Değişken Temelli Modeller El Kitabı VIP

Afshine Amidi ve Shervine Amidi

September 14, 2019

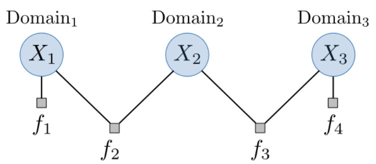
Başak Buluz ve Ayyüce Kızrak tarafından çevrilmiştir

Kısıt memnuniyet problemleri

Bu bölümde hedefimiz değişken-temelli modellerin maksimum ağırlık seçimlerini bulmaktır. Durum temelli modellerle kıyaslandığında, bu algoritmaların probleme özgü kısıtları kodlamak için daha uygun olmaları bir avantajdır.

Faktör grafikleri

❒ **Tanımlama** – Markov rasgele alanı olarak da adlandırılan faktör grafiği, $X_i\in\textrm{Domain}_i$ ve herbir $f_j(X)\geqslant 0$ olan $f_1,...,f_m$ m faktör olmak üzere $X=(X_1,...,X_n)$ değişkenler kümesidir.



❒ **Kapsam ve ilişki derecesi** – f_j faktörünün kapsamı, dayandığı değişken kümesidir. Bu kümenin boyutuna ilişki derecesi (arity) denir.

Not: faktörlerin ilişki derecesi 1 ve 2 olanlarına sırasıyla tek ve ikili denir.

❒ **Atama ağırlığı** – Her atama $x=(x_1,...,x_n)$, o atamaya uygulanan tüm faktörlerin çarpımı olarak tanımlanan bir $\textrm{Weight}(x)$ ağırlığı verir. Şöyle ifade edilir:

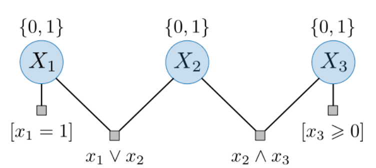
$$\boxed{\textrm{Weight}(x)=\prod_{j=1}^mf_j(x)}$$

❒ **Kısıt memnuniyet problemi** – Kısıtlama memnuniyet problemi (CSP, constraint satisfaction problem), tüm faktörlerin ikili olduğu bir faktör grafiğidir; bunları kısıt olarak adlandırıyoruz:

$$\boxed{\forall j\in[\![1,m]\!],\quad f_j(x)\in\{0,1\}}$$

Burada, j kısıtlı x ataması ancak ve ancak $f_j(x)=1$ olduğunda uygundur denir.

❒ **Tutarlı atama** – Bir CSP'nin bir x atamasının, yalnızca $\textrm{Weight}(x)=1$ olduğunda, yani tüm kısıtların yerine getirilmesi durumunda tutarlı olduğu söylenir.



Dinamik düzenleşim

❒ **Bağımlı faktörler** – X_i değişkeninin kısmi atamaya sahip bağımlı x değişken faktörlerinin kümesi $D(x,X_i)$ ile gösterilir ve X_i'yi önceden atanmış değişkenlere bağlayan faktörler kümesini belirtir.

❒ **Geri izleme araması** – Geri izleme araması (backtracking search), bir faktör grafiğinin maksimum ağırlık atamalarını bulmak için kullanılan bir algoritmadır. Her adımda, atanmamış bir değişken seçer ve değerlerini özyineleme ile arar. Dinamik düzenleşim (yani değişkenlerin ve değerlerin seçimi) ve bakış açısı (yani tutarsız seçeneklerin erken elenmesi), en kötü durum çalışma süresi üssel olarak olsa da grafiği daha verimli aramak için kullanılabilir: $O(|\textrm{Domain}|^n)$.

