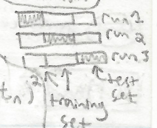
SSE Loss Func. $E(\vec{w}) = \frac{1}{2}\sum_{n=1}^{N}(y(\vec{x}_n,\vec{w}) - t_n)^2 + \frac{\lambda}{2}\|\vec{w}\|^2$ → reg. parameter (don't penalize w_0!) → decrease complexity, chose λ by Cross Validation
→ weight decay / Ridge reg.
bad λ / good λ → training set, validation set, test set ← S-fold cross validation



→ linear models → linear in w → **Least Squares** – $y(\vec{x}_n,\vec{w}) = \vec{x}_n^T\vec{w}$
→ soln is $\vec{w} = (\underline{X}^T\underline{X})^{-1}\underline{X}^T\vec{t}$, $X \in \mathbb{R}^{N\times d+1}$ → **polynomial fit** – $y(\vec{x},\vec{w}) = \sum_{j=0}^{M} w_j x^j$, M picked by Model Selection – Avoid Over-fitting, achieve Generalization.

assumed to be sampled iid so training data and test data have same dist! → **Probabilistic Curve Fit** – $\vec{t} = y(\vec{x},\vec{w}) + \vec{\varepsilon}$, $\vec{\varepsilon} \perp \vec{x}$, $\vec{t} \sim \prod_{n=1}^{N} N(t_n | y(\vec{x}_n,\vec{w}), \beta^{-1}) = p(\vec{t}|\underline{X},\vec{w},\beta)$, $\vec{w}_{ML} = E(\vec{w})$
$\varepsilon_i \sim N(0,\beta^{-1})$ → $\beta E(\vec{w})$, $\beta_{ML}^{-1} = \frac{1}{N}\sum_{n}(y(\vec{x}_n,\vec{w}_{ML}) - t_n)^2$

RVs $\mathrm{Cov}(\vec{x},\vec{y}) = E\{[\vec{x} - E(\vec{x})][\vec{y}^T - E(\vec{y}^T)]\}$
→ $\ln p(\vec{t}|\underline{X},\vec{w},\beta) = -\frac{\beta}{2}\sum_{n=1}^{N}(y(\vec{x}_n,\vec{w}) - t_n)^2 + \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi)$ maximize log likelihood w.r.t $\vec{w}$ ⟺ Minimizing Least Squares.

Beta Dist. $\mathrm{Beta}(x|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}x^{a-1}(1-x)^{b-1}$, $x \in [0,1]$, $E(x) = \frac{a}{a+b}$, $\mathrm{var}(x) = \frac{ab}{(a+b)^2(a+b+1)}$, $\Gamma(x) = \int_0^\infty u^{x-1}e^{-u}du$.

Multinomial $\mathrm{Mult}(m_1,\dots,m_K|\vec{\mu},N) = \binom{N}{m_1\cdots m_K}\prod_{k=1}^{K}\mu_k^{m_k}$ m_k → size of group, $\sum m_k = N$; μ_k → prob of being in group, $\sum \mu_k = 1$
Dirichlet $\mathrm{Dir}(\vec{\mu},\vec{\alpha}) = \frac{\Gamma(\sum_k \alpha)}{\Gamma(\alpha_1)\cdots\Gamma(\alpha_K)}\prod_{k=1}^{K}\mu_k^{\alpha_k - 1}$

Multivariate Gaussian $N(\vec{x}|\vec{\mu},\Sigma) = \left(\frac{1}{2\pi}\right)^{D/2}\frac{1}{|\Sigma|^{1/2}}\exp\{-\frac{1}{2}(\vec{x}-\vec{\mu})^T\Sigma^{-1}(\vec{x}-\vec{\mu})\}$, $\vec{x} \in \mathbb{R}^d$, $\vec{y} = \underline{U}(\vec{x}-\vec{\mu})$, $\Sigma\vec{u}_i = \lambda_i\vec{u}_i$, so $\vec{x}$ shifted by $\vec{\mu}$, rotated by U, and stretched in U_i by $\lambda_i^{1/2}$
→ $\ln p(\underline{X}|\vec{\mu},\Sigma) = -\frac{ND}{2}\ln(2\pi) - \frac{N}{2}\ln|\Sigma| - \frac{1}{2}\sum_{n=1}^{N}(\vec{x}_n - \vec{\mu})^T\Sigma^{-1}(\vec{x}_n - \vec{\mu})$, $\vec{\mu}_{ML} = \frac{1}{N}\sum_{n=1}^{N}\vec{x}_n$, $\Sigma_{ML} = \frac{1}{N}\sum_{n=1}^{N}(\vec{x}_n - \vec{\mu}_{ML})(\vec{x}_n - \vec{\mu}_{ML})^T$ length of x → # of sufficient statistics

