Lecture 1, Jan 8, 2024

Events and Probability

- The *sample space* S is the set of all outcomes for an experiment
  - An outcome s is a member of the sample space: $s \in S$
- An *event* is a set of outcomes that satisfy a certain condition
  - Event A is a subset of S: $A \subseteq S$
- The *complement* of A is the set of all elements in S that are not in A, denoted A^c; note $A \cup A^C = S$ and $A \cap A^C = \varnothing$
- Given two events A and B, the event of either occurring is denoted $A \cup B$; both occurring is $A \cap B$
  - We can break $A \cup B$ into 3 parts: outcomes in A only, outcomes in both A and B, and outcomes in B only; these are mutually exclusive
  - $A \cup B = (A \cap B^C) \cup (A^C \cap B) \cup (A \cap B)$
- $C \subseteq A$ means event $C \implies A$
- Associate with each event a *probability*, satisfying the following:
  1. $P[A] \geq 0 \forall A$
  2. $P[S] = 1$
  3. $A \cap B = \varnothing \iff P[A \cup B] = P[A] + P[B]$
     - Note if A and B are not mutually exclusive then $P[A \cup B] = P[A] + P[B] - P[A \cap B]$

Conditional Probability and Bayes' Rule

- The probability of A *conditioned on* B is defined as $P[A|B] = \dfrac{P[A \cap B]}{P[B]}$ (assuming $P[B] \neq 0$)
  - $P[A|B]$ is known as the *a posteriori* probability
  - By symmetry, $P[B|A] = \dfrac{P[A \cap B]}{P[A]}$
- This gives the *product formula*: $P[A \cap B] = P[A|B]P[B] = P[B|A]P[A]$
- Events A and B are *independent* (denoted $A \perp B$) iff $P[A|B] = P[A]$, equivalently $P[A \cap B] = P[A]P[B]$

Partitioning

- A *partition* of S is sets $H_1, \dots, H_n$ such that $S = H_1 \cup H_2 \dots \cup H_n$ and $i \neq j \implies H_i \cap H_j = \varnothing$
- Since the H sets are mutually exclusive, $A \cap H_i$ are also mutually exclusive
  - Then $P[A] = P[A \cap H_1] + \cdots + P[A \cap H_n] = P[A|H_1]P[H_1] + \cdots + P[A|H_n]P[H_n]$
  - This is the *total probability theorem*
- Now we can find $P[H_i|A] = \dfrac{P[A \cap H_i]}{P[A]} = \dfrac{P[A|H_i]P[H_i]}{\sum_j P[A|H_j]P[H_j]}$
  - This is the *Bayesian* approach
  - The *frequentist* approach assumes no knowledge about the underlying $P[H_i]$ so we can only maximize $P[A|H_i]$

Lecture 2, Jan 12, 2024

Joint Random Variables

- A *random variable* is a function that assigns one or more numbers to the outcome of an experiment
  - Random numbers can be multi-dimensional: $\boldsymbol{X}\colon s \mapsto \mathbb{R}^2 \iff \boldsymbol{X}(s) = (X(s), Y(s))$
- The probability mass function is denoted $P[X = x_i, Y = y_i] = p_{\boldsymbol{X}}(x_i, y_i)$ for discrete random variables
  - Probability of a set/event is the sum of the PMF over the events
- The probability density function is denoted $P[x < X < x + \mathrm{d}x, y < Y < y + \mathrm{d}y] \approx f_{\boldsymbol{X}}(x, y)\,\mathrm{d}x\,\mathrm{d}y$ for continuous random variables
  - Probability of a set/event is the integral of the PDF over the continuous region that defines the event

  - Note we denote PMFs by p, PDFs by f
- *Marginal probabilities* can be computed as $p_Y(y_j) = \sum_j p_{\boldsymbol{X}}(x_i, y_j), p_X(x_i) = \sum_j p_{\boldsymbol{X}}(x_i, y_j)$ (discrete)
  - $f_X(x) = \int_{-\infty}^{\infty} f_{\boldsymbol{X}}(x, y')\,\mathrm{d}y', f_Y(y) = \int_{-\infty}^{\infty} f(x', y)\,\mathrm{d}x'$
  - In isolation the marginals don't have all the information that the joint PMF provides
- Conditional probabilities are given by $p_{Y|X}(y_j|x_i) = \dfrac{p_{\boldsymbol{X}}(x_i, y_j)}{p_X(x_i)}, f_{Y|X}(y|x) = \dfrac{f_{\boldsymbol{X}}(x, y)}{f_X(x)}$
  - The discrete version follows directly from the definitions
  - The continuous version requires a limiting procedure
  - Rearranging gives the *product rule*: $p_{\boldsymbol{X}}(x_i, y_j) = p_{Y|X}(y_j|x_i)p_X(x_i) = p_{X|Y}(x_i|y_j)p_Y(y_j)$ (same with continuous version)

Expectation, Mean and Variance

- The *expected value* of a function $Z = g(X, Y)$ is $E[Z] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x', y')f_{\boldsymbol{X}}(x', y')\,\mathrm{d}x'\,\mathrm{d}y'$
  - For a function dependent on only one of the variables, this is equivalent to integrating on the marginal
- The *mean* is simply $m_X = E[X] = \int_{-\infty}^{\infty} x' f_X(x')\,\mathrm{d}x'$
- The *variance* is defined as $\sigma_X^2 = \mathrm{Var}[X] = E[(X - E[X])^2]$
  - This is a measure of spread
  - Expanding this out gives $\sigma_X^2 = E[X^2] - (E[X])^2$
- $$\begin{aligned} E[g(Y)] &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(y')f_{Y|X}(y'|x')f_X(x')\,\mathrm{d}y\,\mathrm{d}x \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(y')f_{Y|X}(y'|x')\,\mathrm{d}y f_X(x')\,\mathrm{d}x \\ &= \int_{-\infty}^{\infty} E[g(Y)|X = x]f_X(x')\,\mathrm{d}x' \\ &= E[E[g(Y)|X]] \end{aligned}$$
  - In other words we can find the expectation of $g(Y)$ assuming X is known, and then find the expectation of that over X, to find the overall expectation of $g(Y)$
  - Special case: if $g(Y) = Y$ then $E[Y] = E[E[Y|X]]$
  - Example: picking X from a uniform $[0, 1]$, and then picking Y from a uniform $[0, x]$
    - $E[Y] = E[E[Y|X]] = E\left[\dfrac{X}{2}\right] = \dfrac{E[X]}{2} = \dfrac{1}{4}$
- The *covariance* of X and Y is defined as $\sigma_{XY} = \mathrm{Cov}(X, Y) = E[(X - E[X])(Y - E[Y])]$
  - If X and Y tend to vary positively together, the covariance is positive; if one varies positively while the other varies negatively,the covariance is negative; if there is no relation, the covariance is zero
  - Expanding gives $E[XY] - E[X]E[Y]$ ($E[XY]$ is known as the *correlation*)
  - Normalizing gives the *correlation coefficient* $\rho_{XY} = E\left[\left(\dfrac{X - m_X}{\sigma_x}\right)\left(\dfrac{Y - m_Y}{\sigma_Y}\right)\right] = \dfrac{\mathrm{Cov}(X, Y)}{\sigma_X\sigma_Y}$
  - X and Y are *uncorrelated* if $\rho_{XY} = 0 \iff \mathrm{Cov}(X, Y) = 0$ (note uncorrelated does not always imply independent)
  - Note covariance is bilinear (i.e. linear in each argument)
- X, Y are independent if $f_{\boldsymbol{X}}(x, y) = f_X(x)f_Y(y)$ or $p_{\boldsymbol{X}}(x_i, y_j) = p_X(x_i)p_Y(y_j)$
  - Independence means $E[g(X)h(Y)] = E[g(X)]E[h(Y)]$
  - This also means $\mathrm{Cov}(X, Y) = 0$ (i.e. independence implies uncorrelated)
  - $f_{X|Y}(x|y) = \dfrac{f_X(x)f_Y(y)}{f_Y(y)} = f_X(x)$
    - The *a posteriori* distribution is the same as the *a priori* distribution

        - i.e. knowing one does not give any information about the other

Lecture 3, Jan 15, 2024

Sum of Random Variables

- Let $S_n = \sum_{i=1}^{n} X_i$
- We can show that $E[S_n] = E[X_1 + \cdots + X_n] = \sum_{i=1}^{n} E[X_i]$
    - Note that although $E[S_n]$ is on the joint PDF of all the X random variables, $E[X_i]$ is on the marginal only, i.e. f_{X_i}
    - The expected value of a sum is always the sum of the expected values in all cases
- For variance: $\mathrm{Var}[S_n] = E[(S_n - E[S_n])^2]$

$$\begin{aligned}
&= E\left[\left(\sum_{i=1}^{n}(X_i - m_{X_i})\right)^2\right] \\
&= E\left[\left(\sum_{i=1}^{n}(X_i - m_{X_i})\right)\left(\sum_{j=1}^{n}(X_j - m_{X_j})\right)\right] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n}\mathrm{Cov}(X_i, X_j) \\
&= \sum_{i=1}^{n}\mathrm{Var}[X_i] + \sum_{i\neq j}\sum_{j}\mathrm{Cov}(X_i, X_j)
\end{aligned}$$

    - If all X_i, X_j are pairwise uncorrelated, then $\mathrm{Var}[S_n] = \sum_{i=1}^{n}\mathrm{Var}[X_i]$
    - But in general, the variance of a sum of RVs is not the sum of the variances
- Suppose that the Xs are *independent and identically distributed* (IID)
    - This means $f_{\mathbf{X}}(x_1, \dots, x_n) = f_{X_1}(x_1)\cdots f_{X_n}(x_n) = f_X(x_1)\cdots f_X(x_n) = \prod_{i=1}^{n} f_X(x_i)$
    - It follows that all the Xs will have the same mean m and variance σ^2
    - Therefore $E[S_n] = nm, \mathrm{Var}[S_n] = n\sigma^2$
- Let the *sample mean* be $M_n = \frac{1}{n}\sum_{i=1}^{n} X_i = \frac{1}{n}S_n$
    - Assuming IID:
        - $E[M_n] = E\left[\frac{1}{n}S_n\right] = \frac{1}{n}E[S_n] = m$
        - $\mathrm{Var}[M_n] = \mathrm{Var}\left[\frac{1}{n}S_n\right] = \frac{1}{n^2}\mathrm{Var}[S_n] = \frac{\sigma^2}{n}$
    - With increasing n, the expected value is unchanged but the variance decreases; this means to estimate $E[X]$, we can repeat the same experiment and take the mean to get a smaller variance in our results
- To formalize this, we can apply Chebyshev's inequality to the mean
    - $P[|X - m_X| \geq \epsilon] \leq \frac{\sigma_X^2}{\epsilon^2}$
    - Applied to the sample mean: $P[|M_n - E[X]| \geq \epsilon] \leq \frac{\mathrm{Var}[M_n]}{\epsilon^2} = \frac{\sigma^2}{n\epsilon^2} = 1 - \delta$
    - Given any error tolerance ϵ and probability $1 - \delta$, we can always select n such that the probability of M_n being within the tolerance of the true mean is $1 - \delta$ or greater

- This is also known as *convergence in probability*

Theorem

Chebyshev's Inequality:

$$P[|X - m_X| \geq \epsilon] \leq \frac{\sigma_X^2}{\epsilon^2}$$

Alternatively stated as

$$P[|X - m_X| \geq k\sigma] \leq \frac{1}{k^2}$$

Theorem

Weak Law of Large Numbers:

$$\lim_{n \to \infty} P[|M_n - E[X]| < \epsilon] = 1$$

That is, as the sample size N increases, the probability of the sample mean being within ϵ of the true mean approaches 1, where ϵ is any arbitrarily small positive number.

Strong Law of Large Numbers: Given IID X_i with finite mean,

$$P\left[\lim_{n \to \infty} |M_n - E[X]| < \epsilon\right] = 1$$

- SLLN asserts a much stronger form of convergence to $E[X]$
  - Notice that for SLLN the limit is outside the probability
  - The weak law states that for a certain value of n, most of the observed values of M_n will be close to $E[X]$
    - WLLN does not address what happens to the sample mean for a specific sequence of random variables
  - The strong law states that every sequence of sample mean calculations will eventually approach and stay close to $E[X]$
- Consider an event A and suppose we want to find $p = P[A]$
  - Let the *indicator function* for A be $I = \begin{cases} 1 & s \in A \\ 0 & s \notin A \end{cases}$
    - Note that $E[I] = 1 \cdot P[A] + 0 \cdot (1 - P[A]) = P[A] = p$
    - $\text{Var}[I] = E[(I - E[I])^2] = E[(I-p)^2] = (1-p)^2 p + (-p)^2(1-p) = p(1-p)$
  - Repeat the experiment n times so we have $S_n = I_1 + I_2 + \cdots + I_n$ equal to the number of times that A occurred
  - The relative frequency of A is $f_n = \dfrac{S_n}{n}$, so $E[f_n] = \dfrac{E[S_n]}{n} = p$
  - $\text{Var}[f_n] = \dfrac{\sigma^2}{n} = \dfrac{p(1-p)}{n} \leq \dfrac{1}{4n}$
    - But we don't know p, so instead we note $p(1-p)$ is bounded by $1/4$
    - Therefore $\text{Var}[f_n] \leq \dfrac{1}{4n}$
  - This gives us a way to estimate p while bounding the variance on our estimate
    - e.g. we want to be within $\dfrac{1}{10}$ of the true probability 90% of the time
      - Chebyshev: $P[|f_n - p| > \underbrace{0.1}_{\epsilon}] \leq \underbrace{0.1}_{\delta}$, then $0.1 = \dfrac{p(1-p)}{n_0\left(\frac{1}{10}\right)^2} \leq \dfrac{1}{4n_0\left(\frac{1}{10}\right)^2}$
      - Solve to get $n_0 > 250$

Introduction to Parameter Estimation

- Given an IID sequence of random variables, we want to estimate a parameter θ of the distribution X

    - The distribution depends on θ; it can be e.g. for Bernoulli it is $\theta = P[X = 1]$; for a Gaussian $\theta = (m_X, \sigma^2)$
    - $\hat{\Theta}_n$ is an estimator of the unknown parameter
- Note that the estimator is a function of the RVs, $\hat{\Theta}_n(\boldsymbol{X})$
- Estimators have the following properties:
    - The *error* is $\hat{\Theta}_n(\boldsymbol{X}) - \theta$
        - This is how much the estimate is off by from the true value
    - The *bias* is $E[\hat{\Theta}_n(\boldsymbol{X})] - \theta$
        - This is whether we get the correct estimate on average
    - An estimator is *unbiased* if the expected value of the error is zero, i.e. the bias is zero
        - i.e. on average, our estimate will be correct
    - An estimator is *asymptotically unbiased* if $\lim\limits_{n \to \infty} E[\Theta_n(\boldsymbol{X})] = \theta$, even if it's not unbiased
    - An estimator is *consistent* if as $n \to \infty$, the distribution of $\hat{\Theta}_n$ converges to θ (weak law)
        - i.e. as the sample size increases, the estimates become more and more concentrated around θ
        - Consistency implies asymptotic unbiasedness (if the estimator has finite variance) but the reverse is not true

Lecture 4, Jan 19, 2024

Maximum Likelihood Estimation

- Let $\boldsymbol{X} = (X_1, \dots, X_n)$ and $\boldsymbol{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$ be our samples
- Given the distribution $p_{\boldsymbol{X}}(\boldsymbol{x}; \theta)$ that depends on θ, and suppose we don't know the distribution of θ (denoted by the semicolon)
    - This is known as the *likelihood function*
- The *maximum likelihood* estimate of θ maximizes the likelihood, $\hat{\Theta}_n = \underset{\theta}{\operatorname{argmax}}\, p_{\boldsymbol{X}}(\boldsymbol{x}; \theta)$
- Sometimes instead of likelihood directly we maximize its log instead
    - If $X_1, \dots, X_n$ are IID, then $p_{\boldsymbol{X}}(\boldsymbol{x}; \theta) = \prod\limits_{i=1}^{n} p_{X_i}(x_i; \theta)$
    - Therefore $\log(p_{\boldsymbol{X}}(\boldsymbol{x}; \theta)) = \sum\limits_{i=1}^{n} \log(p_{X_i}(x_i; \theta))$
    - Instead of maximize the total likelihood we can maximize the sum of the logs of the marginals
- Example: Bernoulli RV, $p_X(0; \theta) = 1 - \theta, p_X(1; \theta) = \theta$
    - If we do this n times, then $p_{\boldsymbol{X}}(x_1, \dots, x_n; \theta) = \theta^k (1 - \theta)^{n-k}$ where k is the number of 1s we got
    - The log likelihood is then $k \log \theta + (n - k) \log(1 - \theta)$
    - Differentiate wrt θ and set to zero: $\dfrac{\mathrm{d}}{\mathrm{d}\theta} \log p_X(\boldsymbol{x}; \theta) = \dfrac{k}{\theta} - \dfrac{n-k}{1-\theta} = 0$
    - The MLE estimation is then $\hat{\Theta}_n = \dfrac{k}{n}$
    - We say that k is a *sufficient statistic* for this ML estimator of θ; instead of holding onto all the data we only need to keep track of k
    - We can take the expected value to see that this goes to θ, so the estimator is unbiased
    - Since this is the sample mean, weak law convergence applies, so the estimator is consistent

Laplace: Will the Sun Rise Tomorrow?

