Additional Material for Machine Learning

2018/05/05

강준하

Contents

Reference

- [Jaejun Yoo] Minimizing the Negative Log-Likelihood, in Korean
  - http://jaejunyoo.blogspot.com/2018/02/minimizing-negative-log-likelihood-in-kor.html
- [Sanghyuk Chun] Machine Learning Study (6) Information Theory
  - http://sanghyukchun.github.io/62/
- Ian Goodfellow, Deep Learning, 2016
  - http://www.deeplearningbook.org/

1. Some Information Theory

Entropy

- 정보의 단위
- 불확실성이 크고 정보량이 많으면 커짐
- 정의 : $H(x) = -\sum_x p(x) \log_2 p(x)$
- $p(x) \to 0$ 이면 $\log_2 p(x) \to -\infty$ 이긴 한데, $p(x)$의 수렴 속도가 더 빨라서 엔트로피는 0이 된다.

Entropy

- Example

```
p = {'rain': .14, 'snow': .37, 'sleet': .03, 'hail': .46}

def entropy(prob_dist):
    return -sum([ p*log(p) for p in prob_dist.values() ])
```

Definition

```
In [1]: entropy(p)
Out[1]: 1.1055291211185652
```

```
p_2 = {'rain': .01, 'snow': .37, 'sleet': .03, 'hail': .59}

p_3 = {'rain': .01, 'snow': .01, 'sleet': .03, 'hail': .95}
```

Strongly Determined World…

```
In [2]: entropy(p_2)
Out[2]: 0.8304250977453105

In [3]: entropy(p_3)
Out[3]: 0.2460287703075343
```

Low entropy!

Entropy

- 왜 $H(x) = -\sum_x p(x) \log_2 p(x)$ 이런 정의?
- 이 이전에 self-information $I(x) = -\log p(x)$ 정의…
- 다음 property를 가지도록 식 설계됨
  - 자주 나타나는 event는 정보량이 적고, 가끔 나타나는 event는 정보량이 많음
  - Independent events는 추가 정보 가짐

    ex) 코인 2회 던지는 시행의 정보량은 1회 던지는 시행의 정보량의 2배여야 함
- 이 self-information 기반으로 Shannon entropy 정의하게 된다.

Entropy

- 아까 entropy 정의 : $H(x) = -\sum_x p(x) \log_2 p(x)$
- 어디선가 많이 본 식인데…? 기대값!
- $E_{X\sim p}[I(x)] = E_{X\sim p}[-\log p(x)] = -\sum_x p(x) \log p(x)$

(위에 기대값 표기는 $p(x)$ 분포 하에서 기대값 계산했음을 의미)

분포가 고르면	분포가 고르지 않으면
확신이 잘 들지 않는다면	확신이 잘 든다면
값 커짐(분포가 가진 정보가 많음)	값 작아짐(분포가 가진 정보가 적음)

Entropy

- 요약하자면!
- 가질 정보의 기대값을 계산할건데, 기대값이 크다는 것은 정보를 많이 가지고 있을 확률이 높다는 의미

  (문장이 꼬였는데 기대값의 정의에 맞춰서 생각을 해본다면…)

Entropy Derivation (The Wallis derivation)

- N개의 object, K개의 bin(category)
- i번째 bin에 들어갈 수 있는 object의 개수 n_i $(i = 1, \cdots, k)$

→ Object들이 bin에 들어가는 permutation 개수는 $W = \frac{N!}{\prod_i n_i!}$

- 이걸 multiplicity(불확실성, 정보량을 의미)라고 함
- 정보량...? Entropy는 이 multiplicity의 log!

Entropy Derivation (The Wallis derivation)

$$H = \frac{1}{N}\log W = \frac{1}{N}\log N! - \frac{1}{N}\sum_i \log n_i!$$

at $N \to \infty, \ln N! \approx N \ln N - N$ (stirling's approximation)

$$= \lim_{n\to\infty}\left(\frac{1}{N}(N\ln N - N) - \frac{1}{N}\sum_i (n_i \ln n_i - n_i)\right)$$

$$= \lim_{n\to\infty}\left(\ln N - \sum_i \left(\frac{n_i}{N}\ln n_i - \frac{n_i}{N}\right) - 1\right)$$

Entropy Derivation (The Wallis derivation)

$$= \lim_{n\to\infty}\left(\sum_i \left(\frac{n_i}{N}\ln N - \frac{n_i}{N}\ln n_i\right) + \sum_i \frac{n_i}{N} - 1\right)$$

$$= -\lim_{n\to\infty}\sum_i \frac{n_i}{N}\ln\frac{n_i}{N}$$

$= -\sum_i p_i \ln p_i$ (p_i : i번째 bin에 공이 들어갈 확률)

→ Entropy란 주어진 bin에 얼마나 비슷한 수의 element가 들어가는지 측정하는 척도가 될 수 있다... (정보량!)