Mixture of Gaussians $p(\vec{x}) = \sum_{k=1}^{K}\pi_k N(\vec{x},\vec{\mu}_k,\Sigma_k)$ → component; mixing coefficient (adds to 1)
Exponential Family → $p(\vec{x}|\vec{\eta}) = h(\vec{x})g(\vec{\eta})\exp\{\vec{\eta}^T\vec{u}(\vec{x})\}$, $h: \mathbb{R}^n \to \mathbb{R}$, $g: \mathbb{R}^m \to \mathbb{R}$, $\eta \in \mathbb{R}^m$, $u: \mathbb{R}^n \to \mathbb{R}^m$, $-\nabla\log(g(\vec{\eta})) = E(\vec{u}(\vec{x}))$

Linear Basis Func. → $y(\vec{x},\vec{w}) = \sum_{j=0}^{M-1} w_j\phi_j(\vec{x})$, $\phi_j: \mathbb{R}^n \to \mathbb{R}$
→ $\phi_j(\vec{x}) = (\vec{x})_j$
→ Can be nonlinear in input space
→ Can make problem linearly seperable
→ Can be global or local.
$\vec{w}_{ML} = (\Phi^T\Phi)^{-1}\Phi^T\vec{t}$, $(\Phi)_{ij} = (\phi_j(\vec{x}_i)) \in \mathbb{R}^{N\times M}$ Dataset $= \{\vec{x}_1,\dots,\vec{x}_N\} \in \mathbb{R}^n$
$\vec{w}_{ML} = (\lambda I + \Phi^T\Phi)^{-1}\Phi^T\vec{t}$ (ridge regression)

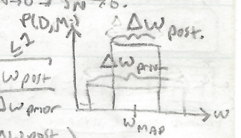
Statistical Decision th. $p(C_k|\vec{x}) = \frac{p(\vec{x}|C_k)p(C_k)}{p(\vec{x})}$ → posterior, likelihood, prior, marginal likelihood; $p(\text{Mistake}) = \int_{R_1}p(\vec{x},C_2)dx + \int_{R_2}p(\vec{x},C_1)dx$, $p(\vec{x},C_1) > p(\vec{x},C_2)$ then assign $\vec{x}$ to C_1.

Loss Function $E[L] = \iint L(t,y(\vec{x}))p(\vec{x},t)\,d\vec{x}\,dt$, **Minkowski Loss** → $E[L] = \iint (t - y(x))^q p(\vec{x},t)\,dx\,dt$, min of
$q=1$ → Conditional median [LASSO]
$q=2$ → $E(t|x)$ [Squared Loss]
$q \to 0$ → Conditional Mode

Bias-var tradeoff $E[L] = \int\{y(x) - h(x)\}^2 p(x)dx + \underbrace{\iint (h(x)-t)^2 p(x,t)\,dx\,dt}_{\text{intrinsic variability of target (noise)}}$ $h(x) = E(t|x) = \int t\,p(t|x)dt$ Bias – Avg diff from model y and actual dist. h, across D
Variance – variability of model y to diff datasets D.
→ beat by Avg many models on diff training sets.
→ $E_D[y(x,D) - h(x)]^2 = \underbrace{E_D[y(x,D) - h(x)]^2}_{\text{Bias}^2} + \underbrace{E_D[\{y(x,D) - E_D[y(x,D)]\}^2]}_{\text{Variance}}$
*small λ → high variance → flexible model → parameters can be big
large λ → high bias → weights tend to 0 → over regularized.
→ Bagging

