KL of 2 Gaussians: $KL(p\|q) = \frac{1}{2}\left(tr(\Sigma_q^{-1}\Sigma_p) + (\mu_p-\mu_q)^T\Sigma_q^{-1}(\mu_p-\mu_q) - d + \log(|\Sigma_q|/|\Sigma_p|)\right)$

1. Data, Measurements & Representations

$R(f,X) = \int_y \mathcal{L}(y, f(X)) P(y|X) dy$, $R(f) = \mathbb{E}_X[R(f,X)]$

1.1 Paradigms:

"Hypothesis Class"

Frequentism: $\mathcal{H} = \{p_\theta : \theta \in \Theta\}$, $\theta^* = \arg\max_{\theta\in\Theta} \mathcal{L}(\theta; \mathcal{D})$

Bayesianism: $\mathcal{H}$, prior $\pi(\theta) \to$ posterior $\pi(\theta|\mathcal{D}) \propto p(X|\theta)\pi(\theta)$

Stat. Learn.: $\min R(f) := \min \mathbb{E}_{(X,Y)}[\mathcal{L}(Y, f(X))] = \mathbb{E}_{(X,Y)}[\mathbb{1}\{f(X)\neq Y\}]$

Problems: 1) Don't know anything about f, 2) don't know $P^*(X,Y)$, 3) R is non-differentiable

Solutions: 1) Def $\mathcal{H} = \{f(.|\theta) : \theta \in \Theta\}$ (f diff-able)

2) Collect $\mathcal{D}_{train} = \{(x_1,y_1),\dots,(x_n,y_n)\} \Rightarrow \mathbb{E}[\dots] \approx \frac{1}{n}\sum \mathcal{L}(f(x_i), y_i)$

3) Chose a loss fct $\mathcal{L}: \mathcal{Y}\times\mathcal{Y} \to \mathbb{R}$ differentiable

$\Rightarrow$ Goal: $\min_{f\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^n \mathcal{L}(f(x_i), y_i) =: \min_{f\in\mathcal{H}} \hat{R}(f) \Rightarrow \hat{f} = \arg\min_{f\in\mathcal{H}} \hat{R}(f)$

$\Rightarrow$ Test error: $\hat{R}(\hat{f}, \mathcal{D}_{test}) = \frac{1}{m}\sum_{j=1}^m \mathcal{L}(y_j^*, \hat{f}(x_j^*))$ $\Rightarrow$ otherwise too optim.

1.2 Recap

Calculus: $\frac{\partial}{\partial x} x^T a = \frac{\partial}{\partial x} a^T x = a$, $\frac{\partial^2}{\partial x^2} x^T A x = A + A^T$

$\frac{\partial}{\partial s}(x-As)^T W (x-As) = -2A^T W(x-As)$, $\frac{\partial}{\partial x}(x-As)^T W(x-As) = 2W(x-As)$

Information Theory: Entropy: $H(X) = -\sum_{x\in\mathcal{X}} P(X=x)\log(P(X=x))$ $= -\int p(x)\log(p(x))dx = \mathbb{E}_X[-\log(p(x))]$

Joint Entropy: $H(X,Y) = \mathbb{E}_{(X,Y)}[-\log(p(x,y))] = -\sum_{x\in\mathcal{X}, y\in\mathcal{Y}} P(X=x,Y=y)\log(P(X=x,Y=y))$

Conditional Entropy: $H[X|Y] = \mathbb{E}_{(X,Y)\sim p(x,y)}[-\log(p(x|y))] = -\sum_{y\in\mathcal{Y}, x\in\mathcal{X}} P(X=x, Y=y)\log\left(\frac{P(X=x,Y=y)}{P(Y=y)}\right)$ $= -\sum_{y\in\mathcal{Y}}\sum_{x\in\mathcal{X}} P(Y=y)P(X=x|Y=y)\log(P(X=x|Y=y))$ $= H(Y) - H(Y|X)$

Mutual Information: $I(X;Y) = I(Y;X) = H(X) - H(X|Y) = H(X) + H(Y) - H(X,Y)$

Cross-Entropy: $H[p\|q] = \mathbb{E}[-\log(q(x))] = H[p] + KL(p\|q) \geq H[p]$

KL Divergence: $KL(p\|q) = H[p\|q] - H[p] = \mathbb{E}_{x\sim p}\left[\log\left(\frac{p(\theta)}{q(\theta)}\right)\right] = const - \mathbb{E}_{x\sim p}[\log q]$ (discrete)