Differential entropy

- 이제까지 discrete한 random variable에 대한 엔트로피
- Continuous하다면? Differential entropy를 정의해서 이용
- By mean value theorem of integral…
- $\int_{i\Delta}^{(i+1)\Delta} p(x)dx = p(x_i)\Delta$ 인 x_i 반드시 존재
- $H_\Delta = -\sum_i p(x_i)\Delta \ln(p(x_i)\Delta)$ 에서 Δ을 0으로 보내면…

Differential entropy

$$\lim_{\Delta\to 0} H_\Delta = \lim_{\Delta\to 0}\left(-\sum_i p(x_i)\Delta \ln(p(x_i)\Delta)\right) = -\int p(x)\ln p(x)dx$$

$$\left(= E_{X\sim p}[-\ln p(x)]\right)$$

$$\therefore H(x) = -\int p(x)\ln p(x)dx$$

잠시 첨언

- 로그의 밑을 자꾸 제멋대로 쓰는 중 (2, e)
- 2 : coding theory (bit), 컴퓨터는 이진법 사용해서
- e : machine learning (logit), 미분/적분 계산 편하게 할려고

Conditional Entropy

- Joint Entropy : $H(x, y) = -\iint p(x, y) \ln p(x, y) dxdy$
- Conditional Entropy -> 확률 정의에 의해 정의 가능
- $H(Y|X) = \sum_x p(x) H(Y|X = x)$
- $H(Y|X) = H(X, Y) - H(X)$

KL divergence

- 두 probability distribution $p(x)$와 $p(y)$의 거리를 Measure 할 수 없을까?
- 왜? 알려지지 않는 probability distribution이 있을 때 우리가 추론한 probability distribution이 얼마나 잘 추론했는지 판단하기 위한 근거 필요함
- 얼마나 차이가 나는지를 measure할 수 없을까? (두 probability distribution 간의 차이를 양적으로 표현하려는 시도)

KL divergence

- $p(x)$: original probability distribution (unknown)
- $q(x)$: our inference about unknown probability distribution

$$KL(p||q) = \underbrace{-\int p(x)\ln q(x)\,dx}_{\text{Cross Entropy}} - \left(-\int p(x)\ln p(x)\,dx\right)$$

$$= -\int p(x)\ln\frac{q(x)}{p(x)}dx$$

두 기대값의 차이 (같은 분포 $p(x)$ 하에서)

KL divergence

- 두 기대값의 차이 (같은 분포 $p(x)$ 하에서)
- 엄밀하게 distance 라고 할 수는 없음…

  $(KL(p||q) \neq KL(q||p)$ 이므로)

Mutual Information

- 서로 다른 두 random variable이 얼마나 mutual dependent한지 measure하고 싶음
- x, y가 independent하면 $p(x, y) = p(x)p(y)$가 성립

→ Dependent한 경우의 $p(x, y)$의 true distribution과 Independent하다고 가정했을 경우의 $p(x)p(y)$의 distribution 간의 KL Divergence

→ 얼마나 mutual하게 information을 많이 가지고 있는가를 measure하는 척도

Mutual Information

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

$$= -\iint p(x,y)\ln\frac{p(x)p(y)}{p(x,y)}dxdy$$

$$= H(x) + H(y) - H(x,y)$$

$$= H(x) - H(x|y) = H(y) - H(y|x)$$

2. Minimizing the Negative Log-Likelihood

TODO

- Why we use identity, sigmoid, softmax function as **output function**?
- How to make **loss function**
- **Regularization** term motivation