Predictive Dist. → Compute via Conjugate prior, MCMC...
→ use Gaussian approx of posterior!
$p(x|D) = \int p(x|D,w)\,p(w|D)\,dw$ posterior distribution $= \frac{p(D|w)p(w)}{p(D)}$ likelihood, ← prior, ← Marginal likelihood
→ $p(t|\vec{t},\vec{x},\underline{X},\alpha,\beta) = N(t|(\beta S_N\Phi^T\vec{t})^T\phi(\vec{x}), \frac{1}{\beta} + \phi(\vec{x})^T S_N\phi(\vec{x}))$ target noise; $S_N^{-1} = (\alpha I + \beta\Phi^T\Phi)^{-1}$ → uncertainty of parameter vals. → $N \to \infty \Rightarrow S_N \to 0$.
if $M_0 = 0$, if $S_0 = \alpha^{-1}I$ → Equivalent Kernel: $y(\vec{x},\vec{m}_n)$
→ $= \sum_{n=1}^{N}\beta\phi(\vec{x})^T S_N\phi(\vec{x}_n)t_n = \sum_{n=1}^{N}k(x,x_n)t_n$

Bayesian Linear Regression $p(\vec{w}|\vec{t},\underline{X},\beta) = \prod_{n=1}^{N}N(t_n|\vec{w}^T\phi(\vec{x}_n),\beta^{-1})\,N(\vec{w}|\vec{m}_0,S_0)$ → posterior over weights, → likelihood, → conjugate prior



This is a mixed model.
Model Selection – use model with most model evidence $p(D|M_i) \approx p(D|w_{map})\frac{\Delta w_{post}}{\Delta w_{prior}}$

Bayesian Model Comparison $\{M_1,\dots,M_L\}$ posterior, prior, (marginal likelihood) model evidence: $p(M_i|D) \propto p(M_i)\,p(D|M_i)$, $p(t|x,D) = \sum_{i=1}^{L}p(t|x,M_i,D)\,p(M_i|D)$ Average, predictive dist. weighted by its posterior
$p(D|M_i)$ → $\int p(D|\vec{w},M_i)p(\vec{w}|M_i)d\vec{w}$
→ $\ln p(D,M_i) \cong \ln p(D|w_{map}) + M\ln\left(\frac{\Delta w_{post}}{\Delta w_{prior}}\right)$
→ increase as $M\uparrow$; → $\uparrow$ as $M\uparrow$; → $\downarrow$ as $M\uparrow$
M = # parameters = complexity.

Evidence Approx. target + input in test case: $p(t^*|\vec{x}^*,D) = \iiint p(t^*|\vec{x}^*,\vec{w},\beta)\,p(w|D,\alpha,\beta)\,p(\alpha,\beta|D)\,dw\,d\alpha\,d\beta$ likelihood, posterior of weights, posterior over hyperparams; precision of output noise, precision of prior, training data
Empirical Bayes → select $(\alpha,\beta)_{MLE}$ of $p(\vec{t}|\underline{X},\alpha,\beta) \propto p(\alpha,\beta|D)$ [assuming flat prior $p(\alpha,\beta)$]
→ Maximize marginal likelihood to determine the values of the hyperparameters from the training data.

Classification ① Linear Discriminant Function: $y_k(\vec{x}_n) = \vec{x}_n^T\vec{w}_k + w_{k0} = W^T x$, K classes, $\underline{X} = \begin{bmatrix}\vec{x}_1\\ \vdots\\ \vec{x}_N\end{bmatrix} \in \mathbb{R}^{N\times(d+1)}$, $T = \begin{bmatrix}\vec{t}_1\\ \vdots\\ \vec{t}_N\end{bmatrix} \in \mathbb{R}^{N\times K}$, t_i is one of K encoded ⇒ optimal W is $(\underline{X}^T\underline{X})^{-1}\underline{X}^T T \in \mathbb{R}^{(d+1)\times K}$ ⇒ assign $\vec{x}_n$ to $\max_k y_k(\vec{x})$ = max row of $W^T x$
→ Sensitive to outliers, bad w/ multiple classes.
Fishers Linear Discriminant – project onto vector $\frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$ ← Maximize btw class variance, ← minimize within class variance *want large dist btw means and little spread.
→ these directly map input into a class.