Remarks: 1) $H(X) \geq 0$ $\forall$ RV X, 2) $H(X,Y) = H(X) + H(Y)$ if $X \perp\!\!\!\perp Y$, on $\mathcal{X}$

3) Let $\mathcal{X}$ be a finite set, then $Unif(\mathcal{X})$ maximizes the entropy among RVs

4) $0 \leq H(X|Y) \leq H(X)$ ("Information never hurts"),

5) $0 \leq I(X;Y) \leq H(X)$, 6) $H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$

7) $H(X|Y) = H(Y|X) + H(X) - H(Y) \leq H(X)$

8) $H[\mathcal{N}(\mu,\Sigma)] = \frac{1}{2}\log((2\pi e)^d \det(\Sigma))$, $H[\mathcal{N}(\mu,\sigma^2)] = \log(\sigma\sqrt{2\pi e})$

9) $KL(p\|q)$: difference between #bits req. for encoding p using a code optimized for q, Inefficiency of using q

2. Anomaly Detection:

1) Project $\mathcal{D}_{train}$ onto low-dim space, 2) Fit GMM, 3) Anomaly Score

2.1 PCA

$\mathcal{X} = \{x_1,\dots,x_n\}$, $x_i \in \mathbb{R}^D$, def $\bar{X} = \frac{1}{n}\sum_{i=1}^n x_i$, let $u \in \mathbb{R}^D$ s.t. $\|u\|_2^2 = u^T u = 1$. Let $\mathcal{X}_u = \{u^T x_1,\dots,u^T x_n\}$. Sample mean: $\frac{1}{n}\sum u^T x_i = u^T\bar{X}$

Sample variance: $\frac{1}{n}\sum(u^T x_i - u^T\bar{X})^2 = u^T S u$, $S = \frac{1}{n}\sum_i (x_i - \bar{X})(x_i - \bar{X})^T$

$u_1 = \arg\max_{u\in\mathbb{R}^D, \|u\|_2=1} u^T S u \iff$ Lagrangian: $\mathcal{L}(u,\lambda) = \underbrace{u^T S u}_{=f(u)} + \lambda(\underbrace{1 - u^T u}_{=g(u)}) = u^T(S - \lambda\mathbb{1})u + \lambda$

$\Rightarrow \frac{\partial\mathcal{L}}{\partial u} = 2(S - \lambda\mathbb{1})u = 0 \Rightarrow Su = \lambda u$. Let u_1 be EV associated with largest eigenvalue $\Rightarrow \lambda_1 = \lambda_1 u_1^T u_1 = u_1^T S u_1 \geq u_i^T S u_i = \lambda_i$

Remaining PCs: Def $\mathcal{X}_1 = \{x - proj_{u_1}(x) : x \in \mathcal{X}\} = \{x - (u_1^T x)\cdot u_1 : x \in \mathcal{X}\}$

$\Rightarrow u_2 := \arg\max_{u\in\mathbb{R}^D, \|u\|_2=1} u^T S_1 u$ where $S_1 = Cov(\mathcal{X}_1)$. Note: $u_i^T u_j = \delta_{ij}$

Alternative perspective: Goal: find $z \in \mathbb{R}$ & $w \in \mathbb{R}^d$ s.t. minimize reconstruction error: $\|x - zw\|_2^2$. Note: $z_i^* = w^T x_i$ (orthogonal proj.)

$\hookrightarrow w^* = \arg\min_{w\in\mathbb{R}^d: \|w\|_2=1} \sum_{i=1}^n \|x_i - ww^T x_i\|_2^2 = \arg\max_{\|w\|_2=1}\sum_{i=1}^n (w^T x_i)^2$

$= \arg\max_{\|w\|_2=1} w^T\left(\sum_{i=1}^n x_i^T x_i\right)w = \arg\max_{\|w\|_2^2=1} w^T\Sigma w = v_1$ (largest EV)

Proof: use $w = \sum_{i=1}^d \alpha_i v_i \Rightarrow w^T\Sigma w = \sum_{i=1}^d \alpha_i^2\lambda_i$, $\|w\|^2 = \sum_{i=1}^d \alpha_i^2 \overset{!}{=} 1$

