Georgia Tech

Machine Learning CS 4641

Optimization

Nimisha Roy

Lecturer, SCI

Director of Online Undergraduate Initiatives

College of Computing

Georgia Tech

Slides adapted from Max Roozbahani, Rod Borela

Georgia Tech

Georgia Tech

Cross Entropy

Cross Entropy: The expected number of bits when a wrong distribution Q is assumed while the data actually follows a distribution P

$$H(p,q) = -\sum_{x \in \mathcal{X}} p(x)\ \log q(x) = H(P) + KL[P][Q]$$

This is because:

$$H(p,q) = \mathrm{E}_p[l_i] = \mathrm{E}_p\left[\log \frac{1}{q(x_i)}\right]$$

$$H(p,q) = \sum_{x_i} p(x_i)\ \log \frac{1}{q(x_i)}$$

$$H(p,q) = -\sum_{x} p(x)\ \log q(x).$$

Labeling target values: Label encoding (ordinal) and One-hot encoding

- **Input Matrix (X):**

- **Target Labels (Y):**

- **Label Encoding (Ordinal):**
cat→1,fish→2,dog→3

- **One-Hot Encoding:**
Cat → [1 0 0], fish → [0 1 0], dog → [0 0 1]

Label encoding (One-hot encoding) Loss Computation

Minimize Objective Function:

Maximize Objective Function:

Interpretation:

- Minimum squared Error works but isn't ideal for classification.
- **Minimize negative sum of dot product is also good but Cross-Entropy / NLL** is better.

Georgia Tech

Why Cross Entropy is better than dot product

$$H(p, q) = -\sum_{x \in \mathcal{X}} p(x) \log q(x)$$

Logarithmic penalty encourages big update leading to faster convergence

Kullback-Leibler Divergence

Another useful information theoretic quantity measures the difference between two distributions.

$$\mathbf{KL}[P(S)\|Q(S)] = \sum_s P(s)\log\frac{P(s)}{Q(s)}$$

$$= \underbrace{\sum_s P(s)\log\frac{1}{Q(s)}}_{\text{cross entropy}} - \mathbf{H}[P] = H(P,Q) - H(P)$$

KL Divergence is a **KIND OF** distance measurement

Excess cost in bits paid by encoding according to Q instead of P.

$$-\mathbf{KL}[P\|Q] = \sum_s P(s)\log\frac{Q(s)}{P(s)}$$

log function is concave or convex?

$$\sum_s P(s)\log\frac{Q(s)}{P(s)} \leq \log\sum_s P(s)\frac{Q(s)}{P(s)}$$

By Jensen Inequality

$$= \log\sum_s Q(s) = \log 1 = 0$$

So $\mathbf{KL}[P\|Q] \geq 0$. Equality iff $P = Q$

When $P = Q$, $KL[P||Q] = 0$

Concave Function: Jensen Inequality

$$\mathbb{E}[f(X)] \leq f(\mathbb{E}[X])$$

Strictly concave functions have global maximum

Convex Function: Jensen Inequality

$$\mathbb{E}[f(X)] \geq f(\mathbb{E}[X])$$

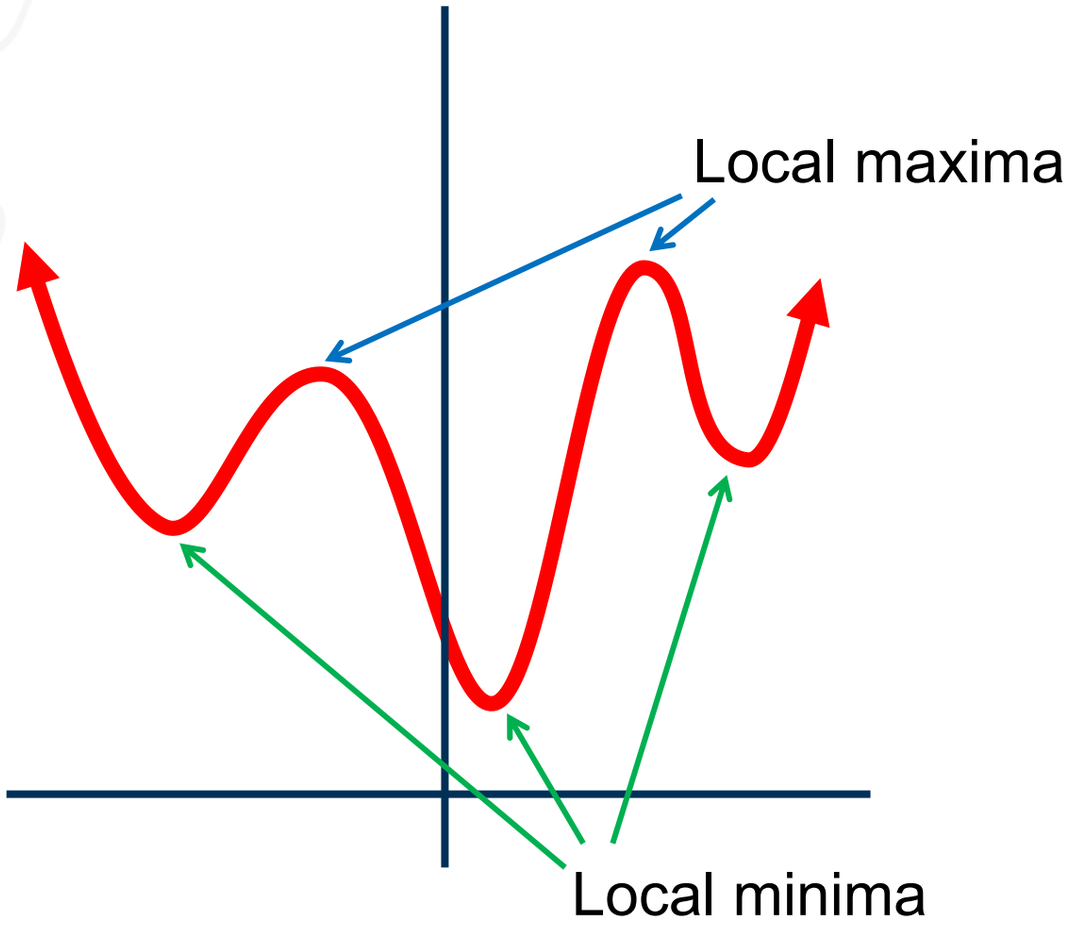
Strictly convex functions have global minimum