② Generative Approach – model $p(C_k|\vec{x})$ indirectly by modelling → infer posterior using Bayes, $\frac{p(\vec{x}|C_k)p(C_k)}{p(\vec{x})} = \frac{p(\vec{x}|C_k)p(C_k)}{\sum_i p(\vec{x}|C_i)p(C_i)} = \frac{\exp(a_k)}{\sum_i\exp(a_i)}$ softmax, $a_k = \ln(p(\vec{x}|C_k)p(C_k))$ → If we assume $\sim N_D(\mu_k,\Sigma^{-1})$ then linear in input space!
→ ML Soln – $p(\vec{t},\underline{X},\pi,\vec{\mu}_1,\vec{\mu}_2,\Sigma) = \prod_{n=1}^{N}[\pi N(\vec{x}_n|\mu_1,\Sigma)]^{t_n}[(1-\pi)N(\vec{x}_n|\mu_2,\Sigma)]^{1-t_n}$, $\pi_{ML} = \frac{N_1}{N_1+N_2}$, $\mu_{1ML} = \frac{1}{N_1}\sum_{n=1}^{N}t_n\vec{x}_n$, $S_{1ML} = \frac{1}{N_1}\sum_{n\in C_1}(\vec{x}_n - \mu_1)(\vec{x}_n - \mu_1)^T$
$S_{ML} = \frac{N_1}{N}S_1 + \frac{N_2}{N}S_2$ → Note: we have to assume data comes from Normal dist.!

③ Discriminative Approach – Model $p(C_k|\vec{x})$ directly. or fix basis function → Fixed nonlinear transformations of input space using vector of basis functions, $\phi(x)$ → Decision boundary linear in feature space ϕ but not in input space.
→ (multiclass) logistic Regression – $p(C_k|\vec{x}_n) = \frac{\exp(\vec{w}_k^T\vec{x}_n)}{\sum_j\exp(\vec{w}_j^T\vec{x}_n)}$ (y_{nk}) likelihood – $p(T|\underline{X},\vec{w}_1,\dots,\vec{w}_K) = \prod_{n=1}^{N}\left[\prod_{k=1}^{K}p(C_k|x_n)^{t_{nk}}\right] = \prod_{n=1}^{N}\left[\prod_{k=1}^{K}y_{nk}^{t_{nk}}\right]$ ($\mathbb{R}^{N\times K}$)
→ -ve log likelihood = C.E. $= -\sum_{n=1}^{N}(y_{nj} - t_{nj})\vec{x}_n$ (for $\vec{w}_j$)
→ (2 class) Logistic Regression – $\prod_{n}^{N}y_n^{t_n}(1-y_n)^{1-t_n}$ $\frac{d}{dy_n} = \frac{y_n - t_n}{y_n(1-y_n)}$ $\frac{dy_n}{dw} = y_n(1-y_n)x_n$ since $\frac{d}{da}\sigma(a) = \sigma(a)(1-\sigma(a))$ $\begin{bmatrix} C & \vec{k} \\ \vec{k}^T & c \end{bmatrix}$
$\sigma(\vec{w}^T\vec{x}) = \frac{1}{1+\exp(-\vec{w}^T\vec{x})}$, $\sigma(-a) = 1 - \sigma(a)$, $\sigma^{-1}(a) = \ln\left(\frac{a}{1-a}\right)$