2.2 GMMs & EM Algo:

Project Data onto d-dim subspace: $x \mapsto (u_1^T x, \dots, u_d^T x)^T$

GMM: $p_\theta(x) = \sum_{k=1}^K \pi_k \mathcal{N}(x|\mu_k,\Sigma_k)$ with $\sum_{i=1}^K \pi_i = 1$, $\theta = \{\pi_i, \mu_i, \Sigma_i\}_{i=1}^K$

$\hookrightarrow \log(p_\theta(X)) = \sum_{i=1}^n \log\left(\sum_{k=1}^K \pi_k\mathcal{N}(x_i|\mu_k,\Sigma_k)\right)$ intractable

$\hookrightarrow$ Sol.: assume you know which distr x_i belongs to: z_i

$\hookrightarrow \log(p_\theta(X,Z)) = \sum_{i=1}^n \log(p_\theta(x_i,z_i)) = \sum_{i=1}^n \log(\pi_{z_i}\mathcal{N}(x_i|\mu_{z_i},\Sigma_{z_i}))$

$= \sum_{i=1}^n \log(\pi_{z_i}) + \sum_{i=1}^n \log(\mathcal{N}(x_i|\mu_{z_i},\Sigma_{z_i}))$

$\hookrightarrow \log(p_\theta(X)) = \mathbb{E}_{Z\sim q}[\log(p_\theta(X))] = \mathbb{E}_{Z\sim q}\left[\log\left(\frac{p_\theta(X,Z)}{p_\theta(Z|X)}\cdot\frac{q(Z)}{q(Z)}\right)\right]$

$= \underbrace{\mathbb{E}_{Z\sim q}\left[\log\left(\frac{p_\theta(X,Z)}{q(Z)}\right)\right]}_{=:M(q,\theta)} + \underbrace{\mathbb{E}_{Z\sim q}\left[\log\left(\frac{q(Z)}{p_\theta(Z|X)}\right)\right]}_{=:E(q,\theta)}$

$\hookrightarrow \log(p_\theta(X)) \geq M(q,\theta)$ $\forall q,\theta \iff E(q,\theta) = KL(q\|p_\theta(Z|X)) \geq 0$

$\log(p_\theta(X)) = M(q^*,\theta)$ if $q^* = p_\theta(.|X)$ $\forall\theta$

EM Algo: 1) Init θ_0, 2) For $t=1,\dots$ update until converg.

$q^t = \arg\min_q E(q,\theta^{t-1}) = p_{\theta^{t-1}}(.|X)$, ← (E-step), (M-step)

$\theta^t = \arg\max_\theta M(q^t,\theta) = \arg\max_\theta \mathbb{E}_{Z\sim q}[\log(p_\theta(X,Z))]$

$\hookrightarrow \{\log(p_{\theta^t}(X))\}_{t\geq 0}$ is non decreasing: $\log(p_{\theta^{t+1}}(X)) \geq M(q^t,\theta^{t+1}) \geq M(q^t,\theta^t) = \log(p_{\theta^t}(X))$

Rem: No guarantee for conv. to global max → only local

2.3 Summary:

Algo for Anom Det.: 1) Project X onto $\mathbb{R}^d$ via PCA, 2) Fit GMM p_θ with K components for low-dim data via EM Algo., 3) For new point x^*, assign anomaly score as $-\log(p_\theta(x^*))$

2.4 Validation:

F1-Score: $2\cdot\left(\frac{1}{precision} + \frac{1}{recall}\right)^{-1}$

Precision: $\frac{\#TP}{\#\{\hat{y}=1\}}$, Recall: $\frac{\#TP}{\#\{y=1\}}$, Accuracy: $\frac{\#TP + \#TN}{n}$

	$y=1$	$y=-1$
$\hat{y}=1$	TP	FP (T1 err)
$\hat{y}=-1$	FN (T2 err)	TN

3. Density Estim:

3.1 Parametric Inference:

Let: Data: $\mathcal{Z} = \{(x_i,y_i) \in \mathbb{R}^d\times\mathbb{R} : 1\leq i\leq n\}$, $X = (X_1,\dots,X_d)$ Features, Y output, ε noise

