Imaj Bolgelerinin Ikili Karsilastirmasi

Bu bolumde bir beyani D ortaya koyacagiz, ki bu beyan, imajdaki iki bilesen (ki imaj gruplamasinin dogru olarak bulmaya calisacagi bilesenler) arasinda bir sinir olup olmadigina dair kanitin olcusu olacak. Beyanin temeli sudur: iki bilesen arasindaki sinirin boyunda yer alan her iki tarafin ogelerinin farkliligina bak, ve onu her bilesenin kendi icindeki farkliliga gore oranla. Yani bu beyan, bir bilesenin ic farkliligini dis farkliligina kiyaslar, ve bu sebeple verinin yerel karakteristikleri gozetmis olur. Kiyaslama mesela, global, verinin her yerinde aynen gecerli olacak bir sabit esik degerine vs. bagli degildir.

Tanim

Bir bilesen $C \subseteq V$, ki C bir bilesendir (component) ve V cizitin tum noktalaridir, *ic farkliligini*, o C'nin minimum kapsayan agacinin, yani $MST(C)$'sinin en buyuk kenar agirligi olarak aliyoruz. Bu ic farkliligi $Int(C)$ olarak belirtirsek,

$$ Int(C) = \max_{e \in MST(C,E)} w(e) $$

ki $w((v_i, v_j))$ bir cizit $G = (V, E)$'yi olusturan bir kenar $(v_i, v_j) \in E$ agirligi olarak belirtilir.

Tanim

Iki bilesen $C_1, C_2 \subseteq V$ arasindaki farki o iki bileseni birlestiren kenarlardan en ufagi olarak aliyoruz. Iki bilesenin arasinda birden fazla baglanti olmasi mumkundur, tum bunlara bakiyoruz, ve en ufagini aliyoruz.

$$ Dif(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} w((v_i, v_j)) $$

Eger C_1, C_2 arasinda bir kenar yok ise $Dif(C_1, C_2) = \infty$ kabul ediliyor.

Prensip olarak iki bilesen arasindaki en minimal baglantinin problem cikartabilecegi dusunulebilirdi, niye en az, niye ortalama vs degil? Fakat pratikte bu olcutun cok iyi isledigini gorduk. Hatta iyi olmaktan ote, bu olcutu minimal yerine medyan, ya da diger ceyreksel (quantile) olcute degistirdigimiz zaman (ki bunu yaparak aykiri degerlere -outlier- karsi daha dayanikli olmasini istemistik), algoritma cetrefilligi NP-Zor haline geliyor. Yani gruplama kriterinde ufacik bir degisiklik problemin cozum zorlulugunda muthis bir degisim ortaya cikartiyor.

Simdi iki bilesenin karsilastirma beyani D'nin tanimina geldik. D olcutu, $Dif(C_1, C_2)$'nin $Int(C_1)$ ya da $Int(C_2)$'den herhangi birinden daha buyuk olup olmadigina bakar. Ayrica bu karsilastirmayi bir esik degeri uzerinden pay ekleyerek yapar, eger irdeleme olumlu ise, iki bilesen arasinda sinir vardir, yoksa yoktur.

$$ D(C_1, C_2) = \begin{cases} Dogru & \text{Eger } Dif(C_1, C_2) > MInt(C_1, C_2) \text{ ise} \\ Yanlis & \text{Diger durumda} \end{cases} $$

Minimum ic fark MInt ise soyle tanimlidir,

$$\mathrm{MInt}(C_1, C_2) = \min(\mathrm{Int}(C_1) + \tau(C_1), \mathrm{Int}(C_2) + \tau(C_2))$$

Esik fonksiyonu τ ustteki irdeledigimiz fark hesaplarinin belli derecelerde disaridan etkilemek icin koyulmustur. Eger bunu kullanmasaydik sadece Int fonksiyonunu kullanmamiz gerekecekti, fakat bu olcut tek basina ufak bir bilesenin yerel karakteristiklerini gostermesi acisindan yeterli degildir. Asiri durumda mesela $|C| = 1, \mathrm{Int}(C) = 0$, yani en kucuk C durumudur bu ($|C|$ bilesenin icindeki oge sayisi), icinde tek oge vardir, ve hicbir kenar yoktur, $\mathrm{Int}(C) = 0$.

Bu sebeple iyi bir τ bilesenin buyuklugunu hesaba katarak, ona ters oranli bir rakam olustururSa iyi olur, mesela bir sabit k uzerinden,

$$\tau(C) = \frac{k}{|C|}$$

Bu demektir ki ufak bilesenler icin daha kuvvetli bir ispat ariyoruz, cunku kucuk $|C|$, τ'yu buyutecektir, ve Dif'in ondan buyuk olmasi daha zorlasacaktir. Tabii dikkat edelim, k bir "bilesen sayisi" degildir, yani fonksiyonuna dikkatli bakarsak, eger bilesenler arasinda yeterince buyuk bir fark var ise ufak bilesenlere hala izin verilmistir.

Algoritma soyledir, girdi olarak $G = (V, E)$ alir, ve V'yi S bilesenlerine ayirir ki her S icinde ona ait olan kenarlar vardir, yani $S = (C_1, .., C_r)$

felzenswalb(G)

E'yi $\pi = (o_1, .., o_m)$ seklinde kucukten buyuge dogru sirala.

Ilk basta S^0 gruplamasini al, ki bu durumda her kenar v_i kendi bileseni icindedir.

for $q = 1, .., m$

S^{q-1} gruplamasini baz alip S^q gruplamasini soyle yarat; q'inci siradaki kenarin birlestirdigi noktalari v_i, v_j oldugunu farz edelim, yani $o_q = (v_i, v_j)$.

return $S = (C_1, ..., C_r)$