPM Book Sections: *30.8*

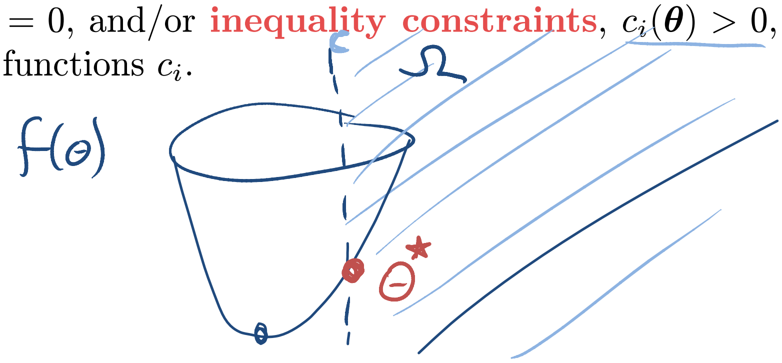
Constrained optimization

- Consider the following **constrained optimization problem**

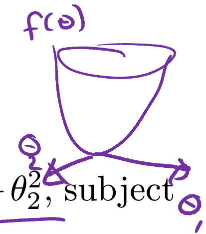
$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\Omega} f(\boldsymbol{\theta})$$

where Ω is some **feasible set**. If the parameters are real-valued, we typically assume $\Omega \subseteq \mathbb{R}^D$, but it could be a more abstract space, such as the set of positive definite matrices.

- The feasible set is then often defined in terms of a set of **equality constraints**, $c_i(\boldsymbol{\theta}) = 0$, and/or **inequality constraints**, $c_i(\boldsymbol{\theta}) > 0$, for certain constraint functions c_i.

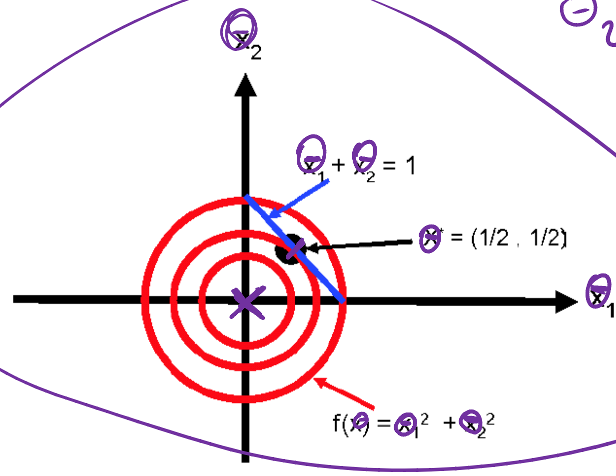


Constrained optimization



- Suppose that we have a single equality constraint $c(\boldsymbol{\theta}) = \mathbf{0}$.
- For example, we might have a quadratic objective, $f(\boldsymbol{\theta}) = \theta_1^2 + \theta_2^2$, subject to a linear equality constraint, $c(\boldsymbol{\theta}) = 1 - \theta_1 - \theta_2 = 0$.
- What we are trying to do is find the point $\boldsymbol{\theta}^*$ that lives on the line, but which is closest to the origin. It is geometrically obvious that the optimal solution is $\boldsymbol{\theta} = (0.5, 0.5)$.

$\theta_2 = 1 - \theta_1$

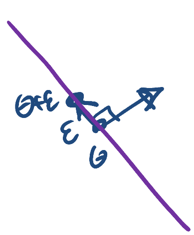


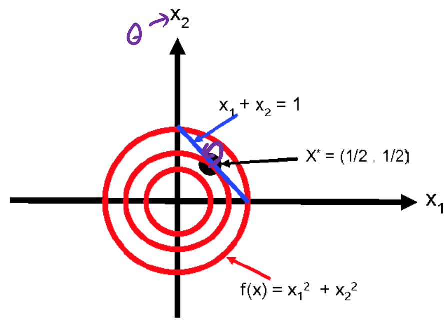
Constrained optimization

- The gradient of the constraint function $\nabla c(\boldsymbol{\theta})$ will be orthogonal to the constraint surface.
- To see why, consider a point $\boldsymbol{\theta}$ on the constraint surface, and another point nearby, $\boldsymbol{\theta} + \boldsymbol{\epsilon}$, that also lies on the surface. If we make a Taylor expansion around $\boldsymbol{\theta}$ we have