- Suppose the sun has risen n consecutive days, $X_1 = 1, \dots, X_n = 1$
- What is the probability that the sun will rise tomorrow?
- A frequentist approach would use the MLE estimate $\hat{\Theta}_n = \dfrac{n}{n} = 1$, so the sun surely rises and this estimate does not change as the number of days increases
- What about a Bayesian approach?
    - Suppose θ is a uniform random variable in the interval $[0, 1]$

  - Now we can find the posterior distribution of θ
  - $f_{\theta|\boldsymbol{X}_n}(\theta|x_1, \dots, x_n) = \dfrac{p_{\boldsymbol{X}_n}(x_1, \dots, x_n|\theta) f_\theta(\theta)}{p_{\boldsymbol{X}_n}(x_1, \dots, x_n)}$
  - $p_{\boldsymbol{X}_n}(x_1, \dots, x_n|\theta) = \theta^n$ if $x_1, \dots, x_n = 1$
  - So the probability of n consecutive 1s is $p_{\boldsymbol{X}}(1, \dots, 1) = \displaystyle\int_0^1 \theta^n f_\theta(\theta)\,\mathrm{d}\theta = \int_0^1 \theta^n\,\mathrm{d}\theta = \frac{1}{n+1}$
  - Therefore $P[X_{n+1} = 1|X_1 = 1, \dots, X_n = 1] = \dfrac{P[X_1 = 1, \dots, X_n = 1 \cap X_{n+1} = 1]}{P[X_1 = 1, \dots, X_n = 1]} = \dfrac{\frac{1}{n+2}}{\frac{1}{n+1}} = \dfrac{n+1}{n+2}$
- Another way is to use the conditional expectation $\hat{\Theta}_n(x) = E[\Theta|\boldsymbol{X} = \boldsymbol{x}]$
  - $P[X_{n=1} = 1|X_1 = 1, \dots, X_n = 1] = E[\Theta|\boldsymbol{X}_n = \mathbf{1}]$

$$
\begin{aligned}
&= \int_0^1 \theta f_{\theta|\boldsymbol{X}_n}(\theta|1, \dots, 1)\,\mathrm{d}\theta \\
&= \int_0^1 \theta \frac{\theta^n}{\frac{1}{n+1}}\,\mathrm{d}\theta \\
&= \frac{n+1}{n+2}
\end{aligned}
$$

Lecture 5, Jan 22, 2024

Maximum A Posteriori (MAP) Estimation

- MAP estimation tries to maximize the probability of the posterior, using a Bayesian approach
- $\hat{\Theta}_n = \underset{\theta}{\operatorname{argmax}}\, p_{\Theta|\boldsymbol{X}}(\theta|\boldsymbol{x}) = \underset{\theta}{\operatorname{argmax}}\, \dfrac{p_{\boldsymbol{X}|\Theta}(\boldsymbol{x}|\Theta) f_\Theta(\theta)}{p_{\boldsymbol{X}}(\boldsymbol{x})}$
  - As with MLE, sometimes it is more convenient to use the log of the posterior instead
  - To simplify the computation we often pick a prior for Θ that matches the form of the likelihood function; this is known as a *conjugate prior*; important ones include:
    - Beta: binomial, geometric
    - Dirichlet: multinomial
    - Gamma: Poisson, exponential
    - Gaussian: Gaussian
  - Note the distribution $p_{\boldsymbol{X}}(\boldsymbol{x})$ usually doesn't matter since it's constant wrt θ
- Example: binomial distribution $p_{X|\Theta}(x|\theta) = \dbinom{n}{k}\theta^k(1-\theta)^{n-k} = \dfrac{n!}{k!(n-k)!}\theta^k(1-\theta)^{n-k}$
  - There are many possible shapes of priors
  - These are all represented by the *beta distribution* $f_\Theta(\theta) = c\theta^{\alpha-1}(1-\theta)^{\beta-1}$ where $\alpha, \beta > 0, 0 \leq \theta \leq 1$ and c is a normalization constant
    - When $\alpha = \beta = 1$ this is uniform
    - $c = \dfrac{1}{B(\alpha, \beta)}$ where $B(\alpha, \beta) = \dfrac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha + \beta)}$
      - Note $\Gamma(m+1) = m!$ for integer m
    - If α, β are integers then $\dfrac{1}{B(\alpha, \beta)} = \dfrac{(\alpha + \beta - 1)!}{(\alpha - 1)!(\beta - 1)!}$

        - This has mean at $\begin{aligned}[t] E[\Theta] &= \frac{1}{B(\alpha,\beta)} \\ &= \int_0^1 \theta f_\Theta(\theta)\,\mathrm{d}\theta \\ &= \int_0^1 \theta^\alpha(1-\theta)^{\beta-1}\,\mathrm{d}\theta \\ &= \frac{B(\alpha+1,\beta)}{B(\alpha,\beta)} \\ &= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\frac{\Gamma(\alpha+1)\Gamma(\beta)}{\Gamma(\alpha+\beta+1)} \\ &= \frac{\alpha}{\alpha+\beta} \end{aligned}$
        - Maximum at $\theta = \dfrac{\alpha-1}{\alpha+\beta-2}$
    - The beta distribution is the conjugate prior of the binomial distribution
    - $p_{\boldsymbol{X}|\Theta}(\boldsymbol{x}|\theta)f_\Theta(\theta) = \dfrac{\binom{n}{k}}{B(\alpha,\beta)}\theta^{k+\alpha-1}(1-\theta)^{n-k+\beta-1}$
    - $p_{\boldsymbol{X}}(\boldsymbol{x}) = \dfrac{\binom{n}{k}}{B(\alpha,\beta)}\displaystyle\int_0^1 \theta^{k+\alpha-1}(1-\theta)^{n-k+\beta-1}\,\mathrm{d}\theta$
        - Note that the integral is just $B(k+\alpha, n-k+\beta)$
        - Therefore $p_{\boldsymbol{X}}(\boldsymbol{x}) = \dfrac{n!}{k!(n-k)!}\dfrac{\Gamma(\alpha+\beta)\Gamma(k+\alpha)\Gamma(n-k+\beta)}{\Gamma(\alpha)\Gamma(\beta)\Gamma(\alpha+n+\beta)}$
    - Solve $\dfrac{\mathrm{d}}{\mathrm{d}\theta}\log f_{\Theta|\boldsymbol{X}}(\theta|\boldsymbol{x}) = 0$
        - $\dfrac{\mathrm{d}}{\mathrm{d}\theta}\log\left(c\theta^{k+\alpha-1}(1-\theta)^{n-k+\beta-1}\right) = \dfrac{k+\alpha-1}{\theta} - \dfrac{n-k+\beta-1}{1-\theta} = 0$
        - $\hat{\theta} = \dfrac{k+\alpha-1}{n+\alpha+\beta-2}$
    - The choice of α and β depends on our knowledge of the prior, e.g. where it peaks, how much variance it has, etc
        - Notice that $\lim\limits_{n\to\infty}\hat{\theta}_{\mathrm{MAP}} = \dfrac{k}{n} = \hat{\theta}_{\mathrm{ML}}$
        - As we take more and more trials, the prior distribution of θ becomes irrelevant since the estimate converges by the weak law

Least Mean Square and Conditional Expectation

- We want to find an estimator that minimizes the mean squared difference between the true value and the estimated value
    - This is another Bayesian approach since we need the prior
- $\hat{\theta}_{\mathrm{LMS}} = \underset{\hat{\theta}}{\operatorname{argmin}}\, E[(\hat{\theta}-\Theta)^2] = E[\Theta|\boldsymbol{X}=\boldsymbol{x}]$
- Suppose we have no data, so we estimate Θ by a constant $\hat{\theta}$:
    - $E[(\hat{\theta}-\Theta)^2] = E[\Theta^2 - 2\Theta\hat{\theta} + \hat{\theta}^2] = \hat{\theta}^2 - 2\hat{\theta}E[\Theta] + E[\Theta]$
    - Differentiate: $2\hat{\theta} - 2E[\Theta] = 0$
    - So in this case the best estimate is $\hat{\theta} = E[\Theta]$
- If we do have data:
    - $E[(\hat{\theta}-\Theta)^2] = E[E[(\hat{\theta}-\Theta)^2|\boldsymbol{x}]] = \displaystyle\int_{-\infty}^{\infty} E[(\Theta-\hat{\theta})|\boldsymbol{X}=\boldsymbol{x}]f_{\boldsymbol{X}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$
    - This can then be minimized by taking $\hat{\theta} = E[\Theta|\boldsymbol{X}=\boldsymbol{x}]$ following the same derivation as the case above

Comparison of MLE, MAP, and LMS Estimation

- Let Θ have a prior uniform on $[0, 1]$ and let X be distributed as uniformly on $[0, \Theta]$
    - The joint distribution covers a triangular area
    - $f(x|\theta)$ is uniform from 0 to θ with value $\frac{1}{\theta}$
    - $f(x, \theta) = f(x|\theta)f(\theta) = \frac{1}{\theta}\frac{1}{1} = \frac{1}{\theta}, 0 < x < \theta < 1$
- For ML:
    - Maximize $f(x|\theta)$
    - We need $\theta \geq x$ because otherwise the value of x couldn't possibly occur
    - And note $f(x|\theta) = \frac{1}{\theta}$ on $x \in [0, \theta]$ so to maximize this we take θ as small as possible
    - Therefore $\hat{\theta}_{\text{ML}} = x$
- For MAP:
    - $f(\theta|x) = \frac{f(x|\theta)f(\theta)}{f(x)} = \frac{f(\theta, x)}{\int_x^1 f(\theta, x)\,\mathrm{d}\theta} = \frac{1}{\theta \ln \frac{1}{x}}, 0 < x < \theta < 1$
    - To maximize this we again take $\hat{\theta}_{\text{MAP}} = x$
    - For this problem, the MAP and ML estimates are the same
- For LMS:
    - $\hat{\theta}_{\text{LMS}} = E[\Theta|x] = \int_x^1 \theta f(\theta|x)\,\mathrm{d}\theta = \int_x^1 \frac{\theta}{\theta \ln \frac{1}{x}}\,\mathrm{d}\theta = \frac{1-x}{\ln \frac{1}{x}}$
    - In this case LMS is less conservative

Lecture 6, Jan 26, 2024

Estimators for Gaussian RVs

- Consider n IID measurements $X_1, \dots, X_n \sim \mathcal{N}(\mu, \sigma^2) = \mathcal{N}(\mu, v)$
- $f_{\boldsymbol{X}}(\boldsymbol{x}; \mu, v) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi v}} e^{-(X_i-\mu)^2} 2v = \frac{1}{(2\pi v)^{\frac{n}{2}}} e^{-\sum_{i=1}^{n} \frac{(X_i-\mu)^2}{2v}}$
- Consider the exponent:

$$\begin{aligned}\sum_{i=1}^{n}(X_i-\mu)^2 &= \sum_i (X_i - M_n + M_n - \mu)^2 \\ &= \sum_i (X_i - M_n)^2 + \sum (M_n-\mu)^2 + 2\sum_i (X_i - M_n)(M_n-\mu) \\ &= nS_n^2 + n(M_n-\mu)^2\end{aligned}$$

    - $\hat{S}_n^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - M_n)^2$ is an estimator for the sample variance
- $f_{\boldsymbol{X}}(\boldsymbol{x}; \mu, v) = \frac{1}{(2\pi v)^{\frac{n}{2}}} e^{-\frac{nS_n^2}{2v}} e^{-n(M_n-\mu)^2} 2v$
    - $\log f_{\boldsymbol{X}}(\boldsymbol{x}; \mu, v) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log v - \frac{nS_n^2}{2v} - \frac{n(M_n-\mu)^2}{2v}$
    - Differentiate wrt μ: $\frac{n}{v}(M_n - \mu) = 0 \implies \hat{\mu}_{\text{ML}} = M_n$
    - Differentiate wrt v: $\frac{n}{2v} + \frac{nS_n^2}{2v^2} + \frac{n(M_n-\mu)^2}{2v^2} = 0 \implies \hat{v}_{\text{ML}} = S_n^2$
- Note:

$$\begin{aligned}E[S_n^2] &= \frac{1}{n}E\left[\sum X_i^2 - 2M_n\sum_i X_i + nM_n^2\right] = E\left[\frac{1}{n}\sum_i X_i^2 - M_n^2\right] \\ &= (v+\mu^2) - \left(\frac{v}{n} + \mu^2\right) \\ &= \frac{n-1}{n}v\end{aligned}$$

  - $E[M_n^2] = \mathrm{Var}[M_n] + E[M_n]^2 = \frac{1}{n}v + \mu^2$
  - This is a biased estimator for the variance!
  - For any finite value of n instead we use ${S'_n}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - M_n)^2$ which is unbiased
  - This applies not just to Gaussians
- Assume the variance is known and the mean has a Gaussian prior; we want to find the MAP estimate
  - Let $X_i = \Theta + W_i$ where W_i is IID noise
  - Assume $E[W_i] = E[W_i|\Theta] = 0$ and $\mathrm{Var}[W_i] = \mathrm{Var}[X_i|\Theta = \theta] = \sigma_w^2$, i.e. noise is independent of θ and zero-mean, known and fixed variance
  - The prior is $f_\Theta(\theta) = c_1 e^{-\frac{(\theta - x_0)^2}{2\sigma^2}}$
  - The likelihood is $f_{\boldsymbol{X}|\Theta}(\boldsymbol{x}|\theta) = c_2 \prod_{i=1}^{n} e^{-\frac{(x_i - \theta)^2}{2\sigma_w^2}}$
    - Knowing θ just gives us the mean of the distribution
    - Note the variance that appears here is different than in the prior!
  - The posterior distribution: $\propto f_{\boldsymbol{X}|\Theta}(\boldsymbol{x}|\theta) f_\Theta(\theta) = c_3 \exp\left(-\frac{1}{2\sigma_w^2}\sum_{i=1}^{n}(x_i - \theta^2) - \frac{1}{2\sigma^2}(\theta - \mu)^2\right)$
  - The exponent becomes $\theta^2\left(\sum_{i=1}^{n}\frac{1}{2\sigma_w^2} + \frac{1}{2\sigma^2}\right) - 2\theta\left(\sum_{i=1}^{n}\frac{x_i}{2\sigma_w^2} + \frac{\mu}{2\sigma^2}\right) + c_4$
  - Completing the square: $\frac{n\sigma^2 + \sigma_w^2}{2\sigma^2\sigma_w^2}\left(\theta - \frac{\sigma^2\sigma_w^2}{n\sigma^2 + \sigma_w^2}\left(\frac{nM_n}{\sigma_w^2} + \frac{\mu}{\sigma^2}\right)\right)^2$
    - This shows that θ is also a Gaussian with mean $\frac{\sigma^2\sigma_w^2}{n\sigma^2 + \sigma_w^2}\left(\frac{nM_n}{\sigma_w^2} + \frac{\mu}{\sigma^2}\right)$ and variance $\frac{\sigma^2\sigma_w^2}{n\sigma^2 + \sigma_w^2}$
  - Since this is a Gaussian we know the maximum occurs at the expectation value
  - The MAP estimate is then $E[\Theta|\boldsymbol{X}] = \frac{n\sigma^2}{n\sigma^2 + \sigma_w^2}M_n + \frac{\sigma_w^2}{n\sigma^2 + \sigma_w^2}\mu$
    - As $n \to \infty$ the first weight approaches 1, the second approaches zero
    - This means as we take more samples, the MAP estimate approaches the ML estimate, as the information from the measurements becomes more important than the prior
  - $\mathrm{Var}[\Theta|\boldsymbol{X}] = \frac{\sigma^2\sigma_w^2}{n\sigma^2 + \sigma_w^2}$
    - Notice that this goes to zero as $n \to \infty$
- In this case, $\hat{\Theta}_{\mathrm{LMS}} = \hat{\Theta}_{\mathrm{MAP}} = E[\Theta|\boldsymbol{X}]$

Lecture 7, Jan 29, 2024

Estimators for Multinomial RVs

- The multinomial distribution is a generalization of the binomial distribution
  - In binomial we had 2 outcomes 0 and 1, so $N_0 + N_1 = n$; in multinomial we have k outcomes, $N_1, \dots, N_K = n$
  - The probability of outcome k is θ_k and $\sum_{k=i}^{K}\theta_k = 1$
  - e.g. tossing a die
- The indicator function for multinomial is a k-tuple $\boldsymbol{X}$, with a 1 in the position that the outcome occurred and 0s everywhere else
  - e.g. $\boldsymbol{X} = (0, 0, 1, 0, \dots, 0)$ indicates outcome is 3
- The probability of $\boldsymbol{X}$ is then $P[\boldsymbol{X} = (b_1, \dots, b_K)] = \prod_{k=1}^{K}\theta_k^{b_k}$ where b_k is the number of occurrences of k

- Again consider n independent trials $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n$ and let $\boldsymbol{\theta} = (\theta_1, \dots, \theta_K)$
- $$\begin{aligned} P[\boldsymbol{X}_1 = \boldsymbol{b}_1, \boldsymbol{X}_2 = \boldsymbol{b}_2, \dots, \boldsymbol{X}_n = \boldsymbol{b}_n; \boldsymbol{\theta}] &= \prod_{j=1}^{n} P[\boldsymbol{X}_j = \boldsymbol{b}_j] \\ &= \prod_{j=1}^{n} \theta_1^{b_{j1}} \dots \theta_K^{b_{jK}} \\ &= \theta_1^{\sum b_{j1}} \dots \theta_K^{\sum b_{jK}} \\ &= \theta_1^{N_1} \dots \theta_K^{N_K} \end{aligned}$$
  - Where $N_k = \sum_j b_{j_k}$ is the number of times outcome k occurred in n trials
  - The vector $\boldsymbol{N} = (N_1, \dots, N_K)$ is a sufficient statistic for our estimators
- Note $E[\boldsymbol{N}; \boldsymbol{\theta}] = (E[N_1], \dots, E[N_K]) = (n\theta_1, n\theta_2, \dots, n\theta_K)$
  - The expected value of the $\boldsymbol{N}$ vector is simply the number of trials times the probability of each trial
- Consider the MLE estimator:
  - $\log P[\boldsymbol{N}; \boldsymbol{\theta}] = \log(\theta_1^{N_1} \dots \theta_K^{N_K}) = \sum_{k=1}^{K} N_k \log \theta_k$
  - Now we need to optimize this sum with respect to $\boldsymbol{\theta}$, with the constraint that all θ_k are positive the sum of all θ_k is 1
  - Lagrangian: $\sum_{k=1}^{K} N_k \log \theta_k + \lambda \left(\sum_{k=1}^{K} \theta_k - 1 \right)$
    - For a particular term θ_j, the derivative is $\frac{N_j}{\theta_k} + \lambda = 0 \implies \frac{N_j}{\theta_j} = -\lambda$
    - Substituting this into the constraint for θ we get $\lambda = -n$
  - Therefore $\hat{\theta}_{j_{\text{ML}}} = -\frac{N_j}{\lambda} = \frac{N_j}{n}$
    - This is expected, since it's the relative frequency of k
- This is for a particular sequence of outcomes; if we only cared about number of occurrences, we have to add the multinomial coefficient
  - $\binom{n}{n_1, n_2, \dots, n_K} = \frac{n!}{n_1! n_2! \dots n_K!}$ where $n_1 + \dots + n_K = n$
  - For $K = 2$, this reduces to the binomial coefficient
- For the MAP estimate we use the Dirichlet prior, which is a generalization of the beta distribution
  - The Dirichlet distribution is $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1) \dots \Gamma(\alpha_K)} \theta_1^{\alpha_1 - 1} \dots \theta_K^{\alpha_K - 1}$ where $\alpha_j > 0, \sum_j \alpha_j = \alpha_0$
    - This is the conjugate prior for the multinomial distribution since it has the same form
  - The posterior is $$\begin{aligned} f(\boldsymbol{\Theta} | n_1, \dots, n_K) &= \frac{p(n_1, \dots, n_K | \boldsymbol{\theta}) f(\boldsymbol{\theta})}{p(n_1, \dots, n_K)} \\ &= c\theta_1^{n_1 + \alpha_1 - 1} \dots \theta_K^{n_K + \alpha_K - 1} \\ &= \frac{\Gamma(\alpha_0 + n)}{\Gamma(\alpha_1 + n) \dots \Gamma(\alpha_K + n_K)} \frac{\prod_{k=1}^{K} \theta_k^{n_k + \alpha_k - 1}}{P(n_1, \dots, n_K)} \end{aligned}$$
  - We again form the Lagrangian and take derivatives to obtain: $\frac{n_j + \alpha_j - 1}{\theta_j} = -\lambda, -\theta_j = \frac{n_j + \alpha_j - 1}{\lambda}$
  - Therefore $\hat{\theta}_{j_{\text{MAP}}} = \frac{n_j + \alpha_j - 1}{n + \alpha_0 - K}$
    - The $-K$ in the denominator gets rid of all the extra 1s in the αs when summed up
    - We can interpret this as a relative frequency, where prior to doing the experiment we did $\alpha_0 - K$ experiments and outcome j occurred $\alpha_j - 1$ times
- Consider the LMS estimator:

  - $E[\mathbf{\Theta}|\boldsymbol{N}] = \int \dots \int (\theta_1, \dots, \theta_K) c\theta_1^{n_1+\alpha_1-1} \dots \theta_K^{n_K+\alpha_K-1} \, \mathrm{d}\theta_1 \dots \mathrm{d}\theta_k$
    $= (E[\Theta_1|n_1 + \alpha_1 - 1], \dots, E[\Theta_K|n_K + \alpha_K - 1]$
    $= \left(\frac{n_1 + \alpha_1}{n + \alpha_0}, \dots, \frac{n_K + \alpha_K}{n + \alpha_0}\right)$
    - Note $E[\Theta_j|n_j + \alpha_j - 1] = c\int_0^1 \theta_j \theta_j^{n_j+\alpha_j-1} \, \mathrm{d}\theta_j = \frac{n_j + \alpha_j}{n + \alpha_0}$
  - Therefore $\hat{\theta}_{j_{\text{LMS}}} = \frac{n_j + \alpha_j}{n + \alpha_0}$
- Again notice that as $n \to \infty$, all 3 of these estimators converge to the ML estimator

Binary Hypothesis Testing

- Hypothesis testing is like a more constrained version of parameter estimation; instead of estimating the value of θ, we are testing whether θ_0 or θ_1 is more likely
- Given two hypotheses H_0 (the *null hypothesis*, or the "default" to be proved or disproved) and H_1 (the *alternative hypothesis*), we want to know which one is more likely
- We would like to find $g\colon S_{\boldsymbol{X}} \mapsto \{ H_0, H_1 \}$ mapping from observations to hypotheses based on $P[\boldsymbol{X} \in A; H_j]$
  - g divides the sample space into 2 parts, the *acceptance region* R^c where H_0 is accepted and *rejection region* R where H_0 is rejected
- If g is not perfect, then 2 types of error can occur:
  - *Type I error*: H_0 is rejected when it is true
    - Also known as the *significance level* of a test
    - $\alpha(R) = P[\boldsymbol{X} \in R; H_0]$
    - We typically pick this to be 10%, 5%, 1%, etc
  - *Type II error*: H_0 is accepted when H_1 is true (i.e. H_0 is false)
    - $\beta(R) = P[\boldsymbol{X} \in R^c; H_1]$
- We can do this partitioning using our 3 estimators
- Using MLE, we simply pick the H that gives us the maximum likelihood
  - We just need to test $p_{\boldsymbol{X}}(\boldsymbol{x}|H_0)$ and $p_{\boldsymbol{X}}(\boldsymbol{x}|H_1)$
  - The *likelihood ratio* is $L(\boldsymbol{x}) = \frac{p_{\boldsymbol{X}}(\boldsymbol{x}|H_1)}{p_{\boldsymbol{X}}(\boldsymbol{x}|H_0)}$ (alternative divided by null)
  - With the maximum likelihood rule we reject H_0 when $L(\boldsymbol{x}) > 1$
  - This can be generalized to rejecting when $L(\boldsymbol{x}) > \xi$ where ξ is the *critical value*
    - Use this when we know one hypothesis is more likely (i.e. a prior)
    - As we increase ξ, α decreases while β increases
- Example: $H_0 : X \sim \mathcal{N}(0, 1), H_1 : X \sim \mathcal{N}(1, 1)$
  - The hypothesis changes the mean of the Gaussian
  - $L(x) = \frac{f_X(x; H_1)}{f_X(x; H_2)} = \frac{e^{-(x-1)^2/2}}{e^{-x^2/2}} = e^{-\frac{1}{2}(-2x+1)}$
  - In this case the threshold rule is $x \lesseqgtr \gamma = \ln \xi + \frac{1}{2}$
  - Type I error: $\alpha(\gamma) = \int_\gamma^\infty \frac{1}{\sqrt{2\pi}} e^{-x'^2/2} \, \mathrm{d}x' = Q(\gamma)$
    - This decreases with γ
  - Type II error: $\beta(\gamma) = \int_{-\infty}^{\gamma} = \frac{1}{\sqrt{2\pi}} e^{-(x'-1)^2/2} \, \mathrm{d}x' = Q(1 - \gamma)$
    - This increases with γ
  - Note $Q(x) = 1 - \Phi(x)$ where $\Phi(x)$ is the standard normal CDF
- So far we've only divided the region into 2, where one side is accept and the other is reject; we could also do a more complex division where we have pockets of accept in the rejection region, etc; is this better?

Theorem

Neyman Pearson Lemma: Given the likelihood ratio test $L(X), \xi$ such that

$$P[L(x) > \xi; H_0] = \alpha \quad \text{and} \quad P[L(X) \leq \xi; H_1] = \beta$$

then for any other test (region R) with $P[X \in R; H_0] \leq \alpha$ it must be that $P[X \notin R; H_1] \geq \beta$ and

$$P[X \in R; H_0] < \alpha \implies P[X \notin R; H_1] > \beta$$

That is, the LRT achieves the best possible tradeoff between α and β.

- The Neyman Pearson lemma states that given any value of α, the likelihood ratio test gives the smallest possible β to achieve that α
  - This is a constrained minimization problem of minimizing β subject to a certain α
    - Lagrangian: $\int_A f_X(x; H_1)\,\mathrm{d}x + \lambda\left(\int_R f_X(x; H_0)\,\mathrm{d}x - \alpha\right) = \lambda(1-\alpha) + \int_A (f_X(x; H_1) - \lambda f_X(x; H_0))\,\mathrm{d}x$
    - To minimize this we include x in A if $\dfrac{f_X(x; H_1)}{f_X(x; H_0)} < \lambda$ to make the term in the integral always negative, which is the LRT

Lecture 8, Feb 2, 2024

Bayesian Hypothesis Testing

- We switch to using a MAP approach instead of ML if we have prior knowledge of the hypotheses
- Assume we know a priori that $P[H_0] = \pi_0$ and $P[H_1] = 1 - \pi_0 = \pi 1$
- MAP rule: compare $p(H_1|x)$ with $p(H_0|x)$
  - $P(H_j|x) = \dfrac{p_x(x|H_j)P[H_j]}{p_x(x|H_0)P[H_0] + p_x(x\ H_1)P[H_1]}$
  - Therefore this is equivalent to $p(x|H_1)\pi_1 \underset{H_0}{\overset{H_1}{\lesseqgtr}} p(x|H_0)\pi_0$
- Example: binary communications
  - If a 0 was sent then $X \sim \mathcal{N}(-1, \sigma^2)$; if a 1 was sent then $X \sim \mathcal{N}(1, \sigma^2)$
  - Compare $\dfrac{1}{\sqrt{2\pi}\sigma} e^{-(x-1)^2} 2\sigma^2 \pi_1$ and $\dfrac{1}{\sqrt{2\pi}\sigma} e^{-(x+1)^2} 2\sigma^2 \pi_0$
  - The threshold is $e^{\frac{4x}{2\sigma^2}} \lesseqgtr \dfrac{\pi_0}{\pi 1}$ or $x \lesseqgtr \dfrac{\sigma^2}{2} \log \dfrac{\pi_0}{\pi_1}$
  - If we assume that the bits are balanced, i.e. $\pi_0 = \pi_1$, then we're simply seeing if x is positive or negative
- Assign different costs, C_{ij} being the cost of accepting H_i if H_j is true
  - The expected cost is $C_{00}P[\text{Accept } H_0|H_0]\pi_0 + C_{01}P[\text{Reject } H_0|H_0]\pi_0 + C_{10}P[\text{Accept } H_1|H_1]\pi_1 + C_{11}P[\text{Reject } H_1|H_1]\pi_1$
  - Define the likelihood $\Lambda(x) = \dfrac{f_X(x|H_1)}{f_X(x|H_0)}$
  - Optimal decision rule: accept H_0 if $\Lambda(x) < \dfrac{\pi_0(C_{01} - C_{00})}{\pi_1(C_{10} - C_{11})}$
    - We can derive this using the same procedure as above
    - A special case is to minimize the probability of error in which case $C_{00} = C_{11} = 0, C_{01} = C_{10} = 1$
    - Equivalent to minimizing $P_E = \int_R f_X(x|H_0)\pi_0\,\mathrm{d}x + \int_{R^c} f_X(x|H_1)\pi_1\,\mathrm{d}x$
- Consider the Gaussian example with minimum error probability as the cost
  - $P_E = \int_R \dfrac{1}{\sqrt{2\pi}\sigma} e_0^{\frac{-(x+1)^2 2\sigma^2}{\pi}}\,\mathrm{d}x + \int_{R^c} \dfrac{1}{\sqrt{2\pi}\sigma} e_1^{\frac{-(x-1)^2 2\sigma^2}{\pi}}\,\mathrm{d}x = 1 + \int_{R^c} \dfrac{1}{\sqrt{2\pi}\sigma}\left(e^{-\frac{(x-1)^2}{2\sigma^2}}\pi_1 - e^{-\frac{(x+1)^2}{2\sigma^2}}\pi_0\right)\mathrm{d}x$

    - To minimize this we want to pick the expression inside brackets so it is always negative
    - Making the expression negative gives us back the MAP rule!
- For multiple hypotheses:
    - Probability of correct is $\sum_{k=1}^{K}\int_{R_k} P[X = x, H_k]\,\mathrm{d}x$
    - Min cost: $\sum_{j}\int_{R_j}\sum_{k=1}^{K} L_{kj}P[X = x, H_k]\,\mathrm{d}x$
        * We need a cost L_{kj} between every pair of hypotheses
        * Pick j to minimize the sum inside the integral
- Naive Bayes assumption: all measurements are independent given θ

Significance Testing

- Sometimes we know what H_0 is but we don't have a clear alternative hypothesis
- We want to bound α, the probability of type I error
- Example: testing if a unit variance Gaussian is zero mean
    - Let $S = \frac{1}{\sqrt{n}}(X_1, \dots, X_n)$
    - Then if H_0 holds then $S \sim \mathcal{N}(0, 1)$
    - Suppose our decision rule is to accept H_0 if $S \in [-\gamma, \gamma]$
    - $\alpha = \int_{-\infty}^{\gamma} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}\,\mathrm{d}x + \int_{\gamma}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}\,\mathrm{d}x = 2Q(\gamma)$
    - If we want to restrict the type I error probability to 5%, then $Q(\gamma) = 0.05 \implies \gamma = 1.96$

Lecture 9, Feb 5, 2024

Joint Gaussian Distributions

Theorem

Central Limit Theorem: Let $X_1, \dots, X_n$ be a sequence of i.i.d. RVs from any distribution with finite mean μ and variance σ^2, and let $S_n = X_1 + \dots + X_n$ be their sum; and let

$$Z_n = \frac{S_n - n\mu}{\sigma\sqrt{n}}$$

which is zero-mean and unit variance, then

$$\lim_{n\to\infty} P[Z_n \leq z] = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{z} e^{-\frac{x^2}{2}}\,\mathrm{d}x$$

i.e. the distribution of Z_n approaches $\mathcal{N}(0, 1)$.

Definition

X and Y are *jointly Gaussian* if their joint PDF is given by

$$f_{X,Y}(x, y) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1 - \rho_{X,Y}^2}} e^{-\frac{1}{2(1-\rho_{X,Y}^2)}\left(\left(\frac{x-\mu_1}{\sigma_1}\right)^2 - 2\rho_{X,Y}\left(\frac{x-\mu_1}{\sigma_1}\right)\left(\frac{y-\mu_2}{\sigma_2}\right) + \left(\frac{y-\mu_2}{\sigma_2}\right)^2\right)}$$

where μ_1, μ_2 are the means, σ_1^2, σ_2^2 are the variances, and $\rho_{X,Y}$ is the correlation coefficient of X, Y.

- Notice that the expression is symmetric in X and Y, and both variables appear in their normalized form
- If X and Y are uncorrelated, then $f_{X,Y}(x,y) = \frac{1}{2\pi\sigma_1\sigma_2} e^{-\frac{1}{2}\left(\left(\frac{x-\mu_1}{\sigma_1}\right)^2 + \left(\frac{y-\mu_2}{\sigma_2}\right)^2\right)} = f_X(x)f_Y(y)$
  - For Gaussian RVs, uncorrelated implies independent
- If we compute marginals by completing the square, we see that both are Gaussian

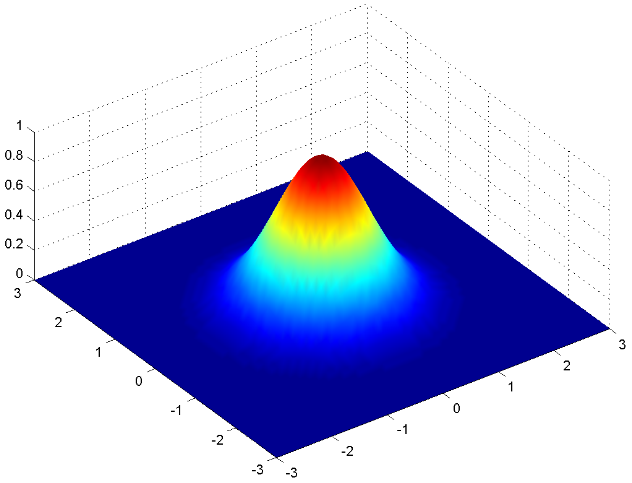


Figure 1: Plot of a joint Gaussian distribution with zero-mean, unit variance and uncorrelated X, Y.

- For the case of zero-mean, unit variance and uncorrelated X, Y above the contours of constant probability are circles centered about the origin
  - Changing the mean shifts the centre of the distribution
  - The exponent is in quadratic form
  - If the variances are not equal (but still uncorrelated), we will get axis-aligned ellipses as the distribution in each dimension gets stretched out
  - If the correlation is nonzero, the axes of the ellipse will no longer be axis-aligned
    - For a positive ρ the ellipse is along the $x = y$ axis
    - For a negative ρ the ellipse is along the $x = -y$ axis
    - The closer ρ is to 1, the more tightly packed the ellipse is along its axis
  - We can always find a transformation that aligns the axes of the ellipse with the x and y axis to make them independent in the new transformed space
- The conditional PDF is $f_{X,Y}(x|y) = \frac{1}{\sqrt{2\pi\sigma_1^2(1-\rho_{X,Y}^2)}} e^{-\frac{1}{2(1-\rho_{X,Y}^2)\sigma_1^2}\left(x-\rho_{X,Y}\frac{\sigma_1}{\sigma_2}(y-\mu_2)-\mu_1\right)^2}$
  - This is another Gaussian with $\mu = \rho_{X,Y}\frac{\sigma_1}{\sigma_2}(y-\mu_2) + \mu_1$ and $\sigma^2 = (1-\rho_{X,Y}^2)\sigma_1^2$
  - Notice the new mean is the normalized y, scaled up by the standard deviation of x, multiplied by the correlation and then add back to mean of x to shift it
  - The variance has no dependence on y but knowing y reduces the variance of x
  - As $\rho_{X,Y} \to \pm 1$, the conditional variance approaches 0 because X is just a linear function of Y
- Consider a linear transformation $\begin{bmatrix} V \\ W \end{bmatrix} = \begin{bmatrix} a & b \\ c & e \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} = \boldsymbol{AX}$ where the determinant is nonzero (invertible)
  - The joint PDF of V and W is given by $f_{V,W}(v,w) = \frac{f_{X,Y}(x,y)}{\det \boldsymbol{A}}$
  - Intuitively an area $\mathrm{d}x$ by $\mathrm{d}y$ is mapped to an area of size $\mathrm{d}P$; this ratio is the determinant
    - $f(x,y)\,\mathrm{d}x\,\mathrm{d}y = f(v,w)\,\mathrm{d}P$ since both are the probability of a small region
  - For a nonlinear transformation the determinant is replaced by a Jacobian
  - Note practically to get this in terms of v, w we need to find the inverse mapping from v, w to x, y

- More generally consider $\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{Z}$ where $A \in \mathbb{R}^{n\times n}$ and is invertible
  - The joint PDF is $f_Z(\boldsymbol{Z}) = f(z_1, \dots, z_n) = \dfrac{f(x_1, \dots, x_n)}{\det \boldsymbol{A}} = \dfrac{f_{\boldsymbol{X}}(\boldsymbol{A}^{-1}\boldsymbol{z})}{\det \boldsymbol{A}}$

Generalization of Expectation and Variance

- Let the *mean vector* of $\boldsymbol{X} = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix}$ be $\boldsymbol{m}_{\boldsymbol{X}} = E[\boldsymbol{X}] = \begin{bmatrix} E[X_1] \\ \vdots \\ E[X_n] \end{bmatrix}$
- Let the *correlation matrix* be $\boldsymbol{R}_{\boldsymbol{X}} = \begin{bmatrix} E[X_1^2] & E[X_1X_2] & \dots & E[X_1X_n] \\ E[X_2X_1] & E[X_2^2] & \dots & E[X_2X_n] \\ \vdots & \vdots & \ddots & \vdots \\ E[X_nX_1] & E[X_nX_2] & \dots & E[X_n^2] \end{bmatrix}$
  - Note that this is symmetric
  - The diagonal elements are second moments
- Let the *covariance matrix* be $\boldsymbol{K}_{\boldsymbol{X}}$ such that entry (i, j) is σ_{X_i,X_j}
  - This is symmetric positive semidefinite
  - The diagonal entries are the variances of each variable
  - If the means are all zero, this is equivalent to the correlation matrix
  - If all X_i, X_j are uncorrelated, then the covariance matrix is diagonal
- Notice that $\boldsymbol{R}_{\boldsymbol{X}} = E[\boldsymbol{X}\boldsymbol{X}^T]$ and $\boldsymbol{K}_{\boldsymbol{X}} = E[(\boldsymbol{X} - \boldsymbol{m}_{\boldsymbol{X}})(\boldsymbol{X} - \boldsymbol{m}_{\boldsymbol{X}})^T] = \boldsymbol{R}_{\boldsymbol{X}} - \boldsymbol{m}_{\boldsymbol{X}}\boldsymbol{m}_{\boldsymbol{X}}^T$
- For any general linear transformation $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}$:
  - $E[\boldsymbol{Y}] = \boldsymbol{A}E[\boldsymbol{X}] = \boldsymbol{A}\boldsymbol{m}_{\boldsymbol{X}}$
  - $\boldsymbol{K}_{\boldsymbol{Y}} = \boldsymbol{A}\boldsymbol{K}_{\boldsymbol{X}}\boldsymbol{A}^T$
- We can apply an eigendecomposition to the covariance matrix
  - Often our covariance matrix will be full rank, which makes it positive definite, and makes the decomposition always possible
  - Find eigenvectors $\boldsymbol{e}_i$ such that $\boldsymbol{K}_{\boldsymbol{X}}\boldsymbol{e}_i = \lambda\boldsymbol{e}_i$ and $\boldsymbol{e}_i^T\boldsymbol{e}_j = \delta_{ij}$ (orthonormal eigenvectors)
  - Let $\boldsymbol{P} = \begin{bmatrix} \boldsymbol{e}_1 & \dots & \boldsymbol{e}_n \end{bmatrix}$ and $\boldsymbol{\Lambda} = \operatorname{diag} \lambda_i$, then $\boldsymbol{K}_{\boldsymbol{X}} = \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^T$
  - For a general Gaussian, this means that if we first transform the variables by $\boldsymbol{P}^T$, then they will all be independent of each other

Lecture 10, Feb 9, 2024

Gaussian Random Vectors

Definition

Gaussian Random Vector: $\boldsymbol{X} \in \mathbb{R}^n$ is Gaussian distributed with mean $\boldsymbol{m}_{\boldsymbol{X}}$ and covariance $\boldsymbol{K}_{\boldsymbol{X}}$ if it has distribution