Initial model

- 세상에는 수많은 random variables가 있음

Ex) temperature, cat or dog, red or green or blue

- 각 random variable들의 실제 probability distribution이 어떻게 생긴지 모름
- 전반적 형태와 형태를 제어하는 parameter들 모름
- 초기 모델을 정해서 추정해보자

Initial model

- 초기 모델 어떻게 정할까
- Temperature : true mean - μ, true variance - σ^2 를 가진

  $x \in (-\infty, \infty)$
- Cat or Dog : 고양이/강아지를 값으로 가짐, 각 결과에 대한 likelihood는 변하지 않음
- Red or Green or Blue : 빨강/초록/파랑을 값으로 가짐, 각 결과에 대한 likelihood는 변하지 않음

Initial model

- 최대한 보수적으로 선택, 가지고 있는 값이 완전하게 probability distribution을 나타내고 있지 않다고 가정
- 동일한 가정이라면 두 random variables 모두 같은 probability distribution 모양 가져야 할 필요 있음
- Maximum Entropy Probability Distribution을 사용하자!

Why Maximum Entropy Probability Distribution?

- 분포가 갖는 entropy값이 해당 class의 probability distribution 들이 가질 수 있는 최대 entropy값과 최소한 같거나 크다.
- 무슨 소리??
- 만약 우리가 어떤 모델을 세울 때 해당 데이터에 알고 있는 정보가 적다면 잘못된 선험적 정보를 부지불식 간에 모델에 넣지 않도록 주의를 기울여야 한다는 것
- 추정할 probability distribution의 entropy를 Maximum으로 놓자

Why Maximum Entropy Probability Distribution?

- (Principle of maximum entropy에 의해) 모양을 정할 때 해당 데이터가 어떤 class에 속한다는 정보 외에 distrubution에 대한 어떠한 정보도 없을 때는 **가장 기본적으로 최소한의 정보만을 사용하여 distribution을 정해야 함**
- 최소한의 정보 → Entropy 최소?
- 우리가 가진 정보량이 아무것도 아니라고 가정 → 정보량이 적다고 가정 → Entropy 최소라고 가정
- Distribution의 Entropy를 최대로 만드는 가정

Why Maximum Entropy Probability Distribution?

- 여기에 해당하는 분포가 Maximum Entropy Probability Distribution!
- 이외에도 많은 physical system이 시간이 지나면 점차 maximum entropy configuration을 향함

→ maximum entropy probability distribution을 가정하는 것이 좋음

Initial model example

- Temperature – Gaussian Distribution

$$P(y \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}$$

- Cat or Dog – Binomial Distribution

$$P(outcome) = \begin{cases} 1-\phi\,, & outcome = cat \\ \phi, & outcome = dog \end{cases}$$

Initial model example

- Red or Green or Blue – Multinomial Distribution

$$P(outcome) = \begin{cases} \phi_{red}, & outcome = red \\ \phi_{green}, & outcome = green \\ 1 - \phi_{red} - \phi_{green}, & outcome = blue \end{cases}$$

- 유도? Lagrange Multipliers 통해서

Functional form

- Gaussian, Binomial, Multinomial distribution 셋 다 같은 functional form으로 나타낼 수 있음
- 이 common functional form에서 세 모델들의 output function(identity, sigmoid, softmax)가 자연스럽게 유도됨

"Exponential Family" distributions

- 고전적인 activation과 loss function을 하나의 틀에서 유도하는 데 매우 좋은 도구
- Mathematical convenience, on account of same useful algebraic properties, etc. (Wikipedia)
- $P(y;\, \eta) = b(y)e^{\eta^T T(y) - a(\eta)}$

“Exponential Family” distributions

$$P(y; \eta) = b(y)\exp(\eta^T T(y) - a(\eta))$$

- η : 분포의 canonical parameter (기준이 되는 매개변수)
- $T(y)$: sufficient statistic (대부분 $T(y) = y$)
- $a(\eta)$: log partition function, 분포를 정규화하는데 사용됨

- $T(y)$, $a(\eta)$, $b(y)$ 정하면 분포의 family 정해지고 η으로 parameterized 됨

Example – Gaussian Dist. (Temperature)

- $\sigma^2 = 1$이라 가정 (이 family는 단일 매개변수만을 다뤄서…)

$$P(y|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right)$$

$$= \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(y-\mu)^2}{2}\right)$$

$$= \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}(y^2 - 2y\mu + \mu^2)\right)$$

