Chain Rule

Bayes Rule

Independence of RV

} read these

Expectation value of $f(x)$ w.r.t dist P

$f: \Omega \to \mathbb{R}$

Expectation value: most likely outcome of $f(x)$

$$E_P[f] = \sum_{x \in \Omega} p(x) f(x) \qquad \text{discrete}$$

$$= \int_\Omega pf \qquad \text{continuous}$$

E.g. $\Omega = \{\{i,j\} \mid 1 \leq i,j \leq 6\}$ two dice

$$f(\{i,j\}) = \begin{cases} i+j & i \neq j \\ 2i & i = j \end{cases}$$

$$E_P[f] = 7$$

Variance

$$\mathrm{Var}_p(f(x)) = E_p\left[\left(f(x) - E_p(f(x))\right)^2\right]$$

Fact: $\mathrm{Var}_p(x) = E_p[x^2] - E_p[x]^2$ Proof: exercise

Standard Deviation:

$$SD_p(f(x)) = \sqrt{\mathrm{Var}_p(f(x))}$$

(measure of clustering around mean)

Covariance:

$$\mathrm{Cov}_p(f(x), g(x)) = E_p\left[\left(f(x) - E_p(f(x))\right)\left(g(x) - E_p(g(x))\right)\right]$$

measures of linear relationship between f, g (also depends on scale of variables)

§3.9 Common Distributions

Bernoulli Distribution

RV x $\Omega = \{0, 1\}$ $\phi \in [0, 1]$

$p(x=0) = \phi$ $p(x=1) = 1 - \phi$

$$p(x) = \phi^x (1-\phi)^{1-x} \qquad x = 0, 1$$

Exer: $E_p[x] = \phi$

$\mathrm{Var}_p(x) = \phi(1-\phi)$

Multinoulli (Categorical)

RV x $\quad \Omega = \{1, \dots, k\} \quad k \in \mathbb{N}$

let $p_1, \dots, p_k \in [0,1]$ s.t. $\sum_i p_i = 1$

$p(x = i) = p_i$

Exer: Compute $E_p[x]$ and $\mathrm{Var}(x)$

Gaussian (Normal) Distribution

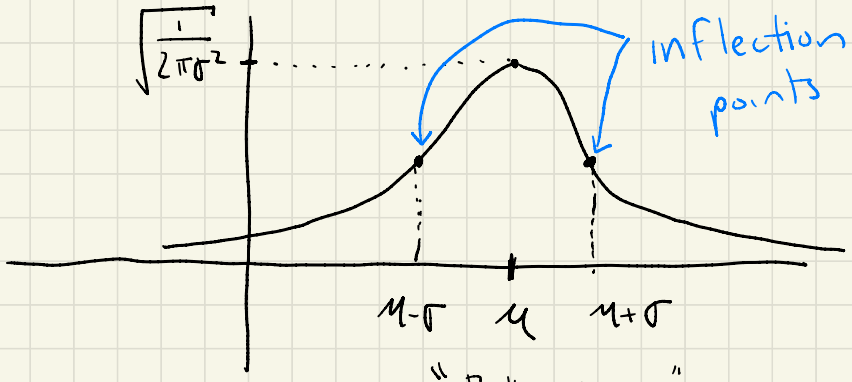
RV x $\quad \Omega = \mathbb{R}$ $\quad$ fix $\mu \in \mathbb{R}$ $\quad \sigma \in \mathbb{R}_{>0}$

$$p(x = x) = \mathcal{N}(x; \mu, \sigma^2) = \sqrt{\frac{1}{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$$

Exer: $E_p[x] = \mu$

$\mathrm{Var}_p(x) = \sigma^2$

$$\int_{\mathbb{R}} \mathcal{N}(x; \mu, \sigma^2)\, dx = 1$$



Def $\exp(x) = e^x$

Central Limit Theorem Sum of many independent random variables is approximately normally distributed.

E.g. 2-dice dist of $i+j$

Multivariate Gaussian

$\Omega = \mathbb{R}^n \quad \mu \in \mathbb{R}^n \quad \Sigma \in \mathbb{R}^{n \times n}$ pos def symmetric

$$\mathcal{N}(\vec{x}; \mu, \Sigma) = \sqrt{\frac{1}{(2\pi)^n \det(\Sigma)}} \exp\left(-\frac{1}{2}(\vec{x}-\mu)^T \Sigma^{-1} (\vec{x}-\mu)\right)$$

Σ covariance matrix (often diagonal)

Others: exponential, Laplace, dirac, empirical

(read in book)

Mixing Distributions

Given distributions labeled by $c = 1, \ldots, k$

and a categorical dist on $1, \ldots, k$

define $p(x) = \sum_i P(c=i) P(x \mid c=i)$

$P(c=i)$: Categorical; $P(x \mid c=i)$: distribution i

Common to mix Gaussian distributions

Chapter 4: Optimization / numerical computation

§4.1 Rounding error (overflow / underflow)

§4.2 Poor conditioning

matrix A $\quad Cond(A) = \max_{i,j} \left| \frac{\lambda_i}{\lambda_j} \right|$

if this is too big, you may run into numerical problems when computing with A.

§4.3 Gradient Based Optimization

In machine learning, often have

$$f: \mathbb{R}^n \to \mathbb{R} \quad (\text{or } f: U \to \mathbb{R} \quad U \subseteq \mathbb{R}^n)$$

and we want

$$x^* = \operatorname{argmin}_x f(x)$$

f: objective / criterion / cost / loss / error function

Recall Given $\hat{u} \in \mathbb{R}^n$ unit vector $(\hat{u}^T \hat{u} = 1)$

directional derivative

$$D_{\hat{u}} f(\vec{x}) = \lim_{h \to 0} \frac{f(\vec{x} + \hat{u}h) - f(\vec{x})}{h}$$

$$= \frac{\partial}{\partial \alpha} f(\vec{x} + \alpha \hat{u}) \Big|_{\alpha = 0}$$

$$\frac{\partial}{\partial x_i} f(\vec{x}) = D_{e_i} f(\vec{x}) \qquad e_i = (0 \dots \underset{\downarrow i}{1} \dots 0)^T$$

$$\nabla f(\vec{x}) = \left(\frac{\partial f}{\partial x_1}(\vec{x}) \quad \dots \quad \frac{\partial f}{\partial x_n}(\vec{x}) \right)^T$$

Question Which direction $\hat{u}$ maximizes

$$D_{\hat{u}} f(\vec{x}) ?$$

Answer: $D_{\hat{u}} f(\vec{x}) = \frac{\partial}{\partial \alpha} f(\vec{x} + \alpha \hat{u}) \Big|_{\alpha = 0}$

$$= \hat{u}^T \nabla_x f(\vec{x}) \quad \text{(chain rule)}$$

$$= \| \nabla_x f(\vec{x}) \| \cos\theta$$

maximized when $\hat{u} = \frac{\nabla_x f(\vec{x})}{\|\nabla_x f(\vec{x})\|}$

So $\nabla_x f(\vec{x})$ is the direction of max'l inc

$-\nabla_x f(\vec{x})$ " " " " max'l dec

Gradient Descent Algorithm

To find $x^* = \text{argmin } f(\vec{x})$ (hopefully)

Pick $\vec{x}$ randomly and choose parameters

$\alpha > 0$ (step size)

$\varepsilon > 0$ (stopping signal)

```
While True:
    prev = x⃗
    x⃗ = x⃗ - α∇_x f(x⃗)
    IF ||x⃗ - prev||_2 < ε:
        break
```

Note: May want to change value of α at each step, and play with value of ε.