GP $\vec{f} = \begin{pmatrix}f(x_1)\\ \vdots\\ f(x_N)\end{pmatrix}$ $[\underline{K}]_{nm} = k(x_n,x_m)$, $\vec{f} \sim N(\vec{0},\underline{K})$ → Marginal of f, $\vec{t}|\vec{f} \sim N(\vec{f},\beta^{-1}I_N)$ → Conditional on f, $\vec{t} = \int p(\vec{t}|\vec{f})p(\vec{f})d\vec{f} \sim N(\vec{0},\underline{C})$ → Marginal of $\vec{t}$, using that $\vec{t},\vec{f}$ gaussian!, $\begin{bmatrix}\vec{t}\\ t_{N+1}\end{bmatrix} \sim N\left(\vec{0}, \begin{bmatrix}k(x_1,x_1) & \cdots & k(x_{N+1},x_1)\\ \vdots & & \vdots\\ k(x_1,x_{N+1}) & \cdots & k(x_{N+1},x_{N+1})\end{bmatrix}\right)$ → predict t_{N+1} from $\vec{x}$ and x_{N+1} and $\vec{t}$
$[\underline{C}]_{nm} = k(x_n,x_m) + \delta_{nm}\beta^{-1}$
$t_n = f_n + \varepsilon_n$, $f \perp \varepsilon_n$
finally, $t_{N+1}|\vec{t} \sim N(\vec{k}^T C^{-1}\vec{t},\ c - \vec{k}^T C^{-1}\vec{k})$. → inverting this is $O(N^3)$ N = # data points or basis functions! if $\vec{f} = \Phi\vec{w}$, $\vec{w} \sim N(\vec{0},\alpha^{-1}I)$ ⇒ $[K]_{nm} = \frac{1}{\alpha}\phi(\vec{x}_n)^T\phi(\vec{x}_m)$, $f \sim N(\vec{0},\underline{K})$ → and you get Bayesian Linear Regression!
→ $\sum_{n=1}^{N}\left[\sum_{k=1}^{N}k(x_k,x_n)t_k\right]$ ← linear combo of targets * Can use parametric family of Covariance functions and infer params from data ≡ learning hyperparameters using Type II ML in standard model!

Hyper Parameters **Type I ML** = Maximize over params, f. $P(f|t) = \frac{P(t|f)P(f)}{P(t)}$ ← prior. **Type II** = max. $P(t|\theta) = \int P(t|f)P(f|\theta)\,df$, integrate over params, maximize over θ hyper parameters.
→ **ARD** – Use Kernel w/ η_i like $k(x,y) = \sum \eta_i (x_i - y_i)^2$ ⟷ η_i small ⇒ dim i irrelevant to predictions; η_i large ⇒ dim i important!

Full introduce $P(\theta)$, infer $P(\theta|t)$ posterior using MCMC, since not tractable!

K Means $\vec{x}_1, \dots, \vec{x}_N \in \mathbb{R}^D$ into K clusters, minimizing $\sum^N \sum^K r_{nk} \|x_n - \mu_k\|^2$
① Given μ_k, set $r_{nk} = 1$ if $k = \arg\min_j \|x_n - \mu_j\|^2$, $r_{nk} = 0$ otherwise [Put x_n in closest μ_k]
② Set $\mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$ [Set μ_k to mean of x_n's in μ_k]

LATENT VARIABLE MODELS

Z continuous → Factor Analysis, PCA. Z discrete, X iid → MoG, Mixture Models. Z discrete, X seq. data → HMM. Data $= \underline{X} \in \mathbb{R}^{N\times D}$, $\underline{Z} \in \mathbb{R}^{N \times K}$ (K latent vars, 1 for each x_n)

MoG $P(Z) = \prod^K \pi_k^{z_k}$, $P(\vec{X}|z_k = 1) = N(\mu_k, \Sigma_k)$, $\gamma(z_{nk})$ ← posterior $= P(z_{nk} = 1|\vec{x}_n) = \frac{P(z_{nk}=1) P(\vec{x}_n|z_{nk}=1)}{\sum_j P(z_{nj}=1)P(x_n|z_{nj}=1)} = \frac{\pi_k N(\vec{x}_n|\mu_k, \Sigma_k)}{\sum_j \pi_j N(\vec{x}_n|\mu_j, \Sigma_j)}$ (π_k ← prior; $N(\dots)$ ← likelihood)

EM Algorithm **Given** $\underline{X}$, $P(\underline{Z}|\underline{X}, \theta)$, $P(\underline{X}, \underline{Z}|\theta)$ **Goal** find θ s.t. $P(X|\theta)$ maximized! Notice $\ln P(X|\theta) = \ln\left(\sum_Z P(X, Z|\theta)\right)$ → Marginal log likelihood → bad

E ① Compute $E_Z(\ln(P(X,Z|\theta))) = \sum_Z \ln[P(X,Z|\theta)] \underbrace{P(Z|X, \theta^{old})}_{\approx P(Z)} = Q(\theta, \theta^{old})$ Expected value of Z given complete data LL. $\ln P(X,Z|\theta) = \sum_Z \ln P(X, Z|\theta)$ → Complete data log likelihood → good

