

Log Lineer Modeller ve Kosulsal Rasgele Alanlar (Log Linear Models and Conditional Random Fields -CRF-)

Ders 2

Charles Elkan ders notlari

Kosulsal Olurluk (Conditional Likelihood)

Diyelim ki elimizde egitim verisi olarak ikili $\langle x, y \rangle$ veri noktaları var. O zaman y 'nin x 'e kosulsal olarak bagli (conditional on) bir dagilimi oldugunu soyleyebiliriz.

$$y \sim f(x; \theta)$$

Yani her x icin farkli bir y dagilimi ortaya cikabilir. Ve tum bu farkli dagilimlerin ortak noktası θ parametresidir. Kosulsal olasilik yani soyle yazilabilir,

$$P(Y = y|X = x; \theta)$$

Usttekiler Y icin bir model ortaya koydu, peki elimizde X 'in dagilimi icin bir olasilik modelimiz var mi? Cevap hayir. Niye? Dusunelim, $p(y, x)$ nedir ?

$$p(x, y) = p(x)p(y|x)$$

Ustte $p(y|x)$ 'i tanimlayacak (θ uzerinden) bir olasilik demeti / ailesi tanimladik, fakat elimizde $p(x)$ dagilimini verecek bir model yok, o zaman $p(x, y)$ 'yi tanimlayacak bir model de yok.

Fakat bu dunyanin sonu degil. Belki de Makine Ogrenimi bransinin bir slogani su ol-mali: “Ogrenmen gerekmeyen seyi ogrenme”. Ustteki ornekte $p(y|x)$ 'i ogrenebiliriz, ama $p(x)$ 'i illa ogrenmemiz gerekir mi?

Siniflayici (classifier) ve takip edilen (supervised) ogrenim durumunu dusunursek, bize egitim amaclı olarak $\langle x, y \rangle$ ikili veri noktaları saglanacak. x kaynak veri, y tahmin edilecek (ya da basta egitim hedefi olan) etiket olacak. y icin bir model ortaya cikartiyoruz, cunku test zamanında y olmayacak, fakat x hep olacak. Yani y 'nin modellenmesi mecburi, cunku “genelleyerek” onun ne oldugunu bulacagiz, ama x hep verili.

Kosulsal Olurluk Maksimum Olurluk Prensibi

Egitim verisi $\langle x_1, y_1 \rangle, \dots, \langle x_n, y_n \rangle$ icin, θ 'yi soyle sec

$$\hat{\theta} = \arg \max_{\theta} \prod_{i=1}^n p(y_i|x_i; \theta)$$

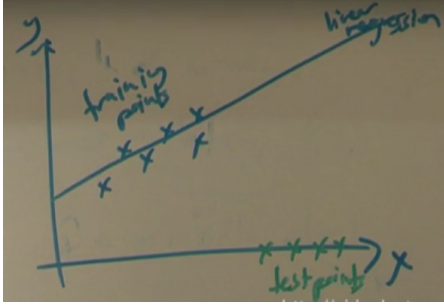
Normal maksimum olurlukta bilindigi gibi olasiliklerin carpimi maksimize edilir, burada maksimize ettigimiz “kosulsal” olasiliklerin carpimi.

Burada önemli bir soru su: bildiğimiz gibi maksimum olurluk hesabı her veri noktasının bir diğerinden bağımsız olduğunu farzeder [çünkü her olurluk hesabını bir diğer ile çarpıyoruz, başka ek çarpım, toplama, vs yapmıyoruz], bu faraziye doğru bir faraziye midir? Bu soru ve ona verilecek cevap çok önemli. Evet, eğer eğitim noktaları birbirinden bağımsız değilse maksimum olurluk kullanmamalıyız. Bağımsızlığı da iyi tanımlamak gerekiyor tabii, eğer üstteki durumda x_i verildikten sonra y_i 'lerin birbirinden bağımsız olması yeterli.