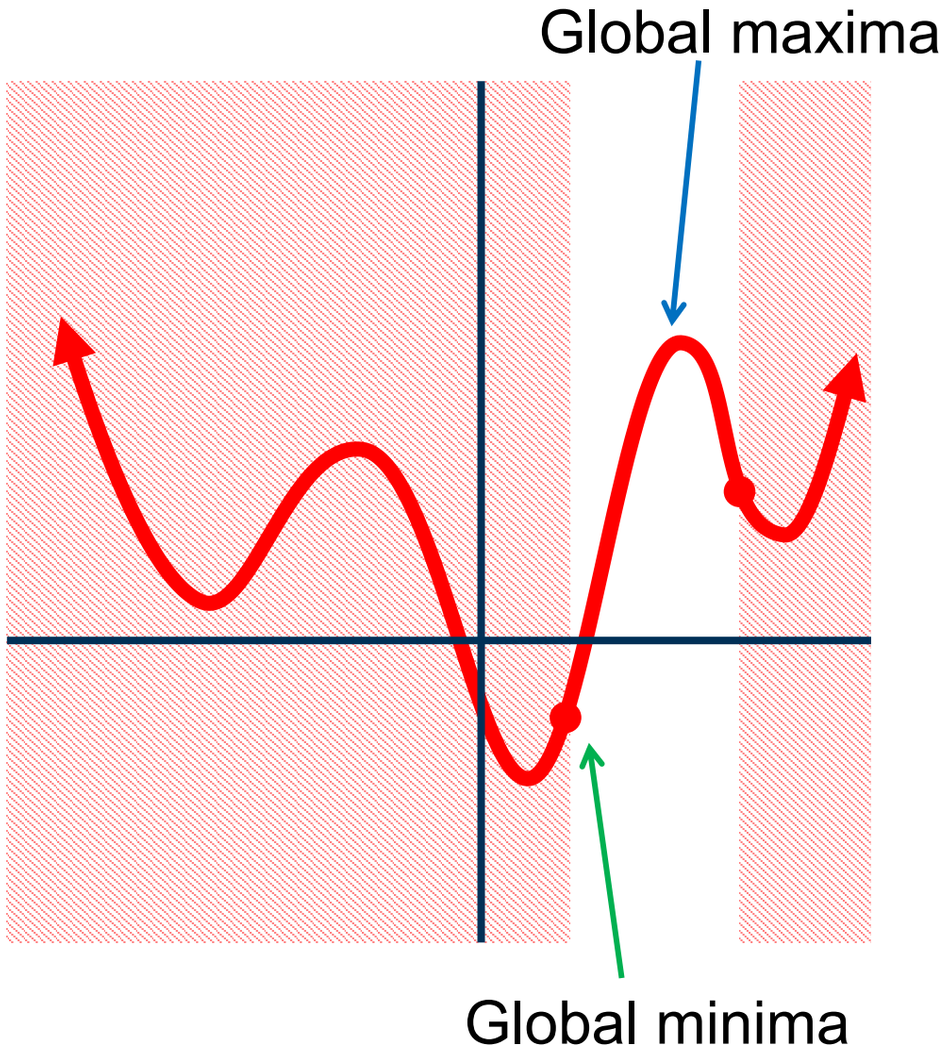
Jensen Inequality

KL Divergence is always non negative

$$-KL[P][Q] = \sum_{x} p(x) \log \frac{q(x)}{p(x)}$$

Unconstrained and constrained optimization





Basic unconstrained optimization problem

- Objective $f(\mathbf{x})$ the quantity we are trying to optimize (maximize or minimize)

- The variables $x_1, x_2, \ldots, x_m$ which can be represented in vector form as $\mathbf{x}$

(Note: the subscript m indicates a feature NOT a datapoint in our dataset)

- Finding the stationary points where the function is no longer increasing or decreasing

Equality constrained optimization

- An equality constrained optimization problem is expressed as:

$$\min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y})$$

$$\text{subject to } \; g(\mathbf{x}, \mathbf{y}) = 0$$

Inequality constrained optimization

- An inequality constrained optimization problem is expressed as:

$$\min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y})$$

$$\text{subject to } \; g(\mathbf{x}, \mathbf{y}) \leq 0$$

Linear, Quadratic, and Nonlinear Programming

Linear Programming (LP)
Objective: Linear function of x.
Constraints: Linear.
Example: $\min 3\,x_1 + 5x_2 \;\; s.t. \;\; x_1 + 2x_2 \leq 6$

Quadratic Programming (QP)
Objective: Quadratic
Constraints: Linear.
Example: $\min x^2 + y^2 + 3x + 5y \quad s.t. \quad x + y \leq 4$