M ② Find $\max_\theta Q(\theta, \theta^{old})$, set θ^{old} to it. (for Gaussian case) $Q(\theta, \theta^{old}) = E_Z\left\{\sum^N \sum^K z_{nk}[\ln \pi_k + \ln N(x_n|\mu_k, \Sigma_k)]\right\} = \sum^N \sum^K \gamma(z_{nk})[\ln \pi_k + \ln N(x_n|\mu_k, \Sigma_k)]$ ↳ $E_Z(z_{nk})$

Computations $E(z_{nk}) = \sum_{z_n} z_{nk} P(z_{nk} = 1|\vec{x}_n) = \gamma(z_{nk})$, μ_j^{new}: $\frac{\partial}{\partial \mu_j} \sum^N \sum^K \gamma(z_{nk})[\ln \pi_k + \ln N(x_n|\mu_k, \Sigma_k)] = \sum^N \gamma(z_{nj})(x_n - \mu_j)\Sigma_j^{-1} = 0$ [Multiply both sides by Σ_j]
$\Rightarrow \sum^N \gamma(z_{nj})x_n - \sum^N \gamma(z_{nj})\mu_j = 0 \Rightarrow \frac{\sum^N \gamma(z_{nj})x_n}{\sum^N \gamma(z_{nj})} = \mu_j$ | π_j^{new}: $\frac{\partial}{\partial \pi_j} \sum^N \left(\gamma(z_{nj})\ln(\pi_j) + \sum_{k\neq j}^K \gamma(z_{nk})\ln(1-\pi_j)\right) = \sum^N \frac{\gamma(z_{nj})}{\pi_j} - \sum^N \sum_{k\neq j}^K \frac{\gamma(z_{nk})}{1-\pi_j} = 0 \Rightarrow \frac{N_j}{\pi_j} = \frac{N - N_j}{1 - \pi_j}$
$\Rightarrow \pi_j^{new} = \frac{N_j}{N}$.

$EM = E_Z(\ln P(X, Z|\pi, \mu, \Sigma)) \to -\frac{1}{2}\sum^N \sum^K r_{nk}\|x_n - \mu_k\|^2 + C = $ kmeans [When $\varepsilon \to 0$, $\Sigma_i = \varepsilon I$ shared]

PCA → Select top eigenvectors of S $\{u_1, \dots, u_M\}$, $z_{n1} = x_n^T u_1$ [project data] ① Max Var Formulation – let $x \in \mathbb{R}^{N \times D}$ $u_1^T u_1 = 1$, want to maximize $\frac{1}{N}\sum^N (u_1^T x_n - u_1^T \bar{x})^2 = u_1^T S u_1$ wrt u_1,
Constrain $\|u_1\| = 1$, $\frac{\partial}{\partial u_1} u_1^T S u_1 + \lambda(1 - u_1^T u_1) \stackrel{=0}{\Rightarrow} S_1 u_1 = \lambda_1 u_1 \Rightarrow$ u_1 eigenvector of S_1, λ_1 eigenval of S_1 ② Min Error Formulation – Use orthonormal basis $\{u_1, \dots, u_D\}$ $\tilde{x}_n = \sum_{i=1}^M z_{ni} u_i + \sum_{i=M+1}^D b_i u_i$
$x_n = \sum_{i=1}^D (x_n^T u_i) u_i$ want to min. $\frac{1}{N}\sum^N \|x_n - \tilde{x}_n\|^2$ to get $z_{nj}^{min} = x_n^T u_j$, $b_j^{min} = \bar{x}^T u_j$ ⟺ $\sum_{i=M+1}^D u_i^T S u_i$ ← min this ⟺ max ①! $= \sum_{i=M+1}^D \lambda_i$ ← Minimized by choosing smallest e-vals.

PCA for D↑ When N<D, from $O(D^3)$ cost to get u_i's reduced to $O(N^3)$, when X centred $\frac{1}{N} X^T X u_i = \lambda_i u_i$ (S, $D\times D$) $\Rightarrow \frac{1}{n} XX^T(Xu_i) = \lambda_i X u_i \Rightarrow Xu_i$ e-val for XX^T ($\in \mathbb{R}^{N\times N}$) w/ same e-val λ_i!