$Y = f(x;\theta) + \varepsilon$, $\mathbb{E}[\varepsilon] = 0$ | $\hat{f} = \arg\min \mathbb{E}_{Y|X}[(Y - f(x))^2] \Rightarrow \hat{f}(x) = \mathbb{E}_{Y|X}[Y|X=x]$

Bayesian view: observations (X & Y) & params θ are RVs

Param. Stat.: functional form of $P(X,Y|\theta)$ a priory known

Non-param Stat.: Sample X & Y to estim the Likelihood

SLT: minimize empirical risk $\min_{f\in\mathcal{C}} \hat{R}(f, \mathcal{Z}^{train})$ without estim $\mathcal{L}$

Parametric Models: $\mathcal{F} = \{f(x;\theta) : \theta\in\Theta\} \Rightarrow$ find best θ

Likelihood fct.: $\mathcal{L}_n(\theta) = \prod_{i=1}^n f(x_i;\theta)$ for datapoints $\mathcal{D} = \{x_1,\dots,x_n\}$

Log-Likelihood: $l_n(\theta) = \log(\mathcal{L}_n(\theta)) \Rightarrow \arg\max_\theta l = \arg\max_\theta \mathcal{L}$

MLE: $\hat{\theta}_{MLE} = \arg\max_{\theta\in\Theta} \mathcal{L}_n(\theta) = \arg\max_{\theta\in\Theta} l_n = \arg\max_{\theta\in\Theta} c\cdot \mathcal{L}^c$ $\forall c\in\mathbb{R}_{\geq 0}$

Exmpl: $f(x;\theta) = \mathcal{N}(x|\mu,\sigma^2) \Rightarrow \mathcal{L}_n(\mu,\sigma^2) \propto \sigma^{-n}\exp\left(\frac{\sum_i(x_i-\mu)^2}{2\sigma^2}\right)$

$\hookrightarrow \mathcal{L}_n \propto \sigma^{-n}\exp\left(-\frac{nS^2}{2\sigma^2}\right)\exp\left(-\frac{n(\bar{X}-\mu)^2}{2\sigma^2}\right)$, $\bar{X} = \frac{1}{n}\sum_i X_i$, $S^2 = \frac{1}{n}\sum_i (X_i - \bar{X})^2$

$\hookrightarrow l_n(\mu,\sigma^2) = -n\cdot\log(\sigma) - \frac{nS^2}{2\sigma^2} - \frac{n(\bar{X}-\mu)^2}{2\sigma^2}$

$\hookrightarrow \frac{\partial l}{\partial\mu} = \frac{\partial l}{\partial\sigma} = 0 \Rightarrow \hat{\mu} = \bar{X}$, $\hat{\sigma} = S$

Jensen: φ convex: $\varphi(\mathbb{E}[X]) \leq \mathbb{E}[\varphi(X)]$

Why MLE?: · Consistency: $\hat{\theta}_n^{MLE} \xrightarrow{P} \theta^*$, θ^*: true value

· Equivariant: $\hat{\theta}_n^{MLE}$ MLE of $\theta \Rightarrow g(\hat{\theta}_n^{MLE})$ MLE of $g(\theta)$

· Asymptotic Normality: $(\hat{\theta} - \theta^*)/\hat{se} \sim \mathcal{N}(0,1) \iff (\hat{\theta}_n^{ML} - \theta^*)/\sqrt{n}$

· Asymptotic Optimality/Efficiency: $\sim\mathcal{N}(0, J^{-1}(\theta)\cdot I^{-1}(\theta))$

$\hat{\theta}_n^{MLE}$ has smallest variance among well behaved estim.

