classical extensions to higher dimensions

- multi-variate standard normal $p(x) = N(\mu=0, \Sigma=\mathbb{I}) = \frac{1}{(2\pi)^{D/2}} \exp\left(-\frac{x x^T}{2}\right)$

  $D = \dim(x)$ x: row vector

  can be expressed as a product of 1-D normal distributions

$$x x^T = \sum_{j=1}^{D} x_j^2 \Rightarrow \exp\left(-\frac{x x^T}{2}\right) = \exp\left(-\frac{\sum_j x_j^2}{2}\right) = \prod_{j=1}^{D} \exp\left(-\frac{x_j^2}{2}\right)$$

$\Rightarrow$ sampling in D dimensions $\Leftrightarrow$ "random number generation" $x \sim N(0, \mathbb{I}) \Leftrightarrow x_j \sim N(0,1)$ ← 1-D Gaussian

- multivariate Gaussian distribution with mean μ, covariance Σ

$$p(x) = \mathcal{N}(\mu, \Sigma) = \frac{1}{\sqrt{\det(2\pi\Sigma)}} \exp\left(-\frac{(x-\mu)\Sigma^{-1}(x-\mu)^T}{2}\right) \qquad \det(r \cdot A) = r^D \det(A)$$

eigendecomposition of $\Sigma = U \Lambda U^T$ $\qquad \Sigma^{-1} = (U^T)^{-1} \Lambda^{-1} U^{-1} = U \Lambda^{-1} U^T$

U: eigenvectors (orthonormal $D \times D$) $\qquad \Lambda$: eigenvalues (diagonal $D \times D$)

$$\Lambda^{-1/2} = \begin{pmatrix} \frac{1}{\sqrt{\lambda_1}} & & \\ & \ddots & \\ & & \frac{1}{\sqrt{\lambda_D}} \end{pmatrix}$$

define new coordinates. $\tilde{x} = (x-\mu) \cdot U \Lambda^{-1/2}$

$$\tilde{x}\tilde{x}^T = (x-\mu)\Sigma^{-1}(x-\mu)^T$$

sampling from $x \sim p(x) = \mathcal{N}(\mu, \Sigma)$

- sample $\tilde{x} \sim \mathcal{N}(0, \mathbb{I})$
- calculate $x = \tilde{x} \cdot \Lambda^{1/2} \cdot U^T + \mu$

"reparameterization trick"

$\hat{=}$ reduce sampling from arbitrary Gaussian to sampling from standard normal, followed by linear transf.

in our language:

$$z = f(x) = (x-\mu) U \Lambda^{-1/2} \sim \mathcal{N}(0, \mathbb{I}) \Rightarrow x = f^{-1}(z) = z \Lambda^{1/2} U^T + \mu \sim \mathcal{N}(\mu, \Sigma)$$

- similar for other analytic multi-variate distributions (scipy.stats offers 16)

if $p(x)$ must be learned: mixture model idea naturally generalize to D dimensions

- histograms: if bins are defined by a regular grid with k levels per dimension
  $\Rightarrow L = k^D$ exponential growth, does not scale
  (requires $N = O(2^D)$ as well to "fill" the bins)
  $\Rightarrow$ define bins by recursive subdivision, e.g. density tree
  quality of models is only medium
  the # of correlations to be considered bounded by tree depth.
  tree depth $= O(\log N)$

- GMM: work as before, but need to learn co-variance instead of variance (outer product)
  $\Rightarrow$ change EM alg: $\Sigma_\ell^{(t)} = \dfrac{\sum_{n=1}^{N} \gamma_{n\ell} \left(x_i - \mu_\ell^{(t)}\right)^T \left(x_1 - \mu_\ell^{(t)}\right)}{\sum_{n=1}^{N} \gamma_{n\ell}}$

- kernel density estimation: unchanged
  but finding a bandwidth σ^2 that works equally well for all x_n is very difficult

$\rightarrow$ become more expensive for growing D) GMM $= O(L \cdot D^2)$, KDE $= O(N D)$

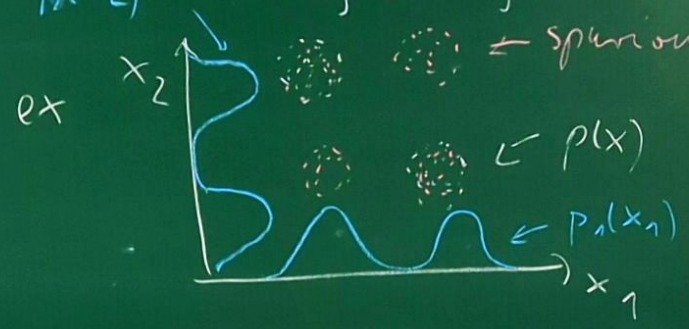
- compute $p(x)$ efficiently: use (approximate) nearest neighbor search to find the non-negligible component $p_\ell(x)$