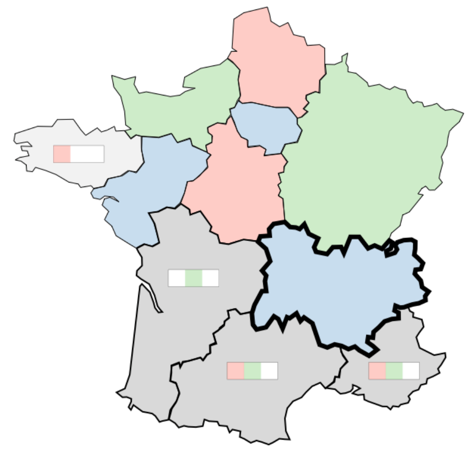
❒ **İleri kontrol** – Tutarsız değerleri komşu değişkenlerin etki alanlarından öncelikli bir şekilde ortadan kaldıran sezgisel bakış açısıdır. Aşağıdaki özelliklere sahiptir:

- Bir X_i değişkenini atadıktan sonra, tüm komşularının etki alanlarından tutarsız değerleri eler.
- Bu etki alanlardan herhangi biri boş olursa, yerel geri arama araması durdurulur.
- Eğer X_i değişkenini atamazsak, komşularının etki alanını eski haline getirilmek zorundadır.

❒ **En kısıtlı değişken** – En az tutarlı değere sahip bir sonraki atanmamış değişkeni seçen, değişken seviyeli sezgisel düzenleşimdir. Bu, daha verimli budama olanağı sağlayan aramada daha önce başarısız olmak için tutarsız atamalar yapma etkisine sahiptir.

❒ **En düşük kısıtlı değer** – Komşu değişkenlerin en yüksek tutarlı değerlerini elde ederek bir sonrakini veren seviye düzenleyici sezgisel bir değerdir. Sezgisel olarak, bu prosedür önce çalışması en muhtemel olan değerleri seçer.

Not: uygulamada, bu sezgisel yaklaşım tüm faktörler kısıtlı olduğunda kullanışlıdır.



Yukarıdaki örnek, en kısıtlı değişken keşfi ve sezgisel en düşük kısıtlı değerin yanı sıra, her adımda ileri kontrol ile birleştirilmiş geri izleme arama ile 3 renk probleminin bir gösterimidir.

❒ **Ark tutarlılığı** – X_l değişkeninin ark tutarlılığının (arc consistency) X_k'ye göre her bir $x_l\in\textrm{Domain}_l$ için geçerli olduğu söylenir:

- X_l'in birleşik faktörleri sıfır olmadığında,

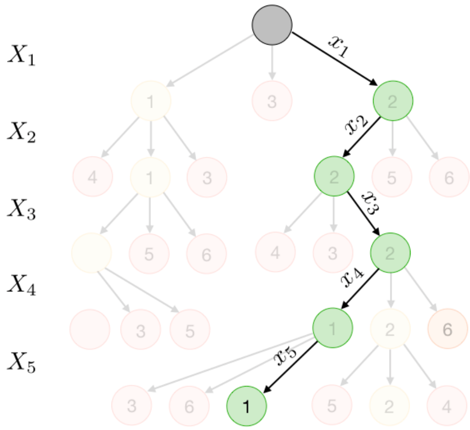
- en az bir $x_k \in \text{Domain}_k$ vardır, öyle ki X_l ve X_k arasında sıfır olmayan herhangi bir faktör vardır.

❒ **AC-3** – AC-3 algoritması, tüm ilgili değişkenlere ileri kontrol uygulayan çok adımlı sezgisel bir bakış açısıdır. Belirli bir görevden sonra ileriye doğru kontrol yapar ve ardından işlem sırasında etki alanının değiştiği değişkenlerin komşularına göre ark tutarlılığını ardı ardına uygular.

Not: AC-3, tekrarlı ve özyinelemeli olarak uygulanabilir.

Yaklaşık yöntemler

❒ **Işın araması** – Işın araması (beam search), her adımda K en üst yollarını keşfederek, $b = |\text{Domain}|$ dallanma faktörünün n değişkeninin kısmi atamalarını genişleten yaklaşık bir algoritmadır.
Aşağıdaki örnek, $K = 2$, $b = 3$ ve $n = 5$ parametreleri ile muhtemel ışın aramasını göstermektedir.