Nonlinear Programming (NLP)
Objective: Nonlinear.
Constraints: Can be nonlinear.
Example: $\min x^2 + y^2 + 3x + 5y \quad s.t. \quad x^2 + y \leq 4$

Recap

- Cross Entropy
- KL Divergence
- Loss Functions
- Jensen Inequality
- Unconstrained and Constrained Optimization
- Linear, Quadratic and Non-Linear Programming

Equality constrained optimization problem

There is a notion of a Lagrangian expression where we combine both the objective function and our constraints

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \lambda g(\mathbf{x})$$

$$s.t.\ \lambda \neq 0$$

Equality constrained optimization problem

- With multiple equality constraints the Lagrangian is expressed as follows:

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \sum_{i=1}^{k} \lambda_i g_i(\mathbf{x})$$

$$s.t. \quad \lambda_i \neq 0$$

Equality constrained optimization problem

- To solve this optimization problem, we only need to satisfy the stationarity condition, which states:

$$\nabla L(\mathbf{x}, \lambda) = 0$$

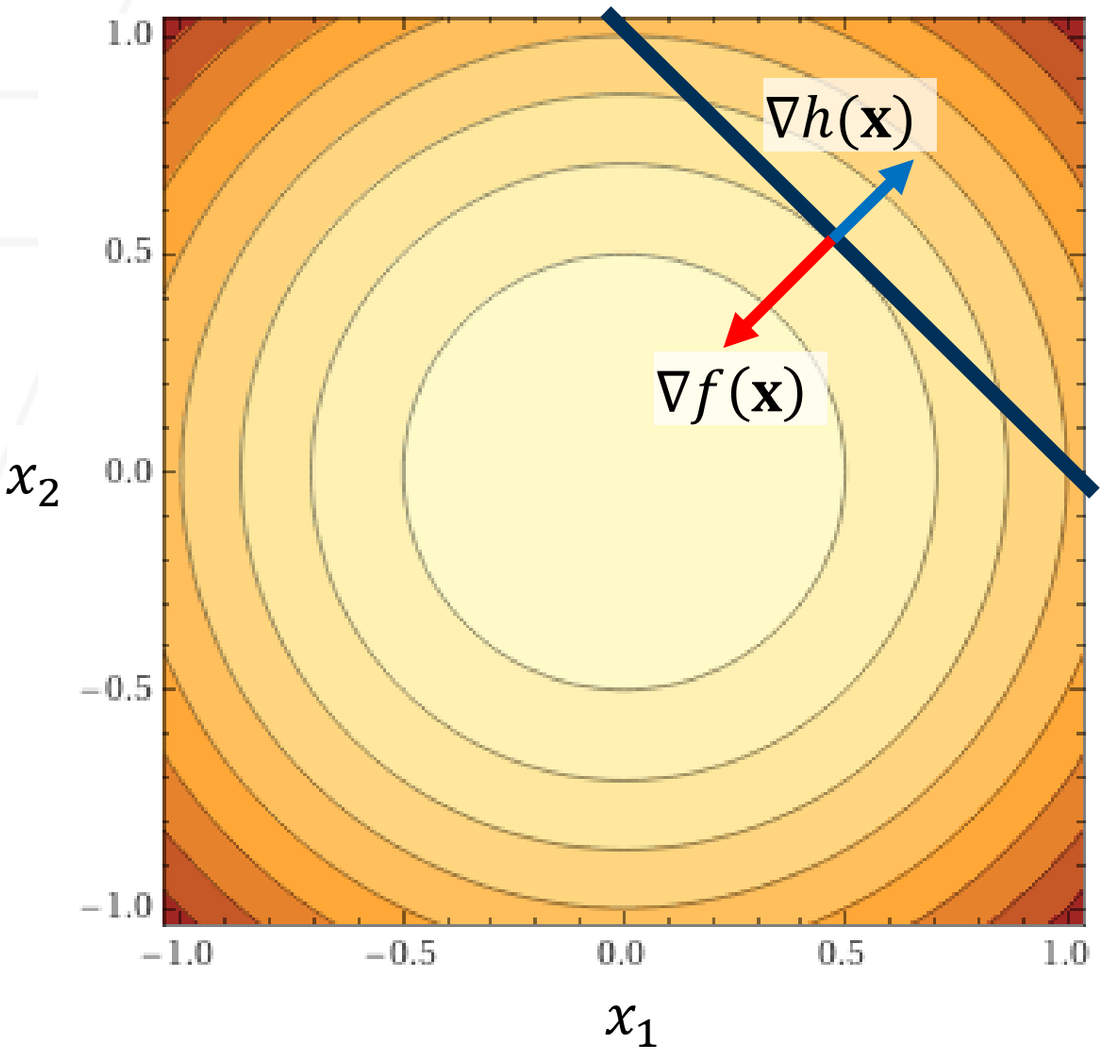
- We also need to satisfy

$$\lambda_i \neq 0$$

- This appeals to the intuition that at the optimal point, the gradient of the equality constraint is proportional to the gradient of the objective function

Equality constrained optimization: *Example 1*

Equality constrained optimization: *Example 2*



$$\max_{\mathbf{x}} \quad 1 - x_1^2 - x_2^2$$

$$s.t. \quad x_1 + x_2 = 1$$

Georgia Tech

Equality constrained optimization: *Example*

$$\max_{\mathbf{x}} \quad 1 - x_1^2 - x_2^2$$

$$s.t. \quad x_1 + x_2 = 1$$

1. Write the Lagrangian for this problem

$$L(\mathbf{x}, \lambda) = 1 - x_1^2 + x_2^2 + \lambda(x_1 + x_2 - 1)$$

2. Calculate the gradient with respect to $\mathbf{x}$

$$\frac{\partial L}{\partial x_1} = -2x_1 + \lambda = 0 \rightarrow \lambda = 2x_1$$

$$\frac{\partial L}{\partial x_2} = -2x_2 + \lambda = 0 \rightarrow \lambda = 2x_2$$