Bu model klasik İstatistik'te çokça kullanılan bir yaklaşımdır, hatta lineer regresyon'un temeli üstteki faraziyedir.

$$y = \alpha + \beta \bar{x} + N(0, \sigma^2)$$

Bu standart lineer regresyon modeli, ve bu modelde her y ona tekabül eden x 'e bağlı, bu sayede x 'ler biliniyorsa y 'ler birbirinden koşulsal olarak bağımsız hale geliyor, böylece x 'ler birbirine bağımlı olsa bile α ve β 'nin bulunması mümkün oluyor.



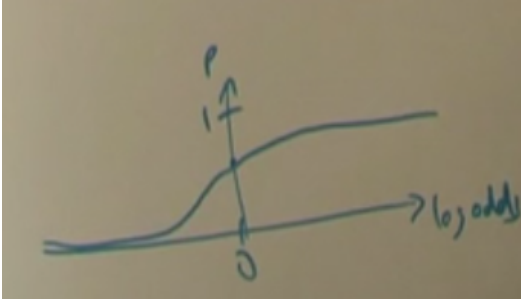
Üstteki resimde eğitim noktaları (training points) mavi olsun, test noktaları yeşil olsun (hemen altında). Bazı Yapay Öğrenim yaklaşımları diyebilir ki eğitim x 'lerinin dağılımı test x 'lerinin dağılımından farklı, bu veri seti öğrenilemez (yani genellenemez, modellenemez). Fakat klasik İstatistik buna bakar ve der ki x 'lerin verildiği durumda y 'ler bağımsızdır, bu şekilde bir koşulsal model öğrenilebilir.

Lojistik Regresyon aynı şekilde işler (lojistik regresyon, log lineer modellerin özel bir halidir, CRF'ler aynı şekilde). Burada da öğrenilen bir

$$p = p(y|x; \alpha, \beta)$$

modeli vardır ve y değerleri sadece 0 ve 1 olabilir. Tahmin edilen olasılık ise y 'nin 1 olma olasılığıdır. Bu model Rasgele Gradyan Çikisi ile eğitilir [detaylar için *Lojistik Regresyon* notlarımıza bakabilirsiniz].

$$\log \frac{p}{1-p} = \alpha + \sum_j \beta_j x_j$$



p log sansinin monotonik bir fonksiyonudur, ve ters yonden bakarsak, log sans p 'nin monotonik bir fonksiyonudur. Yani linear bir fonksiyon (sag taraf) ne kadar buyurse, olasilik / log sans o kadar buyuyecektir. Bu buyume durumu mesela β_j katsayisini veri analizi baglaminda yorumlanabilir hale getirir. Diyelim ki β_4 katsayisi pozitif, o zaman diger tum sartlarin esit oldugu durumda (with all else being equal) x_4 ne kadar buyurse 1 olma olasiligi o kadar artar.

Lojistik modellerin onemli bazi avantajlari var, ki bu avantajlar log lineer modellere de sirayet ediyor (bu iyi).

1) Degiskenler arasi ilinti (correlation) probleme yol acmaz: Bu fayda aslinda daha once belirttigimiz x 'lerin birbirine bagimli olabilmesi ile alakali. Bagimsizlik onsarti aranmadigi icin istedigimiz kadar x 'i problemin uzerine atabiliriz, egitici algoritma bunlardan cikartabildigi kadar iyi bir model bulacaktır.

Kiyasla mesela Naive Bayes boyle degildir, eger bir NB siniflayicisini egitiyorsak, ve ogelerin (feature) arasinda ilinti var ise, siniflayicinin dogrulugu (accuracy) azalabilir.

2) LR ile “1 olma olasiligini”, yani “bir sayisal skoru”, elde ediyoruz, bu sadece 1/0 degerinden daha fazla bir bilgi demektir.