$$c(\boldsymbol{\theta} + \boldsymbol{\epsilon}) \approx c(\boldsymbol{\theta}) + \boldsymbol{\epsilon}^T \nabla c(\boldsymbol{\theta})$$

Since both $\boldsymbol{\theta}$ and $\boldsymbol{\theta} + \boldsymbol{\epsilon}$ are on the constraint surface, we must have $c(\boldsymbol{\theta}) = c(\boldsymbol{\theta} + \boldsymbol{\epsilon})$ and hence $\boldsymbol{\epsilon}^T \nabla c(\boldsymbol{\theta}) \approx 0$. Since $\boldsymbol{\epsilon}$ is parallel to the constraint surface, we see that the vector ∇c is normal to the surface.





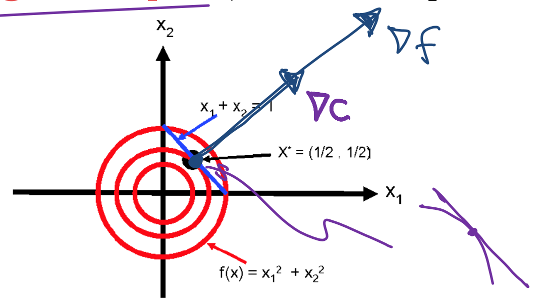
Constrained optimization

- We seek a point $\boldsymbol{\theta}^*$ on the constraint surface such that $f(\boldsymbol{\theta})$ is minimized. Such a point must have the property that $\nabla f(\boldsymbol{\theta})$ is also orthogonal to the constraint surface, as otherwise we could decrease $f(\boldsymbol{\theta})$ by moving a short distance along the constraint surface.

- Since both $\nabla c(\boldsymbol{\theta})$ and $\nabla f(\boldsymbol{\theta})$ are orthogonal to the constraint surface at $\boldsymbol{\theta}^*$, they must be parallel (or anti-parallel) to each other. Hence there must exist a constant $\lambda^* \neq 0$ such that

$$\boxed{\nabla f(\boldsymbol{\theta}^*) = \lambda^* \nabla c(\boldsymbol{\theta}^*)}$$

λ^* is called a **Lagrange multiplier**, and can be positive or negative, but not zero.



Lagrangian

- We can now convert our constrained optimization problem into an unconstrained one by defining a new function called the **Lagrangian**:

$$\boxed{L(\boldsymbol{\theta}, \lambda) := f(\boldsymbol{\theta}) - \lambda c(\boldsymbol{\theta})}$$

We now have $D+1$ equations in $D+1$ unknowns, which we can solve for $\boldsymbol{\theta}^*$ and λ. Why? Since we are only interested in $\boldsymbol{\theta}^*$, we can "throw away" the value λ; hence it is sometimes called an **undetermined multiplier**.

