Monte Carlo, Entegraller, MCMC

Fizik, biyoloji ve ozellikle makina ogrenimi problemlerinde bazen cok boyutlu bir fonksiyon uzerinden entegral almak gerekebiliyor. En basit ornek, mesela bir dagilimin baska bir fonksiyon ile carpiminin beklentisini (expectation) hesaplamak gerektiginde, ki bu

$$E(f) = \int p(x)f(x)dx$$

entegralidir, $x \in \Re^n$, $p(x)$ dagilim fonksiyonu, $f(x)$ herhangi bir baska fonksiyon olmak uzere, o zaman tum x degerlerini teker teker gecerek entegral hesabini yapmak gerekecekti.

Fakat $p(x)$ bir dagilim olduguna gore, bizim gectigimiz her x icin bir olasilik degeri var ise, bu isi tersine cevirerek, $p(x)$'teki olasiliklara gore belli (az) sayida x urettirirsek, ve sadece bu x'leri entegral hesabinda kullanirsak yaklasiksal acidan gercek entegral hesabina yaklasmis oluruz. Bu mantikli degil mi? Dusunursek, mesela 10 degeri 0.4 olasiliginda ise, 5 degeri 0.1 olasiliginda ise, hem sayi, hem olasiligi ile carpmak yerine "daha fazla 10 degeri uretmek" ve bu degerleri f'e gecmek, toplamak, sonra bolmek, vs. yaklasiksal olarak ayni kapiya cikar. Yani

$$E_N = \frac{1}{N} \sum_{1=1}^{N} f(x^{(i)})$$

ustteki entegralin yaklasiksal temsilidir, $x^{(i)}$ $p(x)$ olasiligina gore uretilen sayilari temsil ediyor.

Iste Monte Carlo entegral hesabinin artasinda yatan numara budur.

MCMC

Demek ki Monte Carlo entegralinin islemesi icin $p(x)$'den ornekleme yapmak gerekiyor. Simdi ikinci numaraya gelelim. Bazen ne yazik ki $p(x)$'den ornekleme yapmak kolay olmuyor. Bu durumlar icin $p(x)$ yerine onu yaklasiksal olarak temsil eden bir $\pi(x)$'i elde etmekle ugrasiliyor. Bu $\pi(x)$ ise bir Markov Zincirinin (Markov Chain -yine MC harfleri!-) duragan dagilimi olarak hayal ediliyor. Niye? Cunku Markov Zincirlerinin duragan dagiliminin hesabi icin gecis olasiliklarinin surekli birbiriyle carpilmasi, burada kullanilan ogeler, ustteki Monte Carlo entegralini olustaran ogeler kullanilarak elde edilebilir. Bu pek bir anlam ifade etmiyorsa, Markov Zincirleri konusuna bakmanizi tavsiye ederim. Klasik Markov Zinciri problemleri ile bizim ihtiyaclarimizin (MC entegrali) farki surada: Markov Zinciri teorisinde bir gecis matrisi, yan Markov Zincirinin kendisi verilir, ve duragan dagilimin hesaplanmasi istenir. MCMC problemlerinde, yani Monte Carlo entegrali icin Markov Zinciri kullanildigi durumlarda elimizde bir $\pi(x)$ dagilimi vardir ve bir Markov Zinciri olusturmamiz gerekir.

Nihai dagilimi biliriz, ve bu dagilima "giden" gecisleri uretiriz. Bu uretim sirasinda bir yandan kenarda entegralimizi hesaplariz, ve sonuca vardigimizda hem nihai, duragan dagilima erismisizdir, hem de entegral hesabimizi dogru bir sekilde yapmis oluruz.

Gecisleri uretmek icin literaturde bir suru teknik oldugunu gorebilirsiniz, Onemsel Ornekleme (Importance Sampling), Ornekleme ve Oneme Gore Tekrar Ornekleme (Sampling Importance Resampling), Metropolis-Hastings, Gibbs Orneklemesi gibi teknikleri vardir, ve detaylari degisik olsa da hepsi de MCMC kategorisine girer, ve yapmaya calistiklari $\pi(x)$'e giderken bir sekilde bir gecisleri, zinciri ortaya cikartmak ve bu gecisleri entegral hesabinda kullanmaktir.

Ustteki tekniklerden en yaygin kullanilani Metropolis-Hastings algoritmasidir.

Not: Bu alandaki makalelerde bir dagilimin "belli bir carpimsal sabite kadar" bilindigi (known up to a multiplicative constant) soylenir. Bu soz aslinda su anlama gelir. Mesela ayriksal bir dagilimimiz var, ama bu dagilimin kendisini, su halini biliyoruz

```
[ 4.3  2.   8.4  8.7  1.8]
```

Bu bir dagilim degil, cunku ogelerin toplami 1 degil. Onu bir dagilim haline cevirmek icin, tum ogeleri toplamak ve bu vektordeki tum sayilari bu toplam ile bolmek gerekir. Toplam 25.2, bolersek

```
[ 0.17063492  0.07936508  0.33333333  0.3452381   0.07142857]
```

Ilk vektor "belli bir carpimsal sabite kadar" bilinen dagilim, carpimsal sabit 25.2. Esas dagilim ikinci vektor.

Peki niye bu sozu soyleyenler toplami hesaplayip gercek dagilimi hesaplamiyorlar? Sebep performans. Bazen ayriksal dagilim o kadar yuksek boyutlu, fazla oge iceren bir halde oluyor ki, performans acisindan bu basit toplam hesabini yapmak bile cok pahali oluyor. Iste MCMC metotlarinin bir guzel tarafi daha burada, dagilimin kendisi olmasa bile belli bir carpimsal sabite kadar bilinen versiyonlari ile gayet rahat bir sekilde isliyorlar.

Kaynaklar

Algorithmic Machine Learning, Stephen Marsland