CS M146 - Week 8

Xinzhu Bei

xzbei@cs.ucla.edu

March 2, 2018

Overview

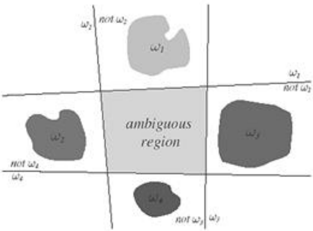
- Multi-class decision boundary
- Boosting, Adaboost
- Deterministic v.s. Probabilistic
- K means v.s. GMM
- Recap: Bayes Theorem, Bayesian Learning
- Maximum a posteriori v.s. Maximum Likelihood Estimation
- KKT v.s. Lagrange multipliers

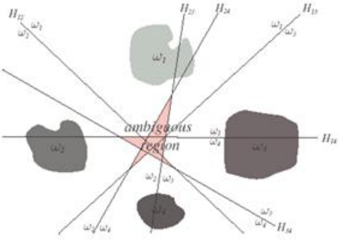
Survey

- How many people think multi-class classification and boosting are difficult?
- How many people think Kmeans and GMM are difficult?
- How many people think Bayesian Learning is difficult?

Decision Boundary of Multi-categorical Classifiers

- **One v.s. All**:
  Training: Decompose into k binary classification tasks
  Ideal Testing: only the correct label will have a positive score
  In class, we use: $\hat{y} = \arg\max_{y \in \{1,\ldots,k\}} w_y^T x$
- **One v.s. One**:
  Training: Decompose into $C(K, 2)$ binary classification tasks
  Testing: each label gets $k - 1$ votes, majority, tournament



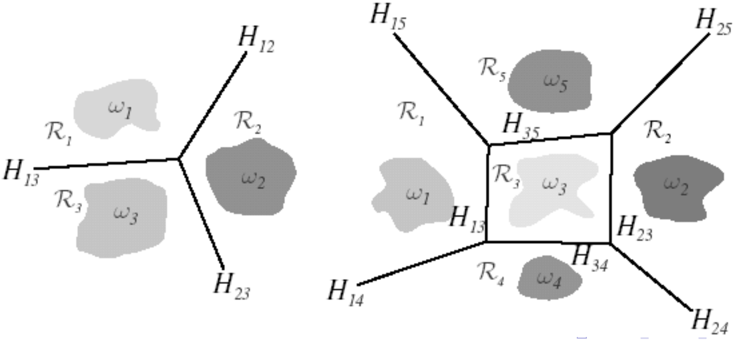


Decision Boundary of Multi-categorical Classifiers

- Both of these approaches can lead to regions in which the classification is undefined.
- In practice, all we need is for ${w_i}^T x$ to be more than all others ⇒ this is a weaker requirement: for examples with label i, we need

$$ {w_i}^T x > {w_j}^T x, \text{ for all } j \neq i $$

- The resulting classifier is called a **linear machine**.

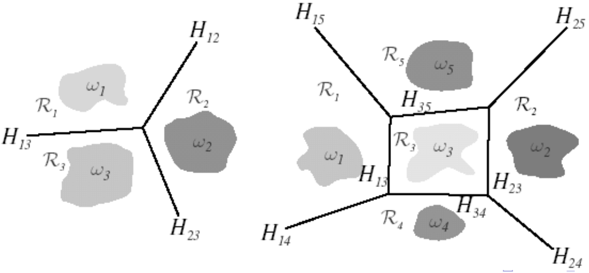


Decision Boundary of Multi-categorical Classifiers

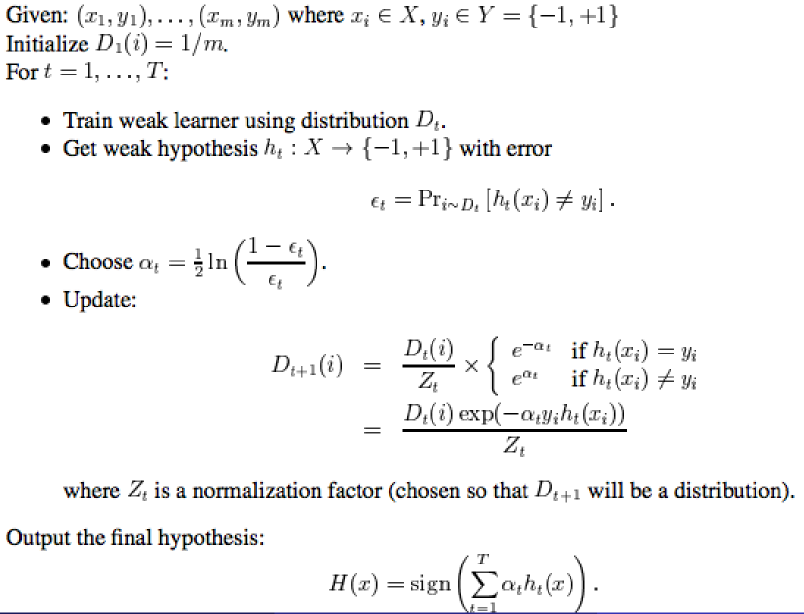
- If $\mathcal{R}_i$ and $\mathcal{R}_j$ are contiguous, the boundary between them is a portion of the hyperplane H_{ij} defined by