Equality constrained optimization: *Example*

3. Solve for $\mathbf{x}^*$ and λ

Using our previous two derivatives and the equality constraint, we obtain:

$$x_1 + x_2 = 1 \rightarrow x_1 + x_1 = 1 \rightarrow x_1^* = x_2^* = \frac{1}{2}$$

From $\frac{\partial L}{\partial x_1} = 0$ we find the value of λ

$$\lambda = 2x_1 \rightarrow \lambda = 1$$

Inequality constrained optimization problem

- Constraints that limit how small or big variables can be. These are inequality constraints noted as $g_j(\mathbf{x})$
- An optimization problem is usually expressed as:

$$\min_{\mathbf{x}} f(\mathbf{x})$$

$$\text{subject to} \quad g_j(\mathbf{x}) \leq 0, j = 1, 2, \ldots, m$$

.

Inequality constrained optimization problem

- The optimization problem can be rewritten as

$$\text{. } \min_{\mathbf{x}} \quad f(\mathbf{x}) + \sum_{j=1}^{m} \lambda_j g_j(\mathbf{x})$$

$$\text{subject to} \qquad \boldsymbol{\lambda} \geq 0$$

- We need to satisfy the Karush-Kuhn-Tucker (KKT) conditions on $\mathbf{x}$, $\boldsymbol{\lambda}$ which is **necessary and sufficient** for optimality

Karush-Kuhn-Tucker Conditions

- Stationarity condition:

$$\min L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_{j=1}^{m} \lambda_j g_j(\mathbf{x})$$

- Primal feasibility

$$g_j(\mathbf{x}) \leq 0, j = 1, 2, \ldots, m$$

- Dual Feasibility

$$\lambda_j \geq 0$$

- Complementary slackness

$$\lambda_j g_j(\mathbf{x}) = 0, j = 1, \ldots, m$$

If g(x) < 0 → **inactive** solution since λ =0
If g(x) = 0 → **active** solution since λ is at play and
> 0, which means constraint is active

Karush-Kuhn-Tucker Interpretations

- The complementary slackness condition applies only to inequality constraints
- It states that for a given inequality constraint…

$$\ldots\text{either } g_j(x) = 0 \quad \text{or} \quad \lambda_j = 0$$

- Whenever the dual variable $\lambda_j > 0$ and therefore $g_i(x) = 0$, we say that the **constraint is active** or that the **constraint is tight at x**
- The reverse $\lambda_j = 0$ and therefore $g_j(x) < 0$ means that the constraint is inactive, meaning that it is not effectively impacting the solution

Rule: For each inequality constraint, **either** it is active and has a positive multiplier, or it is inactive and its multiplier is zero.

Primal vs. Dual Form

Primal vs Dual Form

Primal Problem
Original optimization problem

Dual Problem
Derived problem as **a function of** λ only.

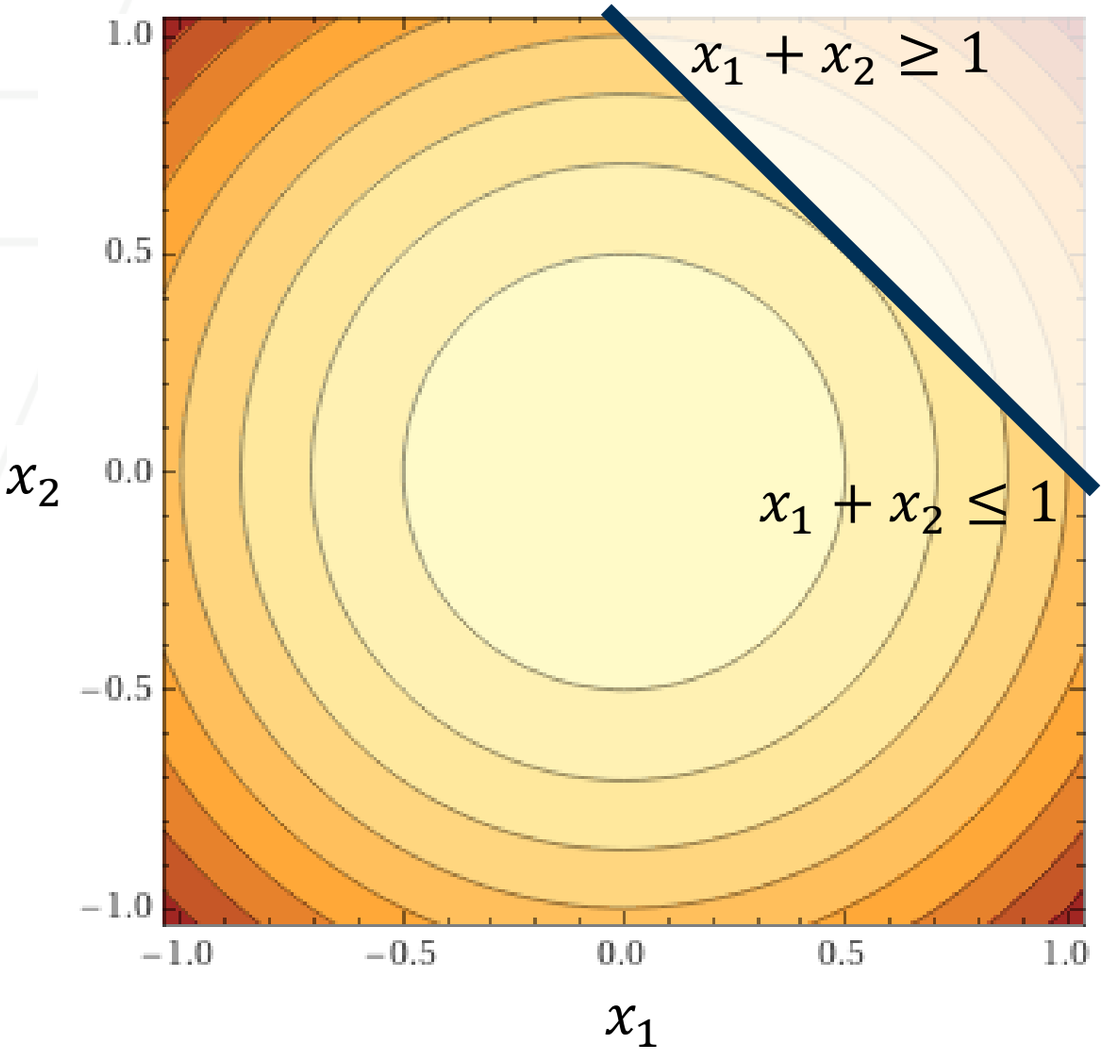
Weak Duality
•Dual optimum ≤ primal optimum (for minimization problems).
•Always true.