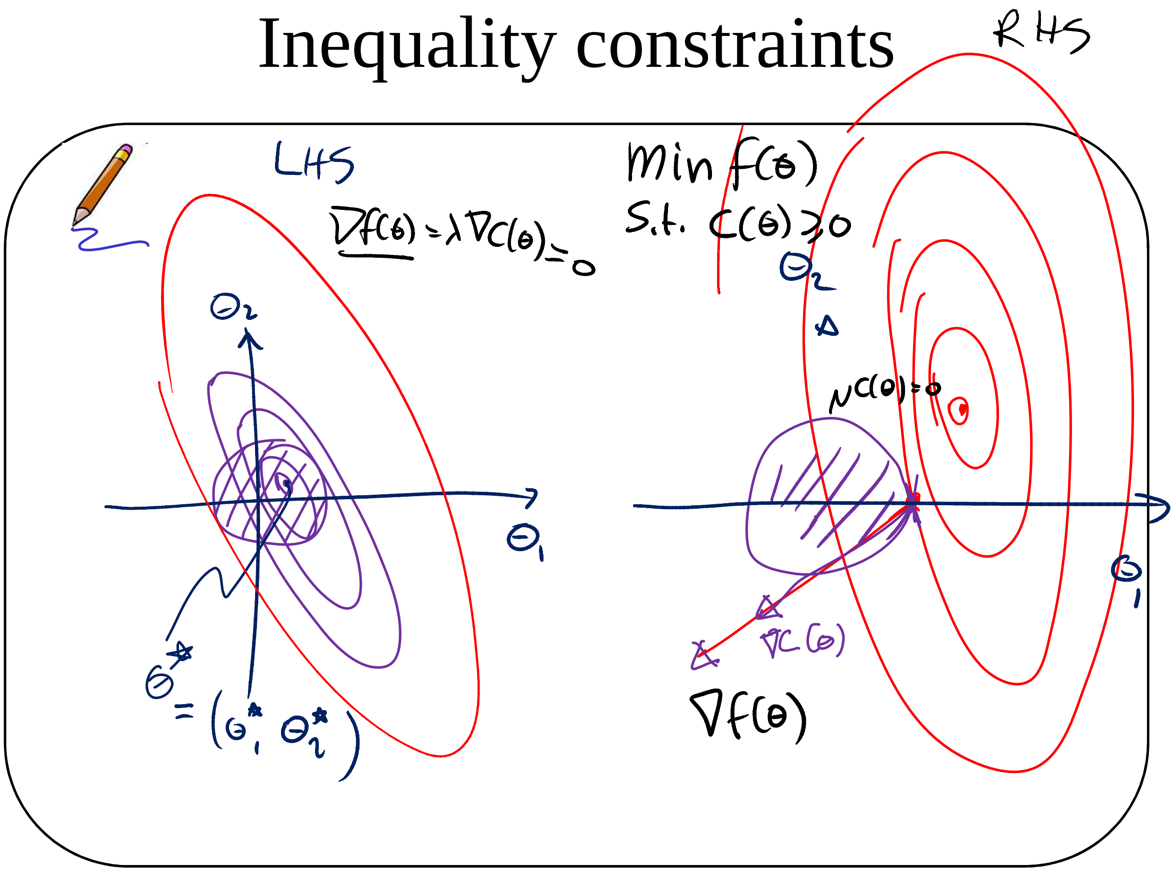
$$\nabla_\theta L(\theta, \lambda) = \nabla_\theta f(\theta) - \lambda \nabla_\theta c(\theta) = 0$$

$$\nabla_\theta f(\theta^*) = \lambda^* \nabla_\theta c(\theta^*)$$

$$\nabla_\lambda L(\theta, \lambda) = -c(\theta) = 0$$

$$c(\theta) = 0$$

Inequality constraints



Inequality constraints

- Now consider the case where we have a single **inequality constraint** $c(\boldsymbol{\theta}) \geq 0$.
- If the solution lies in the region where $c(\boldsymbol{\theta}) > 0$, the constraint is **inactive**, so we have the usual stationarity condition $\nabla f(\boldsymbol{\theta}^*) = \mathbf{0}$. Our equations still hold, provided we set $\lambda^* = 0$. LHS
- If the solution lies on the boundary where $c(\boldsymbol{\theta}) = 0$, the constraint is **active**, so $\nabla c(\boldsymbol{\theta})$ and $\nabla f(\boldsymbol{\theta})$ must be parallel, as for the equality constraint case. RHS
- However, this time we require that $\lambda^* > 0$, so the gradients point in the *same* direction. Since the gradients of c and f point in the same direction, we will follow c to its minimum, where $c(\boldsymbol{\theta}^*) = 0$.
- We can summarize these two cases by writing $\boxed{\lambda^* c(\boldsymbol{\theta}^*) = 0}$: either $\lambda^* = 0$ or $c(\boldsymbol{\theta}^*) = \mathbf{0}$ (or both). This is called the **complementarity condition**.