$$\begin{aligned} \mathbf{w}_i^T\mathbf{x} + \mathbf{w}_{i0} &= \mathbf{w}_j^T\mathbf{x} + w_{j0} \\ (\mathbf{w}_i - \mathbf{w}_j)^T\mathbf{x} + (w_{i0} - w_{j0}) &= 0 \\ \rightarrow \mathbf{w}_i - \mathbf{w}_j &\text{ is normal to } H_{ij} \end{aligned} \tag{1}$$

- Thus, with the linear machine it is not the weight vectors themselves but their differences that are important.



Boosting Algorithm: Adaboost



Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X, y_i \in Y = \{-1, +1\}$
Initialize $D_1(i) = 1/m$.
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : X \to \{-1, +1\}$ with error

$$\epsilon_t = \Pr_{i \sim D_t}\left[h_t(x_i) \neq y_i\right].$$

- Choose $\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \epsilon_t}{\epsilon_t}\right)$.
- Update:

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases} \\ &= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t} \end{aligned}$$

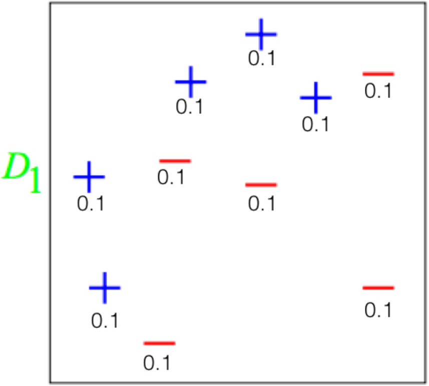
where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

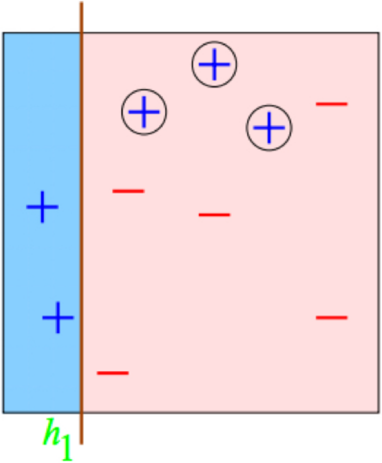
$$H(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right).$$

Boosting Algorithm: Adaboost

- **Weak learner** or **weak learning algorithm** is applied to find a **weak hypothesis** $h_t : \mathcal{X} \rightarrow \{-1, +1\}$, where the aim of the weak learner is to find a weak hypothesis with low weighted error ϵ_t relative to D_t
- Advanced topic: How to derive α_t
- Minimize $Z_t = \sum_i D_t(i) \exp(-\alpha_t y_i h_t(x_i)) \leftrightarrow$ minimize the error bound
  Choose $\alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$ to minimize Z_t
  Choose h_t that minimize the weighted error to minimize Z_t