Strong Duality
•Dual optimum = primal optimum.
•Holds for **convex problems** with feasible solution.

We like quadratic programming because the objective functions shape allows easier optimization and more chances of minimization due to convex function. So it will always have strong duality.

Why care?
•Sometimes easier to solve the dual.
•Many ML models (e.g., SVMs) are solved via the dual.

Inequality constrained optimization: *Example 1*



$$\max_{\mathbf{x}} \quad 1 - x_1^2 - x_2^2$$

$$s.t. \quad x_1 + x_2 \leq 1$$

Inequality constrained optimization: *Example 1*

1. Write the Lagrangian for this problem

$$L(\mathbf{x}, \lambda) = -1 + x_1^2 + x_2^2 + \lambda(x_1 + x_2 - 1)$$

2. Calculate the gradient with respect to $\mathbf{x}$ (stationarity condition)

$$\frac{\partial L}{\partial x_1} = 2x_1 + \lambda = 0 \rightarrow \lambda = -2x_1$$

$$\frac{\partial L}{\partial x_2} = 2x_2 + \lambda = 0 \rightarrow \lambda = -2x_2$$

$$x_1 = x_2$$

Inequality constrained optimization: *Example 1*

3. Use the complementary slackness

Case 1: $\lambda > 0$ and $g(x) = 0$

$$x_1 + x_2 - 1 = 0 \rightarrow 2x_1 = 1$$

$$x_1 = x_2 = \frac{1}{2}$$

4a. Check if the dual feasibility condition is satisfied

$$\lambda = -2x_1 = -1$$

This solution **does NOT satisfy the dual feasibility condition** and it is thus **unfeasible**

Inequality constrained optimization: *Example 1*

3. Use the complementary slackness

Case 2: $\lambda = 0 \quad \text{and } g(x) < 0$

$$\lambda = -2x_1$$

$$x_1 = x_2 = 0$$

4b. Check if the primal feasibility condition is satisfied:

$$g(x_1, x_2) = 0 + 0 - 1 = -1 < 0$$

The solution is **feasible** and therefore the optimal!

Inequality constrained optimization: *Example 1*

Summary of the KKT conditions for the problem:

$$\left.\begin{aligned}\lambda &= -2x_1\\ \lambda &= -2x_2\end{aligned}\right\}\ \textbf{Stationarity}$$

$$x_1 + x_2 - 1 \leq 0 \quad \textbf{Primal feasibility}$$

$$\lambda \geq 0 \quad \textbf{Dual feasibility}$$

$$\lambda(x_1 + x_2 - 1) = 0 \quad \textbf{Complementary slackness}$$

Gradient descent

- In machine learning, some of the optimization problems suffer from the following challenges:
  - A closed form solution may be computationally intractable
    - Functions are complex: no easy derivative = 0 solution.
    - High dimensions: solving $\nabla f(x)=0$ analytically is impossible.
  - All the data may not be available at the time of training (e.g. streaming data)

- Gradient descent is an iterative minimization technique for differentiable functions on a domain

- **Intuition:** the function $f(\mathbf{x})$ decreases the fastest by going from x_n to x_{n+1} in the opposite direction of the gradient of $f(\mathbf{x})$

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \alpha_n \nabla f(\mathbf{x}_n)$$

Learning rate

- The choice of the learning rate α_n plays an important role in the success of the minimization:
  - If α_n is **too small** then the number of iterations necessary to reach the minimum will be too large
  - If α_n is **too large** then we may overshoot the minimum
- It can be set to a constant or varied iteratively to improve the performance of the algorithm

Momentum

- The gradient descent technique is very susceptible to local minima, therefore, it we can modified by adding a momentum term

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \alpha_n g(\mathbf{x}_n)$$

Where:

$$g(\mathbf{x}_n) = \mu g(\mathbf{x}_{n-1}) + \nabla f(\mathbf{x}_n)$$

$\mu =$ momentum constant

- The momentum constant also needs to be fine tuned along with the learning rate to each problem, typically picked as a value between 0 and 1

