Nauczyliśmy się podstaw Łańcuchy Markowa i metody Monte Carlo. Dzisiaj poznamy konkretny model ale za nim omówimy structured probalistic models czyli ujęcie sieci jako teorii grafów czyli węzłów i brzegów czyli połączeń węzłów- modele graficzne.

Directed Models czyli ukierunkowane mamy wiedze o zależności między elementami. Modele te są stosowane gdy wiem że wartość w węzłach zależą od jednostronnie od innych węzłów. Czyli że tak w jedną stronę i mamy pewność że nie w obie a->b->c. Liczymy tutaj prawdopodobieństwa oraz we wzorach mamy warunkowe bo b|a c|b.

Undirected Models czyli modele nieukierunkowane inaczej nazywane sieciami Markowa sygnał może przepływać w obu kierunkach a-b-c. Musimy zrezygnować z liczenia prawdopodobieństwa.

Klika to podzbiór bezpośrednio połączonych ze sobą węzłów (a,b) i (b,c). w przykładzie (v1,h1), (v1,h2,h3) ale nie będzie klikom (v1,v2) czy (v1,h1,v2) bo v nie są połączone ze sobą.

Maksymalna klika nie można dołożyć żadnego węzła z grafu czyli ab i bc . w drugim przypadku v1h1h2h3 to jedna z maksymalnych klik.

Potencjał kliki funkcja przypisana każdej kulce(klika nawet maksymalnych) grafu która modeluj lokalne zależności między zmiennymi w tej kulce(klice)

Przykład\_1

Zakładamy że może przyjmować wartości binarne, mamy graf i są dwie kliki, funkcja potencjału zdefiniowane na maksymalnych kilkach, Unnormalized probability distribution, Czynnik normalizujący.

Przykład\_2

Sieć społecznościowa, rozkład poparcia dla danej idei 0 lub 1, definiujemy na dwóch wartościach nie na maksymalnych klikach. Mamy funkcje potencjału i liczymy rozkład prawdopodobieństwa.

Lepiej się dodaje niż mnoży więc wyprowadzono wzór tak o normalnie każdy go zapamięta. To co w mianowniku to energia sieci. Więc prawdopodobieństwo sprowadza się do energii wydzielonej przez sumę wszystkich możliwości. Rozkład z tym e powinien się kojarzyć z soft max i rozkładem Boltzmanna

Ograniczona Maszyna Boltzmana

Prosta dwuwarstwowa sieć neuronowa wyróżniamy w niej warstwę widoczną oraz warstwę ukrytą. Na widoczną podajemy wartości a ukryta ma generować pewną reprezentację danych wektor lepszy niż oryginalny. W oryginalnych maszynach boltzamana neurony mogą przyjmować tylko dwie wartości 0 albo 1, na połączeniach są klasycznie wagi. Połączenia działają dwukierunkowo, wyróżnia się tym że maja probabilistyczny charakter to czy w neuronie pojawi się 0 lub 1 nie jest kwestią neuronu tylko losowane z rozkładu prawdopodobieństwa. Wartość leży między 0 a 1 czyli dwa razy podając ten sam wektor mamy różne rezultaty

Przykład SR w warstwie widocznej mamy wszystkie filmy użytkownik mógł się podobać lub nie albo zobaczyć albo nie i oznaczamy 1 lub 0, powstaje taki wektor wejściowy 111000 możemy stwierdzić że Alice jest fanką SF/fantasy i tak dalej możemy wrzucić w kategorię. Sieć nauczyła się grupować wydobywać ukryte cechy z filmów.

Jak to działa-> podajemy wejścia sieci i na tej podstawie obliczamy prawdopodobieństwa aktywacji neuronów i aktywujemy lub nie dany neuron a potem faza w dół czyli bierzmy 0 i 1 i aktywujemy neurony warstwy widocznej. Dostaniemy wynik podobny do naszych danych ale może się różnić co pozwoli na uzupełnienie luk.

Kryterium uczenia-> maksymalizacja funkcji wiarygodności. Jeżeli maksymalizujemy to wynik jest taki sam jak maksymalizujemy logarytm z funkcji wiarygodności, a jeśli mamy logarytm iloczynów otrzymujemy sumę logarytmów więc możemy maksymalizować sumę logarytmów prawdopodobieństw. Jeżeli znalibyśmy prawdziwy rozkład to byłaby to wartość oczekiwana prawdopodobieństwa pod warunkiem q które znalibyśmy. Prowadzi to do przekształceń do minimum z sumy KL ponoć było już ale chyba kolejność wykładów jest zła.

Uczenie RBM-> punktem wyjścia jest prawdopodobieństwo aktywacji a dokładnie rozkład i rozważmy sobie prawdopodobieństwo aktywacji konkretnego neuronu warstwy widocznej. Na tą chwilę ograniczamy się że zbiór uczący składa się z jednego elementu. Chcąc policzyć pochodną po konkretnym elemencie to rozważymy logarytm z l nie liczymy samych ekstremów. Gradient funkcji wiarygodności generalnie liczymy gradienty pochodne, teraz liczymy pochodne funkcji energii, jeżeli weźmiemy jakiś mini batch to są inne wzory.

Alternatywne metody uczenia

Największą wadą próbkowania Gibbsa jest fakt że długo musimy czekać na stan stacjonarny, najpopularniejszą metodą ulepszającą Contrastive divergence.

Rozważamy minibatch, bierzemy pierwszy element, obliczamy aktywację warstwy ukrytej i zapamiętujemy to jako E[vihj] i wykonujemy to k razy (góra dół w zasadzie naokrągło) i otrzymujemy nowe wartości i zapamiętujemy ich iloczyn jak E[vihk]recon. Pomaga to w obliczaniu gradientu w tej metodzie mamy element zależny od danych i to prawdopodobieństwo zależne od wygenerowanego elementu widocznego. Im wyższe k tym lepsze przybliżenie ale w praktyce wystarczy nawet k=1. Aktualizacja wag-> całe wartości można zastąpić tymi elementami obliczonymi

Pewną modyfikacją dla CD jest Persistent CD-> zamiast losować od tego samego punktu tworzymy sobie R łańcuchów Markowa i wykonujemy dla każdego k Gibbs sampling. Różnica jest taka że nie musimy stosować tego od pczątku jeżeli łańcuchy osiągnęły stan stabilny to git, chodzi o to że nie musimy zaczynać od początku.

Parallel tempering-> ustalamy ilość łańcuchów zależne jest to od temperatury i dla każdej z osobna tworzymy rozkład inicjalizujemy R łańcuchów i próbkowanie Gibbsa. Możemy wymienić stan o wyższej temperaturze ale robimy to tylko z jakimś prawdopodobieństwem.

Poza klasyczny RBM

Deep belief network -> nakładanie kolejnej warstwy i uczenie tych ostatnich połączeń

Gaaussian-Bernoulli RBM -> chodzi o rozkład w warstwie ukrytej

Urestricted Boltzmann Machines -> dwa neurony są ze sobą skorelowane w jednej warstwie