Foundations of Machine Learning (CS 725)

FALL 2024

Lecture 22:

- Clustering / PCA

Instructor: Preethi Jyothi

Consider the following abstract algorithm for approximating $\arg\min_{(\phi,\theta)\in\Phi\times\Theta} C(\phi,\theta)$.

The subroutines `minphi` and `mintheta` are such that $\texttt{minphi}(\theta) = \arg\min_{\phi'\in\Phi} C(\phi',\theta)$ and $\texttt{mintheta}(\phi) = \arg\min_{\theta'\in\Theta} C(\phi,\theta')$ (breaking ties arbitrarily).

```
1:  cost ← ∞
2:  flag ← true
3:  Initialize φ ∈ Φ
4:  repeat
5:      if flag = true then
6:          θ ← mintheta(φ)
7:      else
8:          φ ← minphi(θ)
9:      end if
10:     prevcost ← cost
11:     cost ← C(φ, θ)
12:     flag ← NOT flag
13: until prevcost = cost
14: return (φ, θ)
```

```
1:  cost ← ∞
2:  flag ← true
3:  Initialize φ ∈ Φ
4:  repeat
5:      if flag = true then
6:          θ ← mintheta(φ)
7:      else
8:          φ ← minphi(θ)
9:      end if
10:     prevcost ← cost
11:     cost ← C(φ, θ)
12:     flag ← NOT flag
13: until prevcost = cost
14: return (φ, θ)
```

Q1: [True or False? Justify] In line number 11, cost will never increase its value

Ans: True. (ϕ, θ) is updated only in line 6 or line 8. In either case, the updated `cost` is no more than `cost` before the update.

```
1:  cost ← ∞
2:  flag ← true
3:  Initialize φ ∈ Φ
4:  repeat
5:      if flag = true then
6:          θ ← mintheta(φ)
7:      else
8:          φ ← minphi(θ)
9:      end if
10:     prevcost ← cost
11:     cost ← C(φ, θ)
12:     flag ← NOT flag
13: until prevcost = cost
14: return (φ, θ)
```

Q2: [True or False? Justify] If θ remains unchanged on executing line number 6 in a particular iteration, then that will be the last iteration before the algorithm terminates.

Ans: True. If Line 6 is executed, then ϕ remains unchanged. If θ is also unchanged, then `prevcost = cost` in line 13, and the loop ends.

```
1:  cost ← ∞
2:  flag ← true
3:  Initialize φ ∈ Φ
4:  repeat
5:      if flag = true then
6:          θ ← mintheta(φ)
7:      else
8:          φ ← minphi(θ)
9:      end if
10:     prevcost ← cost
11:     cost ← C(φ, θ)
12:     flag ← NOT flag
13: until prevcost = cost
14: return (φ, θ)
```

Q3: [True or False? Justify] If θ is assigned a value θ_0 in line 6 at a particular iteration, then it will never again be assigned the same value in a later iteration.

Ans: False. Consider the following example: $\Phi = \{\phi_0, \phi_1\}$, $\Theta = \{\theta_0, \theta_1\}$ such that $C(\phi_0, \theta_0) = 2$, $C(\phi_0, \theta_1) = 3$, $C(\phi_1, \theta_0) = 1$, $C(\phi_1, \theta_1) = 3$. Then, if at Line 3, $\phi = \phi_0$, then Line 6 will be executed and θ is set to θ_0. In the next iteration, ϕ is set to ϕ_1 and in the next (and last) iteration, θ is again set to θ_0 .

```
cost ← ∞
flag ← true
Initialize φ ∈ Φ
repeat
    if flag = true then
        θ ← mintheta(φ)
    else
        φ ← minphi(θ)
    end if
    prevcost ← cost
    cost ← C(φ, θ)
    flag ← NOT flag
until prevcost = cost
return (φ, θ)
```

Q4: [True or False? Justify] Suppose Θ is a finite set, but Φ is infinite. Then the algorithm could run forever if $\arg\min_{\phi' \in \Phi} C(\phi', \theta)$ is not unique for some θ, depending on how `minphi` breaks ties.

Ans: False. θ can be repeated at most twice (with the first repetition leading to termination of the loop). Since Θ is finite, and every two iterations θ must be updated, there can be at most $2|\Theta| + 1$ iterations of the loop.