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{n}{2}} \det(\boldsymbol{K}_{\boldsymbol{X}})^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{m}_{\boldsymbol{X}})^T \boldsymbol{K}_{\boldsymbol{X}}^{-1}(\boldsymbol{x} - \boldsymbol{m}_{\boldsymbol{X}})\right]$$

- The exponent is in quadratic form and specifies an ellipsoid in $\mathbb{R}^n$
- Note that if $X_1, \dots, X_n$ are all uncorrelated then $\boldsymbol{K}_{\boldsymbol{X}}$ is diagonal
  - $\boldsymbol{K}_{\boldsymbol{X}} = \begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_n^2 \end{bmatrix}$

  - $\boldsymbol{K}_{\boldsymbol{X}}^{-1} = \begin{bmatrix} \frac{1}{\sigma_1^2} & & \\ & \ddots & \\ & & \frac{1}{\sigma_n^2} \end{bmatrix}$
  - Multiply this by $\boldsymbol{x} - \boldsymbol{m}_{\boldsymbol{X}}$ and we get $\left(\frac{x_1 - m_1}{\sigma_1}\right)^2 + \cdots + \left(\frac{x_n - m_n}{\sigma_n}\right)^2$
  - This expression is in the exponent, so we can split it up into a product of exponentials
  - The resulting distribution is a product of distributions in each X, so they are all independent
- Consider some linear transformation $\boldsymbol{A}$ so that $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}$ is the transformed version of $\boldsymbol{X}$, which are jointly Gaussian
  - $f_{\boldsymbol{Y}}(\boldsymbol{y}) = \dfrac{f_{\boldsymbol{X}}(\boldsymbol{x})}{\det \boldsymbol{A}} = \dfrac{f_{\boldsymbol{X}}(\boldsymbol{A}^{-1}\boldsymbol{y})}{\det \boldsymbol{A}}$
  - Substitute this into the Gaussian for $\boldsymbol{X}$, in the exponent we get $(\boldsymbol{A}^{-1}\boldsymbol{y} - \boldsymbol{m}_{\boldsymbol{X}})^T \boldsymbol{K}_{\boldsymbol{X}}^{-1} (\boldsymbol{A}^{-1}\boldsymbol{y} - \boldsymbol{m}_{\boldsymbol{X}})$
  - Factor out $\boldsymbol{A}$: $(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{m}_{\boldsymbol{X}})^T \boldsymbol{A}^{-T} \boldsymbol{K}_{\boldsymbol{X}}^{-1} \boldsymbol{A}^{-1} (\boldsymbol{y} - \boldsymbol{A}\boldsymbol{m}_{\boldsymbol{X}}) = (\boldsymbol{y} - \boldsymbol{A}\boldsymbol{m}_{\boldsymbol{X}})^T (\boldsymbol{A}\boldsymbol{K}_{\boldsymbol{X}}\boldsymbol{A}^T)^{-1} (\boldsymbol{y} - \boldsymbol{A}\boldsymbol{m}_{\boldsymbol{X}})$
  - Therefore $\boldsymbol{A}\boldsymbol{K}_{\boldsymbol{X}}\boldsymbol{A}^T$ is the new covariance matrix and $\boldsymbol{A}\boldsymbol{m}_{\boldsymbol{X}}$ is the new mean; the result is still Gaussian
  - Since $\boldsymbol{K}_{\boldsymbol{X}}$ is real and symmetric we can find $\boldsymbol{A}$ such that $\boldsymbol{A}\boldsymbol{K}_{\boldsymbol{X}}\boldsymbol{A}^T = \boldsymbol{\Lambda}$, then the resulting Gaussian will be independent in its variables
- Suppose $\boldsymbol{X}$ is IID, can we find a linear transformation $\boldsymbol{A}$ such that the resulting $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}$ has covariance $\boldsymbol{K}_{\boldsymbol{Y}}$?
  - $\boldsymbol{K}_{\boldsymbol{Y}} = \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^T = \boldsymbol{P}\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{P}^T$
  - Let $\boldsymbol{A} = \boldsymbol{P}\boldsymbol{\Lambda}^{\frac{1}{2}}$
  - Then $\boldsymbol{K}_{\boldsymbol{Y}} = \boldsymbol{A}\boldsymbol{K}_{\boldsymbol{X}}\boldsymbol{A}^T = \boldsymbol{A}\boldsymbol{1}\boldsymbol{A}^T = \boldsymbol{A}\boldsymbol{A}^T = \boldsymbol{P}\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{P}^T = \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^T$

Lecture 11, Feb 12, 2024

Gaussian Discriminant Analysis

- Consider a classification problem where we have classes $c \in C$, each having a prior π_c, jointly Gaussian distributed with a mean of $\boldsymbol{\mu}_c$ and a covariance of $\boldsymbol{\Sigma}_c$
- Gaussian discriminant analysis is a special case of hypothesis testing for this type of classification problem; given an observation of the vector $\boldsymbol{X}$, we would like to know which class it came from (i.e. which hypothesis is true)
- Consider the case where all the classes have the same covariance $\boldsymbol{\Sigma}$, so they differ only by their mean
  - The posterior is $P[y = c|\boldsymbol{x}] = \dfrac{f(\boldsymbol{x}|c)\pi_c}{\sum_{c'} f(\boldsymbol{x}|y = c')\pi_c'}$
  - The numerator becomes $\dfrac{e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_c)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_c)}}{(2\pi)^{\frac{D}{2}}\sqrt{\det \boldsymbol{\Sigma}}}\pi_c = \dfrac{e^{-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x}}}{(2\pi)^{\frac{D}{2}}\sqrt{\det \boldsymbol{\Sigma}}} \exp\left[-\frac{1}{2}\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c + \boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} + \log \pi_c\right]$
  - Let $\boldsymbol{\beta}_c^T = \boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}$ and $\boldsymbol{\gamma}_c = \log \pi_c - \dfrac{1}{2}\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c$
  - The exponential can then be written as $p(y = c|\boldsymbol{x}) = \dfrac{\exp(\boldsymbol{\beta}_c^T\boldsymbol{x} + \boldsymbol{\gamma}_c)}{\sum_{c'} \exp(\boldsymbol{\beta}_{c'}^T\boldsymbol{x} + \boldsymbol{\gamma}_{c'})}$
    - This is a softmax function
    - The exponential in the softmax makes it so that the largest term dominates while all other terms are usually much smaller
    - Each class has an associated $\boldsymbol{\beta}_c$ and $\boldsymbol{\gamma}_c$, which contains all the info of the class
  - With this, we have $p(y = c|x) \approx \begin{cases} 1 & \boldsymbol{\beta}_c^T\boldsymbol{x} + \boldsymbol{\gamma}_c \gg \boldsymbol{\beta}_{c'}^T\boldsymbol{x} + \boldsymbol{\gamma}_{c'} \\ 0 & \text{otherwise} \end{cases}$
  - The decision rule is $\hat{y}(\boldsymbol{x}) = \underset{c}{\operatorname{argmax}}\, \boldsymbol{\beta}_c^T + \boldsymbol{\gamma}_c$
  - This is referred to as *linear Gaussian discriminant analysis*, since the decision boundary is a linear function of $\boldsymbol{x}$

    - * The boundary occurs where $\boldsymbol{\beta}_1^T\boldsymbol{x}+\boldsymbol{\gamma}_1=\boldsymbol{\beta}_0^T\boldsymbol{x}+\boldsymbol{\gamma}_0$ which forms a hyperplane

- More generally, the covariances of the classes are different
  - $\log p(y=c|\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_c)^T\boldsymbol{\Sigma}_c^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_c) - \frac{1}{2}\det\boldsymbol{\Sigma}_c + \log\pi_c - \frac{D}{2}\log 2\pi$
  - Consider the boundary between two regions:
    - * $-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{\Sigma}_1^{-1}\boldsymbol{x}+\boldsymbol{\mu}_1^T\boldsymbol{\Sigma}_1^{-1}\boldsymbol{x}-\frac{1}{2}\log\det\boldsymbol{\Sigma}_1+\log\pi_1 = -\frac{1}{2}\boldsymbol{x}^T\boldsymbol{\Sigma}_0^{-1}\boldsymbol{x}+\boldsymbol{\mu}_0^T\boldsymbol{\Sigma}_1^{-1}\boldsymbol{x}-\frac{1}{2}\log\det\boldsymbol{\Sigma}_0+\log\pi_0$
    - * $\boldsymbol{x}^T(\boldsymbol{\Sigma}_1^{-1}-\boldsymbol{\Sigma}_0^{-1})\boldsymbol{x}+2(\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0-\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1)^T\boldsymbol{x}+(\boldsymbol{\mu}_0^T\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0-\boldsymbol{\mu}_1^T\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1)+\log\frac{\det\boldsymbol{\Sigma}_0}{\det\boldsymbol{\Sigma}_1}+2\log\frac{\pi_1}{\pi_0}=0$
    - * This is a quadratic form that can define a parabola, hyperbola, or even circles and ellipses
- To obtain the parameters of the Gaussian distribution of each class, we can use ML estimation
  - $\hat{\pi}_c = \frac{n_c}{n}$ is given by the relative frequency of class c
  - $\hat{\boldsymbol{\mu}}_c = \frac{1}{n_c}\sum_i \boldsymbol{x}_i^c$ is given by the sample mean
  - $\hat{\boldsymbol{\Sigma}}_c = \frac{1}{n_c}(\boldsymbol{x}_i^c-\hat{\boldsymbol{\mu}}_c)^T(\boldsymbol{x}_i^c-\hat{\boldsymbol{\mu}}_c)$
  - Note that the variance estimate here is biased; use the version with n_c-1 in the denominator for unbiased

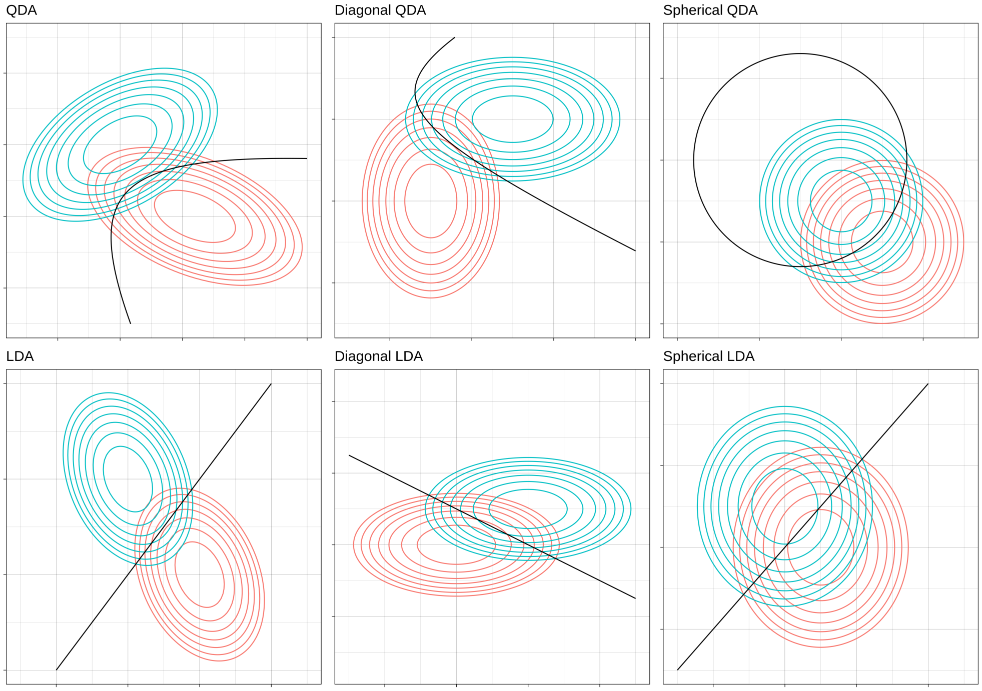


Figure 2: Illustration of the different kinds of Gaussian discriminant analysis.

Gaussian Parameter Estimation

- Let $\boldsymbol{X},\boldsymbol{Y}$ be jointly Gaussian and let $\boldsymbol{w}=\begin{bmatrix}\boldsymbol{X}\\\boldsymbol{Y}\end{bmatrix}$; we want to find the MAP estimator for $\boldsymbol{X}$ given $\boldsymbol{Y}$
  - i.e. we want to find the distribution of $\boldsymbol{X}$ conditioned on $\boldsymbol{Y}$
  - We will make use of the covariances between elements of $\boldsymbol{X}$ and $\boldsymbol{Y}$
  - Strategy: expand out the exponent of the joint PDF and rearrange it into a form with $\boldsymbol{X}$ as the variable
- The mean of $\boldsymbol{w}$ is $\boldsymbol{\mu}=\begin{bmatrix}\boldsymbol{\mu}_X\\\boldsymbol{\mu}_Y\end{bmatrix}$

- The covariance is $\boldsymbol{\Sigma}_w = E[(\boldsymbol{w}-\boldsymbol{\mu}_w)(\boldsymbol{w}-\boldsymbol{\mu}_w)^T] = E\left[\begin{bmatrix}\boldsymbol{X}-\boldsymbol{\mu}_X \\ \boldsymbol{Y}-\boldsymbol{\mu}_Y\end{bmatrix}\begin{bmatrix}(\boldsymbol{X}-\boldsymbol{\mu}_X)^T \\ (\boldsymbol{Y}-\boldsymbol{\mu}_Y)^T\end{bmatrix}\right] = \begin{bmatrix}\boldsymbol{\Sigma}_{\boldsymbol{XX}} & \boldsymbol{\Sigma}_{\boldsymbol{XY}} \\ \boldsymbol{\Sigma}_{\boldsymbol{YX}} & \boldsymbol{\Sigma}_{\boldsymbol{YY}}\end{bmatrix}$
  - The overall dimension is $N \times N$; the $\boldsymbol{\Sigma}_{\boldsymbol{XY}}, \boldsymbol{\Sigma}_{\boldsymbol{YX}}$ matrices are in general rectangular
  - Let the *precision matrix* $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}_w^{-1} = \begin{bmatrix}\boldsymbol{\Lambda}_{\boldsymbol{XX}} & \boldsymbol{\Lambda}_{\boldsymbol{XY}} \\ \boldsymbol{\Lambda}_{\boldsymbol{YX}} & \boldsymbol{\Lambda}_{\boldsymbol{YY}}\end{bmatrix}$
    * This is the opposite of variance; the larger the precision, the more tightly concentrated the distribution
    * Note $\boldsymbol{\Sigma}_{\boldsymbol{XX}} \neq \boldsymbol{\Lambda}_{\boldsymbol{XX}}^{-1}$, and $\boldsymbol{\Lambda}_{\boldsymbol{XY}} = \boldsymbol{\Lambda}_{\boldsymbol{YX}}^T$
- Now consider the exponent of the joint PDF
  - $$\begin{aligned} & -\frac{1}{2}(\boldsymbol{w}-\boldsymbol{\mu}_w)^T\boldsymbol{\Sigma}_w^{-1}(\boldsymbol{w}-\boldsymbol{\mu}_w) \\ &= -\frac{1}{2}\begin{bmatrix}\boldsymbol{X}-\boldsymbol{\mu}_X \\ \boldsymbol{Y}-\boldsymbol{\mu}_Y\end{bmatrix}^T \begin{bmatrix}\boldsymbol{\Lambda}_{\boldsymbol{XX}} & \boldsymbol{\Lambda}_{\boldsymbol{XY}} \\ \boldsymbol{\Lambda}_{\boldsymbol{YX}} & \boldsymbol{\Lambda}_{\boldsymbol{YY}}\end{bmatrix}\begin{bmatrix}\boldsymbol{X}-\boldsymbol{\mu}_X \\ \boldsymbol{Y}-\boldsymbol{\mu}_Y\end{bmatrix} \\ &= -\frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu}_X)^T\boldsymbol{\Lambda}_{\boldsymbol{XX}}(\boldsymbol{X}-\boldsymbol{\mu}_X) - \frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu}_X)^T\boldsymbol{\Lambda}_{\boldsymbol{XY}}(\boldsymbol{Y}-\boldsymbol{\mu}_Y) - \frac{1}{2}(\boldsymbol{Y}-\boldsymbol{\mu}_Y)^T\boldsymbol{\Lambda}_{\boldsymbol{YX}}(\boldsymbol{X}-\boldsymbol{\mu}_X) - \frac{1}{2}(\boldsymbol{Y}-\boldsymbol{\mu}_Y)^T \\ &= -\frac{1}{2}\left(\boldsymbol{X}^T\boldsymbol{\Lambda}_{\boldsymbol{XX}}\boldsymbol{X} - \boldsymbol{X}^T\boldsymbol{\Lambda}_{\boldsymbol{XY}}(\boldsymbol{Y}-\boldsymbol{\mu}_Y) + (\boldsymbol{Y}-\boldsymbol{\mu}_Y)^T\boldsymbol{\Lambda}_{\boldsymbol{YX}}\boldsymbol{X} - \boldsymbol{\mu}_X^T\boldsymbol{\Lambda}_{\boldsymbol{XX}}\boldsymbol{X} - \boldsymbol{X}^T\boldsymbol{\Lambda}_{\boldsymbol{XX}}\boldsymbol{\mu}_X + \dots\right) \\ &= -\frac{1}{2}\left(\boldsymbol{X}^T\boldsymbol{\Lambda}_{\boldsymbol{XX}}\boldsymbol{X} - 2\boldsymbol{X}^T(\boldsymbol{\Lambda}_{\boldsymbol{XX}}\boldsymbol{\mu}_X - \boldsymbol{\Lambda}_{\boldsymbol{XY}}(\boldsymbol{Y}-\boldsymbol{\mu}_Y)) + \dots\right) \end{aligned}$$
  - This gives us $f(\boldsymbol{x}|\boldsymbol{y})$ and implies that it is jointly Gaussian
  - Let $f(\boldsymbol{x}|\boldsymbol{y}) = c\exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_{X|Y})^T\boldsymbol{\Sigma}_{X|Y}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_{X|Y})\right] = c\exp\left[-\frac{1}{2}\left(\boldsymbol{x}\sigma_{\boldsymbol{XY}}^{-1}\boldsymbol{x} - 2\boldsymbol{x}^T\boldsymbol{\Sigma}_{\boldsymbol{XY}}^{-1}\boldsymbol{\mu}_{X|Y}\right)\right]$
  - By matching terms we see $\boldsymbol{\Sigma}_{\boldsymbol{XY}} = \boldsymbol{\Lambda}_{\boldsymbol{XX}}^{-1}$ and $\boldsymbol{\mu}_{X|Y} = \boldsymbol{\mu}_X - \boldsymbol{\Lambda}_{\boldsymbol{XX}}^{-1}\boldsymbol{\Lambda}_{\boldsymbol{XY}}(\boldsymbol{y}-\boldsymbol{\mu}_Y)$
  - Therefore $f_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}|\boldsymbol{y}) \sim \mathcal{N}(\boldsymbol{\mu}_{X|Y}, \boldsymbol{\Sigma}_{X|Y})$
- Given this PDF, we can see that the MAP estimate is simply $\boldsymbol{\mu}_{X|Y}$
  - We can show that this is the same as the LMS estimate
  - However, we only have this in terms of the precision matrix; can we find it in terms of $\boldsymbol{\Sigma}$?
- Using the Schur complement on $\boldsymbol{\Sigma}^{-1}$ we can find a general expression for each of the $\boldsymbol{\Lambda}$
  - $\boldsymbol{\Lambda}_{\boldsymbol{XX}} = (\boldsymbol{\Sigma}_{\boldsymbol{XX}} - \boldsymbol{\Sigma}_{\boldsymbol{XY}}\boldsymbol{\Sigma}_{\boldsymbol{YY}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{YX}})^{-1}$
  - $\boldsymbol{\Lambda}_{\boldsymbol{XY}} = (\boldsymbol{\Sigma}_{\boldsymbol{XX}} - \boldsymbol{\Sigma}_{\boldsymbol{XY}}\boldsymbol{\Sigma}_{\boldsymbol{YY}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{YX}})^{-1}\boldsymbol{\Sigma}_{\boldsymbol{XY}}\boldsymbol{\Sigma}_{\boldsymbol{YY}}^{-1}$
  - Therefore $\boldsymbol{\mu}_{X|Y} = \boldsymbol{\mu}_X + \boldsymbol{\Sigma}_{\boldsymbol{XY}}\boldsymbol{\Sigma}_{\boldsymbol{YY}}^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_Y)$, $\boldsymbol{\Sigma}_{X|Y} = \boldsymbol{\Sigma}_{\boldsymbol{XX}} - \boldsymbol{\Sigma}_{\boldsymbol{XY}}\boldsymbol{\Sigma}_{\boldsymbol{YY}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{YX}}$
  - We get both the mean of the estimate and its sprea