3) Bu skor, anlami olan bir olasiliksal degerdir: Sonucta SVM siniflayicilari da $-\infty$ ve $+\infty$ arasinda degerler dondururler, ve bu degerler siralama (ranking) amaclil kullanilabilir, fakat olasilik matematigi acisindan anlami olan bir degerin olmasi bundan bile iyidir. Naive Bayes 0 ve 1 arasinda deger dondurebilir, fakat bu degerlerin de olasiliksal olarak aslinda anlami yoktur, pratikte goruldu ki bu degerler cok uc noktalarda, ya sifra cok yakin, ya bire cok yakin. Literaturde NB skorlarinin “iyi kalibre edilmiş olmadığı” soyleneir.

X_1, \dots, X_n test ornekleri ve tahmin edilen olasiliklar $P(Y = 1|x_i) = v_i$ olsun. Diyelim ki $s = \sum_i v_i$ ve t sayisi $1, \dots, n$ tane ogenin icinden $y = 1$ degerini tasiyan ogelerin sayisi olsun. Ornek, elimizde 100 tane egitim noktasi var, bunlari 60'i 1 degerinde. Bu durumda s yaklasik 60 olacaktir (rasgele gurultuyu hesaba katarsak tabii), yani $E[t] = s$ denebilecektir ve bu sadece eger olasiliklar iyi kalibre edilmissse soylenebilir.

4) Dengesiz egitim verisi kullanilabilir: pek cok egitim setinde mesela 1 degeri tasiyan degerleri 0 degeri tasiyanlardan cok daha fazla. Lojistik regresyon bu tur veriyle rahatca calisabilir.

Ders 3

Lojistik regresyon için log olurlugun (LCL) turevini almak lazim. Once basitlestirme amaclı $\alpha = \beta_o$, ve $x_0 = 1$. O zaman log sansin eski hali (altta esitligin sol tarafı) soyle yazilabilir (sag taraf), daha derli toplu bir formül olur,

$$\alpha + \sum_j \beta_j x_j = \sum_{j=0}^d \beta_j x_j$$

Bulmak istedigim her j için $\frac{d}{d\beta_j} LCL$ lazim

$$\frac{d}{d\beta_j} LCL = \sum_{i:y_i=1} \frac{d}{d\beta_j} \log p(1|..) + \sum_{i:y_i=0} \frac{d}{d\beta_j} \log p(0|..) \quad (3)$$

Eger ustteki bir bolumu p digerine $1 - p$ dersem, yani soyle

$$= \sum_{i:y_i=1} \frac{d}{d\beta_j} \underbrace{\log p(1|..)}_p + \sum_{i:y_i=0} \frac{d}{d\beta_j} \underbrace{\log p(0|..)}_{1-p}$$

O zaman

$$= \sum_{i:y_i=1} \frac{d}{d\beta_j} \log p + \sum_{i:y_i=0} \frac{d}{d\beta_j} \log(1 - p)$$

Biliyoruz ki

$$\frac{d}{d\beta_j} \log p = \frac{1}{p} \frac{d}{d\beta_j} p \quad (1)$$

$$\frac{d}{d\beta_j} \log(1 - p) = \frac{1}{1 - p} (-1) \frac{d}{d\beta_j} p \quad (2)$$

Ustteki son iki formülün her ikisinde de $d/d\beta_j p$ kısmi olduguna dikkat.

Notasyon

$$e = \exp \left[- \sum_{j=0}^n \beta_j x_j \right]$$

$$p = \frac{1}{1 + e}$$

$$1 - p = \frac{1 + e - 1}{1 + e} = \frac{e}{1 + e}$$

Simdi $d/d\beta_j p$ 'e donelim, ve p 'nin ustteki gibi oldugundan hareketle,

$$\begin{aligned}\frac{d}{d\beta_j} p &= (-1)(1+e)^{-2} \frac{d}{d\beta_j} e \\ &= (-1)(1+e)^{-2} (e) \frac{d}{d\beta_j} (x_j) \\ &= \frac{1}{1+e} \frac{e}{1+e} x_j = p(1-p)x_j\end{aligned}$$

Son ifade kodlama icin oldukca uygun, $d/d\beta_j p$ hesabini yine icinde p iceren bir ifadeye bagladik, ayrica turev x_j ile orantili.