	D1		
1	0.1		
2	0.1		
3	0.1		
4	0.1		
5	0.1		
6	0.1		
7	0.1		
8	0.1		
9	0.1		
10	0.1		

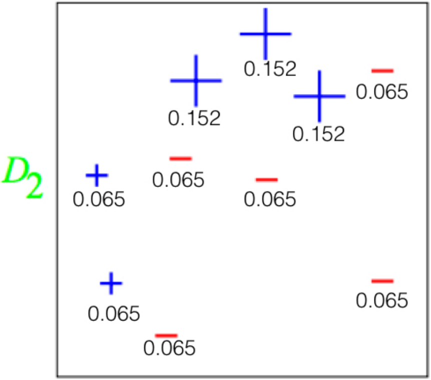


ε_1=0.30

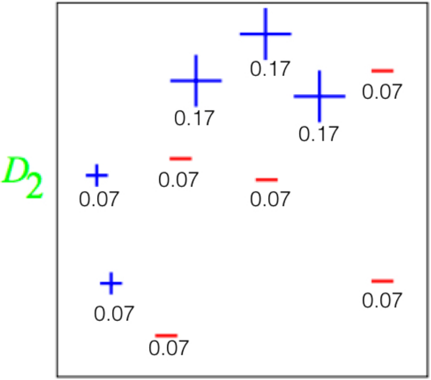
α_1=0.42

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$

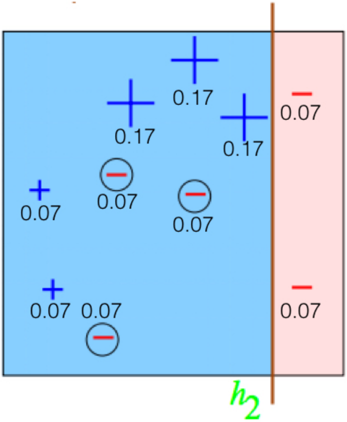
$$D_{t+1} = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t} \qquad e^{\alpha_t} = 1.52 \qquad e^{-\alpha_t} = 0.65$$



	D1	D2'
1	0.1	0.10*0.65
2	0.1	0.10*0.65
3	0.1	0.10*1.52
4	0.1	0.10*1.52
5	0.1	0.10*1.52
6	0.1	0.10*0.65
7	0.1	0.10*0.65
8	0.1	0.10*0.65
9	0.1	0.10*0.65
10	0.1	0.10*0.65