Example – Gaussian Dist. (Temperature)

$$= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y^2\right) \exp\left(\mu y - \frac{1}{2}\mu^2\right)$$

$$\boxed{P(y;\, \eta) = b(y)\exp(\eta^T T(y) - a(\eta))}$$

- $\eta = \mu \rightarrow \boxed{\mu = \eta}$
- $T(y) = y$
- $a(\eta) = \frac{1}{2}\mu^2 = \frac{1}{2}\eta^2$
- $b(y) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y^2\right)$

Example – Binomial Dist. (Cat or Dog)

$$P(y|\phi) = \phi^y(1-\phi)^{1-y}$$

$$= \exp(\log(\phi^y(1-\phi)^{1-y}))$$

$$= \exp(y\log\phi + \log(1-\phi) - y\log(1-\phi))$$

$$= \exp\left(\log\left(\frac{\phi}{1-\phi}\right)y + \log(1-\phi)\right)$$

Example – Binomial Dist. (Cat or Dog)

$$= \exp\left(\log\left(\frac{\phi}{1-\phi}\right)y + \log(1-\phi)\right)$$

$$\boxed{P(y;\,\eta) = b(y)\exp(\eta^T T(y) - a(\eta))}$$

- $\eta = \log\left(\frac{\phi}{1-\phi}\right)$
- $T(y) = y$
- $a(\eta) = -\log(1-\phi)$
- $b(y) = 1$

Example – Binomial Dist. (Cat or Dog)

- $\eta = \log\left(\frac{\phi}{1-\phi}\right)$

  $\rightarrow \boxed{\phi = \frac{1}{1+e^{-\eta}}}$ (sigmoid function!)

- $a(\eta) = -\log(1-\phi) = -\log\left(1 - \frac{1}{1+e^{-\eta}}\right) = \log(1 + e^{-\eta})$

Example – Multinomial Dist. (Color)

- π : K classes에서 각 class가 될 확률들의 vector, k는 각 class
- Color의 예시에서는 $\pi = [\phi_{red} \quad \phi_{green} \quad 1 - \phi_{red} - \phi_{green}]^t$

$$P(y|\pi) = \prod_{k=1}^{K} \pi_k^{y_k}$$

$$= \exp\left[\log\left(\prod_{k=1}^{K} \pi_k^{y_k}\right)\right] = \exp\left(\sum_{k=1}^{K} y_k \log \pi_k\right)$$

Example – Multinomial Dist. (Color)

$$= \exp\left[\sum_{k=1}^{K-1} y_k \log \pi_k + \underbrace{\left(1 - \sum_{k=1}^{K-1} y_k\right)}_{y_K} \log \underbrace{\left(1 - \sum_{k=1}^{K-1} \pi_k\right)}_{\pi_K}\right]$$

$$= \exp\left(\sum_{k=1}^{K-1} y_k \log \pi_k + \log \pi_K - \sum_{k=1}^{K-1} y_k \log \pi_K\right)$$

$$= \exp\left[\sum_{k=1}^{K-1} y_k \log \frac{\pi_k}{\pi_K} - \log \pi_K\right]$$

Example – Multinomial Dist. (Color)

$$= \exp\left[\sum_{k=1}^{K-1} y_k \log\frac{\pi_k}{\pi_K} - \log\pi_K\right]$$

$$\boxed{P(y;\,\eta) = b(y)\exp(\eta^T T(y) - a(\eta))}$$

- $\eta_k = \log\frac{\pi_k}{\pi_K}$
- $T(y) = y$
- $a(\eta) = -\log\pi_K$
- $b(y) = 1$

Example – Multinomial Dist. (Color)

$$\eta_k = \log \frac{\pi_k}{\pi_K} \Leftrightarrow \frac{\pi_k}{\pi_K} = e^{\eta_k}$$

$$\Leftrightarrow \sum_{k=1}^{K} \frac{\pi_k}{\pi_K} = \sum_{k=1}^{K} e^{\eta_k} \Leftrightarrow \frac{1}{\pi_K} \underbrace{\sum_{k=1}^{K} \pi_k}_{= 1} = \sum_{k=1}^{K} e^{\eta_k}$$

$$\Leftrightarrow \pi_K = \frac{1}{\sum_{k=1}^{K} e^{\eta_k}}$$

$$\boxed{\pi_k = \frac{e^{\eta_k}}{\sum_{k=1}^{K} e^{\eta_k}}}$$

(softmax function!)

