10/14 **Review**

<u>Learning in BNs</u>

Case I. fixed DAG, complete data "lookup" CPTs

Maximum likelihood (ML) estimation: $P_{ML}(X_i = x \mid pa_i = \pi) = \frac{\text{count}(X_i = x, pa_i = \pi)}{\text{count}(pa_i = \pi)}$

Ex: Markov models of language

product rule

* Let $w_\ell = \ell^{th}$ word in sentence. In general: $P(w_1, w_2, \cdots, w_L) = \prod_\ell P(w_\ell \mid w_1, \cdots, w_{\ell-1})$

* Markov model: $P(w_1, w_2, \cdots, w_L) = \prod_\ell P(w_\ell \mid \underbrace{w_{\ell-(n-1)}, \cdots, w_{\ell-2}, w_{\ell-1}}_{n-1 \text{ previous words}})$

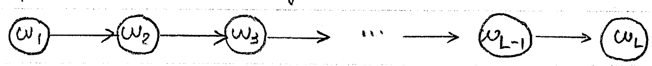
* Models of different orders

$n=1$ unigram

$n=2$ bigram

$n=3$ trigram

* special case (bigram)



Same CPT $P(w_\ell = w' \mid w_{\ell-1} = w)$ used at each node $\ell > 1$.

* How to learn?

- Collect large corpus of text ($\sim 10^8$ words)
- Vocabulary size ($10^3 - 10^5$)
- Count c_{ij} = # times that word j follows word i

  c_i = # times that word i appears

Estimate: $P_{ML}(w_\ell = j \mid w_{\ell-1} = i) = c_{ij}/c_i$ for bigram model.

* Problems w/ n-gram models:

- no generalize to unseen n-grams.
- n-gram counts become increasingly sparse as n increases.

<u>Ex</u>: Naive Bayes models for document classification.

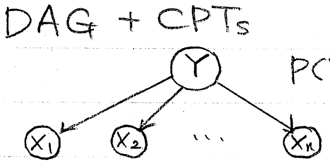
* Variables

$Y \in \{1, 2, \cdots, m\}$ possible document topics

$X_i \in \{0, 1\}$ Does the i^{th} word in dictionary appear in document?

Represent every document as bit vector.

* BN = DAG + CPTs



$P(Y=y)$

$P(X_i = 1 \mid Y = y)$

* Document classification

$$P(Y=y \mid \vec{X}=\vec{x}) = P(\vec{X}=\vec{x} \mid Y=y)\, P(Y=y) \big/ P(\vec{X}=\vec{x})$$ Bayes rule

$$= \left[\prod_{i=1}^{n} P(X_i=x_i \mid Y=y)\right] P(Y=y) \big/ P(\vec{X}=\vec{x})$$ Conditional independence

"Naive" Bayes

* Strengths:

(i) easy to estimate $P(y)$ and $P(X_i=1 \mid Y=y)$ from labeled corpus of text

$P_{ML}(y)$ = proportion of topics

$P_{ML}(X_i=1 \mid Y=y)$ = fraction of documents on topic y that contain i^{th} word

(ii) useful baseline.

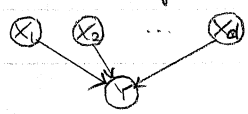
* Weaknesses

(i) assumption that words appear independently given topic.

(ii) "Bag-of-words" representation (bit vector) ignores word order, word count, ...

Case II. fixed DAG, complete data, parametric CPTs.

Case IIa: linear regression



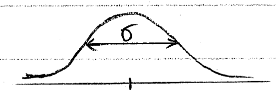
How to predict a real-valued child $y \in \mathbb{R}$ from real-valued parents $\vec{x} \in \mathbb{R}^d$?

* Gaussian CPT

$$P(Y=y \mid \vec{X}=\vec{x}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y-\sum_{i=1}^{d} w_i x_i)^2/2\sigma^2}$$

variance

weights w_i



Intuitively: model input-output relation by noisy linear map

$$y = \sum_{i=1}^{d} w_i x_i + \text{noise}$$

$$E[y] = \vec{w}\cdot\vec{x}$$

* Training data

$\{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \cdots (\vec{x}_T, y_T)\}$ T examples

* Probability of IID data:

$$P(y_1, y_2, y_3, \cdots, y_T \mid \vec{x}_1, \vec{x}_2, \cdots, \vec{x}_T) = \prod_{t=1}^{T} P(y_t \mid \vec{x}_t)$$