Inequality constraints

- Putting it all together, the problem of minimizing $f(\boldsymbol{\theta})$ subject to $c(\boldsymbol{\theta}) \geq 0$ can be obtained by optimizing the Lagrangian subject to the following constraints:

$$\begin{aligned} c(\boldsymbol{\theta}) &\geq 0 \\ \lambda^* &\geq 0 \\ \lambda^* c(\boldsymbol{\theta}^*) &= 0 \end{aligned}$$

Many constraints

- In general, if we have multiple equality constraints, $c_i(\boldsymbol{\theta}) = 0$ for $i \in \mathcal{E}$, and multiple inequality constraints, $c_i(\boldsymbol{\theta}) \geq 0$ for $i \in \mathcal{I}$, we can define the feasible set as

$$\Omega = \{\boldsymbol{\theta} \in \mathbb{R}^D : \underbrace{c_i(\boldsymbol{\theta}) = 0, i \in \mathcal{E}}_{\text{eq.}}, \underbrace{c_i(\boldsymbol{\theta}) \geq 0, i \in \mathcal{I}}_{\text{ineq.}}\}$$

and the Lagrangian as

$$L(\boldsymbol{\theta}, \boldsymbol{\lambda}) = f(\boldsymbol{\theta}) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(\boldsymbol{\theta})$$

- The **active set** is defined as the contraints that are active at a point:

$$\mathcal{A}(\boldsymbol{\theta}) = \mathcal{E} \cup \{i \in \mathcal{I} : c_i(\boldsymbol{\theta}) = 0\}$$

Karush-Kuhn-Tucker conditions

- We have the following necessary first-order conditions for being at a local minimum:

$$\boxed{\begin{aligned} \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \boldsymbol{\lambda}) &= \mathbf{0} \\ c_i(\boldsymbol{\theta}^*) &= 0 \; \forall i \in \mathcal{E} \;\checkmark \\ c_i(\boldsymbol{\theta}^*) &\geq 0 \; \forall i \in \mathcal{I} \;\checkmark \\ \lambda_i^* &\geq 0 \; \forall i \in \mathcal{I} \;\checkmark \\ \lambda_i^* c_i(\boldsymbol{\theta}^*) &= 0 \; \forall i \in \mathcal{I} \cup \mathcal{E} \;\checkmark \end{aligned}}$$

- These are called the **Karush-Kuhn-Tucker** or **KKT** conditions.
- If f and the c_i are convex, the KKT conditions are sufficient for a minimum as well.

Example

$1 - \theta_1 - \theta_2 = 0$

- Maximize $f(\boldsymbol{\theta}) = 1 - \theta_1^2 - \theta_2^2$ subject to the constraint that $\theta_1 + \theta_2 = 1$.

(i) $L(\theta_1, \theta_2, \lambda) = f(\theta_1, \theta_2) - \lambda\, C(\theta_1, \theta_2)$