$\hookrightarrow \hat{\theta}_n^{ML}$ minimizes $\mathbb{E}[(\hat{\theta}_n - \theta^*)^2]$ as $n\to\infty$

Cramer-Rao Lower Bound: For any estim $\hat{\theta}$ of θ it holds:

$\mathbb{E}_{x|\theta}[(\hat{\theta}-\theta)^2] \geq \left(\frac{\partial}{\partial\theta}b_{\hat{\theta}} + 1\right)^2 / I(\theta) + b_{\hat{\theta}}^2$ where:

$I(\theta) = \mathbb{E}_{x|\theta}[\Lambda^2]$, $\Lambda = \frac{\partial}{\partial\theta}\log(p(x|\theta)) = \frac{1}{p(x|\theta)}\frac{\partial}{\partial\theta}p(x|\theta)$, $b_{\hat{\theta}} = \mathbb{E}_{x|\theta}[\hat{\theta}] - \theta$

Proof Idea: $\mathbb{E}_{x|\theta}[\Lambda] = 0$, $\mathbb{E}_{x|\theta}[\Lambda\hat{\theta}] = \frac{\partial}{\partial\theta}b_{\hat{\theta}} + 1$, $Cov(\Lambda,\hat{\theta})^2 = (\mathbb{E}_{x|\theta}[\Lambda\hat{\theta}])^2$ ← Cauchy Schwarz

Rem: $I(\theta)$ Fisher Information

· For unbiased estim $\hat{\theta}$: $\mathbb{E}[(\hat{\theta} - \theta_0)^2] \geq \frac{1}{I(\theta)}$

· If $\frac{\partial}{\partial\theta}b_{\hat{\theta}} < 0 \Rightarrow$ bias variance trade-off: $\frac{\partial}{\partial\theta}b_{\hat{\theta}} < 0$ vs. $b_{\hat{\theta}}^2$

Stein Estimator: For finite samples MLE not necessarily efficient. Consider $\mathcal{N}(\theta_0, \sigma^2\mathbb{1})$ in $\mathbb{R}^d$, $d\geq 3$.

$\hat{\theta}_{JS} := \left(1 - \frac{(d-2)\sigma^2}{\|y\|^2}\right)y$ for a sample $y\sim\mathcal{N}(\theta_0,\sigma^2\mathbb{1})$

$\hookrightarrow \mathbb{E}[(\hat{\theta}_{JS} - \theta_0)^2] \leq \mathbb{E}[(\hat{\theta}_{ML} - \theta_0)^2]$ $\forall\theta_0$

3.2 Bayesian Inference:

Let $\mathcal{X}$ = Data, $|\mathcal{X}| = n$

$p(X=x|\mathcal{X}) = \int p(x,\theta|\mathcal{X})d\theta = \int p(x|\theta)p(\theta|\mathcal{X})d\theta$

Asymptotically: $p(\theta|\mathcal{X}) \xrightarrow{n\to\infty} \delta(\theta - \hat{\theta}) \Rightarrow p(X=x|\mathcal{X}) \to p(x|\hat{\theta})$

Exmpl 1: $p(x|\theta) = \mathcal{N}(\theta,1)$, $p(\theta)\sim\mathcal{N}(m_0,s_0)$, $\mathcal{X} = \{x_1,\dots,x_n\}$

$\hookrightarrow p(\theta|\mathcal{X}) \propto p(\mathcal{X}|\theta)p(\theta) \propto \exp\left(-\frac{1}{2}\sum_{i=1}^n(x_i-\theta)^2\right)\cdot\exp\left(-\frac{(\theta-m_0)^2}{2s_0}\right)$

$\propto \exp\left(-\frac{1}{2}\left(n + \frac{1}{s_0}\right)\theta^2 + \left(n\bar{X} + \frac{m_0}{s_0}\right)\theta\right)$

$\Rightarrow \sigma_{post}^2 = \frac{1}{n + \frac{1}{s_0}} = \frac{s_0}{1 + ns_0}$, $\theta_{post} = \frac{n\bar{X} + \frac{m_0}{s_0}}{n + 1/s_0} = \frac{ns_0\bar{X} + m_0}{1 + ns_0}$

Exmpl 2: $Y = Xw + \varepsilon$, $w\sim\mathcal{N}(\mu_0,\Sigma_0)$, $\varepsilon\sim\mathcal{N}(0,\sigma^2\mathbb{1})$

$\Rightarrow p(Y|w) = \mathcal{N}(Y|Xw,\sigma^2\mathbb{1})$, $p(w) = \mathcal{N}(\mu_0,\Sigma_0)$, $p(w|Y) = \mathcal{N}(\mu_n,\Sigma_n)$

where $\Sigma_n = \left(\Sigma_0^{-1} + \frac{1}{\sigma^2}XX^T\right)^{-1}$, $\mu_n = \Sigma_n\left(\Sigma_0^{-1}\mu_0 + \frac{1}{\sigma^2}X^T Y\right)$