Example – Multinomial Dist. (Color)

$$a(\eta) = -\log \pi_K$$

$$= \log \pi_K^{-1}$$

$$= \log \sum_{k=1}^{K} e^{\eta_k}$$

Summary

- 각 모델에서 우리가 관심있는 response variable들을 η 에 대해 하나로 모아 정리해보면:
- Linear Regression (Gaussian Dist.) : $\mu = \eta$
- Logistic Regression (Binomial Dist.) : $\phi = \frac{1}{1+e^{-\eta}}$
- Softmax Regression (Multinomial Dist.) : $\pi_k = \frac{e^{\eta_k}}{\sum_{k=1}^{K} e^{\eta_k}}$
- η : data → μ, ϕ, π 계산 (데이터를 통해 분포 예측!)

Generalized Linear Models

- 각 모델은 output으로 response variable 뱉음
- 이 response variable은 어떤 exponential family dist.을 따름
- 이 분포의 canonical parameter η은 관측값, 매 관측마다 변화

Ex) cat or dog를 예측하는 logistic regression model

고양이 그림을 넣으면 $1-\phi \approx 1$이 나와야 함

개 그림을 넣으면 $\phi \approx 1$이 나와야 함

$$P(outcome) = \begin{cases} 1-\phi\ , outcome = cat \\ \phi, \quad\quad outcome = dog \end{cases}$$

Generalized Linear Models

- η 통해서 ϕ을 계산 → ϕ은 η을 parameter로 갖는 값
- 다른 input을 넣으면 나올 output probability가 그때그때 다르다는 얘기

- $y_i \sim \mathcal{N}(\mu_i, {\sigma_i}^2)$ 일 때, linear regression에서 μ_i란?
- $y_i \sim \mathcal{B}(\phi_i, 1)$ 일 때, logistic regression에서 ϕ_i란?
- $y_i \sim \mathcal{M}(\pi_i, 1)$ 일 때, softmax regression에서 π_i란?

Generalized Linear Models

- 주어진 모델에서 각 입력(input data)에 따라 해당하는 canonical parameter가 정해지고 이것이 response variable(our inference)의 분포에 영향을 미침
- Feature vector : input x
- How to x be canonical parameter η?

→ 가장 간단한 linear combination을 사용하곤 함

$$\eta = \theta^t x$$

Model Example – Linear Regression

$$\eta = \theta^t x = \mu$$

Model Example – Logistic Regression

$$\eta = \theta^t x = \log \frac{\phi_i}{1-\phi_i}$$

$$\phi_i = \frac{1}{1+e^{-\eta}} = \boxed{\frac{1}{1+e^{-\theta^t x}}}$$

$$\boxed{\phi = \frac{1}{1+e^{-\eta}}}$$

익숙한 식...?

Model Example – Softmax Regression

$$\eta_k = \theta^t x = \log \frac{\pi_k}{\pi_K}$$

$$\pi_k = \frac{e^{\eta_k}}{\sum_{k=1}^{K} e^{\eta_k}}$$

$$\pi_{k,\,i} = \frac{e^{\eta_k}}{\sum_{k=1}^{K} e^{\eta_k}} = \boxed{\frac{e^{\theta^t x}}{\sum_{k=1}^{K} e^{\eta_k}}}$$

익숙한 식...?

Why linear model?

- "모델 디자인" 혹은 "선택"의 문제
- Andrew Ng 교수님 曰…
  - 아마도 선형 조합이 canonical parameter에 대한 각 feature에 영향을 줄 수 있는 가장 쉬운 방법일 것이기 때문
  - 선형 조합이 단순한 x 뿐만 아니라 x에 대한 함수에 대해서도 η와 선형적으로 조합한다면 더 복잡한 형태로 구축 가능
  - $\eta = \theta^t \Phi(x)$와 같은 모델을 구축 가능하다는 의미

    (여기서 Φ는 우리의 feature에 복잡한 변형(transformation)을 주는 operator, 이를 통해 선형 조합의 단순함을 덜 수 있음)

Loss function

- 이제까지 한 것 : 각 response variable이 어떻게 만들어지는지, 분포들의 parameter가 각 input에 대해 어떻게 계산되는지
- 이제 남은 것 : 어떤 parameter가 좋은지 어떻게 알 수 있을까? 어떤 parameter를 선택했을 때, 이게 얼마나 좋은지를 measure 할 수 없을까?