$= 1 - \theta_1^2 - \theta_2^2 - \lambda \left[1 - \theta_1 - \theta_2 \right]$

(ii) $\nabla_{\theta_1} L(\theta_1, \theta_2, \lambda) = 0 \Rightarrow$

$\nabla_{\theta_2} L(\theta_1, \theta_2, \lambda) = 0 \Rightarrow$

$\nabla_{\lambda} L(\theta_1, \theta_2, \lambda) = 0 \Rightarrow$

Example

$\theta^* = \left(\frac{1}{2}, \frac{1}{2}\right)$

Quadratic programs

- A generic **quadratic program** or **QP** has the form

$$\min_{\boldsymbol{\theta}} \frac{1}{2}\boldsymbol{\theta}^T \mathbf{H}\boldsymbol{\theta} + \mathbf{d}^T\boldsymbol{\theta} \;\text{ s.t. }\; \mathbf{A}\boldsymbol{\theta} \le \mathbf{b},\; \mathbf{A}_{eq}\boldsymbol{\theta} = \mathbf{b}_{eq},\; \mathbf{b}_l \le \boldsymbol{\theta} \le \mathbf{b}_u$$

The constraints $\mathbf{b}_l \le \boldsymbol{\theta} \le \mathbf{b}_u$ are known as **box constraints**, and can always be rewritten as linear inequality constraints.

- QPs arise in several areas of machine learning, including **support vector machines** and **lasso** .

- Assume we want to minimize:

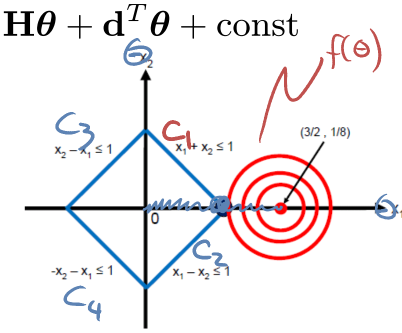
$$f(\boldsymbol{\theta}) = (\theta_1 - \frac{3}{2})^2 + (\theta_2 - \frac{1}{8})^2 = \frac{1}{2}\boldsymbol{\theta}^T\mathbf{H}\boldsymbol{\theta} + \mathbf{d}^T\boldsymbol{\theta} + \text{const}$$

where $\mathbf{H} = 2\mathbf{I}$ and $\mathbf{d} = -(3, 1/4)$, subject to

L1 Norm:

$\|\theta\|_1 = |\theta_1| + |\theta_2|$

$$|\theta_1| + |\theta_2| \leq 1$$



We can rewrite the constraints as

$1\cdot\theta_1 + \theta_2 \geq 0$

$$\theta_1 + \theta_2 \leq 1, \quad \theta_1 - \theta_2 \leq 1, \quad -\theta_1 + \theta_2 \leq 1, \quad -\theta_1 - \theta_2 \leq 1$$

$1 - \theta_1 - \theta_2 \geq 0$ C_1 C_2 C_3 C_4

which we can write more compactly as

$$\boxed{\mathbf{b} - \mathbf{A}\boldsymbol{\theta} \geq \mathbf{0}}$$

where $\mathbf{b} = \mathbf{1}$ and

$$\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 1 \\ -1 & -1 \end{pmatrix}$$

$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$

Quadratic programs

- The Lagrangian is

$$L(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{\theta}^T\mathbf{H}\boldsymbol{\theta} + \mathbf{d}^T\boldsymbol{\theta} + \boldsymbol{\lambda}^T(\mathbf{A}\boldsymbol{\theta} - \mathbf{b})$$

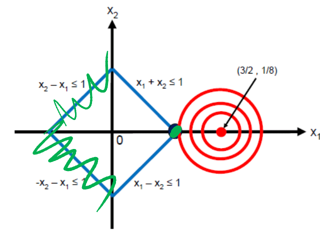
and the KKT conditions are

$$\begin{aligned} \mathbf{H}\boldsymbol{\theta} + \mathbf{d} + \mathbf{A}^T\boldsymbol{\lambda} &= \mathbf{0} \\ \mathbf{b} - \mathbf{A}\boldsymbol{\theta} &\geq \mathbf{0} \end{aligned}$$

If we treat the inequality as an equality, we can write

$$\begin{pmatrix} \mathbf{H} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\theta} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} -\mathbf{d} \\ \mathbf{b} \end{pmatrix}$$