* Log-likelihood

$$\mathcal{L} = \log P(\text{data}) = \sum_{t=1}^{T} \log P(y_t \mid \vec{x}_t)$$

* Estimate $\vec{w}$ and σ^2 by maximizing log-likelihood:

$$\mathcal{L} = \sum_{t=1}^{T} \left\{ -\frac{1}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(y_t - \vec{w}\cdot\vec{x}_t)^2 \right\}$$: Same as minimizing mean squared error fit to data

* To maximize $\mathcal{L}(\vec{w})$:

$$0 = \frac{\partial \mathcal{L}}{\partial w_\alpha} = \sum_t \left\{ -\frac{1}{2\sigma^2} \cdot 2(y_t - \vec{w}\cdot\vec{x}_t)\, x_{t\alpha} \right\}$$

$\alpha = 1, 2, \cdots, d$

α^{th} component of $\vec{x}_t$

Linear equations : $\sum_t y_t x_{t\alpha} = \sum_t (\vec{w} \cdot \vec{x}_t)\, x_{t\alpha}$ for $\alpha = 1, 2, \cdots, d$

$$= \sum_t \left(\sum_{\beta=1}^{d} w_\beta \cdot x_{t\beta} \right) x_{t\alpha}$$

In matrix-vector form : $d \times d$ matrix $A_{\alpha\beta} = \sum_t x_{t\beta} x_{t\alpha}$

$$A = \sum_t \vec{x}_t \vec{x}_t^{\,T}$$

$d \times 1$ vector $b_\alpha = \sum_t y_t x_{t\alpha}$

$$\vec{b} = \sum_t y_t \vec{x}_t$$

Set of linear equations : $\boxed{A\vec{w} = \vec{b}}$

$\boxed{\vec{w} = A^{-1}\vec{b}}$ solution (ML)

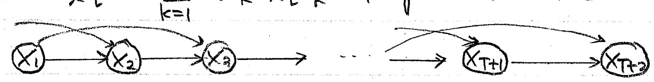
* Ill-conditioned problems arise when :
  - input dimensionality exceeds # examples ($d > T$)
  - inputs not in general position
  - option : minimum norm solution

    $\min \|\vec{w}\|$ such that $\frac{\partial \mathcal{L}}{\partial \vec{w}} = \vec{0}$ (always unique)

* example : time series prediction

  time series : $\{x_1, x_2, \cdots, x_T\}$ $x_t \in \mathbb{R}$.

  model : $x_t = \sum_{k=1}^{d} w_k x_{t-k}$ + gaussian noise

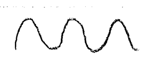


(d=2)

Q : If x_t is a linear function of $x_{t-1}, \cdots, x_{t-d}$,

is x_t a linear function of "time" t ? No.

Ex. $x_t = \sin(\omega t)$



$x_t = 2(\cos\omega)\, x_{t-1} - x_{t-2}$

$\boxed{\text{DETOUR} - \text{numerical optimization}}$

* How to maximize (or minimize) function $f(\vec{\theta})$ over $\vec{\theta} = (\theta_1, \theta_2, \cdots, \theta_d) \in \mathbb{R}^d$?
* Not always possible to solve analytically?

  $\frac{\partial f}{\partial \vec{\theta}} = \left(\frac{\partial f}{\partial \theta_1}, \frac{\partial f}{\partial \theta_2}, \cdots, \frac{\partial f}{\partial \theta_d} \right) = (0, 0, \cdots, 0)$ in closed form.

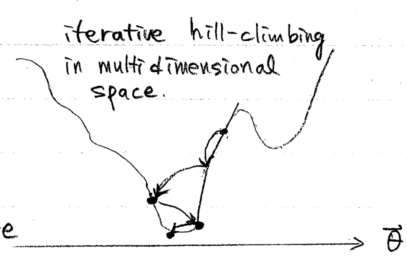
* Turn to numerical methods :

(i) gradient descent (or ascent)

iterative update rule

$$\vec{\theta} \longleftarrow \vec{\theta} - \eta \frac{\partial f}{\partial \vec{\theta}}$$

$\eta > 0$ scalar learning rate



* Cons
  - tuning $\eta > 0$ can be tricky ; – no guarantee of monotonic convergence
  - local vs. global optima

* Pros

- simple, generic procedure for differentiable function.
- asymptotic convergence to local optima.