Recap

- Cross Entropy
- Loss Functions
- KL Divergence
- Jensen Inequality
- Unconstrained and Constrained Optimization
- Linear, Quadratic and Non-Linear Programming

Georgia Tech

Equality constrained optimization problem

There is a notion of a Lagrangian expression where we combine both the objective function and our constraints

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \lambda g(\mathbf{x})$$

$$s.t.\ \lambda \neq 0$$

Equality constrained optimization problem

- With multiple equality constraints the Lagrangian is expressed as follows:

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \sum_{i=1}^{k} \lambda_i g_i(\mathbf{x})$$

$$s.t. \quad \lambda_i \neq 0$$

Equality constrained optimization problem

- To solve this optimization problem, we only need to satisfy the stationarity condition, which states:

$$\nabla L(\mathbf{x}, \lambda) = 0$$

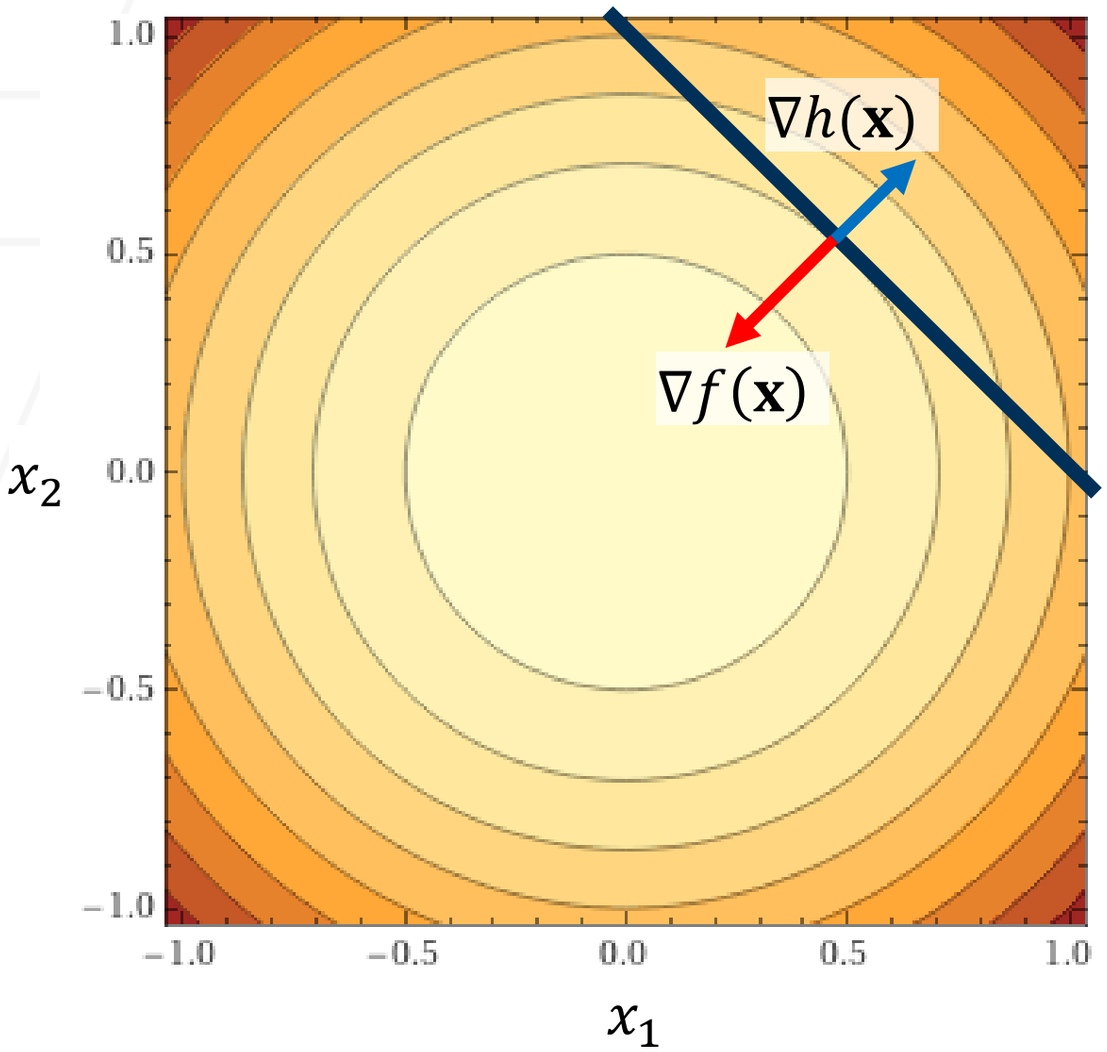
- We also need to satisfy

$$\lambda_i \neq 0$$

- This appeals to the intuition that at the optimal point, the gradient of the equality constraint is proportional to the gradient of the objective function

Equality constrained optimization: *Example 1*

Equality constrained optimization: *Example 2*



$$\max_{\mathbf{x}} \quad 1 - x_1^2 - x_2^2$$

$$s.t. \quad x_1 + x_2 = 1$$

Equality constrained optimization: *Example*

$$\max_{\mathbf{x}} \quad 1 - x_1^2 - x_2^2$$

$$s.t. \quad x_1 + x_2 = 1$$

1. Write the Lagrangian for this problem

$$L(\mathbf{x}, \lambda) = 1 - x_1^2 + x_2^2 + \lambda(x_1 + x_2 - 1)$$

2. Calculate the gradient with respect to $\mathbf{x}$

$$\frac{\partial L}{\partial x_1} = -2x_1 + \lambda = 0 \rightarrow \lambda = 2x_1$$

$$\frac{\partial L}{\partial x_2} = -2x_2 + \lambda = 0 \rightarrow \lambda = 2x_2$$