	D1	D2
1	0.1	0.07
2	0.1	0.07
3	0.1	0.17
4	0.1	0.17
5	0.1	0.17
6	0.1	0.07
7	0.1	0.07
8	0.1	0.07
9	0.1	0.07
10	0.1	0.07



ε_2=0.21

α_2=0.65

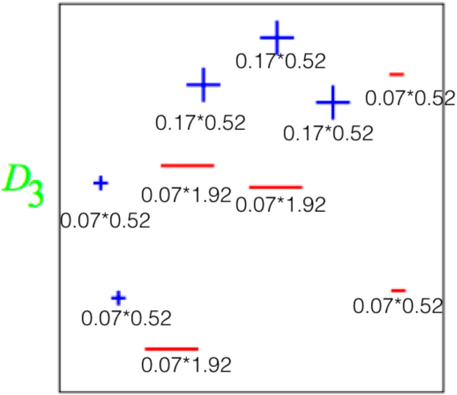
$$e^{\alpha_t} = 1.92$$

$$e^{-\alpha_t} = 0.52$$

$$D_{t+1} = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

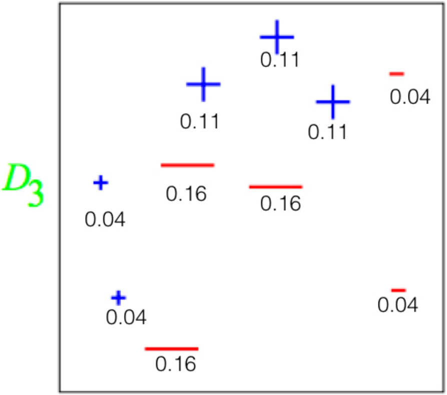
$$e^{\alpha_t} = 1.92$$

$$e^{-\alpha_t} = 0.52$$

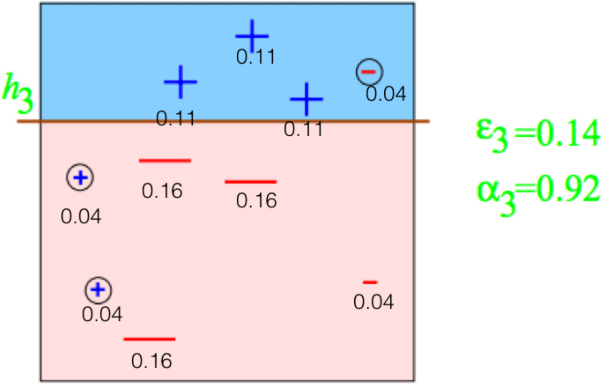


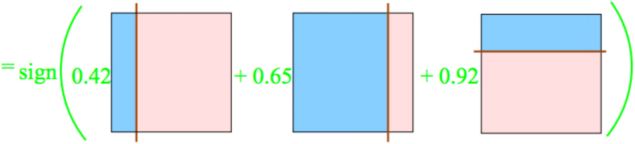
	D1	D2	D3'
1	0.1	0.07	0.07*1.92
2	0.1	0.07	0.07*0.52
3	0.1	0.17	0.17*0.52
4	0.1	0.17	0.17*0.52
5	0.1	0.17	0.17*0.52
6	0.1	0.07	0.07*1.92
7	0.1	0.07	0.07*1.92
8	0.1	0.07	0.07*1.92
9	0.1	0.07	0.07*1.92
10	0.1	0.07	0.07*1.92

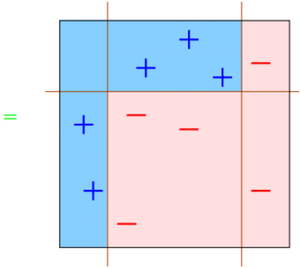
$$D_{t+1} = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t} \qquad \begin{aligned} e^{\alpha_t} &= 1.92 \\ e^{-\alpha_t} &= 0.52 \end{aligned}$$



	D1	D2	D3
1	0.1	0.07	0.04
2	0.1	0.07	0.04
3	0.1	0.17	0.11
4	0.1	0.17	0.11
5	0.1	0.17	0.11
6	0.1	0.07	0.16
7	0.1	0.07	0.16
8	0.1	0.07	0.16
9	0.1	0.07	0.04
10	0.1	0.07	0.04


h_3
0.11
0.11
0.11
0.04
0.04
0.16
0.16
0.04
0.04
0.16
$\varepsilon_3 = 0.14$
$\alpha_3 = 0.92$


= sign
0.42
+ 0.65
+ 0.92


=

Deterministic v.s. Probabilistic

Deterministic: All data is known beforehand

- Once you start the system, you know exactly what is going to happen.
- Example. Predicting the amount of money in a bank account.
  - If you know the initial deposit, and the interest rate, then:
  - You can determine the amount in the account after one year.

Probabilistic: Element of chance is involved

- You know the likelihood that something will happen, but you dont know when it will happen.
- Example. Roll a die until it comes up 5.
  - Know that in each roll, a 5 will come up with probability 1/6.
  - Dont know exactly when, but we can predict well.

Deterministic v.s. Probabilistic

- Are classifiers we learned until now (like SVM, perceptron, KNN, etc.) deterministic or probabilistic?
  - Unfortunately, they are deterministic.
  - A deterministic approach does not model the distribution of classes but rather separates the feature space and return the class associated with the space where a sample originates from.
- It's important to point out that probabilism and determinism are not mutually exclusive.
  - It is possible for every probabilistic method to simply return the class with the highest probability and therefore seem deterministic. (e.g. Logistic Regression, $p(y|x, \theta) > 0.5$)
  - Based on the distance to the seperating hyperplane in SVMs a probability can be computed and returned for each class.

Deterministic v.s. Probabilistic

What about K-means v.s. GMM?

- K-Means make hard assignments of points to clusters:

$$\gamma_{nk} \in \{0, 1\}$$

- GMM is an probabilistic approach of clustering:

$$\gamma_{nk} = P(z_n = k|x_n) \tag{2}$$

$$p(z_n = k|\boldsymbol{x}_n) = \frac{p(\boldsymbol{x}_n|z_n = k)p(z_n = k)}{p(\boldsymbol{x}_n)} = \frac{\overset{\boxed{N(x|\mu_k, \Sigma_k)}}{p(\boldsymbol{x}_n|z_n = k)}\,\overset{\boxed{\omega_k}}{p(z_n = k)}}{\sum_{k'=1}^{K} p(\boldsymbol{x}_n|z_n = k')p(z_n = k')}$$

- Both of them are **unsupervised** learning algorithm.