Not: $K = 1$ açgözlü aramaya (greedy search) karşılık gelirken $K \to +\infty$, BFS ağaç aramasına eşdeğerdir.

❒ **Tekrarlanmış koşullu modlar** – Tekrarlanmış koşullu modlar (ICM, iterated conditional modes), yakınsamaya kadar bir seferde bir değişkenli bir faktör grafiğinin atanmasını değiştiren yinelemeli bir yaklaşık algoritmadır. i adımında, X_i'ye, bu değişkene bağlı tüm faktörlerin çarpımını maksimize eden v değeri atanır.

Not: ICM yerel minimumda takılıp kalabilir.

❒ **Gibbs örneklemesi** – Gibbs örneklemesi (Gibbs sampling), yakınsamaya kadar bir seferde bir değişken grafik faktörünün atanmasını değiştiren yinelemeli bir yaklaşık yöntemdir. i adımında:

- her bir $u \in \text{Domain}_i$ olan öğeye, bu değişkene bağlı tüm faktörlerin çarpımı olan bir ağırlık $w(u)$ atanır,
- v'yi w tarafından indüklenen olasılık dağılımından örnek alır ve X_i'ye atanır.

Not: Gibbs örneklemesi, ICM'nin olasılıksal karşılığı olarak görülebilir. Çoğu durumda yerel minimumlardan kaçabilme avantajına sahiptir.

Faktör grafiği dönüşümleri

❒ **Bağımsızlık** – A, B, X değişkenlerinin bir bölümü olsun. A ve B arasında kenar yoksa A ve B'nin bağımsız olduğu söylenir ve şöyle ifade edilir:

$$\boxed{A,B \text{ bağımsız} \Longleftrightarrow A \perp\!\!\!\perp B}$$

Not: bağımsızlık, alt sorunları paralel olarak çözmemize olanak sağlayan bir kilit özelliktir.

❒ **Koşullu bağımsızlık** – Eğer C'nin şartlandırılması, A ve B'nin bağımsız olduğu bir grafik üretiyorsa A ve B verilen C koşulundan bağımsızdır. Bu durumda şöyle yazılır:

$$\boxed{A \perp\!\!\!\perp B | C}$$

❒ **Koşullandırma** – Koşullandırma, bir faktör grafiğini paralel olarak çözülebilen ve geriye doğru izlemeyi kullanabilen daha küçük parçalara bölen değişkenleri bağımsız kılmayı amaçlayan bir dönüşümdür. $X_i = v$ değişkeninde koşullandırmak için aşağıdakileri yaparız:

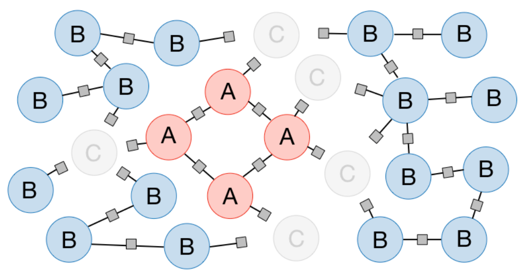
- X_i'ye bağlı tüm $f_1,...,f_k$ faktörlerini göz önünde bulundurun
- X_i ve $f_1,...,f_k$ öğelerini kaldırın
- $j \in \{1,...,k\}$ için $g_j(x)$ ekleyin:

$$\boxed{g_j(x) = f_j(x \cup \{X_i : v\})}$$

❒ **Markov blanket** – $A \subseteq X$ değişkenlerin bir alt kümesi olsun. $\text{MarkovBlanket}(A)$'i, A'da olmayan A'nın komşuları olarak tanımlıyoruz.