Lecture 12, Feb 16, 2024

Gaussian Systems

- Let $\boldsymbol{X}$ be jointly Gaussian and let $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{b} + \boldsymbol{Z}$, where $\boldsymbol{Z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{Z}})$
  - Assume that $\boldsymbol{A}, \boldsymbol{b}$ are fixed and known, and $\boldsymbol{Z}, \boldsymbol{X}$ are independent (zero-mean, independent noise)
  - We would like to estimate $\boldsymbol{X}$ from $\boldsymbol{Y}$
- Again let $\boldsymbol{W} = \begin{bmatrix}\boldsymbol{X} \\ \boldsymbol{Y}\end{bmatrix} = \begin{bmatrix}\boldsymbol{X} \\ \boldsymbol{A}\boldsymbol{X} + \boldsymbol{b} + \boldsymbol{Z}\end{bmatrix} = \begin{bmatrix}\boldsymbol{1} & \boldsymbol{0} \\ \boldsymbol{A} & \boldsymbol{1}\end{bmatrix}\begin{bmatrix}\boldsymbol{X} \\ \boldsymbol{Z}\end{bmatrix} + \begin{bmatrix}0 \\ \boldsymbol{b}\end{bmatrix}$
  - Since $\boldsymbol{W}$ is obtained through a linear transformation from $\begin{bmatrix}\boldsymbol{X} \\ \boldsymbol{Z}\end{bmatrix}$, we know it is jointly Gaussian
  - We've converted this to the conditional PDF problem we found last time
  - $\hat{\boldsymbol{x}}_{\text{MAP/LMS}}(\boldsymbol{y}) = \boldsymbol{\mu}_{\boldsymbol{X}} + \boldsymbol{\Sigma}_{\boldsymbol{XY}}\boldsymbol{\Sigma}_{\boldsymbol{YY}}^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_{\boldsymbol{Y}})$
- $\boldsymbol{\mu}_{\boldsymbol{Y}} = E[\boldsymbol{A}\boldsymbol{X} + \boldsymbol{b} + \boldsymbol{Z}] = \boldsymbol{A}\boldsymbol{\mu}_{\boldsymbol{X}} + \boldsymbol{b}$
- $$\begin{aligned}\boldsymbol{\Sigma}_{\boldsymbol{XY}} &= E[(\boldsymbol{X}-\boldsymbol{\mu}_{\boldsymbol{X}})(\boldsymbol{A}\boldsymbol{X} + \boldsymbol{b} + \boldsymbol{Z} - \boldsymbol{A}\boldsymbol{\mu}_{\boldsymbol{X}} - \boldsymbol{b})^T] \\ &= E[(\boldsymbol{X}-\boldsymbol{\mu}_{\boldsymbol{X}})(\boldsymbol{X}-\boldsymbol{\mu}_{\boldsymbol{X}})^T\boldsymbol{A}^T] + E[(\boldsymbol{X}-\boldsymbol{\mu}_{\boldsymbol{X}}) + \boldsymbol{Z}] \\ &= \boldsymbol{\Sigma}_{\boldsymbol{X}}\boldsymbol{A}^T\end{aligned}$$
- $$\begin{aligned}\boldsymbol{\Sigma}_{\boldsymbol{YY}} &= E[(\boldsymbol{A}(\boldsymbol{X}-\boldsymbol{\mu}_{\boldsymbol{X}}) + \boldsymbol{Z})(\boldsymbol{A}(\boldsymbol{X}-\boldsymbol{\mu}_{\boldsymbol{X}}) + \boldsymbol{Z})^T] \\ &= \boldsymbol{A}\boldsymbol{\Sigma}_{\boldsymbol{X}}\boldsymbol{A}^T + \boldsymbol{\Sigma}_{\boldsymbol{Z}}\end{aligned}$$

- Substituting these in we get $\hat{\boldsymbol{x}}_{\text{MAP/LMS}} = \boldsymbol{\mu_X} + \boldsymbol{\Sigma_X A}^T(\boldsymbol{A\Sigma_X A}^T + \boldsymbol{\Sigma_Z})^{-1}(\boldsymbol{y} - \boldsymbol{A\mu_X} - \boldsymbol{b})$
$= (\boldsymbol{\Sigma_X}^{-1} + \boldsymbol{A}^T\boldsymbol{\Sigma_Z}^{-1}\boldsymbol{A})^{-1}(\boldsymbol{A}^T\boldsymbol{\Sigma_Z}^{-1}(\boldsymbol{y} - \boldsymbol{b}) + \boldsymbol{\Sigma_X}^{-1}\boldsymbol{\mu_X})$
  - Note the second form can be derived using the matrix inversion formula
    - It only uses the inverse covariance (precision) matrices
  - Also $\boldsymbol{\Sigma}_{X|y} = (\boldsymbol{\Sigma_X}^{-1} + \boldsymbol{A}^T\boldsymbol{\Sigma_Z}^{-1}\boldsymbol{A})^{-1}$
- Example: consider $\begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \Theta + \begin{bmatrix} W_1 \\ \vdots \\ W_n \end{bmatrix} \iff \boldsymbol{Y} = \boldsymbol{A}X + \boldsymbol{Z}$
  - Let $\theta \sim \mathcal{N}(\boldsymbol{x}_0, \sigma_0^2)$ and $w_i \sim \mathcal{N}(0, \sigma_i^2)$
  - θ is some true value, plus zero-mean Gaussian noise w_i; we measure this n times
  - Compute terms:
    - $\boldsymbol{b} = \boldsymbol{0}$
    - $\boldsymbol{\mu_X} = x_0$
    - $\boldsymbol{\mu_Y} = \boldsymbol{A\mu_X} + \boldsymbol{b} = \begin{bmatrix} x_0 \\ \vdots \\ x_0 \end{bmatrix}$
    - $\boldsymbol{\Sigma_{YY}} = \boldsymbol{A\Sigma_X A}^T + \boldsymbol{\Sigma_Z} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \sigma_0^2 \begin{bmatrix} 1 & \dots & 1 \end{bmatrix} + \begin{bmatrix} \sigma_1^2 & & \\ & \vdots & \\ & & \sigma_n^2 \end{bmatrix}$
    - $\boldsymbol{\Sigma_{XY}} = \boldsymbol{\Sigma_X A}^T = \begin{bmatrix} \sigma_0^2 & \dots & \sigma_0^2 \end{bmatrix}$
  - Substituting, we get $\dfrac{\frac{x_0}{\sigma_0^2} + \sum_{i=1}^n \frac{x_i}{\sigma_i^2}}{\frac{1}{\sigma_0^2} + \sum_{i=1}^n \frac{1}{\sigma_i^2}}$
    - This is the same result that we would get through regular MAP estimation

Lecture 13, Feb 26, 2024

Linear Regression

- Consider a linear model $Y = \boldsymbol{w}^T\boldsymbol{X} + Z$ where we have n noisy measurements y_i from n inputs $\boldsymbol{x}_i$
  - Assume Z is some IID Gaussian random noise
  - Given these measurements, our goal is to find the best set of weights $\boldsymbol{w}^T = \begin{bmatrix} w_1 & \dots & w_D \end{bmatrix}$
  - Each weight w_j corresponds to the jth coefficient of $\boldsymbol{x}$, which has dimension D
- Form the *design matrix* $\begin{bmatrix} \boldsymbol{x}_1^T & y_1 \\ \vdots & \vdots \\ \boldsymbol{x}_n^T & y_n \end{bmatrix}$
- Consider the MLE $\hat{\boldsymbol{w}}_{\text{ML}} = \operatorname*{argmax}_{\boldsymbol{w}\in\mathbb{R}^D} \log p((\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)|\boldsymbol{w})$

$$\begin{aligned}
&= \operatorname*{argmax}_{\boldsymbol{w}\in\mathbb{R}^D} \log \prod_{i=1}^n p(\boldsymbol{x}_i, y_i|\boldsymbol{w}) \\
&= \operatorname*{argmax}_{\boldsymbol{w}\in\mathbb{R}^D} \sum_{i=1}^n \log\left(\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(y_i - \boldsymbol{w}^T\boldsymbol{x}_i)^2}\right) \\
&= \operatorname*{argmax}_{\boldsymbol{w}\in\mathbb{R}^D} -\sum_{i=1}^n (y_i - \boldsymbol{w}^T\boldsymbol{x}_i)^2 \\
&= \operatorname*{argmin}_{\boldsymbol{w}\in\mathbb{R}^D} \boldsymbol{e}^T(\boldsymbol{w})\boldsymbol{e}(\boldsymbol{w})
\end{aligned}$$

- Where the error vector is $\boldsymbol{e}(\boldsymbol{w}) = \begin{bmatrix} y_1 - \boldsymbol{w}^T\boldsymbol{x}_1 \\ \dots \\ y_n - \boldsymbol{w}^T\boldsymbol{x}_n \end{bmatrix} = \boldsymbol{y} - \begin{bmatrix} \boldsymbol{x}_1^T \\ \vdots \\ \boldsymbol{x}_n^T \end{bmatrix} \boldsymbol{w}$
- This is now a *least squares regression problem*
- Let $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^T \\ \vdots \\ \boldsymbol{x}_n^T \end{bmatrix}$ then we have $\underset{\boldsymbol{w}\in\mathbb{R}^D}{\operatorname{argmin}} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})$
- Expand: $\frac{1}{2}\boldsymbol{w}^T\boldsymbol{X}\boldsymbol{X}\boldsymbol{w} - \boldsymbol{w}\boldsymbol{X}^T\boldsymbol{y} + \frac{1}{2}\boldsymbol{y}^T\boldsymbol{y}$ (note a factor of $\frac{1}{2}$ was added)
- Derivative: $\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \boldsymbol{X}^T\boldsymbol{y} = 0 \implies \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} = \boldsymbol{X}^T\boldsymbol{y}$
- Therefore $\hat{\boldsymbol{w}}_{\text{ML}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$

- Another way to write this is $\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}) = 0$, meaning we can interpret this as making the error vector orthogonal to all the input data
  - This is the *normal equation*
- $\boldsymbol{X}^T\boldsymbol{X}$ is the *scatter matrix*
  - This is an estimate of the covariance/correlation matrix of the data
- Regression can be performed in any general vector space, so our model can be nonlinear in $\boldsymbol{x}$ (but still linear in $\boldsymbol{w}$)
  - In general given any basis function $\boldsymbol{\phi}(\boldsymbol{x}_i)$ we can try to fit $y_i = \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_i) + z_i$
  - Let $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{\phi}^T(\boldsymbol{x}_1) \\ \vdots \\ \boldsymbol{\phi}^T(\boldsymbol{x}_n) \end{bmatrix}$ then $\hat{\boldsymbol{w}}_{\text{ML}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$
  - e.g. we can work in the vector space of polynomials to perform polynomial regression, or the space of sinusoids for a Fourier series
    - For d-degree polynomial regression we'd have $\boldsymbol{\phi}^T(x_i) = \begin{bmatrix} 1 & x_i & x_i^2 & \dots & x_i^d \end{bmatrix}$
  - Example: measuring the height of a cannonball h_i vs. time t_i for $i = 1, \dots, n$
    - Use the model $h_i = w_1 t_i + w_2 t_i^2 + z_i = \boldsymbol{w}^T\boldsymbol{x}_i + z_i$ where $\boldsymbol{x}_i = \begin{bmatrix} t_i \\ t_i^2 \end{bmatrix}$

Bayesian Regression – Regularization

- If we make the model too complex, i.e. too high of a dimension for $\boldsymbol{\phi}$, we will get overfitting
- Typically when the model overfits, we get very large weights that are not physically realistic for our system
  - To keep the weights down, we can use regularization
  - Here we show a way to derive the same result by instead assuming a prior on $\boldsymbol{w}$
- Assume that each weight has a prior $w_i \sim \mathcal{N}(0, \tau^2)$; now can find the MAP estimate
- $$\begin{aligned} \hat{\boldsymbol{w}}_{\text{MAP}} &= \underset{\boldsymbol{w}\in\mathbb{R}^D}{\operatorname{argmax}}\, p((\boldsymbol{\phi}(x_1), y_1), \dots, (\boldsymbol{\phi}(x_n), y_n))p(\boldsymbol{w}) \\ &= \underset{\boldsymbol{w}\in\mathbb{R}^D}{\operatorname{argmax}} \sum_{i=1}^{n} \left[\log\left(\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(y_i - \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_i))^2} \right) + \sum_{j=1}^{D} \log \frac{1}{\sqrt{2\pi}\tau} e^{-\frac{w_j^2}{2\tau^2}} \right] \\ &= \underset{\boldsymbol{w}\in\mathbb{R}^D}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_i))^2 + \frac{\sigma^2}{\tau^2}\|\boldsymbol{w}\|^2 \\ &= \underset{\boldsymbol{w}\in\mathbb{R}^D}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_i))^2 + \lambda\|\boldsymbol{w}\|^2 \end{aligned}$$
  - The first term is the same least squares term as before, but now we have an additional term that penalizes the norm of $\boldsymbol{w}$, effectively keeping the weights small

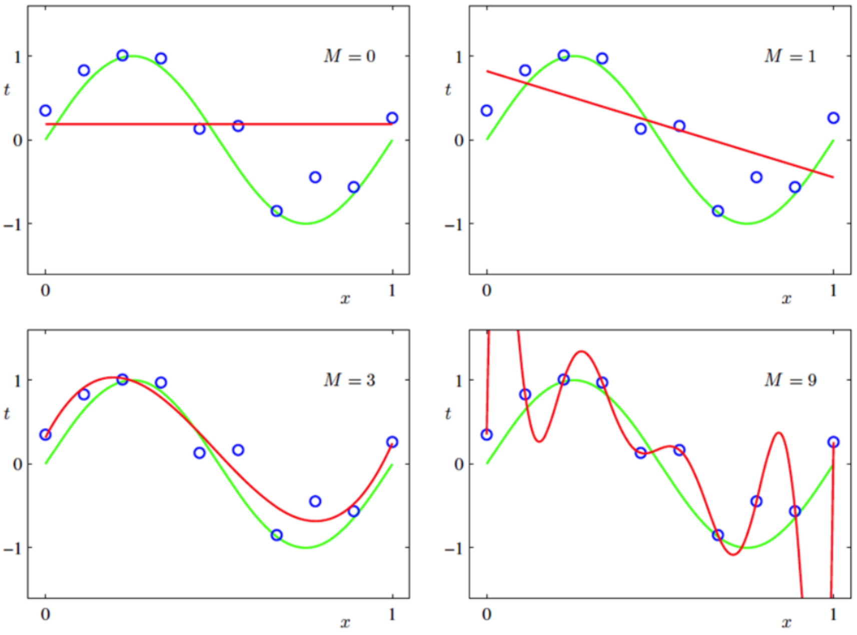


Figure 3: Polynomial regression for different degrees. Green is the underlying function we're trying to approximate.

- Let $\boldsymbol{e}(\boldsymbol{w}) = \begin{bmatrix} y_1 \\ \vdots \\ y_n \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\phi}^T(\boldsymbol{x}_1) \\ \vdots \\ \boldsymbol{\phi}^T(\boldsymbol{x}_n) \\ -\sqrt{\lambda}\mathbf{1} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_D \end{bmatrix}$ and $\tilde{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{\phi}^T(\boldsymbol{x}_1) \\ \vdots \\ \boldsymbol{\phi}^T(\boldsymbol{x}_n) \\ -\sqrt{\lambda}\mathbf{1} \end{bmatrix}$
  - * The error can again be written as $\boldsymbol{e}^T(\boldsymbol{w})\boldsymbol{e}(\boldsymbol{w})$
- Using the same derivation as before, $\hat{\boldsymbol{w}}_{\text{MAP}} = (\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{X}})^{-1}\tilde{\boldsymbol{X}}^T \begin{bmatrix} \boldsymbol{y} \\ \mathbf{0} \end{bmatrix} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\mathbf{1})^{-1}\boldsymbol{X}^T\boldsymbol{y}$
- Notice that this result is almost the same as the MLE solution, except with the addition of $\lambda\mathbf{1}$
- This is known as *ridge regression*

• We can also solve this by writing it as a Gaussian system
  - $\begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_1^T \\ \vdots \\ \boldsymbol{x}_n^T \end{bmatrix} \begin{bmatrix} W_1 \\ \vdots \\ W_D \end{bmatrix} + \begin{bmatrix} Z_1 \\ Z_n \end{bmatrix} \iff \boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{z}$
  - $\hat{\boldsymbol{w}}_{\text{MAP}}(\boldsymbol{y}) = (\boldsymbol{\Sigma}_W^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}_Z^{-1}\boldsymbol{X})^{-1}\left(\boldsymbol{X}^T\frac{1}{\sigma^2}\boldsymbol{y} + \frac{1}{\tau^2}\cdot 0\right) = (\boldsymbol{X}^T\boldsymbol{X} + \frac{\sigma^2}{\tau^2}\mathbf{1})^{-1}\boldsymbol{X}^T\boldsymbol{y}$
  - This gives us the conditional precision $\boldsymbol{\Sigma}_{X|Y}^{-1} = \frac{1}{\tau^2}\mathbf{1} + \frac{1}{\sigma^2}\boldsymbol{X}^T\boldsymbol{X}$
  - So $\boldsymbol{\Sigma}_{X|Y} = \sigma^2\left(\boldsymbol{X}^T\boldsymbol{X} + \frac{\sigma^2}{\tau^2}\mathbf{1}\right)^{-1}$
    - * Notice that $\boldsymbol{X}^T\boldsymbol{X} + \frac{\sigma^2}{\tau^2}\mathbf{1}$ is related to the covariance
    - * As we collect more data the added term becomes negligible
    - * $\boldsymbol{X}^T\boldsymbol{X}$ becomes bigger so the covariance shrinks