Quadratic programs



- The KKT matrix on the LHS is singular. Note constraints c_3 and c_4 (corresponding to the two left faces of the diamond) are inactive, so $c_3(\boldsymbol{\theta}^*) > 0$ and $c_4(\boldsymbol{\theta}^*) > 0$ and hence, by complementarity, $\lambda_3^* = \lambda_4^* = 0$. We can therefore remove these inactive constraints to get the following:

$$\begin{pmatrix} 2 & 0 & 1 & 1 \\ 0 & 2 & 1 & -1 \\ 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \\ \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 1/4 \\ 1 \\ 1 \end{pmatrix}$$

We see that the solution is

$$\boldsymbol{\theta}^* = (1, 0)^T, \boldsymbol{\lambda}^* = (0.875, 0.125, 0, 0)^T$$

Notice that the optimal value of $\boldsymbol{\theta}$ occurs at one of the vertices of the L1 **simplex**. Consequently the solution vector is **sparse**.

Lasso for feature selection

$$\min_{\theta} \|Y - X\theta\|_2^2 + \lambda \|\theta\|_1$$

$$\min_{\theta : \|\theta\|_1 = t} \|Y - X\theta\|_2^2$$

$$t = g(\lambda)$$

Lasso for feature selection

Duality

- **Duality theory** provides an alternative way to express optimization problems that can often lead to faster algorithms, as well as new insights into a problem. It also relaxes some of the differentiation conditions.

Duality

- Consider the **primal problem**

$$\min_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) \text{ s.t. } \mathbf{c}(\boldsymbol{\theta}) \geq \mathbf{0}$$

The Lagrangian is

$$L(\boldsymbol{\theta}, \boldsymbol{\lambda}) = f(\boldsymbol{\theta}) - \lambda^T \mathbf{c}(\boldsymbol{\theta})$$

$f(\theta) = \lambda c(\theta) + L$

We define the **dual** objective function as

$$g(\boldsymbol{\lambda}) = \min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \min_{\boldsymbol{\theta}} \underbrace{f(\boldsymbol{\theta})}_{B} - \lambda^T \underbrace{\mathbf{c}(\boldsymbol{\theta})}_{A} = -f^*(\boldsymbol{\lambda})$$

where f^* is the **Fenchel conjugate** of f.

- We see that the dual objective g is a concave function, since it is a minimum over an affine function of $\boldsymbol{\lambda}$. The corresponding **dual problem** is

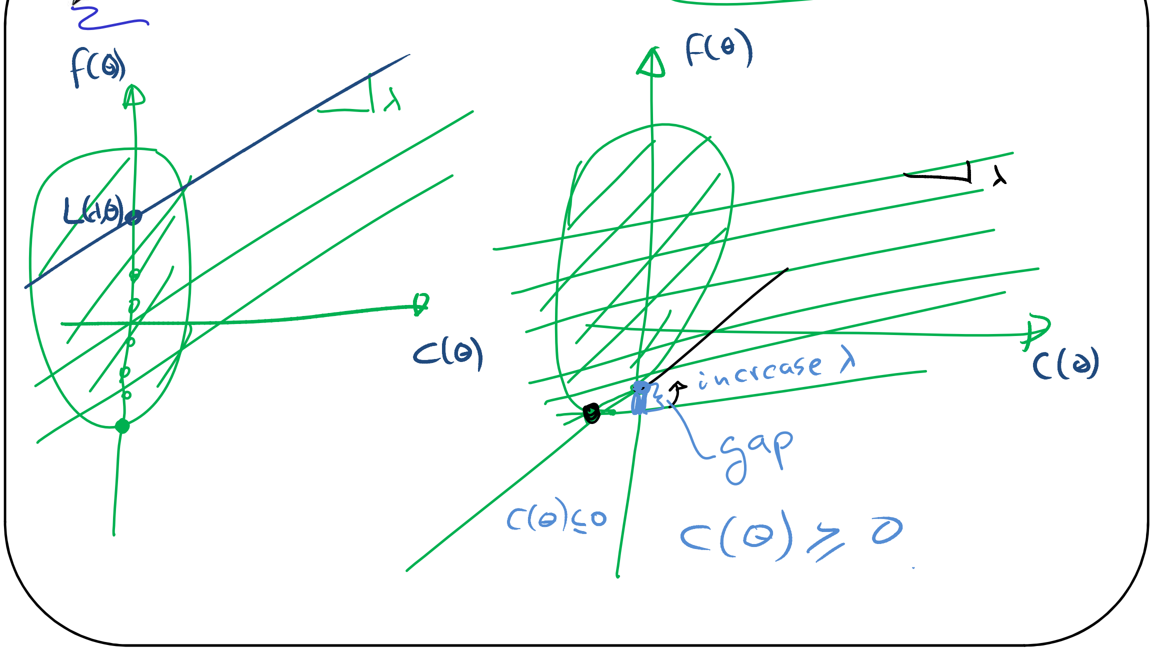
$$\max_{\boldsymbol{\lambda}} g(\boldsymbol{\lambda}) \text{ s.t. } \boldsymbol{\lambda} \geq \mathbf{0}$$