K-means algorithm
a.k.a Llyod's algorithm

- Step 0: randomly assign the cluster centers $\{\mu_k\}$
- Step 1: Minimize J over $\{r_{nk}\}$ -- Assign every point to the closest cluster center

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\boldsymbol{x}_n - \boldsymbol{\mu}_j\|_2^2 \\ 0 & \text{otherwise} \end{cases}$$

- Step 2: Minimize J over $\{\mu_k\}$ -- update the cluster centers

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk}\boldsymbol{x}_n}{\sum_n r_{nk}}$$

- Loop until it converges

GMM

- Let θ represent all parameters $\{w_k, \nu_k, \Sigma_k\}$
- Step 0: Initialize θ with some values (random or otherwise)
- Step 1: Computer $\gamma_{nk} = p(z_n = k|\mathbf{x}_n)$ using the current θ

$$p(z_n = k|\boldsymbol{x}_n) = \frac{p(\boldsymbol{x}_n|z_n = k)p(z_n = k)}{p(\boldsymbol{x}_n)} = \frac{\overset{\boxed{N(x|\mu_k,\Sigma_k)}}{p(\boldsymbol{x}_n|z_n = k)}\overset{\boxed{\omega_k}}{p(z_n = k)}}{\sum_{k'=1}^{K} p(\boldsymbol{x}_n|z_n = k')p(z_n = k')}$$

- Step 2: Update θ using the just computer γ_{nk}

$$\omega_k = \frac{\sum_n \gamma_{nk}}{\sum_k \sum_n \gamma_{nk}}, \quad \boldsymbol{\mu}_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} \boldsymbol{x}_n$$

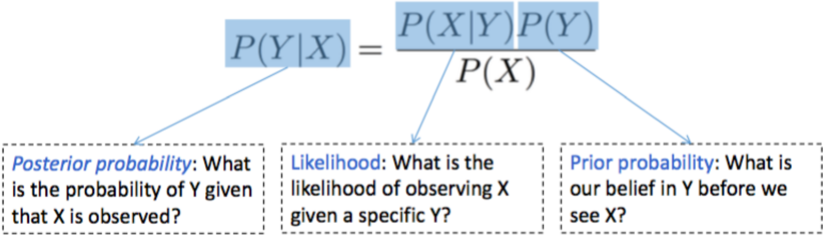
$$\boldsymbol{\Sigma}_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}$$

- Loop until converge

Kmeans v.s. GMM

	K means	GMM
Supervision	Unsupervised	Unsupervised
Model	Deterministic	Probabilistic
Hyperparameter	k	k
Objective	$\min \sum_{n,k} \gamma_{nk} \lVert x_n - \mu_k \rVert_2^2$	$\max p(x \mid \theta) = \prod_n p(x_n \mid \theta)$ $= \prod_n \sum_k p(z_n = k) p(x_n \mid z_n = k, \theta_k)$
Parameter	γ_{nk}, μ_k $\gamma_{nk} \in \{0, 1\}$ $\mu_k = \frac{\sum_n \gamma_{nk} x_n}{\sum_n \gamma_{nk}}$ -	w_k, μ_k, Σ_k $\gamma_{nk} = p(z_n = k \mid x_n)$ $\mu_k = \frac{\sum_n \gamma_{nk} x_n}{\sum_n \gamma_{nk}}$ w_k, Σ_k
Testing	$k = \arg\min_j \lVert x_n - \mu_j \rVert_2^2$	$k = \arg\max_j p(Z_n = j \mid x_n)$

Bayes Theorem



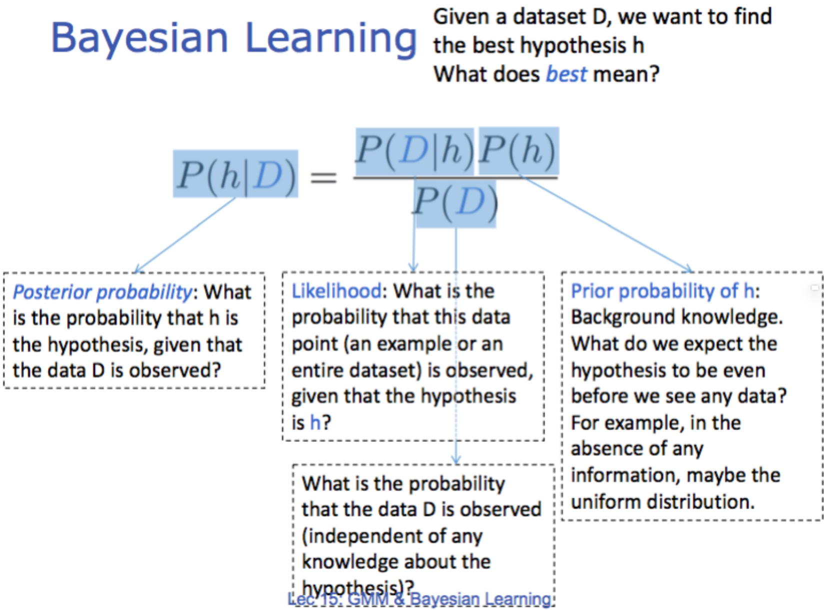
$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

Posterior probability: What is the probability of Y given that X is observed?