PPCA Generative, assume $P(z) = N_M(z|0, I)$, $P(x|z) = N_D(x|Wz + \mu, \sigma^2 I)$, $z \in \mathbb{R}^M$, $x, \mu \in \mathbb{R}^D$, $W \in \mathbb{R}^{D \times M}$ marginal data density / pred. dist. $\to P(x) = \int P(z)P(x|z)\,dz = N(x|\mu, WW^T + \sigma^2 I)$ $E(X) = E(\mu + Wz + \varepsilon) = \mu$
$C(X) = E((\mu + Wz + \varepsilon - \mu)(Wz + \varepsilon)^T)$
* rotations in latent space ⇒ redundancy in parameterization. $P\left(\begin{bmatrix} z \\ x\end{bmatrix}\right) = N\left(\begin{bmatrix} 0 \\ \mu\end{bmatrix}\begin{bmatrix} I & W^T \\ W & WW^T + \sigma^2 I\end{bmatrix}\right)$, $P(z|x) = N(M^{-1}W^T(x - \mu), \sigma^2 M^{-1})$ $M = W^T W + \sigma^2 I$ → Can now solve for W and σ^2 directly with ML or use EM algorithm!

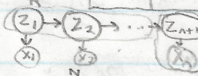
EM PCA faster than PPCA, use complete data log likelihood $\ln(p(x, z|\mu, W, \sigma^2)) = \sum^N [\ln P(x_n|z_n) + \ln P(z_n)]$ * Does not require constructing cov matrix. * $O(NDM)$ | $\sigma^2 \to 0 \Rightarrow$ PPCA → PCA ML params same!

FA $P(X|z) = N(Wz + \mu, \Psi)$, $\Psi \in \mathbb{R}^{D \times D}$ diagonal. $P(X) = N(\mu, WW^T + \Psi)$
Factors non-identifiable, can find many diff params to get same ML → degeneracy, interpretation of factors impossible!

Name	Solve directly w/ ML?	looks for directions of large	rotation invariant	Scale invariant
PCA	Yes	Variance	Yes	No
FA	No	Correlations	No	Yes

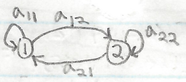
Autoencoders – Neural nets w/ same # of inputs as outputs. M<D hidden units (bottleneck) ↳ nonlinear extension of PCA.

HMM – x_i's not iid
– Markov Assumption: $P(x_1, \dots, x_n) = P(x_1)\prod_{n=2}^N P(x_n|x_{n-1})$ ↗ $P(x_n|x_1, \dots, x_{n-1}) = P(x_n|x_{n-1})$ [1st order]
Markov model w/ random measures



→ mixture model w/ states coupled across time

$A = \begin{bmatrix} P(\text{from 1 to 1}) & P(\text{from 1 to 2}) \\ P(\text{from 2 to 1}) & P(\text{from 2 to 2})\end{bmatrix}$ ↳ all cols same ⇒ Standard mixture model!



State Space Model – introduce latent vars which form Markov chain. ↳ using $z_{n+1} \perp\!\!\!\perp z_{n-1} | z_n$
The model $\theta = \{\pi, A, \phi\}$ $P(X, Z|\theta) = P(z_1|\pi)\prod_{n=2}^N P(z_n|z_{n-1}, A)\prod_{n=1}^N P(x_n|z_n, \phi)$
$[\pi]_k = P(z_{1k} = 1)$ ↳ $[A]_{jk} = P(z_{nk} = 1|z_{n-1,j} = 1)$, $\sum^K A_{jk} = 1$ ↳ emission probs $= \prod^K P(x_n|\phi_k)^{z_{nk}}$

EM for HMM's Note $P(X|\theta) = \sum_{z_1, \dots, z_n} P(X, Z|\theta) = \sum_{z_1} P(z_1)P(x_1|z_1) \cdots \sum_{z_n} P(z_n|z_{n-1})P(x_n|z_n)$ ← No closed form sol'n, usually $P(X|\theta) = \sum_{i=1}^N P(z_i)P(X|z_i)$ ← Closed form sol'n! Since z_i indp!