Duality

$\lambda c(\theta) = 0$

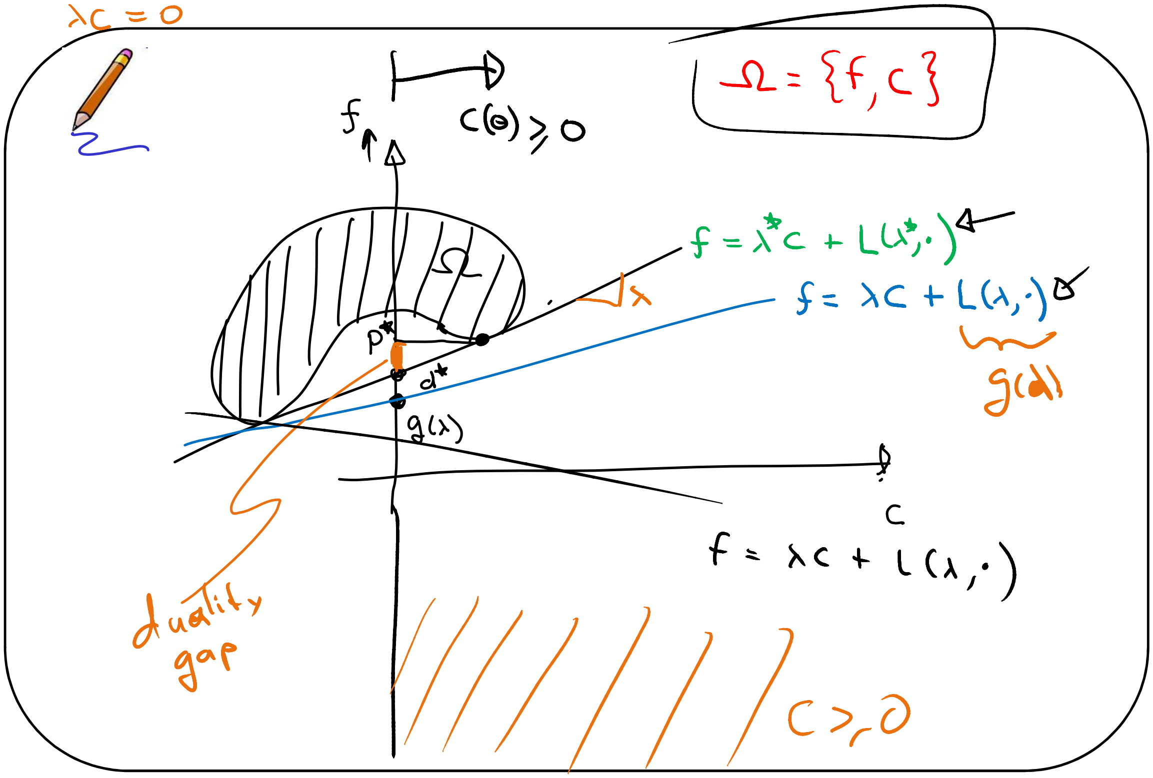
$L(\theta, \lambda) = f(\theta) - \lambda c(\theta)$