Bu hesapla aslinda (1) icindeki $d/d\beta_j p$ kismini hesaplamis olduk. Eger yerine koyarsak,

$$\frac{d}{d\beta_j} \log p = \frac{1}{p} p(1-p)x_j$$

p 'ler iptal olur

$$= (1-p)x_j$$

Ayni sekilde (2) icin

$$\begin{aligned}\frac{d}{d\beta_j} \log(1-p) &= \frac{1}{1-p} (-1)p(1-p)x_j \\ &= -px_j\end{aligned}$$

Ustteki turevler tek bir egitim veri noktası icin. Tum egitim veri setinin turevi her noktanin turevlerinin toplami olacak, (3)'de goruldugu gibi.

$$\frac{d}{d\beta_j} LCL = \sum_{i:y_i=1} (1-p_i)x_{ij} + \sum_{i:y_i=0} -p_i x_{ij} \quad (4)$$

x_{ij} notasyonunda j, j^{inci} oge / ozellik anlamına geliyor. Simdi notasyonel bir numara kullanacagim,

$$= \sum_{tum\ i} (y_i - p_i)x_{ij}$$

Bunu niye yaptım? (4) formülünde eşitliğin sağ tarafı, birinci terim içinde 1 sayısı var, sonraki terimde 1 yok. Eğer 1 olup olmaması yerine y_i kullanırsam, ki zaten 1'in olup olmaması y_i 'nin 1 olup olmamasına bağlı, tek bir terimde işi halledebilirim. $y_i = 1$ olduğu zaman üstteki ifade $1 - p_i$ olacaktır, olmadığı zaman $-p_i$ olacaktır.

Eristigimiz sonucu analiz etmemiz gerekirse, nihai formül gayet basit ve temiz çıktı.

[24:10] kalibrasyonla alakalı bir yorum

Rasgele Gradyan Çikisi (Stochastic Gradient Ascent)

Fikir: türevi eğitim noktası başına hesapla, ve modeli hemen güncelle.

Eğitim noktaları $\langle x, y \rangle$ olarak gelsinler. Her nokta için, ve her β_j için

$$\frac{d}{d\beta_j} p(y|x; \beta) = g_j$$

hesapla.

$$\beta_j := \beta_j + \alpha g_j$$

Gradyanın ne olduğunu hatırlayalım, bir fonksiyonun maksimumuna “doğru” olan bir gidis yönünü gösterir, ve bu gidis yönü o fonksiyonu oluşturan değişkenlerin (parçalı türevleri) üzerinden belirtilir. O zaman elimizdeki gradyan o iç değişkenlerin maksimum yandaki değişim şeklini bize tarif eder.

Algoritmanın tamamı: alttaki formül için

$$\frac{d}{d\beta_j} p(y|\bar{x}; \bar{\beta}) = (y - p)x_j$$

Her x için

- O anki modele göre p 'yi hesapla

- Her $j = 0, \dots, d$ için

- $\beta_j := \beta_j + \alpha \underbrace{(y - p)x_j}_{\text{kısmi türev}}$ hesapla

Peki metotun ismindeki “rasgele (stochastic)” tanımı nereden geliyor? İyi bir soru bu çünkü metotta rasgele sayı üretimi gibi şeyler görmüyoruz. Cevap, metot yine de rasgele, çünkü her noktayı ayrı ayrı işliyoruz, ve bu noktaların eğitim algoritmasını gelişi bir nevi “veriye örneklemek” gibi sanki, ek olarak veriyi eğitime almadan önce rasgele şekilde karıştırmak da iyi olabilir.