Lecture 14, Mar 8, 2024

Logistic Regression

- Try to estimate $P[Y = i|\boldsymbol{x}, \boldsymbol{w}]$ where $\boldsymbol{y} \in C = \{1, 2, \dots, c\}$ are classes, $\boldsymbol{x}$ is a feature, and $\boldsymbol{w}$ are linear model weights
- Example: binary hypothesis ($c = 2$) with Bernoulli probabilities
    - Then $p(y = 1|x) = \dfrac{p(x|y = 1)p(y = 1)}{p(x|y = 1)p(y = 1) + p(x|y = 0)p(y = 0)} = \dfrac{1}{1 + \frac{p(x|y=0)p(y=0)}{p(x|y=1)p(y=1)}}$
    - We can write this as $\dfrac{1}{1 + e^{-\alpha}}$ where $\alpha = \log \dfrac{p(x|y = 0)p(y = 0)}{p(x|y = 1)p(y = 1)}$
- $\sigma(\alpha) = \dfrac{1}{1 + e^{-\alpha}}$ is the *sigmoid function*, which maps $\mathbb{R} \to (0, 1)$ which is useful for probabilities
    - Has an S shape with value of $\dfrac{1}{2}$ at 0
    - Note $\sigma(-\alpha) = 1 - \sigma(\alpha)$ and $\alpha = \log \dfrac{\sigma(\alpha)}{1 - \sigma(\alpha)}$
    - $\dfrac{\mathrm{d}\sigma}{\mathrm{d}\alpha} = \sigma(\alpha)(1 - \sigma(\alpha))$
    - We can classify $\hat{y} = 1$ if $\sigma(\alpha) > \dfrac{1}{2}$ or $\hat{y} = 0$ otherwise
- Our model is then $\hat{p}(y = 1|\boldsymbol{x}) = \dfrac{1}{1 + e^{-\boldsymbol{w}^T\boldsymbol{x}}} = \sigma(\boldsymbol{w}^T\boldsymbol{x})$, where we try to find the best weights $\boldsymbol{w}$
- Compared to Gaussian discriminant analysis, which has $2D$ for means and $D(D + 1)/2$ for covariances and priors, we only have D parameters and a lot less computation overall
- Consider a Bernoulli trial with $\theta = P[y = 1]$, so $P[y] = \theta^y(1 - \theta)^{1-y}$
    - Let $\theta = P[y = 1|\boldsymbol{x}, \boldsymbol{w}] = \sigma(\boldsymbol{w}^T\boldsymbol{x})$
    - For n trials, the NLL is $-\log \prod_{i=1}^{n} P[y_i|\boldsymbol{x}_i, \boldsymbol{w}] = -\sum_{i=1}^{n} \log(\theta_i^y(1 - \theta_i)^{1-y_i})) = -\sum_{i=1}^{n} y_i \log \theta_i + (1 - y_i)\log(1 - \theta_i)$ where $\theta_i = \sigma(\boldsymbol{w}^T\boldsymbol{x}_i) = \sigma_i$
    - This is the *cross-entropy* loss function
    - $\dfrac{\mathrm{d}}{\mathrm{d}w}\mathrm{NLL} = -\sum_{i=1}^{n}\left(y_i\dfrac{1}{\theta_i}\theta_i' + (1 - y_0)\dfrac{1}{1 - \theta_i}(-\theta_i')\right) = -\sum_{i=1}^{n}\left(y_i\dfrac{\theta_i'}{\theta_i} - (1 - y_0)\dfrac{\theta_i'}{1 - \theta_i}\right) = 0$
    - $\theta_i' = \sigma_i(1 - \sigma_i)\dfrac{\mathrm{d}}{\mathrm{d}w_j}\boldsymbol{w}^T\boldsymbol{x}_i = \sigma_i(1 - \sigma_i)x_{ij}$
    - $\dfrac{\theta_i'}{\theta_i} = (1 - \sigma_i)x_{ij} \implies \dfrac{\theta_i'}{1 - \theta_i'} = \sigma_i x_{ij}$
    - Therefore $\dfrac{\mathrm{d}}{\mathrm{d}w}\mathrm{NLL} = \sum_{i=1}^{n}(y_i - \sigma_i)x_{ij} = 0$
        - This can be interpret as the error multiplied by the observation
    - No closed-form solution; we can use methods such as gradient descent
        - The gradient vector is $\sum_{i=1}^{n}(y_i - \sigma_i)\boldsymbol{x}_i$
- Just like in linear regression, we're not restricted to just a single basis; we can change to e.g. a polynomial basis
    - Change of basis can make the space more linearly separable
    - Sometimes the problem is unsolvable as-is due to the data not being linearly separable
- For multiple classes, $p(c_k|\boldsymbol{x}, \boldsymbol{w}) = \dfrac{e^{\alpha_k}}{\sum_i e^{-\alpha_i}}$ where $\alpha_k = \boldsymbol{w}_k^T\boldsymbol{x}$
    - This is a softmax
    - This reduces to the same sigmoid function if we only have 2 classes
- We can also replace the sigmoid with tanh (and rescale to between 0 and 1)

Lecture 15, Mar 11, 2024

Discrete-Time Markov Chains

- A *Markov chain* is a discrete-valued random sequence X_n where the future is of the process given the present is independent of the past, i.e. $P[X_{n+1}|X_1, \dots, X_n] = P[X_{n+1}|X_n]$
  - The present X_n is known as the *state*
- Example: sum process $S_n = X_1 + \cdots + X_n = S_{n-1} + X_n$ where X_i are IID, $S_0 = 0$
  - $P[S_{n+1} = s_{n+1}|S_n = s_n, \dots, S_1 = s_1] = P[X_n = s_{n+1} - s_n] = P[S_{n+1} = s_{n+1}|S_n = s_n]$
- $P[X_3, X_2, X_1] = P[X_3|X_2]P[X_2|X_1]P[X_1]$ due to the Markov property
  - The latter is a lot easier to store since we don't have to go over all possible comminations of the 3 variables
  - $P[X_3|X_2]$ and $P[X_2|X_1]$ are *transition probabilities*
  - $P[X_1]$ is the *initial probability*
  - The joint PMF of the values is the product of the initial probability and all intermediate transition probabilities
- These transition probabilities could be time-dependent, but often they are constant, in which case the Markov chain has *homogeneous transition probabilities*
  - We only need to store a single version of the transition probability matrix
- X_n is completely specified by $p_i(0)$ and the *transition probability matrix*, where the entry ij denotes the probability of transitioning to state j while in state i
  - Each row sums up to 1 since it s a PMF
  - We can also use a *state transition diagram* to visualize this
- Example: speech activity using a Markov model; if packet n was silent, then the probability of silence in the next packet is $1 - \alpha$ and probability of speech activity is α
  - Transition matrix: $P = \begin{bmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{bmatrix}$

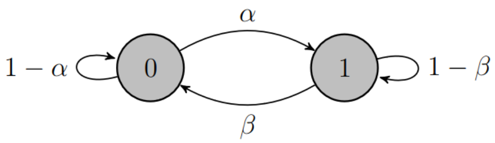


Figure 4: State transition diagram for example.

- The transition probability matrix after n steps, $P(n)$, is the original single-step transition matrix raised to the power n; $p_{ij}(n)$ denotes the probability of transition for n steps
  - $P(n) = P(n-1)P = P^n$
  - Suppose we start with some initial state PMF $\boldsymbol{p}(0)$, then after n steps the new distribution will be $\boldsymbol{p}(n) = \boldsymbol{p}(0)P^n$
  - To find a closed-form expression we can diagonalize P
- Some Markov chains will have $\lim_{n\to\infty} P(n)$ exist; this will give the asymptotic or steady-state PMF
- Instead of finding an expression for P^n, we can find the steady-state PMF more directly, assuming one exists
  - Let the steady-state PMF be $\pi = (\pi_1, \dots, \pi_n)$ and so $p_{ij}(n) = \pi_j$
  - To find the steady-state PMF we solve $\boldsymbol{\pi} = \boldsymbol{\pi}P$ such that $\sum_i \pi_i = 1$
    - Since $\boldsymbol{\pi}$ is a PMF, the first equation only gives $n - 1$ independent equations
    - i.e. when we have this PMF, it remains unchanged by the Markov chain
  - These together are known as the *global balance equations*

Recurrence Relations in Markov Chains

- When does a Markov chain have steady-state probabilities?

- We can break states into separate *classes*, where each one is of a different type
  - State j is *accessible* from i if there is a sequence of transitions from i to j with nonzero probability
  - States i and j *communicate* if they are accessible from each other (i.e. we can go from i to j and back); this is denoted $i \leftrightarrow j$
    - * A state always communicates with itself by definition (even if there are no self edges)
    - * $i \leftrightarrow j, j \leftrightarrow k \implies i \leftrightarrow k$
  - States that communicate with each other are in the same *class*
    - * States in the same class share the same fate – they have the same limiting behaviour
  - Classes are always disjoint (i.e. one state cannot be in two different classes; in that case the two classes would communicate with each other so they would be the same)
    - * However, states in different classes aren't necessarily independent, since we can still have one-way accessible connections
- The states of a Markov chain consists of one or more disjoint classes; if it has a single class, it is *irreducible*
  - Intuitively this means that we can go from one state to any other state
- A state is *periodic* with period d if it can only re-occur at times that are multiples of d, i.e. $p_{ii}(n) = 0$ if n is not a multiple of d
  - If the Markov chain is periodic, it's similar to having multiple chains
  - Let $\tau(x)$ be all the possible times that we can visit x (starting from x at time 0); the period of x is the GCD of $\tau(x)$ (same across an entire class)
  - If all states/classes have period 1, the chain is *aperiodic*
- A Markov chain that is irreducible and aperiodic will converge to a stationary distribution (we see this later)

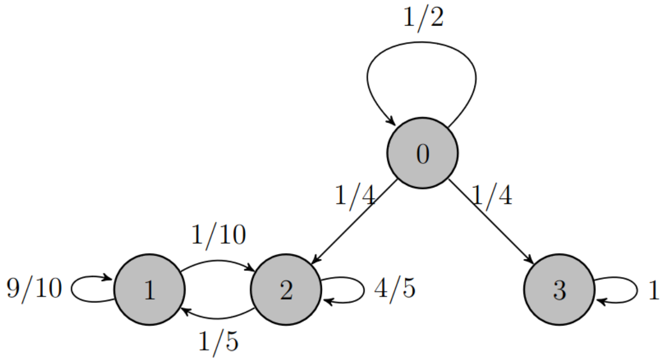


Figure 5: State transition diagram for a Markov chain with 3 classes: $\{0\}, \{1, 2\}, \{3\}$. This is aperiodic.

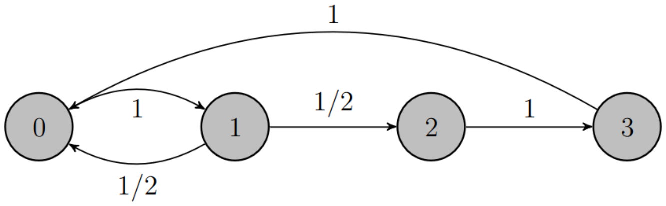


Figure 6: State transition diagram for a Markov chain with a single class. This is periodic with period 2.

- A state i is *recurrent* if the process returns to the state with probability 1, or *transient* if the probability is less than 1
  - For any recurrent state, if we leave it, we know eventually we will come back
  - Whenever we come back we will go through the exact same cycle again
  - For a transient state, when we leave it, it's possible that we won't reach this state again

- Recurrence/transience is a class property; if a state is recurrent then all states in its class will also be recurrent
- To find if a state is recurrent, we sum the probability of returning to the state after all possible number of steps
  - State i is recurrent iff $\displaystyle\sum_{n=1}^{\infty} p_{ii}(n) = \infty$
  - State i is transient iff $\displaystyle\sum_{n=1}^{\infty} p_{ii}(n) < \infty$
  - e.g. if $p_{00}(n) = \left(\dfrac{1}{2}\right)^n$, then the sum converges to 2 so it is finite and the state is transient

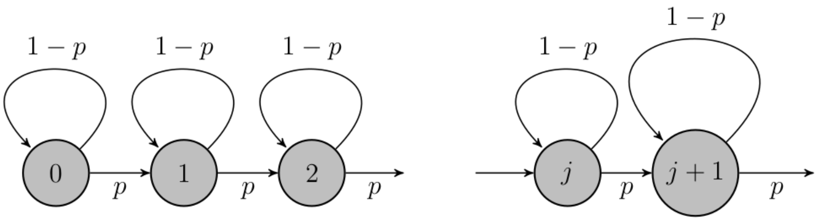


Figure 7: Binomial counting process state transition diagram. Each state is its own class and every class/state is transient.

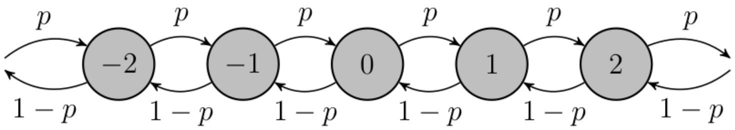


Figure 8: Random walk process state transition diagram. All states are in the same class. This is also periodic.

Lecture 16, Mar 15, 2024

Convergence of Markov Chains

- We would like to know the following:
  1. When does a Markov chain have steady-state probabilities?
  2. Are the steady-state probabilities unique?
  3. When does P^n converge to the steady-state probabilities?
- Suppose that state i is recurrent; let $T_i(k)$ be the number of steps between the kth visit and $k-1$th visit
  - The proportion of time spent in i is $\dfrac{k}{T_i(1) + \cdots + T_i(k)}$
  - Note that this is the reciprocal of $\displaystyle\frac{1}{k}\sum_{j=1}^{k} T_i(j) \to E[T_i]$
    - * This is because the return times T_i are IID
  - Therefore this converges to $\dfrac{1}{E[T_i]} = \pi_i$
    - * This satisfies the global balance equations
- State i is *positive recurrent* if $E[T_i] < \infty$; in this case all states in the class will have nonzero probability since $\pi_i > 0$
  - A positive recurrent, aperiodic state is *ergodic*

- State i is *null recurrent* if $E[T_i] = \infty$; in this case $\pi_i = 0$ and states will have zero probability
    - This arises with infinite state Markov chains
    - Even though we always come back (the state is recurrent), if the chain is infinite, it is possible to have zero probability if we don't return often enough

Theorem

For an irreducible (single class), aperiodic, and positive recurrent Markov chain,

$$\lim_{n \to \infty} p_{jj}(n) = \pi_j \quad \forall j$$

that is, there is a steady-state PMF, obtainable by solving the global balance equations.

Theorem

For an irreducible, periodic, and positive recurrent Markov chain with period d,

$$\lim_{n \to \infty} p_{jj}(nd) = d\pi_j \quad \forall j$$

where the "steady-state" probabilities exist over the period instead of individual steps.

- The factor of d comes in because there are only $\frac{1}{d}$ of the total states that can occur
- For a periodic Markov chain the transition probabilities form multiple sub-matrices

Lecture 17, Mar 18, 2024

Bayesian Networks

- Given $X_1, \dots, X_n$ where each $X_i \in \{1, 2, \dots, S\}$, then fully specifying $P[X_1, \dots, X_n]$ requires specifying $S^n - 1$ values, which is very expensive
- However, we can reduce this if not all the variables depend on each other
    - e.g. if X_n is Markov then we only need to specify the transition probability matrix and initial PMF, which is only $S + S^2 - 1$ values
- We model the dependence relationship between variables as a graph, where the nodes are random variables and a directed edge from X to Y means Y depends on X; these are known as *Bayesian networks*
    - e.g. for a Markov chain, the graph is just a long chain, since each variable only depends on the previous one
    - We assume that the inter-dependencies among RVs can be factorized into the form $p(y|\text{pa}\{y\})$, where $\text{pa}\{y\}$ denotes all parents of y
    - Note an edge from $X \to Y$ does not mean X does not depend on Y, but we specify the conditional probabilities as $P[Y|X]$ instead of $P[X|Y]$
    - This can be used to describe causal relationships (but causality is not necessary in a graph)
- These graphs can have complex relationships that are not necessarily linear like a Markov chain, so we can no longer just specify state transition matrices
    - In general these are DAGs
    - The more parents a node has, the more information we need to fully specify its conditional probability, since it depends on more things
        * For a node that depends on k other nodes, we need to specify S^k values; this is often much less than S^n
        * The time/space complexity of storing a Bayesian network is therefore related to the node with the most connections
- We are interested in the conditional independence between RVs in the network – given some set of

variables, are two variables conditionally independent?
  - e.g. in a Markov chain, given the present state, all future states are independent of all past states
  - Recall that conditional independence of a and b given c means $p(a|b,c) = p(a|c) \iff p(a,b|c) = p(a|c)p(b|c)$
- Consider $p(a,b|c) = p(a|b,c)p(b|c) = p(a|c)p(b|c)$; graphically this corresponds to c pointing to a and b, with no edge between a and b
  - In this configuration c is known as a *tail-to-tail* node
  - Without observing c, a and b are dependent; if c is given, then they are independent
  - We can think of c as a "gate" – when it's nonblocking/not observed, we have dependence between a and b; but when it is observed, it "blocks" the dependence between a and b

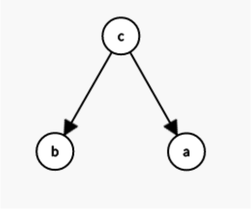


Figure 9: Tail-to-tail configuration.

- Consider another configuration, where a points to c points to b
  - Without c, a and b are dependent: $p(a,b) = \sum_{c'} p(a,b,c') = p(a) \sum_{c'} p(b|c')p(c'|a) = p(a)p(b|a)$
  - If we observe c, a and b are independent: $p(a,b|c) = \dfrac{p(a)p(c|a)p(b|c)}{p(c)} = \dfrac{p(a,c)p(b|c)}{p(c)} = p(a|c)p(b|c)$
  - Having c again blocks the dependence between a and b
  - This is known as *head-to-tail* configuration

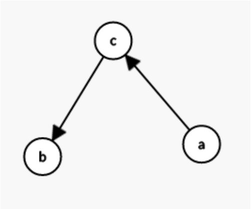


Figure 10: Head-to-tail configuration.

- Now consider a pointing to c and b pointing to c
  - Now without c, a and b are independent since $p(a,b) = \sum_{c'} p(a)p(b)p(c|a,b) = p(a)p(b) \sum_{c'} p(c|a,b) = p(a)p(b)$
  - However, once c is observed, a and b are no longer independent since $p(a,b|c) = \dfrac{p(a)p(b)p(c|a,b)}{p(c)} \neq p(a)p(b)$
  - This is the opposite of the two previous cases; not having c blocks the dependence
  - This is a *head-to-head* configuration
- For n random variables, $p(x_1,\dots,x_n) = p(x_n|x_1,\dots,x_{n-1})p(x_{n-1}|x_1,\dots,x_{n-2})\dots p(x_2|x_1)p(x_1)$
  - In this form, the relation holds for all random variables; to simplify, we need to make assumptions about independence, i.e. removing edges so that the graph is more sparse
- If we have a Bayesian network then we can show $p(x_1,\dots,x_n) = \prod_{i=1}^{n} p(x_i|\text{pa}\{x_i\})$
  - The joint probability distribution is the product of the PMF of each variable given its parents

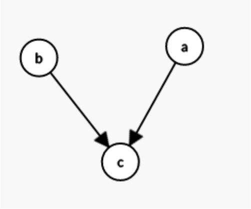


Figure 11: Head-to-head configuration.