$f(\theta) = \lambda C(\theta) + L(\lambda, \theta)$



Duality

$c \geq 0$
$\lambda \geq 0$
$\lambda c = 0$



Duality

- Solving the dual has several advantages:
  1. It is always convex, even if the primal is not;
  2. The number of variables in the dual is equal to the number of constraints in the primal, which is often less than the number of variables in the primal
  3. I might enable us to deal with non-differentiable problems.

Duality

$L(\lambda, \theta) = f(\theta) - \lambda c(\theta) +$
$c(\theta) = 0$
$\lambda \geq 0$

- The key question is, do the two methods give the same results? Let $\boxed{p^* = f(\boldsymbol{\theta}^*)}$ be the optimal primal value, and $\boxed{d^* = g(\boldsymbol{\lambda}^*)}$ be the optimal dual value. We have the following two important theorems:

$p^* = L(\lambda, \theta^*) = f(\theta^*) + 0$

  – **Weak duality**: $\boxed{d^* \leq p^*}$ This always holds. To see this, note that for $\lambda \geq 0$, since $\mathbf{c}(\boldsymbol{\theta}) \geq \mathbf{0}$,

$$f(\boldsymbol{\theta}) \geq L(\boldsymbol{\theta}, \boldsymbol{\lambda}) \geq \min_{\boldsymbol{\theta}'} L(\boldsymbol{\theta}', \boldsymbol{\lambda}) = g(\boldsymbol{\lambda})$$

  – **Strong duality**: $d^* = p^*$. This only holds for convex problems. The reason is that a convex function can be precisely represented either in primal or dual form.

Put another way, for any real function $L(\boldsymbol{\theta}, \boldsymbol{\lambda})$, weak duality says we always have

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\lambda}} L(\boldsymbol{\theta}, \boldsymbol{\lambda}) \geq \max_{\boldsymbol{\lambda}} \min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \boldsymbol{\lambda})$$

If strong duality holds, the two terms are equal, so the **duality gap**, $p^* - d^*$, is zero. In this case, $L(\boldsymbol{\theta}^*, \boldsymbol{\lambda}^*)$ is a **saddle point**.

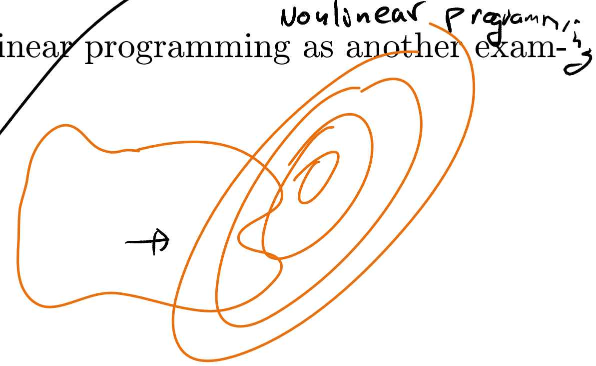
Further reading

- Nocedal & Wright
- Stephen Boyd
- Bertsekas
Nonlinear Programming

- Please read the book section about linear programming as another example.

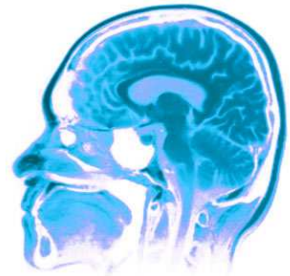
- Read on the algorithms

  1. Interior point methods
  2. Active set methods
  3. Projected gradient



Mark
Scmidt

Next class



Bayesian Learning

Nando de Freitas
2011
KPM Book Sections: *4*