Bazı Tavsiyeler (Heuristics)

1) Her özellik (feature) x_j 'i ölçeklemek, yani aynı ortalama (mean) ve varyansa sahip olacak şekilde tekrar ayarlamak. Yani mesela 0 ile 100 arasında olabilecek “yas” gibi

bir özelliği, 0 ve 1 arasında değişen özellikler ile aynı ortalama ve varyansa sahip olacak şekilde ayarlamak. Bunun sebebi güncelleme hesabındaki λ 'nin tek bir sabit olması, ve bu sabit her j için aynıdır, o sebeple λ 'nin her öğeye “aynı şekilde” uygulanabilmesi için öğelerin birbirine yakın olması iyidir. Ek olarak, genellikle eğitim verisinde 0 ile 1 arasında ikisel türden öğeler vardır, o sebeple bu şekilde olmayan diğer öğeleri 0 ve 1 arasında çekmek daha uygun ve kolay olur.

2) Veriyi rasgele şekilde sıralamak. Terminoloji: eğitim veri seti üzerinden bir geçiş yapmak bir “cag” (epoch) olarak bilinir.

3) λ 'yi deneme / yanılma yöntemi ile bulun (bu sabiti bulmanın sistemik bir yöntemi yok). Belki verinin içinden alınan daha ufak bir örneklem üzerinde bu deneme / yanılma işlemi yapılabilir.

4) Deneme yanılma işlemi şöyle yapabilirsiniz: büyük bir λ ile işe başlarsınız, ve her cagda λ değerini azaltabilirsiniz (mesela her cag sonunda 1/2 ile çarparak).

Ders 4

Log Lineer Modeller

Bu modeller lojistik regresyonun yapıya sahip (structured) girdiler ve çıktılar için genellenmiş halidir. Lojistik regresyonda girdi $\bar{x} \in \mathbb{R}^d$ ve çıktı $y \in 0, 1$ idi, yani çıktı ikiseldi. Fakat biz bundan daha genel makine öğrenimi problemlerini çözmek istiyoruz, yani istediğimiz $x \in \mathbb{X}$, ki \mathbb{X} herhangi bir uzay olabilmeli, ve $y \in \mathbb{Y}$ ki \mathbb{Y} aynı şekilde herhangi bir uzay olabilmeli.

Mesela x bir cümle olabilmeli, diyelim ki $x = \text{“he sat on the mat”}$, tercümesi “adam paspasın üzerinde oturdu”. Buna karşılık olan y ise mesela şöyle olabilmeli, $y = \text{“pronoun verb article noun”}$, yani her kelimenin hangi gramer ögesi olduğunu gösteren bir ibare. Mesela “sat” yani oturmak, bir fiil (verb), “mat” paspas, bir isim (noun), ve y içinde gelen eğitim verisinde bunlar olabilmeli (üstteki örnekte ikinci öge), sadece 0/1 değerleri değil.

Bu tabii ki takip edilen (supervised) bir eğitim şekli olacak. Fakat dikkat bazı makine öğrenimi uygulamalarında “çok sınıftan gelen” ama tek bir değer vardır, mesela $y \in 1, 2, 3$ olabilir, 3 sınıflı bir çıktı yani. Bazen çıktı gerçek sayı (real number) olabilir, ama yine de tek bir y değeri vardır. Üstteki durum böyle değildir. Potansiyel olarak y 'nin büyüklüğü x ile birebir aynı bile olmayabilir. Bu tür bir karışık eslemeden bahsediyoruz. Tek sınırlamamız \mathbb{Y} 'nin sonlu (finite) olması.

Model şöyle (notasyonu biraz değiştirdik, β yerine w kullanıyoruz mesela, w modelin “ağırlıklarını (weights)” temsil ediyor.

$$p(y|x; w) = \frac{\exp[\sum_j w_j F_j(x, y)]}{Z(x, w)}$$

Yakından bakarsak model LR modeline benziyor. Bir lineer fonksiyonun exp'si alınıyor ve bu değer olasılık hesabında kullanılıyor. İleride zaten göreceğiz ki LR üstteki yaklaşımın bir “özel durumu”, yani üstteki model daha genel bir tanım.