Loss function

$$P(outcome) = \begin{cases} 1-\phi, & outcome = cat \\ \phi, & outcome = dog \end{cases}$$

- 다시 cat or dog example에서…
- 고양이를 넣으면 $\phi \approx 0$이 되도록 계산해야 함
- 이 계산 후, loss function이 우리가 얼마나 정확한 분포에 가까이 갔는지 정량화해줌(measure를 내뱉어준다)

MLE (Maximum Likelihood Estimation)

- μ, ϕ, π(parameter)가 주어지면 y가 나타날 probability distribution이 계산됨
- 그런데 이 y를 고정시켜 놓고 parameter가 바뀌도록 하면?

(y는 현실세계에서 label과 같은 것이므로 일종의 현실세계의 probability distribution을 나타낸다고 할 수도 있을 듯...)

- 같은 함수가 likelihood function이 됨

→ 고정된 y 값에 대하여 현재 parameter의 likelihood에 대해 알려주는 함수

MLE (Maximum Likelihood Estimation)

- 현재 우리가 갖고 있는 데이터가 가장 나올 법한 parameter를 고르고 싶음
- $\arg\max_{parameter} P(y|parameter)$
- y는 분포가 받는 parameter에 따라 변함
- 이 parameter는 η의 함수, $\eta = \theta^t x$

MLE (Maximum Likelihood Estimation)

- 관측된 데이터((x, y) 쌍)는 고정되어 있으므로, 우리가 바꿀 수 있는 부분은 θ밖에 없음
- 이에 맞게 MLE formalize하면 $\arg\max_\theta P(y|x;\theta)$
- 근데 Negative Log-Likelihood를 Minimize한다고 했던 거 같은데... 왜 Logarithm 이용?
- 확률함수 P는 $[0, 1]$ 사이의 값만 내뱉는데 이 값들을 여러 번 곱하면 값이 매우 빠르게 작아짐 → 이런 현상을 방지

MLE Example – Linear Regression

Gaussian Distribution

$$\begin{aligned}
\log P(y|x;\theta) &= \log \prod_{i=1}^{m} P(y^{(i)}|x^{(i)};\theta) \\
&= \sum_{i=1}^{m} \log P(y^{(i)}|x^{(i)};\theta) \\
&= \sum_{i=1}^{m} \log \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)}-\theta^T x^{(i)})^2}{2\sigma^2}\right) \\
&= \sum_{i=1}^{m} \log \frac{1}{\sqrt{2\pi}\sigma} + \sum_{i=1}^{m} \log\left(\exp\left(-\frac{(y^{(i)}-\theta^T x^{(i)})^2}{2\sigma^2}\right)\right) \\
&= m \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2\sigma^2} \sum_{i=1}^{m} (y^{(i)}-\theta^T x^{(i)})^2 \\
&= C_1 - C_2 \underline{\sum_{i=1}^{m} (y^{(i)}-\theta^T x^{(i)})^2}
\end{aligned}$$

$\theta^t x = \mu$

MLE Example – Logistic Regression

Binomial Distribution

$$
\begin{aligned}
-\log P(y|x;\theta) &= -\log \prod_{i=1}^{m} (\phi^{(i)})^{y^{(i)}} (1-\phi^{(i)})^{1-y^{(i)}} \\
&= -\sum_{i=1}^{m} \log \left((\phi^{(i)})^{y^{(i)}} (1-\phi^{(i)})^{1-y^{(i)}} \right) \\
&= -\underline{\sum_{i=1}^{m} y^{(i)} \log (\phi^{(i)}) + (1-y^{(i)}) \log (1-\phi^{(i)})}
\end{aligned}
$$

$$\phi_i = \frac{1}{1+e^{-\theta^t x}}$$

Minimizing the binary cross-entropy (i.e. binary log loss)