❒ **Önerme** – $C = \text{MarkovBlanket}(A)$ ve $B = X \backslash (A \cup C)$ olsun. Bu durumda:

$$\boxed{A \perp\!\!\!\perp B | C}$$



❒ **Eliminasyon** – Eliminasyon, X_i'yi grafikten ayıran ve Markov blanket de şartlandırılmış küçük bir alt sorunu çözen bir faktör grafiği dönüşümüdür:

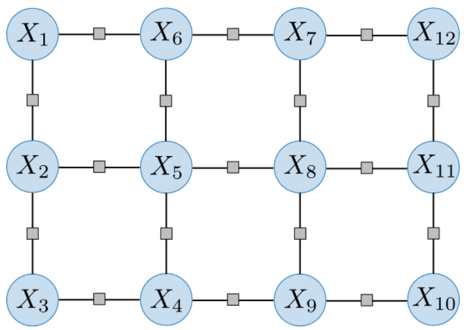
- X_i'ye bağlı tüm $f_{i,1},...,f_{i,k}$ faktörlerini göz önünde bulundurun
- X_i ve $f_{i,1},...,f_{i,k}$ kaldır
- $f_{\text{new},i}(x)$ ekleyin, şöyle tanımlanır:

$$\boxed{f_{\text{new},i}(x) = \max_{x_i} \prod_{l=1}^{k} f_{i,l}(x)}$$

❒ **Ağaç genişliği** – Bir faktör grafiğinin ağaç genişliği (treewidth), değişken elemeli en iyi değişken sıralamasıyla oluşturulan herhangi bir faktörün maksimum ilişki derecesidir. Diğer bir deyişle,

$$\boxed{\textrm{Treewidth} = \min_{\textrm{orderings}} \max_{i\in\{1,...,n\}} \textrm{arity}(f_{\textrm{new},i})}$$

Aşağıdaki örnek, ağaç genişliği 3 olan faktör grafiğini gösterir.



Not: en iyi değişken sıralamasını bulmak NP-zor (NP-hard) bir problemdir.

Bayesçi ağlar

Bu bölümün amacı koşullu olasılıkları hesaplamak olacaktır. Bir sorgunun kanıt verilmiş olma olasılığı nedir?

Giriş

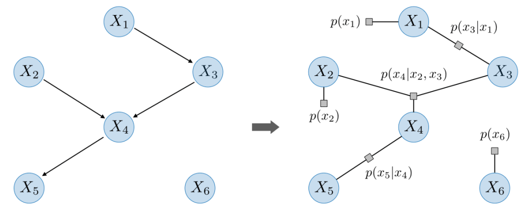
❒ **Açıklamalar** – C_1 ve C_2 sebeplerinin E etkisini yarattığını varsayalım. E etkisinin durumu ve sebeplerden biri (C_1 olduğunu varsayalım) üzerindeki etkisi, diğer sebep olan C_2'nin olasılığını değiştirir. Bu durumda, C_1'in C_2'yi açıkladığı söylenir.

❒ **Yönlü çevrimsiz çizge** – Yönlü çevrimsiz bir çizge (DAG, directed acyclic graph), yönlendirilmiş çevrimleri olmayan sonlu bir yönlü çizgedir.

❒ **Bayesçi ağ** – Her düğüm için bir tane olmak üzere, yerel koşullu dağılımların bir çarpımı olarak, $X=(X_1,...,X_n)$ rasgele değişkenleri üzerindeki bir ortak dağılımı belirten yönlü bir çevrimsiz çizgedir:

$$\boxed{P(X_1=x_1,...,X_n=x_n)\triangleq\prod_{i=1}^n p(x_i|x_{\textrm{Parents}(i)})}$$

Not: Bayesçi ağlar olasılık diliyle bütünleşik faktör grafikleridir.



❒ **Yerel olarak normalleştirilmiş** – Her $x_{\textrm{Parents}(i)}$ için tüm faktörler yerel koşullu dağılımlardır. Bu nedenle yerine getirmek zorundalar:

$$\boxed{\sum_{x_i} p(x_i|x_{\textrm{Parents}(i)})=1}$$

Sonuç olarak, alt-Bayesçi ağlar ve koşullu dağılımlar tutarlıdır.

Not: yerel koşullu dağılımlar gerçek koşullu dağılımlardır.