Aklımıza birçok soru geliyor herhalde, mesela “ Z nedir?” ya da “ F_j nasıl hesaplanır?” gibi. Z şöyle tanımlanır

$$Z(x, w) = \sum_{y'} \exp \left[\sum_j w_j F_j(x, y') \right]$$

Tüm y' 'lere bakılıyor, yani tüm mümkün \mathbb{Y} değerleri teker teker y' üzerinden toplamda kullanılıyor. \mathbb{Y} 'nin sonlu olma faraziyesi burada önemli hale geliyor, toplamı sonsuz bir kume üzerinden yapamayız.

Z normalizasyon için kullanılıyor, çünkü olasılık teorisinde eğer elimizde çoklu bir hedef var ise, bu hedeflere olan olasılık değerlerinin toplamı 1 olmalıdır. Z iste bunu garantiler, bu sebeple bölen (denominator) bölümün (nominator) toplamı olmalıdır.

Her $F_j(x, y)$ bir özellik fonksiyonudur (feature function). Niye? Çünkü elimdeki x 'ler illa bir vektor olmayabilir, yani x_j “vektörünü” alıp w_j “vektörü” ile carpamam, bu sebeple önce bir fonksiyon ile bir numerik değer üretmem gerekiyor. Kume olarak

$$F_j : \mathbb{X} \times \mathbb{Y} \rightarrow \mathbb{R}$$

Eğer $F_j(x, y) > 0$ ve $w_i > 0$ ise, o zaman $F_j(x, y) = 0$ 'a kıyasla $p(y|x; w)$ artar. Sezgisel olarak tarif edersek özellik fonksiyonun (OF) soylediği sudur, eğer ağırlık pozitif ise OF'in değeri ne kadar büyürse elimizdeki y , x ile o kadar “uyumludur” (tabii ki belli bir özellik yani j için). Negatif ilinti bunun tam tersi olurdu.

Eğitim w_j ağırlıklarını bulmamızı sağlar. F önceden tanımlıdır (yani eğitime bile başlamadan önce), bu fonksiyonun ne olacağı “secilir”. Secilirken tabii ki x, y arasındaki ilintiye göre fazla / az sonuç geri getirebilecek şekilde secilmelidir.

Kelime örneğine geri dönersek, bir F şöyle olabilir,

$F_{15}(x, y) =$ “eğer ikinci kelimenin baş harfi büyük ve ikinci etiket isim (noun)”. OF'ler reel değerlidir. Bunun özel durumu 0/1 değeri veren OF'lerdir. Biraz önceki örnek mesela 0/1 donduruyor.

Ya da $F_{14}(x, y)$ diyelim ki şöyle “ilk kelimenin baş harfi büyük, ve ilk etiket bir isim”. Tahmin edebiliriz ki eğitim setimizde ilk kelimesinin baş harfi büyük *olan* ama o kelimesi isim olmayan pek çok örnek olacaktır. Bu durumda w_{14} küçük olur.

Dedğimiz gibi F reel değeri olabilir, mesela

$$F_{16}(x, y) = \text{length}(y) - \text{length}(x)$$

yani bu fonksiyonda x 'nin uzunluğunu y 'nin uzunluğundan çıkartıyoruz. Bu ne işe yarar? Diyelim ki otomatik tercüme yapması için bir yapay öğrenim programı yazıyoruz, x, y eğitim noktaları birbirinin tercümesi olan İngilizce/Fransızca cümleler. Cogunlukla Fransızca cümleler tekabül ettikleri İngilizce cümlelerden çok daha uzun oluyorlar, yani üstteki çıkarma cogunlukla pozitif sonuç verecek. Değişik bir

acidan bakarsak, pozitif bir sonuc, bir tercumenin dogru oldugu yonunde bir isaret olarak kabul edilebilir, ve ustteki OF uzerinden egitim algoritmasi bunu kullanir. Egitim sonrasi w_{16} pozitif bir agirlik alacaktır.