MLE Example – Softmax Regression

Binomial Distribution

$$-\log P(y|x;\theta) = -\log \prod_{i=1}^{m} \prod_{k=1}^{K} \pi_k^{y_k}$$

$$= -\sum_{i=1}^{m} \sum_{k=1}^{K} y_k \log \pi_k$$

Minimizing the categorical cross-entropy (i.e. multi-class log loss)

MAP(Maximum a posteriori estimation)

- 이제까지 한 것 : θ를 MLE 이용해 estimate
- 별다른 제약 없었음 → 나올 수 있는 범위가 매우 넓어짐(어떤 값이 나와도 받아들임)
- 실제로는 이런 가정이 너무 비현실적임(θ가 유한 범위 안에서 값을 갖길 바람)
- 또한 θ가 너무 커진다는 것은 overfitting을 의미

MAP(Maximum a posteriori estimation)

- 이를 위해 θ에 prior(선험적 지식? 우리의 사전적인 가정, 제약) 두게 됨
- MLE가 계산하는 것 : $\arg\max_{\theta} P(y|x;\theta)$
- MAP가 계산할 것 : $\arg\max_{\theta} P(y|x;\theta)\,P(\theta)$

MAP(Maximum a posteriori estimation)

- MLE 계산할 때처럼 prior와 함께 joint likelihood를 풀면

$$\theta_{MAP} = \arg\max_{\theta}\left[\log\prod_{i=1}^{m} P\left(y^{(i)}|x^{(i)};\ \theta\right)P(\theta)\right]$$

$$= \arg\max_{\theta}\left[\underbrace{\sum_{i=1}^{m}\log P\left(y^{(i)}|x^{(i)};\ \theta\right)}_{\text{MLE때 이미 form 도출함}} + \underbrace{\log P(\theta)}_{\text{이쪽 부분만 살펴보자}}\right]$$

MAP(Maximum a posteriori estimation)

- θ의 모든 항이 continuous-valued 실수 값이므로 평균 0과 분산 V를 갖는 Gaussian Distribution을 할당해 보자

$$\theta \sim \mathcal{N}(0, V)$$

$$\log P(\theta|0, V) = \log\left(\frac{1}{\sqrt{2\pi}V}\exp\left(-\frac{(\theta-0)^2}{2V^2}\right)\right)$$

$$= \log C_1 - \frac{\theta^2}{2V^2}$$

$$= \log C_1 - C_2\theta^2$$

MAP(Maximum a posteriori estimation)

$$\log C_1 - C_2\theta^2 \propto -C_2\theta^2$$

$$\propto C\|\theta\|_2^2$$

- L2 regularization!
- θ의 prior distribution을 변경하면 또다른 regularization 가능

  Ex) Laplace prior 주면 L1 regularization

Regularization

- Machine Learning에서 weight를 regularize한다는 것은 “no weight becomes too large”하겠다는 것
- y를 예측할 때 너무 큰 영향을 미치지 못하게 만드는 것
- 통계적인 관점에서 보면 prior항이 값을 주어진 범위 내에서 나오도록 제한하는 역할을 한다고 생각할 수 있음
- 이 범위가 scaling constant C로 표현되고, prior distribution 자체를 parameter화 함

Ex) L2 Regularization에서는 Gaussian dist.의 분산을 정함

Conclusion

- 모든 것이 완벽하다면…
  - θ에 대한 full distribution 계산
  - 이 분포의 값들과 새로운 관측값 x를 가지고 y를 계산할 수 있음

    Ex) 여기서 θ가 weights이므로 10-feature linear regression에서는 10개의 원소를 갖는 벡터가 됨 (신경망에서는 수백만이…)
  - 이로부터 가능한 모든 response y에 대한 full distribution을 얻을 수 있음

Conclusion

- 복잡한 시스템에서는 weights의 원소 개수가 매우 많기 때문에 위와 같이 함수 형태를 계산할 수 없음
- 따라서 fully Bayesian modeling에서는 이런 분포들을 보통 근사하여 사용하곤 함
- 전통적인 machine learning에서는 a single value (point estimate)를 할당하곤 함
- ~~썩 맘에 들지는 않음~~

Next Additional Things?

- GAN
- Dropout & Ensemble
- Restricted Boltzmann Machine
- (Paper Review) Ashia C. Wilson et al., "The Marginal Value of Adaptive Gradient Methods in Machine Learning", 2017, Advances in Neural Information Processing Systems