- Let A, B, C be disjoint sets of nodes in a DAG; a path from A to B is *blocked* with respect to C if the path passes through a node in C that is not head-to-head, or it passes through a head-to-head node for which neither the node nor its descendants are in C
  - If every path from A to B is blocked, then A is *D-separated* (*directed separated*)
  - If A and B are D-separated, then all nodes in A are independent of all nodes in B given all nodes in C

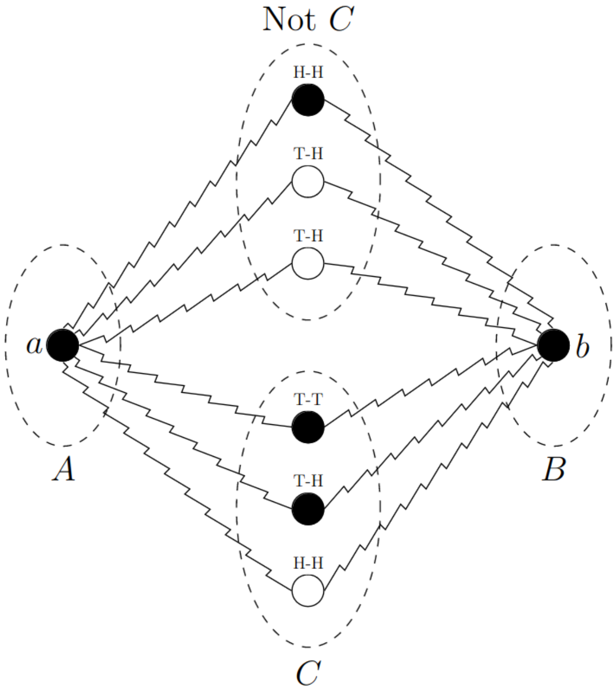


Figure 12: D-separation.

- Example: consider the network in the figure below
  - Are a and b independent given c?
    - * The path from a passes through e (head-to-head) and f (tail-to-tail)
    - * e is head-to-head, and its descendant c is given; f is tail-to-tail but it is not given
    - * Therefore a and b are not independent
  - Are a and b independent given f?
    - * f is tail-to-tail and it is given; e is head-to-head and neither it nor its descendants are given
    - * Therefore a and b are independent

- For $x_1, \ldots, x_n$, consider x_i given all others x_j
  - $p(x_i|\boldsymbol{x}_{j\neq i}) = \dfrac{p(x_1, \ldots, x_n)}{\sum_{x_i} p(x_1, \ldots, x_n)} = \dfrac{\prod_k p(x_k|\text{pa}\{x_k\})}{\sum_{x_i} \prod_{k'} p(x'_k|\text{pa}\{x'_k\})}$
  - Any factor in the denominator that does not depend on x_i can be taken out of the summation,

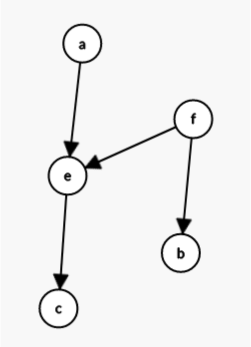


Figure 13: Example Bayesian network.

  which cancels a corresponding term in the numerator
  - The terms that remain are $p(x_i|\text{pa}\{x_i\})$, and all (direct) children of x_i, and co-parents of these descendants
  - These terms are known as the *Markov blanket* and separates x_i from the rest of the nodes

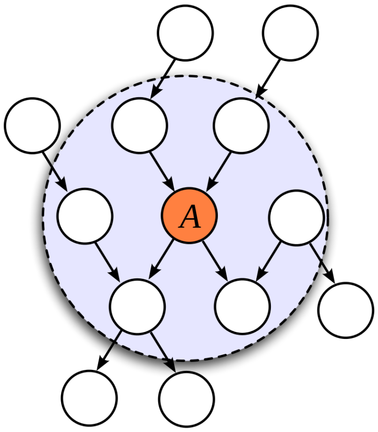


Figure 14: Illustration of the Markov blanket.

Lecture 18, Mar 22, 2024

Markov Random Fields (Undirected Graphs)

- Unlike Bayesian networks, these graphs are undirected, so we no longer have to worry about subtleties such as head-to-head nodes
- An edge does not necessarily indicate dependence, but rather related behaviour between nodes; conditional independence depends on path connectivity
- Factorization is done differently

- We want the conditional independence property: given disjoint sets of nodes A, B, and C, where C is observed
  - If all paths between A and B pass through C, then they are conditionally independent
  - If at least one path is not blocked, then conditional independence is not guaranteed
  - Alternatively we can remove all nodes in C and check for connectivity between the two sets
- How should we factor the probabilities so that we get the above properties?
- The Markov blanket in the case of Markov random fields is just the immediate neighbours of the node (no more descendants or co-parents)

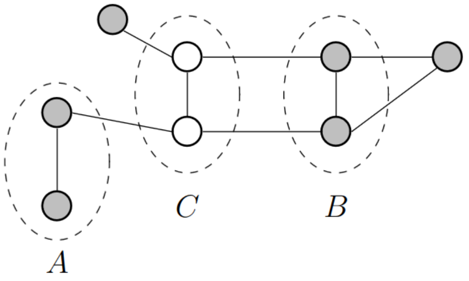


Figure 15: Example undirected graph.

- Consider x_i, x_j; suppose that they are conditionally independent, then $p(x_i, x_j | x_{\setminus\{i,j\}}) = p(x_i|x_{\setminus\{i,j\}})p(x_j|x_{\setminus\{i,j\}})$
  - This requires that there is no direct path between x_i and x_j and all other paths are blocked
  - x_i and x_j cannot be in the same factor
- A *clique* is a subset of nodes where all pairs are connected by a link (i.e. they're all direct neighbours)
  - A *maximal clique* is a subset of nodes where no additional node can be added while remaining a clique
  - Every maximal clique must form its own factor, since the nodes inside it cannot be separated by intermediate nodes, so they are not independent
- The joint distribution of all $\boldsymbol{x}$ is a product of the *potential function* on all the maximal cliques
  - $p(\boldsymbol{x}) = \dfrac{1}{Z}\prod_C \psi_C(\boldsymbol{x}_C)$ where Z is a normalization and C are the maximal cliques
    - $Z = \sum_{\boldsymbol{x}}\prod_C \psi_C(\boldsymbol{x}_C)$ is the partition function
    - We do this over maximal cliques because as per the discussion above, nodes in a maximal clique must all be in the same factor, because they are directly connected
  - The potential functions ψ_C are all nonnegative, but they need not be conditional PDFs
  - In this way we factorize the joint distribution
- The *Hammerly-Clifford theorem* states that we can always construct these distributions this way over maximal cliques
- Since potentials are exponential, we express $\psi_C(\boldsymbol{x}_C) = e^{-E(\boldsymbol{x}_C)}$
  - $E(\boldsymbol{x}_C)$ is the *energy*
- Therefore the joint distribution is $p(\boldsymbol{x}) = \dfrac{1}{Z}\prod_C e^{-E(\boldsymbol{x}_C)} = \dfrac{1}{Z}\exp\left(-\sum_C E(\boldsymbol{x}_C)\right)$
  - Note that the energy function for each clique is possibly different
  - To maximize the joint probability, we need to minimize the total energy $\sum_C E(\boldsymbol{x}_C)$
- Example: suppose we scan a monochrome image (each pixel $x_i \in \{1, -1\}$), and we get $y_i \in \{1, -1\}$; the process introduces some noise which possibly flips the pixels, so we would like to denoise the image by recovering x_i from y_i
  - We assume that for the most part, $x_i = y_i$ and noise occurs relatively rarely
  - The pixels are scanned in a rectangular grid; we assume that adjacent pixels tend to have the same sign
    - The maximal cliques in the image are adjacent pixels, and each x_i with its corresponding y_i
    - Each pair will have its own potential function
  - Consider $\psi(x_i, y_i) = e^{-\eta x_i y_i}$ and $\psi(x_i, x_j) = e^{-\beta x_i x_j}$
    - This is defined so that if x_i, y_i (or x_i, x_j) have the same value/sign, the potential is lower than the case of the pixels having different signs
    - The more frequent case of the pixels being the same sign has a lower potential
    - η and β are relative weightings
  - Let $\psi(x_i) = e^{-hx_i}$, which biases the pixels (if we know that there are more +1s than -1s or

otherwise)

  - $E(\boldsymbol{x}, \boldsymbol{y}) = h\sum_i x_i - \beta\sum_{i,j} x_i x_j - \eta\sum_i x_i y_i$ and $p(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{Z}e^{-E(\boldsymbol{x},\boldsymbol{y})}$
  - Now given $\boldsymbol{y}$, we wish to find $\boldsymbol{x}$ that minimizes the energy $E(\boldsymbol{x}, \boldsymbol{y})$
  - In this case, we do it by brute force:
    * Set $x_i = y_i$ for all i initially
    * Select a pixel x_i to change to the opposite polarity, and keep the change if the energy is reduced
    * Continue until a local minimum or maximum iterations is reached

Directed to Undirected Graphs

- Suppose we have a simple Markov chain with each X_i pointing to X_{i+1}
  - This factors as $p(\boldsymbol{x}) = p(x_1)p(x_2|x_1)\dots p(x_n|x_{n-1})$
  - Each pair of two nodes except the first is a clique
- If we have a node that has multiple parents, we "marry" the parents (*moralizing*) by connecting them, and all the parents and the child gives a maximal clique
- We can always convert a directed graph to an undirected graph this way
- However, it's not always possible to convert an undirected graph to a directed graph (we can't find a directed graph that satisfies all the conditional independence properties of the original graph)
  - If the undirected graph is a tree then we can do this, but if it has cycles then it's not possible

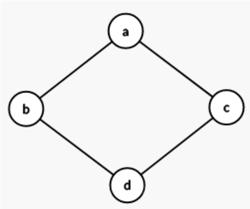


Figure 16: A case where an undirected graph cannot be converted into directed.

Lecture 19, Mar 25, 2024

Inference on Graphical Models

- Consider a Markov chain; we may want to perform inference tasks, such as finding the most likely state at time n, or finding the most likely sequence until time n, etc
  - All of these require us to find the marginal distribution, either over one variable or multiple variables
- Suppose we have N variables and we want to find the marginal of one of the variables
  - The brute force approach needs to sum over all the other variables, resulting in a summation with S^{N-1} terms and so exponential complexity
  - However this could be simplified drastically with assumptions about the graph
- Our summation goes over all the variables to marginalize over, but the summed terms may not all depend on these variables
  - This lets us factor out terms and simplify the computation drastically
  - The factored sums can be computed, into functions over the variables in the expression that aren't summed over
  - This results in *messages*
  - We represent each of the messages as a vector, and the factors as matrices, so we can simply do a vector matrix multiplication to perform the summation
  - The messages start at each end and get passed towards the middle to the variable that we are finding the marginal for; when we get both messages, we multiply them to get the final marginal

distribution

- *Message passing algorithm*: for a linear Markov model, messages are passed from either side inwards
  - This results in a complexity of only NS^2 instead of S^{N-1}
  - By letting the messages pass through the entire chain instead of stopping it at a node, we can get the marginals for all nodes
  - This is the *sum-product* algorithm
- To find the maximum likelihood sequence, we need more than just the marginals, due to dependence between variables
- We want to find $\underset{\boldsymbol{x}}{\operatorname{argmax}} \frac{1}{z}\psi_{12}(x_1, x_2) \dots \psi_{N-1,N}(x_{N-1}, x_N)$
- Consider finding the max of just two variables, $g(x_1, x_2)$
  - The distribution can be represented in a table
  - We first find the maximum value of each row, and then find the maximum of all the row maxima
  - Now we go back to the row that the overall max came from and find the column that gave the max
  - This is the *sum-product* algorithm
- When we have multiple variables, we can factor out terms just as we did in a summation, since the max function also distributes
  - This also involves message passing in the same way, completely analogous to the sum-product algorithm
  - This is the *max-product* algorithm
- In practice, we normally work with the log probabilities to avoid over/underflow

Factor Graphs

- In a factor graph, in addition to nodes for random variables, we also have nodes that are factors of the joint distribution that explicitly state the relationships between nodes
  - Noes are only connected directly to nodes of the other type (variable nodes are only connected to factor nodes and vice versa)
  - Instead of using edges connecting nodes to state the relationships between nodes, we use the factor nodes to do this explicitly
    - Each factor node is the keeper of a factor ψ
  - This changes the graph to a star pattern instead
  - This results in a *bipartite graph* (we can put all variables in one partition and all factors in the other, and there will be no connections within the two partitions)
- To convert an undirected graph to a factor graph, we associate a factor node with each maximal clique
  - Links between variables in the maximal clique are replaced by links to the factor node
  - Note that we don't have to use maximal cliques; by using non-maximal cliques we can get alternative factor graphs for the same undirected graph
    - These other graphs will have more factors, which makes the factorization finer
  - Therefore factor graphs are in general not unique

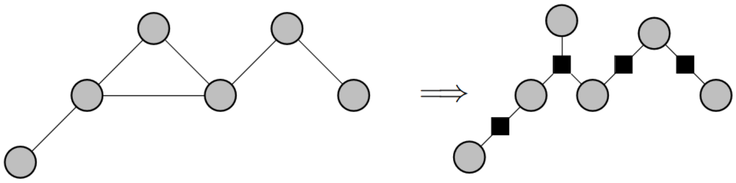
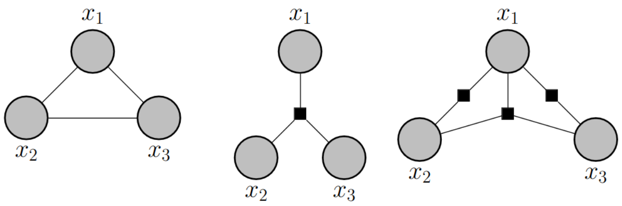


Figure 17: Conversion of an undirected graph to a factor graph by maximal cliques.

- We are particularly interested in factor graphs that are trees (between any pair of nodes, there is only one path)
  - An undirected graph that is a tree will always have a factor graph that is a tree
- For directed graphs, we replace each conditional probability with a factor node
- A directed graph is a tree if every node has only one parent; in a *polytree*, nodes can have multiple



$$p(x_1, x_2, x_3) = \psi_a(x_1, x_2)\, \psi_b(x_1, x_3)\, \psi_c(x_1, x_2, x_3)$$

Figure 18: Alternative factor graphs for an undirected graph.

parents, but there is still only a single path between nodes (ignoring directions)
  - Directed trees and polytrees can both be converted into factor graphs that are trees

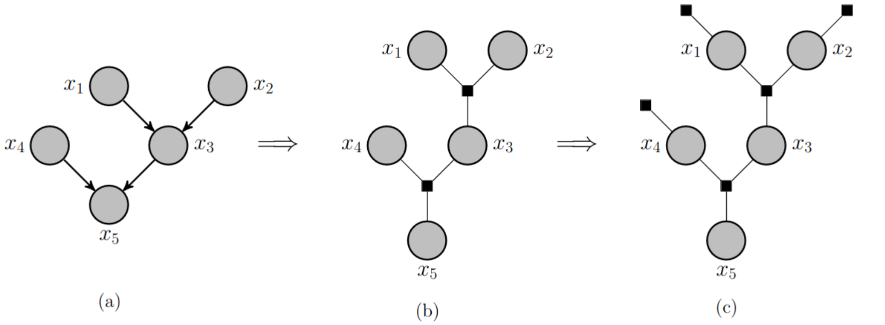


Figure 19: Conversion of a polytree directed graph to a factor graph (two variations).

- In a factor tree, we can also pass messages
  - Starting at the lowest factor nodes, we sum over its child variables and pass the message to the parent
  - When the parent has multiple factors passing it messages, the messages are multiplied
  - This is analogous to factoring out terms/moving sums
  - Since our graph is no longer linear, the way we can move the sums is more complicated and depends on the factors

- Generally we see two types of activity: either at variable nodes or factor nodes
- At variable nodes, we have a product over all the factors that are coming in
  - One or more factor nodes feed the variable x with their messages $\mu f_{l_k \to x}(x)$
    - * These messages are functions of the variable
  - The output produced is a product, the message $\mu_{x \to f}(x)$
  - For variable nodes that are leaves, we let its message $\mu_{x \to f}(x) = 1$ as initialization
- At factor nodes, we take the product of all incoming messages, and then multiply and marginalize over the factor at the node (for all variables except the one that the resulting message will be passed to)
  - One or more variables feed it with the messages $\mu_{x_{m_j} \to f}(x_{m_j})$
  - For factor nodes that are leaves, the message will be a function of the next variable as initialization
- This algorithm lets us find all marginal PMFs if we pass messages both from the leaves to the root and from the root to al leaves
- The result is exact for tree graphs, but if the graph has cycles, this is only an approximation
- For max-product, the variable nodes simply take the product as before, but the factor nodes take the maximum over the local factor instead of marginalizing (summing) over it
  - This goes all the way to the root, and then we trace back the maxima at each step to find the maximum likelihood sequence

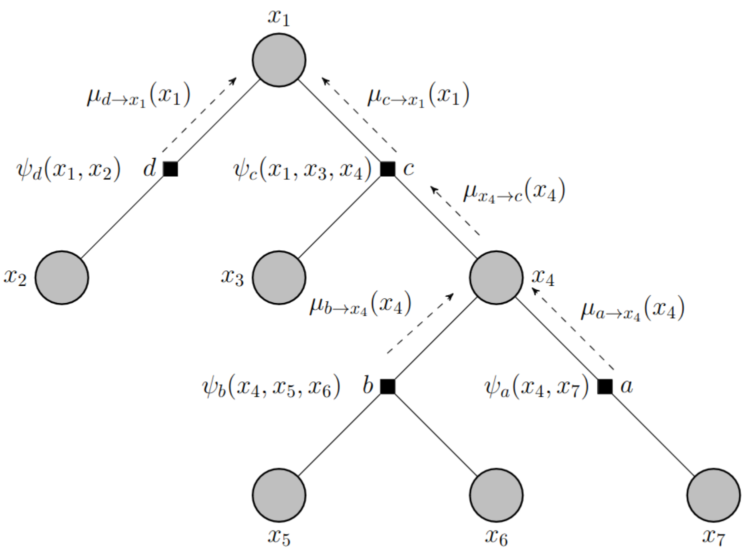


Figure 20: Message passing in a factor tree.

Lecture 20, Apr 1, 2024

Hidden Markov Models

- Suppose we have a Markov chain $Z_1, \dots, Z_N$, and instead of being able to observe Z_i directly, we instead observe $X_1, \dots, X_N$, where $p(x_i|z_i)$ is known, i.e. we observe the states with some noise
  - This is known as a *hidden Markov model*
  - e.g. sequence of speech, robot locations, pixels in handwriting, etc
  - $p(x_i|z_i)$ are the *emission probabilities* (what we can observe)
  - We'd like to perform inference on this, such as MAP estimation like we did with graphical models before
  - $p(z_1, \dots, z_N, x_1, \dots, x_N) = p(\boldsymbol{z}|\boldsymbol{x})p(\boldsymbol{x}) = p(z_1)\prod_{n=2}^{N} p(z_n|z_{n-1})\prod_{m=1}^{N} p(x_m|z_m)$
    - This gives the joint distribution of states and measurements

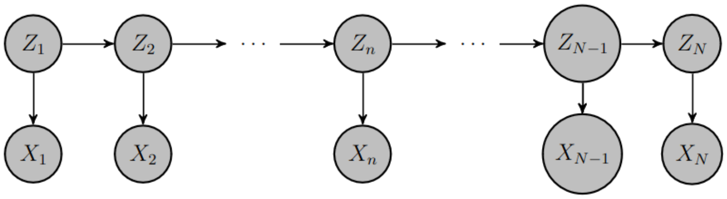


Figure 21: Illustration of a hidden Markov model.