Bir log lineer modelde (buna CRF'ler de dahil) ilk yapilan is probleminiz icin onemli olan OF'leri ortaya cikartmak.

F tanimlamanin degisik bir baska yolu:

$a(x)$ bir fonksiyon olsun. Her $v \in \mathbb{Y}$ icin

$$F_j(x, y) = a(x)I(y = v)$$

tanimlayalim.

$$p(y|x; w) = \frac{\exp \sum_j w_j F_j(x, y)}{Z}$$

Simdi lineer zincirli CRF konusuna bakalim. Yine $x \in \mathbb{X}$ ve $y \in \mathbb{Y}$. x bir girdi zinciri, y bir cikti zinciri ve en basit durumda x ile ayni uzunlukta. Konusma bolumlerini etiketlemek bu kategoriye dahil, ama bir diger uygulama kelimeyi arasina eksi isaretleri koyarak bolme (hyphenation).

Mesela girdi $x = \text{"beloved"}$, cikti $y = \text{"00100000"}$ cunku bu kelime "be-loved" olarak bolunur.

Bu uygulama icin bir OF

$$F_j(x, y) = \frac{\text{kac tane 1 var}}{x \text{ uzunlugu}}$$

$x = \text{"beloved"}$, cikti $y = \text{"00100000"}$ icin sonuc $1/7$ olurdu.

Lineer zincir CRF icin hangi OF'lerin bazi sinirlari var.

$$F_j(\bar{x}, \bar{y}) = \sum_i f_j(y_{i-1}y_i\bar{x}i)$$

ki sembol uzeri duz cizgiyi (\bar{x} gibi) bu sefer bir sirali veri temsil etmek icin kullaniyoruz)

Mesela

$$f_{18} = f_j(y_{i-1}y_i\bar{x}i) = \text{"}i = 2, y_{i-1} = 0, y_i = 1, x_1x_2 = \text{"as"}}$$

Mesela "async" kelimesi "a-sync" olarak bolumelir, ve egitim setinde "async" ile " $y = 01\dots$ " gelirse ustteki OF bu bolunmeyi odullendirir / ogrenir.

Simdi CRF olmayan bir Lineer Model'e bakalim,

Mesela cok etiketli takip edilen ogrenim. "Cok etiketli" ne demektir? Dikkat, "cok sinifli (multi label)" degil, yani tek ogenin iki veya daha fazla deger arasindan birini secmesinden bahsetmiyoruz. Birde fazla etiket alabilmekten bahsediyoruz, mesela bir Internet sayfasi, bir veya daha fazla kategoriye ayni anda ait olabilir, mesela hem Spor, hem Is Dunyasi. Diyelim ki 10 mumkun etiket var, bir dokuman kac degisik sekilde etiketlenebilir?

$2^{10} = 1024$ sekilde (bu sayi, hesap bir kumenin kac degisik sekilde alt kumesi olabilir hesabini yansitiyor ayni zamanda, yani siralama onemli olmadan belli sayida ogenin kac degisik sekilde alt kumeleri olabilir sorusu). Bu buyuk bir rakam. Ve bu kadar cok olasilik var ise, egitim verisi tum kombinasyonlar icin ornek veri icermeyebilir. Fakat muhakkak algoritmamizin bu kombinasyonlari tahmin edebilmesini tercih ederiz.

Cozum? 10 degisik siniflayici kurarak bu problemi cozebiliriz (ayri ayri, tek basina tek sinifa bakilince yeterli veri cikar herhalde), fakat bu sekilde "siniflararasi" iliskileri yakalayamayiz. Log lineer model yaklasiminda oyle bir ikisel (binary) OF yaratirsiniz ki, mesela,

$$F_{19}(x, y) = "Spor \in y, Is Dunyasi \in y"$$

Dikkat edersek OF sadece y 'ye bakiyor. Bu OF'yi iceren algoritma egitilince ustteki OF icin bir pozitif agirlik ogrenilebilecektir.