E ① Compute $\gamma(z_n) = P(z_n|X, \theta^{old}) = \frac{P(X|z_n)P(z_n)}{P(X)} = \frac{P(x_1, \dots, x_n|z_n)P(z_n)P(x_{n+1}, \dots, x_N|z_n)}{P(X)} = \frac{P(x_1, \dots, x_n, z_n)P(x_{n+1}, \dots, x_N|z_n)}{P(X)} = \frac{\alpha(z_n)\beta(z_n)}{P(X)}$ (α ← joint prob of data up to point z_n + z_n; β ← cond. prob. of data after z_n; $P(X)$ ← conditioned on θ^{old} so = likelihood)

$\alpha(z_n) = P(x_n|z_n)P(x_1, \dots, x_{n-1}, z_n) = P(x_n|z_n)\sum_{z_{n-1}} P(x_1, \dots, x_{n-1}, z_{n-1}, z_n) = P(x_n|z_n)\sum_{z_{n-1}} P(x_1, \dots, x_{n-1}, z_{n-1})P(z_n|z_{n-1}) = P(x_n|z_n)\sum_{z_{n-1}} \alpha(z_{n-1})P(z_n|z_{n-1})$

$\beta(z_n) = \sum_{z_{n+1}} P(x_{n+1}, \dots, x_N, z_{n+1}|z_n) = \sum_{z_{n+1}} P(x_{n+1}, \dots, x_N|z_{n+1})P(z_{n+1}|z_n) = \sum_{z_{n+1}} P(x_{n+2}, \dots, x_N|z_{n+1})P(x_{n+1}|z_{n+1})P(z_{n+1}|z_n) = \sum_{z_{n+1}} \beta(z_{n+1})P(x_{n+1}|z_{n+1})P(z_{n+1}|z_n)$

also $\alpha(z_1) = P(x_1, z_1) = P(z_1)P(x_1|z_1) = \prod^K \pi_k P(x_1|\phi_k)^{z_{1k}}$, $\beta(z_N) = P(z_N|X) = \frac{P(X, z_N)\beta(z_N)}{P(X)}$ ($P(X, z_N) \to \alpha(z_N)$) $\Rightarrow \beta(z_N) = 1$

Notice $\sum_{z_n} \frac{\alpha(z_n)\beta(z_n)}{P(X)} = \sum_{z_n} P(z_n|X, \theta) = 1 \Rightarrow P(X|\theta) = \sum_{z_n} \alpha(z_n)\beta(z_n)$, $P(X|\theta) = \sum_{z_n} P(X, z_n|\theta) = \sum_{z_n} \alpha(z_n)\beta(z_n) = \sum_{z_N} \alpha(z_N)$ ↳ set $n = N$, $\beta(z_N) = 1$

$\xi(z_{n-1}, z_n) = P(z_{n-1}, z_n|X) = \frac{P(X|z_{n-1}, z_n)P(z_{n-1}, z_n)}{P(X)} = \frac{\alpha(z_{n-1})P(x_n|z_n)P(z_n|z_{n-1})\beta(z_n)}{P(X)}$

M ② Maximize expected complete data log likelihood $\sum_Z P(Z|X, \theta^{old})\log P(X, Z|\theta)$ for θ and replace θ^{old} w/ θ.

$Q(\theta, \theta^{old}) = \sum_{k=1}^K \gamma(z_{1k})\log \pi_k + \sum_{n=2}^N \sum_{k=1}^K \sum_{j=1}^K \xi(z_{n-1,j}, z_{nk})\log A_{jk} + \sum_{n=1}^N \sum_{k=1}^K \gamma_{nk}\log P(x_n|z_n)$, $\pi_k^{new} = \frac{\gamma(z_{1k})}{\sum_j \gamma(z_{1j})}$

Viterbi Decoding → Choose $\gamma(z_n)$ w/ largest probability instead of sums to get path of max expected # of correct states.

HMM Ext. → Autoregressive → better at catching long range correlations
→ Input-Output HMM → emission probs and transition probs depend on sequence $u_1, \dots, u_N$
→ Factorial HMM → 2 Markov chains of latent variables → Exact inference intractible
→ Regularizing HMM's: high dim outputs = lots of params in emission model → ↑ training data, tie params across states.
→ Many states = transition matrix has many entries. → force some transitions to = 0

prior – I want all weights = 0
likelihood – I want to fit data!