Equality constrained optimization: *Example*

3. Solve for $\mathbf{x}^*$ and λ

Using our previous two derivatives and the equality constraint, we obtain:

$$x_1 + x_2 = 1 \rightarrow x_1 + x_1 = 1 \rightarrow x_1^* = x_2^* = \frac{1}{2}$$

From $\frac{\partial L}{\partial x_1} = 0$ we find the value of λ

$$\lambda = 2x_1 \rightarrow \lambda = 1$$

Inequality constrained optimization problem

- Constraints that limit how small or big variables can be. These are inequality constraints noted as $g_j(\mathbf{x})$
- An optimization problem is usually expressed as:

$$\min_{\mathbf{x}} f(\mathbf{x})$$

$$\text{subject to} \quad g_j(\mathbf{x}) \leq 0, j = 1, 2, \ldots, m$$

.

Inequality constrained optimization problem

- The optimization problem can be rewritten as

$$. \min_{\mathbf{x}} \quad f(\mathbf{x}) + \sum_{j=1}^{m} \lambda_j g_j(\mathbf{x})$$

$$\text{subject to} \qquad \boldsymbol{\lambda} \geq 0$$

- We need to satisfy the Karush-Kuhn-Tucker (KKT) conditions on $\mathbf{x}$, $\boldsymbol{\lambda}$ which is **necessary and sufficient** for optimality

Karush-Kuhn-Tucker Conditions

- Stationarity condition:

$$\min L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_{j=1}^{m} \lambda_j g_j(\mathbf{x})$$

- Primal feasibility

$$g_j(\mathbf{x}) \leq 0, j = 1, 2, \ldots, m$$

- Dual Feasibility

$$\lambda_j \geq 0$$

- Complementary slackness

$$\lambda_j g_j(\mathbf{x}) = 0, j = 1, \ldots, m$$

If g(x) < 0 → **inactive** solution since λ =0
If g(x) = 0 → **active** solution since λ is at play and
> 0, which means constraint is active

Karush-Kuhn-Tucker Interpretations

- The complementary slackness condition applies only to inequality constraints
- It states that for a given inequality constraint…

$$\ldots\text{either } g_j(x) = 0 \quad \text{or} \quad \lambda_j = 0$$

- Whenever the dual variable $\lambda_j > 0$ and therefore $g_i(x) = 0$, we say that the **constraint is active** or that the **constraint is tight at x**
- The reverse $\lambda_j = 0$ and therefore $g_j(x) < 0$ means that the constraint is inactive, meaning that it is not effectively impacting the solution

Rule: For each inequality constraint, **either** it is active and has a positive multiplier, or it is inactive and its multiplier is zero.

Primal vs. Dual Form

Primal vs Dual Form

Primal Problem
Original optimization problem

Dual Problem
Derived problem as **a function of** λ only.

Analogy:
"Instead of directly solving for x and y together, we solve for what x and y *should look like* given a certain λ (like a candidate penalty price). Then we search over λ to find the best price that satisfies the constraint. This flips a 2D search into a 1D search."

Motivation:
In big problems (many x's), solving for x in terms of λ can be done analytically or numerically, but then the dual problem might be much lower-dimensional → computationally cheaper.

Primal vs Dual Form

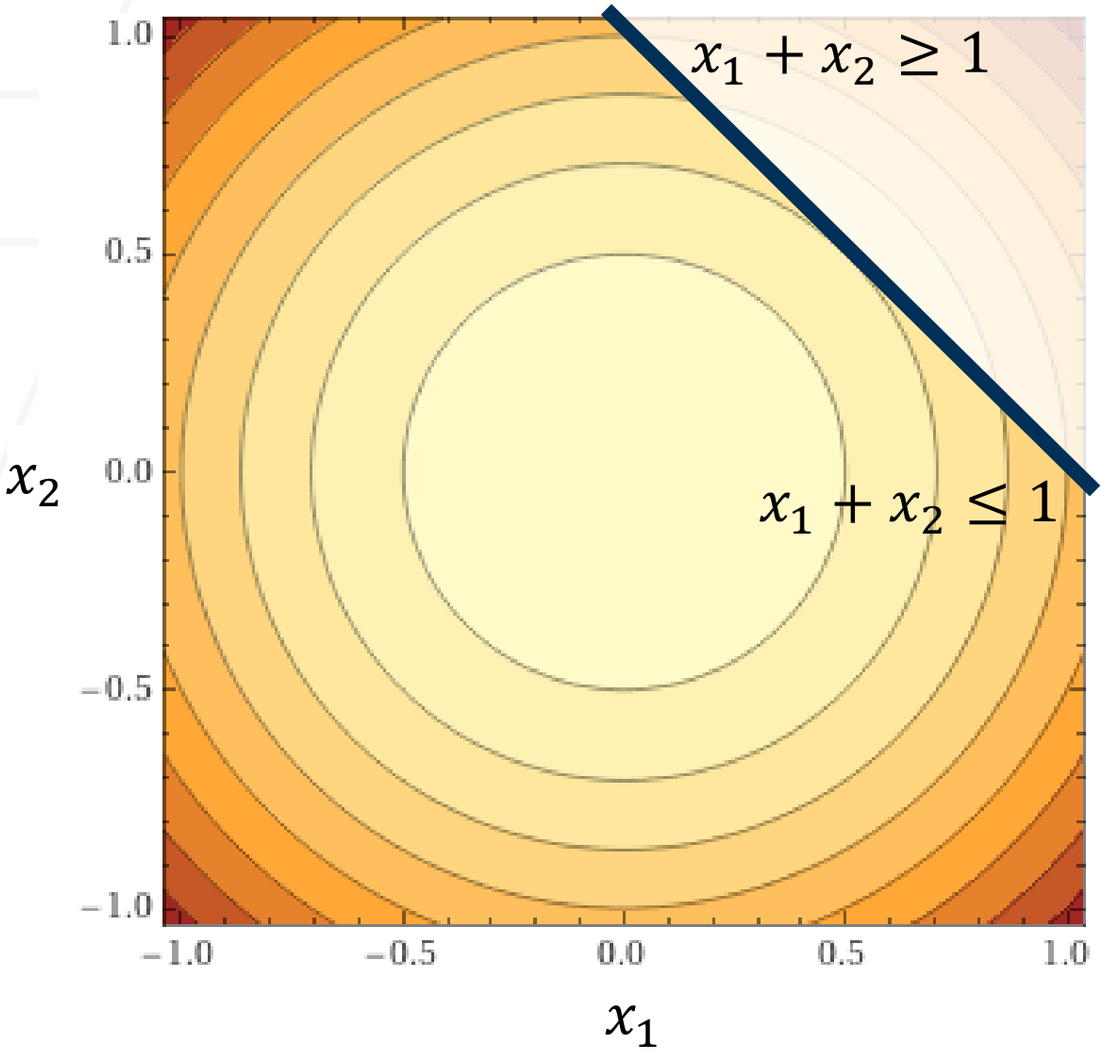
Weak Duality
- Dual optimum ≤ primal optimum (for minimization problems).
- Always true.