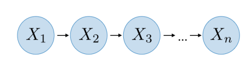
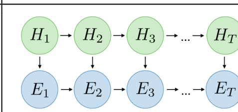
❒ **Marjinalleşme** – Bir yaprak düğümünün marjinalleşmesi, o düğüm olmaksızın bir Bayesçi ağı sağlar.

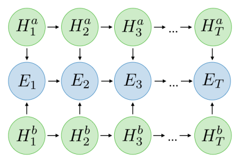
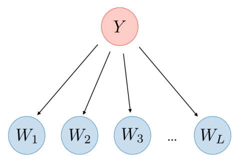
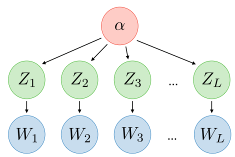
Olasılık programları

❒ **Konsept** – Olasılıklı bir program değişkenlerin atanmasını randomize eder. Bu şekilde, ilişkili olasılıkları açıkça belirtmek zorunda kalmadan atamalar üreten karmaşık Bayesçi ağlar yazılabilir.

Not: Olasılık programlarına örnekler arasında Gizli Markov modeli (HMM, hidden Markov model), faktöriyel HMM, naif Bayes (naive Bayes), gizli Dirichlet tahsisi (LDA, latent Dirichlet allocation), hastalıklar ve semptomları belirtirler ve stokastik blok modelleri bulunmaktadır.

❒ **Özet** – Aşağıdaki tablo, ortak olasılıklı programları ve bunların uygulamalarını özetlemektedir:

Program	Algoritma	Gösterim	Örnek
Markov Modeli	$X_i\sim p(X_i\vert X_{i-1})$		Dil modelleme
Gizli Markov Modeli (HMM)	$H_t\sim p(H_t\vert H_{t-1})$ $E_t\sim p(E_t\vert H_t)$		Nesne izleme

Faktöriyel HMM	$H_t^o \underset{o\in\{a,b\}}{\sim} p(H_t^o\|H_{t-1}^o)$ $E_t \sim p(E_t\|H_t^a,H_t^b)$		Çoklu nesne izleme
Naif Bayes	$Y \sim p(Y)$ $W_i \sim p(W_i\|Y)$		Belge sınıflandırma
Gizli Dirichlet Tahsisi (LDA)	$\alpha \in \mathbb{R}^K$ dağılım $Z_i \sim p(Z_i\|\alpha)$ $W_i \sim p(W_i\|Z_i)$		Konu modelleme

Çıkarım

❒ **Genel olasılıksal çıkarım stratejisi** – $E=e$ kanıtı verilen Q sorgusunun $P(Q|E=e)$ olasılığını hesaplama stratejisi aşağıdaki gibidir:

- Adım 1: Q sorgusunun ataları olmayan değişkenlerini ya da marjinalleştirme yoluyla E kanıtını silin
- Adım 2: Bayesçi ağı faktör grafiğine dönüştürün
- Adım 3: kanıtın koşulu $E=e$
- Adım 4: Q sorgusu ile bağlantısı kesilen düğümleri marjinalleştirme yoluyla silin
- Adım 5: olasılıklı bir çıkarım algoritması çalıştırın (kılavuz, değişken eleme, Gibbs örneklemesi, parçacık filtreleme)

❒ **İleri-geri algoritma** – Bu algoritma, L boyutunda bir HMM durumunda herhangi bir $k\in\{1,...,L\}$ için $P(H=h_k|E=e)$ (düzeltme sorgusu) değerini hesaplar. Bunu yapmak için 3 adımda ilerlenir:

- Adım 1: için $i\in\{1,...,L\}$, hesapla $F_i(h_i)=\sum_{h_{i-1}}F_{i-1}(h_{i-1})p(h_i|h_{i-1})p(e_i|h_i)$
- Adım 2: için $i\in\{L,...,1\}$, hesapla $B_i(h_i)=\sum_{h_{i+1}}B_{i+1}(h_{i+1})p(h_{i+1}|h_i)p(e_{i+1}|h_{i+1})$
- Adım 3: için $i\in\{1,...,L\}$, hesapla $S_i(h_i)=\frac{F_i(h_i)B_i(h_i)}{\sum_{h_i}F_i(h_i)B_i(h_i)}$