Likelihood: What is the likelihood of observing X given a specific Y?

Prior probability: What is our belief in Y before we see X?

Posterior / Likelihood £ Prior



Bayesian Learning

Given a dataset D, we want to find the best hypothesis h
What does *best* mean?

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

Posterior probability: What is the probability that h is the hypothesis, given that the data D is observed?

Likelihood: What is the probability that this data point (an example or an entire dataset) is observed, given that the hypothesis is h?

What is the probability that the data D is observed (independent of any knowledge about the hypothesis)?

Prior probability of h: Background knowledge. What do we expect the hypothesis to be even before we see any data? For example, in the absence of any information, maybe the uniform distribution.

Lec 15: GMM & Bayesian Learning

Karush-Kuhn-Tucker conditions

Given general problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{subject to} \quad & h_i(x) \leq 0, \quad i = 1, \ldots m \\ & \ell_j(x) = 0, \quad j = 1, \ldots r \end{aligned}$$

The **Karush-Kuhn-Tucker conditions** or **KKT conditions** are:

- $0 \in \partial f(x) + \sum_{i=1}^{m} u_i \partial h_i(x) + \sum_{j=1}^{r} v_j \partial \ell_j(x)$ (stationarity)
- $u_i \cdot h_i(x) = 0$ for all i (complementary slackness)
- $h_i(x) \leq 0, \ \ell_j(x) = 0$ for all i, j (primal feasibility)
- $u_i \geq 0$ for all i (dual feasibility)

Optimization - Lagrange multipliers

- In mathematical optimization, the method of Lagrange multipliers is a strategy for finding the local maxima and minima of a function subject to equality constraints.
- Consider an optimization problem:

$$\begin{aligned} &\text{minimize} && f(x_1, \cdots, x_n) \\ &\text{subject to} && g_k(x_1, \cdots, x_n) = 0, \quad k = 1, \ldots, M \end{aligned} \tag{3}$$

The Lagrangian takes the form

$$\mathcal{L}(x_1, \cdots, x_n, \lambda_1, \cdots, \lambda_M) = f(x_1, \ldots, x_n) - \sum_{k=1}^{M} \lambda_k g_k(x_1, \ldots, x_n)$$

Optimization - Lagrange multipliers

Methods of solving optimizaiton using Lagrangian multipliers:

- Step 1: Solve the following system of equations.

$$\begin{aligned} \frac{\partial L(x_1, \cdots, x_n, \lambda_1, \cdots, \lambda_M)}{\partial x_i} &= 0 \text{ , where } i = 1 \cdots n \\ \frac{\partial L(x_1, \cdots, x_n, \lambda_1, \cdots, \lambda_M)}{\partial \lambda_k} &= 0 \text{ , where } k = 1 \cdots M \end{aligned} \tag{4}$$

$$g_k(x_1, \cdots, x_n) = 0 \text{ , where } k = 1 \cdots M$$

- Step 2:Plug in all solutions $x_1, \cdots, x_n$, from the first step into $f(x_1, \cdots, x_n)$ and identify the minimum and maximum values, provided they exist.

Optimization - Lagrange multipliers

Find the extrema of the function $f(x, y) = 2y + x$ subject to the constraint $0 = g(x, y) = y^2 + xy - 1$.
Solution: Set $\mathcal{L}(x, y, \lambda) = f(x, y) + \lambda g(x, y)$, then

$$\begin{aligned} \frac{\partial L}{\partial x} &= 1 + \lambda y \\ \frac{\partial L}{\partial y} &= 2 + 2\lambda y + \lambda x \\ \frac{\partial L}{\partial \lambda} &= y^2 + xy - 1 \end{aligned} \tag{5}$$

Setting these equal to zero, we see from the third equation that $y \neq 0$, and from the first equation that $\lambda = \frac{-1}{y}$, so that from the second equation $0 = \frac{-x}{y}$ implying that $x = 0$. From the third equation, we obtain $y = \pm 1$.

** Note that it doesn't matter if you are using $f(\cdot) \pm \lambda g(\cdot)$, since all that changes is the sign of λ^*, where $(\lambda^*, x^*, \cdots)$ is the critical point.

The End