Strong Duality
- Dual optimum = primal optimum.
- Holds for **convex problems** with feasible solution.

This is why **Quadratic Programming** is popular: QP is convex and has efficient solvers that work on the dual.

Why care?
Many ML models (e.g., SVMs) are solved via the dual.

Inequality constrained optimization: *Example 1*



$$\max_{\mathbf{x}} \quad 1 - x_1^2 - x_2^2$$

$$s.t. \quad x_1 + x_2 \leq 1$$

Inequality constrained optimization: *Example 1*

1. Write the Lagrangian for this problem

$$L(\mathbf{x}, \lambda) = -1 + x_1^2 + x_2^2 + \lambda(x_1 + x_2 - 1)$$

2. Calculate the gradient with respect to $\mathbf{x}$ (stationarity condition)

$$\frac{\partial L}{\partial x_1} = 2x_1 + \lambda = 0 \rightarrow \lambda = -2x_1$$

$$\frac{\partial L}{\partial x_2} = 2x_2 + \lambda = 0 \rightarrow \lambda = -2x_2$$

$$x_1 = x_2$$

Inequality constrained optimization: *Example 1*

3. Use the complementary slackness

Case 1: $\lambda > 0$ and $g(x) = 0$

$$x_1 + x_2 - 1 = 0 \rightarrow 2x_1 = 1$$

$$x_1 = x_2 = \frac{1}{2}$$

4a. Check if the dual feasibility condition is satisfied

$$\lambda = -2x_1 = -1$$

This solution **does NOT satisfy the dual feasibility condition** and it is thus **unfeasible**

Inequality constrained optimization: *Example 1*

3. Use the complementary slackness

Case 2: $\lambda = 0$ and $g(x) < 0$

$$\lambda = -2x_1$$

$$x_1 = x_2 = 0$$

4b. Check if the primal feasibility condition is satisfied:

$$g(x_1, x_2) = 0 + 0 - 1 = -1 < 0$$

The solution is **feasible** and therefore the optimal!

Inequality constrained optimization: *Example 1*

Summary of the KKT conditions for the problem:

$$\left.\begin{aligned}\lambda &= -2x_1\\ \lambda &= -2x_2\end{aligned}\right\}\ \textbf{Stationarity}$$

$$x_1 + x_2 - 1 \leq 0 \quad \textbf{Primal feasibility}$$

$$\lambda \geq 0 \quad \textbf{Dual feasibility}$$

$$\lambda(x_1 + x_2 - 1) = 0 \quad \textbf{Complementary slackness}$$

Gradient descent

- In machine learning, some of the optimization problems suffer from the following challenges:
  - A closed form solution may be computationally intractable
    - Functions are complex: no easy derivative = 0 solution.
    - High dimensions: solving $\nabla f(x)=0$ analytically is impossible.
  - All the data may not be available at the time of training (e.g. streaming data)

- Gradient descent is an iterative minimization technique for differentiable functions on a domain

- **Intuition:** the function $f(\mathbf{x})$ decreases the fastest by going from x_n to x_{n+1} in the opposite direction of the gradient of $f(\mathbf{x})$

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \alpha_n \nabla f(\mathbf{x}_n)$$

Learning rate

- The choice of the learning rate α_n plays an important role in the success of the minimization:
  - If α_n is **too small** then the number of iterations necessary to reach the minimum will be too large
  - If α_n is **too large** then we may overshoot the minimum
- It can be set to a constant or varied iteratively to improve the performance of the algorithm

Momentum

- The gradient descent technique is very susceptible to local minima, therefore, it we can modified by adding a momentum term

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \alpha_n g(\mathbf{x}_n)$$

Where:

$$g(\mathbf{x}_n) = \mu g(\mathbf{x}_{n-1}) + \nabla f(\mathbf{x}_n)$$

$\mu =$ momentum constant

- The momentum constant also needs to be fine tuned along with the learning rate to each problem, typically picked as a value between 0 and 1