$F_0=B_{L+1}=1$ kuralı ile. Bu prosedürden ve bu notasyonlardan anlıyoruz ki

$$\boxed{P(H=h_k|E=e)=S_k(h_k)}$$

Not: bu algoritma, her bir atamada her bir kenarın $h_{i-1}\rightarrow h_i$'nin $p(h_i|h_{i-1})p(e_i|h_i)$ olduğu bir yol olduğunu yorumlar.

❒ **Gibbs örneklemesi** – Bu algoritma, büyük olasılık dağılımını temsil etmek için küçük bir dizi atama (parçacık) kullanan tekrarlı bir yaklaşık yöntemdir. Rasgele bir x atamasından Gibbs örneklemesi, $i\in\{1,...,n\}$ için yakınsamaya kadar aşağıdaki adımları uygular:

- Tüm $u\in\textrm{Domain}_i$ için, x atamasının $w(u)$ ağırlığını hesaplayın, burada $X_i=u$
- Örnekleme w : $v\sim P(X_i=v|X_{-i}=x_{-i})$ ile uyarılmış olasılık dağılımından
- Set $X_i=v$

Not: X_{-i}, $X\backslash\{X_i\}$ ve x_{-i}, karşılık gelen atamayı temsil eder.

❒ **Parçacık filtreleme** – Bu algoritma, bir seferde K parçacıklarını takip ederek gözlem değişkenlerinin kanıtı olarak verilen durum değişkenlerinin önceki yoğunluğuna yaklaşır. K boyutunda bir C parçacığı kümesinden başlayarak, aşağıdaki 3 adım tekrarlı olarak çalıştırılır:

- Adım 1: teklif - Her eski parçacık $x_{t-1}\in C$ için, geçiş olasılığı dağılımından $p(x|x_{t-1})$ örnek x'i alın ve C' ye ekleyin.
- Adım 2: ağırlıklandırma - C' nin her x değerini $w(x)=p(e_t|x)$ ile ağırlıklandırın, burada e_t t zamanında gözlemlenen kanıttır.
- Adım 3: yeniden örnekleme - w ile indüklenen olasılık dağılımını kullanarak C' kümesinden örnek K elemanlarını C cinsinden saklayın: bunlar şuanki x_t parçacıklarıdır.

Not: bu algoritmanın daha pahalı bir versiyonu da teklif adımındaki geçmiş katılımcıların kaydını tutar.

❒ **Maksimum olabilirlik** – Yerel koşullu dağılımları bilmiyorsak, maksimum olasılık kullanarak bunları öğrenebiliriz.

$$\boxed{\max_\theta\prod_{x\in\mathcal{D}_{\textrm{train}}}p(X=x;\theta)}$$

❒ **Laplace yumuşatma** – Her d dağılımı ve $(x_{\textrm{Parents}(i)},x_i)$ kısmi ataması için, $\textrm{count}_d(x_{\textrm{Parents}(i)},x_i)$'a λ ekleyin, ardından olasılık tahminlerini almak için normalleştirin.

❒ **Beklenti-Maksimizasyon** – Beklenti-Maksimizasyon (EM, expectation-maximization) algoritması, olasılığa art arda bir alt sınır oluşturarak (E-adım) tekrarlayarak ve bu alt sınırın (M-adımını) optimize ederek θ parametresini maksimum olasılık tahmini ile tahmin etmede aşağıdaki gibi etkin bir yöntem sunar:

- E-adım: her bir e veri noktasının belirli bir h kümesinden geldiği gerideki $q(h)$ durumunu şu şekilde değerlendirin:

$$\boxed{q(h)=P(H=h|E=e;\theta)}$$

- M-adım: maksimum olasılığını belirlemek için e veri noktalarındaki küme özgül ağırlıkları olarak gerideki olasılıklar $q(h)$ kullanın.