- sampling from a mixture model
  - sample $\ell \sim \text{discrete}(\underbrace{\pi_1}_{=N_1/N}, \dots, \underbrace{\pi_L}_{=N_L/N})$, sample $x \sim p_\ell(x)$

methods to reduce $p(x)$ to a collection of 1-D distributions

- naive. $p(x) = \prod_{j=1}^{D} p_j(x_j)$, learn a 1-D model for each coordinate x_j

$\Rightarrow x_j \perp x_{j'}$ (independent coordinates $\Rightarrow$ all correlations lost $\Rightarrow$ highly unrealistic)



← spurious peaks of $p(x) = p_1(x_1)\, p_2(x_2)$

- exact: <u>auto-regressive models</u> : decompose $p(x)$ by Bayesian chain rule

$$p(x) = p_1(x_1)\, p_2(x_2|x_1)\, p_3(x_3|x_1,x_2) \cdots p_D(x_D|x_{1:D-1})$$

$$= p_2(x_2)\, p_D(x_D|x_2)\, p_3(x_3|x_2,x_D) \cdots p_j(\,.\,)$$

any ordering of chain rule is valid for $p(x)$

each $p_j(x_j|x_{<j})$ is a collection of 1-D distributions (one distribution per value of $x_{<j}$)

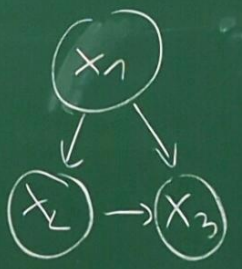
$\Rightarrow$ use conditional inverse transform method

$$x_j \sim p(x_j|x_{<j}) \iff z_j = p(z_j)\,,\; x_j = f_j^{-1}(z_j; x_{<j})$$

$$f^{-1}(z) = \begin{pmatrix} x_1 = f_1^{-1}(z_1) \\ x_2 = f_2^{-1}(z_2; x_1) \\ \vdots \\ x_D = f_D^{-1}(z_D; x_1, \ldots, x_{D-1}) \end{pmatrix}$$

auto-regressive model

$\rightarrow$ visualize as directed acyclic graph (DAG)



- problem: $f_j^{-1}(z_j, x_{<j})$ is a different 1-D function for each value of $x_{<j}$

  ⇒ if $x_{<j}$ is defined on a regular grid with k levels per dimension

  ⇒ $x_{<j}$ can take $k^{(j-1)}$ different values ⇒ does not scale well

- general solution: learn f_j^{-1} with a neural networks, which generalizes from a few seen (TS) values of $x_{<j}$ to all possible values

- if $x_j \perp x_{j'}$ for $j' < j$, then $x_{j'}$ can be dropped from $f_j^{-1}(\,)$ ⇒ drop arrow in DAG

  $x_j \perp x_{j''} \mid x_{j'}$ for $j', j'' < j$, $x_{j''}$ can be dropped ⇒ —//—

  $x_j \perp x_{j'''} \mid x_{j'}, x_{j''}$

  etc.

⇒ drop as many arrows from DAG as possible ⇒ $f_j^{-1}(\,)$ become easier to learn

① Markov chain: $X_j \perp X_{j'} \mid X_{j-1}$ for all $j' < j-1$

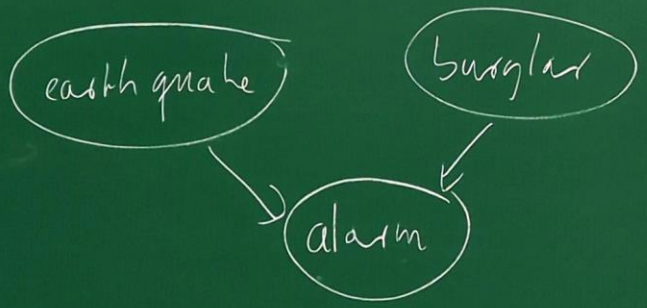
$X_1 \to X_2 \to \dots \to X_0$

typical: j = time step
"future only depends on present, not the past"

$f_j^{-1}(Z_j; X_{j-1})$ for all j ⇒ easy to learn

② causal model: $p(x) = \prod_{j=1}^{0} p_j(x_j \mid PA(x_j))$

causal parents of x_j (set of direct causes)



property: causal DAG has fewest edges
⇒ easier to learn

but: finding the causal DAG is very hard
hot research topic "causal discovery"