- Example: two-state HMM: Z_i are binary variables; observation X_i is equal to Z_i with probability $1-\epsilon$ and its complement with probability ϵ
  - $\boldsymbol{p} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$
  - $\boldsymbol{P} = \begin{bmatrix} \beta & 1-\beta \\ 1-\alpha & \alpha \end{bmatrix}$
  - $P_e = \begin{bmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{bmatrix}$
    - This is the matrix of emission probabilities

  - The figure below shows a *trellis diagram*, which has one column for each time, one row for each state, and transition probabilities
    - Every possible realization of $Z_1, \ldots, Z_N$ corresponds to a path across the Trellis diagram
    - The probability of the sequence is the product of its initial state and the corresponding transition probabilities
    - The "length" of a path is its log probability, equal to the sum of the logs of the probabilities of its transitions
  - Observing $X_i = k$ gives a hint about the likelihood of $Z_n = j$ through the emission probability $P[X_n = k | Z_n = j]$

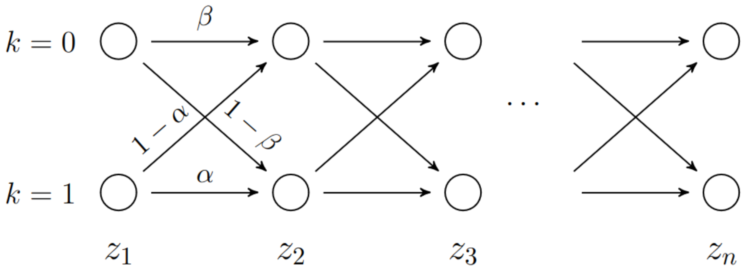


Figure 22: Trellis diagram for the example.

- Once we make the observations, $\boldsymbol{x}$ is no longer a random variable, but known observations
- Note $p(\boldsymbol{z}, \boldsymbol{x}) = p(z_1)p(x_1|z_1)p(z_2|z_1)p(x_2|z_2) \ldots p(x_N|z_N)p(z_N|z_{N-1}) = \psi(z_1, x_1)\psi(z_1, z_2, x_2) \ldots \psi(z_{n-1}, z_N, x_N)$
  - We group together every pair of transition probability and emission probability
- With this factorized form, we can use message passing to find the most likely value at time n, $z_n^* = \underset{z_n}{\operatorname{argmax}}\, p(z_n, \boldsymbol{x})$ or the most likely sequence $\boldsymbol{z}^* = \underset{z_1, \ldots, z_N}{\operatorname{argmax}}\, p(\boldsymbol{z}, \boldsymbol{x})$

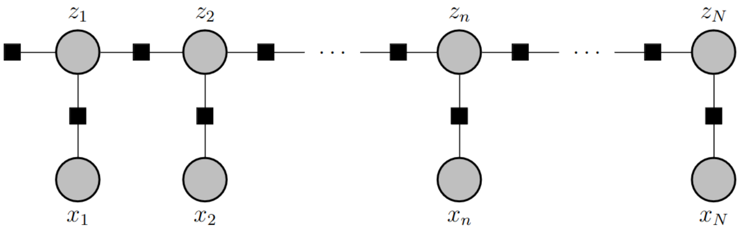


Figure 23: HMM converted to a factor graph.

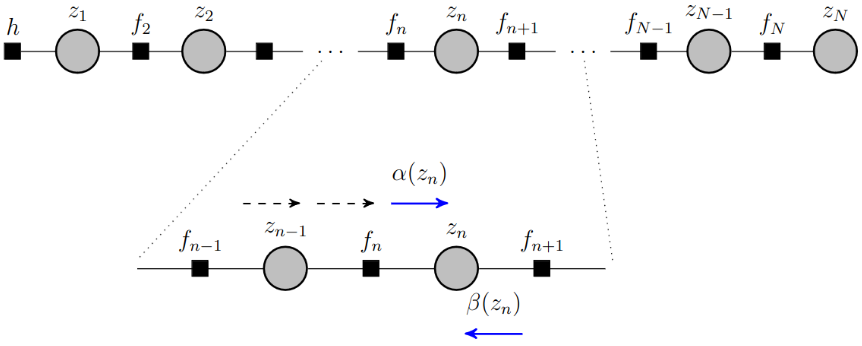


Figure 24: Simplified factor graph after observing $\boldsymbol{x}$.

- The HMM can be converted into a factor graph
  - Since x_i are observed, they can be combined into the factors between z_i
  - $h(z_1) = p(z_1)p(x_1|z_1)$ is the initial factor

  - $f_n(z_{n-1}, z_n) = p(z_n|z_{n-1})p(x_n|z_n)$
  - Factor node f_{n-1} comes before the variable node z_{n-1}
- At each node the following happens:
  - At a variable node z_{n-1}, we only have a single factor coming in, so the message created is simply the message from the preceding factor node, $\mu_{z_{n-1}\to f_n}(z_{n-1}) = \mu_{f_{n-1}\to z_{n-1}}(z_{n-1})$
    - Messages are passed right through without modification
  - At a factor node f_n we marginalize the product of its factor and the incoming message, so $\mu_{f_n\to z_n}(z_n) = \displaystyle\sum_{z_{n-1}} f_n(z_{n-1}, z_n)\mu_{z_{n-1}\to f_n}(z_{n-1})$
    - Multiply the factor by the incoming message and marginalize the variable of the incoming message
    - This is similar to a matrix multiplication
  - Messages that pass from right to left can be simplified similarly
- Let $\alpha(z_n) = \mu_{f_n\to z_n}(z_n)$ (left to right message) and $\beta(z_n) = \mu_{f_{n+1}\to z_n}(z_n)$ (right to left message)
  - $\alpha(z_n) = p(x_n|z_n)\displaystyle\sum_{z_{n-1}} \alpha(z_{n-1})p(z_n|z_{n-1})$
    - Obtained by substituting what we had above
    - This is like computing the next state probability normally, but multiplying by the hint, $p(x_n|z_n)$, for each possible state
    - Each possible value of state z_n is weighed by the hint
  - $\beta(z_n) = \displaystyle\sum_{z_{n+1}} \beta(z_{n+1})p(x_{n+1}|z_{n+1})p(z_{n+1}|z_n)$
    - In this case we cannot move the hint outside the summation, since the hint is from the node we came from, z_{n+1}, which is the one we sum over
  - $\alpha(z_1) = p(z_1)p(x_1|z_1)$
  - $\beta(z_N) = 1$
- Therefore the joint distribution when the messages meet is $p(z_n, \boldsymbol{x}) = \alpha(z_n)\beta(z_n)$
  - Let $\gamma(z_n) = \dfrac{p(z_n, \boldsymbol{x})}{p(\boldsymbol{x})} = \dfrac{\alpha(z_n)\beta(z_n)}{p(\boldsymbol{x})} = p(z_n|\boldsymbol{x})$
  - This is the marginal of z_n given the observations $\boldsymbol{x}$
  - This can also be a form of normalization, since $p(z_n, \boldsymbol{x})$ is going to have very small values due to the large number of observations
- If we condition on z_n, it disconnects the chain on its two sides
  - $\alpha(z_n) = p(x_1, \dots, x_n, z_n)$
  - $\beta(z_n) = p(x_{n+1}, \dots, x_N|z_n)$
  - $\alpha(z_n)$ encapsulates all the information of the observations prior to z_n, and $\beta(z_n)$ encapsulates all the information provided by observations after z_n

Forward-Backward (Baum-Welch) Algorithm

- Both $\alpha(z_n)$ and $\beta(z_n)$ involve multidimensional distributions, so computing them deals with very small probabilities; this introduces round-off errors when working with numbers of normal magnitudes
- For numerical stability it's better to normalize α and β:
  - $\hat{\alpha}(z_n) = p(z_n|x_1, \dots, x_n) = \dfrac{\alpha(z_n)}{p(x_1, \dots, x_n)}$
  - $\hat{\beta}(z_n) = \dfrac{p(x_{n+1}, \dots, x_N|z_n)}{p(x_{n+1}, \dots, x_N|x_1, \dots, x_n)} = \dfrac{\beta(z_n)}{p(x_{n+1}, \dots, x_N|x_1, \dots, x_n)}$
  - Note $\hat{\alpha}(z_n)\hat{\beta}(z_n) = \dfrac{\alpha(z_n)\beta(z_n)}{\left(\prod_{m=1}^{n} c_m\right)\left(\prod_{m=n+1}^{N} c_m\right)} = \dfrac{\alpha(z_n)\beta(z_n)}{p(\boldsymbol{x}_N)} = \gamma(z_n) = p(z_n|\boldsymbol{x})$
    - This is why we define β as above, so it complements α
- We need the normalization $p(\boldsymbol{x}_n)$
  - Let $c_n = p(x_n|\boldsymbol{x}_{n-1}) = p(x_n|x_1, \dots, x_{n-1})$

  - Then $p(\boldsymbol{x}_n) = p(x_1)p(x_2|x_1)\dots p(x_n|x_1, \dots, x_{n-1}) = \prod_{m=1}^{n} c_m$
- Substituting $\hat{\alpha}$ into the old recursive relation for α gives us the recursive relationship for $\hat{\alpha}$
  - $c_n\hat{\alpha}(z_n) = p(x_n|z_n)\sum_{z_{n-1}} \hat{\alpha}(z_{n-1})p(z_n|z_{n-1})$
- For β we have $c_{n+1}\hat{\beta}(z_n) = \sum_{z_{n+1}} \hat{\beta}(z_{n+1})p(x_{n+1}|z_{n+1})p(z_{n+1}|z_n)$
- Using the fact that $\sum_{z_n} \hat{\alpha}(z_n) = 1$, we obtain an expression for c_n
  - $c_n = \sum_{z_n}\left(p(x_n|z_n)\sum_{z_{n-1}} \hat{\alpha}(z_{n-1})p(z_n|z_{n-1})\right)$
- For each node n, we first calculate $\hat{\alpha}(z_n)$ for all values of z_n, and then find c_n through the summation; do this for all nodes up until z_N, and then go in the reverse direction to calculate $\hat{\beta}(z_n)$
  - We need to go in reverse for $\hat{\beta}$ since it requires c_{n+1}, which we can only get through the forward pass
  - This is called the *forward-backward algorithm*
- To compute $\hat{\alpha}(z_n)$: for state $z_n = k$, we do the following to find $\hat{\alpha}(z_n = k)$:
  - Use the previous values of $\hat{\alpha}(z_{n-1})$, multiply by the transition probabilities $p(z_n = k|z_{n-1} = j)$ for each previous state $z_{n-1} = j$ and sum
  - The sum is multiplied by $p(x_n|z_n = k)$, the hint
  - After we compute this for each k, we find the normalization c_n, by summing over the values we computed for all k and taking the inverse
  - Using c_n we find $\hat{\alpha}(z_n)$ by dividing each of the previous results by c_n

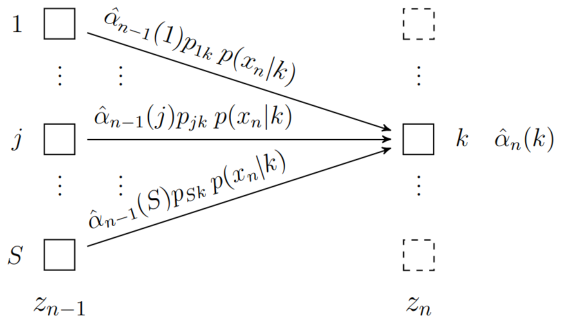

1
$\hat{\alpha}_{n-1}(1)p_{1k}\, p(x_n|k)$
j
$\hat{\alpha}_{n-1}(j)p_{jk}\, p(x_n|k)$
k
$\hat{\alpha}_n(k)$
$\hat{\alpha}_{n-1}(S)p_{Sk}\, p(x_n|k)$
S
z_{n-1}
z_n

Figure 25: Calculation of $\hat{\alpha}(z_n)$.

- To compute $\hat{\beta}(z_n)$: for state $z_n = k$, do the following to find $\hat{\beta}(z_n = k)$:
  - Use the values of $\hat{\beta}(z_{n+1})$, multiply by transition probabilities $p(z_{n+1} = j|z_n = k)$ for each j, and then multiply by the hint $p(x_{n+1}|z_{n+1} = j)$ (note in this case the hint does not distribute), and sum
  - Use c_{n+1}, obtained from the forward pass for $\hat{\alpha}$, to normalize and find $\hat{\beta}(z_n = k)$

Example: Robot Position Estimation

- Consider a robot with position described by (x_n, y_n)
- At each time it takes an action a_n to stay put, or move up, down, left, or right according to some transition probability
  - The action depends on the previous action; if the robot was previously moving in some direction, it will continue along that same direction unless it decides to stop
- The robot lives in a rectangular grid; at the end, the transition probabilities are such that the robot never leaves the grid
- We have noisy observations of the robot's position, $\hat{\boldsymbol{x}}_n$ (but not its actions)

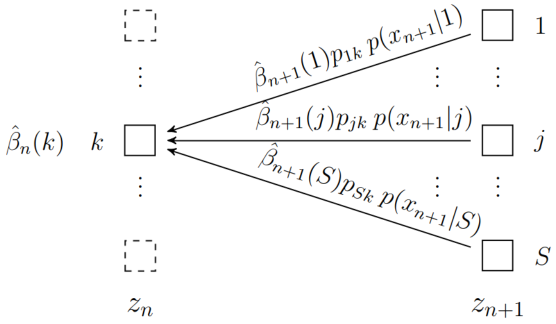

$\hat{\beta}_{n+1}(1)p_{1k}\,p(x_{n+1}|1)$
1
$\hat{\beta}_{n+1}(j)p_{jk}\,p(x_{n+1}|j)$
$\hat{\beta}_n(k)$
k
j
$\hat{\beta}_{n+1}(S)p_{Sk}\,p(x_{n+1}|S)$
S
z_n
z_{n+1}

Figure 26: Calculation of $\hat{\beta}(z_n)$. Note the order of indices on the transition probabilities p should be reversed.

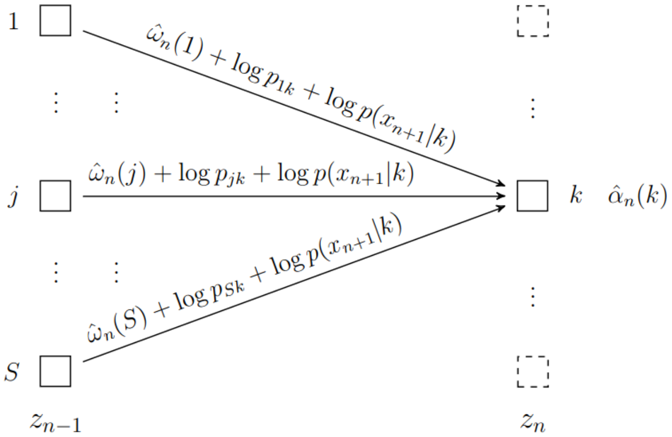
- Let $z_n = (x_n, y_n, a_n)$ where (x_n, y_n) is the position at time n and a_n is the previous action
    - The state is augmented by the previous action
- We have a lot of possible states, but since the actions are local, the transition probability $p(z_n|z_{n-1})$ will eliminate many prior states as impossible
    - Having the noisy observations broadens the set of possible state transitions

Viterbi Algorithm

- This algorithm is used to find the most likely sequence of hidden random variables given the observations
- We are now looking for $\boldsymbol{z}^* = \underset{z_1,\ldots,z_N}{\operatorname{argmax}}\, p(\boldsymbol{z}, \boldsymbol{x})$
- We create the factor graph in the exact same way as before
- In this case, instead of summing to find marginals, we are only interested in the maximums; therefore we can take the log, since it is a monotonic function
    - We couldn't before because we needed to sum probabilities
- We now look for $\displaystyle\max_{x_1,\ldots,x_N} \left(\log \frac{1}{Z} + \sum_{i=2}^{N} \log \psi(x_{i-1}, x_i) \right)$ where Z is normalization
- Recall message passing for the max-sum algorithm:
    - At each variable node, take sum (due to logs) $\displaystyle\mu_{x\to f}(x) = \sum_{k=1}^{L} \mu_{f_{l_k}\to x}(x)$
        - At each leaf we initialize to $\log 1 = 0$
    - At each factor node, we find $\displaystyle\mu_{f\to x} \max_{x_{m_1},\ldots,x_{m_M}} \left(\log \psi_f(x, x_{m_1}, \ldots, x_{m_M}) + \sum_{j=1}^{M} \mu_{x_{m_j}\to f}(x_{m_j}) \right)$
- For a hidden Markov model this simplifies:
    - $\mu_{h\to z_1}(z_1) = \log p(z_1) + \log(p(x_1|z_1) \equiv \omega_1(z_1)$
        - This is initialization of the messages
    - $\mu_{z_n\to f_{n+1}}(z_n) = \mu_{f_n\to z_n}(z_n)$
        - Just like the forward-backward algorithm, only a single factor feeds into each variable, so it's just passed along
    - $\mu_{f_{n+1}\to z_{n+1}}(z_{n+1}) = \max_{z_n} \{ \log f_{n+1}(z_n, z_{n+1}) + \mu_{z_n\to f_{n+1}}(z_n) \}$
        - Take the maximum value of the message over all possible values for z_n
- Let $\omega_{n+1}(z_{n+1})$ be the function that records the max at z_n, computed for each possible value of z_{n+1}
    - $\omega_{n+1}(z_{n+1}) = \max_{z_n} \{ \log p(z_{n+1} \mid z_n) + \log p(x_{n+1}|z_{n+1}) + \omega_n(z_n) \}$

      $= \log p(x_{n+1}|z_{n+1}) \max_{z_n} \{ \log p(z_{n+1} \mid z_n) + \omega_n(z_n) \}$
    - This recursive relationship for ω_n is the core of the Viterbi algorithm
- At each step, we compute the best path and store it along with the state that was chosen that led to the best path; the probability is summed due to the logs
- To calculate $\omega_{n+1}(k)$:

  - Take $\omega_n(j)$, the transition probability $\log p(z_{n+1} = k | z_n = j)$ and hint $\log p(x_{n+1} | z_{n+1} = k)$ and add, for each state j at the previous time
  - Find the value of j that leads to the max; this is the value of $\omega_{n+1}(k)$
  - Keep track of which j led to the max $\omega_{n+1}(k)$, so we can backtrack later



1
j
S
$\hat{\omega}_n(1) + \log p_{1k} + \log p(x_{n+1}|k)$
$\hat{\omega}_n(j) + \log p_{jk} + \log p(x_{n+1}|k)$
$\hat{\omega}_n(S) + \log p_{Sk} + \log p(x_{n+1}|k)$
k
$\hat{\alpha}_n(k)$
z_{n-1}
z_n

Figure 27: Illustration of the Viterbi algorithm.

Lecture 21, Apr 5, 2024

Forward-Backward and Viterbi Algorithm Examples