$\Rightarrow p(y_{new}|x_{new},Y) = \mathcal{N}(y_{new}|x_{new}^T\mu_n, x_{new}^T\Sigma_n x_{new} + \sigma^2)$

Rem: $n=0$: σ_{post}^2 & $\mu_{post} = \sigma_{prior}^2$ & μ_{prior}, $n\to\infty$: σ_{post}^2 & $\mu_{post} = \sigma_{MLE}^2$ & μ_{MLE}

4. Regression:

$y_i = f(x_i) + \varepsilon_i \Rightarrow$ estim f: $\hat{f}$

4.1 Linear Regression:

Multicolinearity ⇒ not full rank ⇒ Not invertible ⇒ $+\lambda\mathbb{1}$: invertible

$y_i = \beta^T x_i + \varepsilon_i$, $\beta\in\mathbb{R}^d$, $x_i\in\mathbb{R}^d$, $y_i\in\mathbb{R}$, $\varepsilon_i\sim\mathcal{N}(0,\sigma^2) \Rightarrow y_i\sim\mathcal{N}(\beta^T x_i, 1)$

OLSE: minimize MSE: $\hat{\beta} = \arg\min_\beta \frac{1}{n}\sum_{i=1}^n (y_i - \beta^T x_i)^2 = \arg\min_\beta \mathcal{L}(\beta)$

where $\mathcal{L}(\beta) = \frac{1}{n}\|y - X\beta\|_2^2 \Rightarrow \hat{\beta} = (X^T X)^{-1}X^T Y$ ← unbiased

If $X = UDV^T \Rightarrow \hat{\beta} = VD^{-1}U^T Y \Rightarrow$ if features correlated $(D^{-1})\nearrow$

Regularization: Ridge: Gaussian prior, LASSO: Laplace prior,

Exmpl Ridge: $\hat{\beta}_\lambda = \arg\min_\beta \|Y - X\beta\|_2^2 + \lambda\|\beta\|_2^2 = (X^T X + \lambda\mathbb{1})^{-1}X^T Y$

4.2 Bias-Variance Tradeoff:

$Y = f(x) + \varepsilon$; ⇒ estim: $\hat{f}$

$MSE = \mathbb{E}[(\hat{f}(x) - y)^2] = \underbrace{(\mathbb{E}[\hat{f}(x)] - f(x))^2}_{Bias^2} + \underbrace{Var[\hat{f}(x)]}_{Variance} + \underbrace{\mathbb{E}[\varepsilon^2]}_{Noise}$

Rem: OLSE is the unique minimum variance unbiased linear estimator.

Linear Models are very biased.

4.3 LinReg w/ Trafos:

$\phi: \mathbb{R}^d \to \mathbb{R}^m \Rightarrow y_i = \beta^T\phi(x_i)$ with $\beta\in\mathbb{R}^m \Rightarrow \hat{\beta} = (\Phi^T\Phi)^{-1}\Phi Y$

Polynomials: Possible to approx (almost) any fct f w/ a polynomial.

Goal: Compute polyn. of arbitrary deg. that minimizes $\mathcal{L}(\gamma;\mathcal{D}) = \sum_{i=1}^n (y_i - \gamma(x_i))^2$

Approach: $\gamma(x) = \langle\beta,\varphi(x)\rangle$ for $\beta\in\mathbb{R}^\infty$ & $\varphi: \mathbb{R}^d\to\mathbb{R}^\infty$

Idea 1: $\beta = (\beta_{j_1,\dots,j_d})_{j_i\in\mathbb{N}}$, $\tilde{\varphi}(x) = (x_1^{j_1},\dots,x_d^{j_d})_{j_i\in\mathbb{N}} \Rightarrow \tilde{\varphi}^T\tilde{\varphi} = \infty$ ↯

Solution: $\varphi(x) = \exp\left(-\frac{\|x\|^2}{2}\right)\left(\frac{x_1^{j_1}\cdots x_d^{j_d}}{\sqrt{j_1!\cdots j_d!}}\right)_{j_i\in\mathbb{N}} \Rightarrow \varphi^T\varphi = \exp\left(\frac{\|x - x'\|^2}{2}\right)$

$\hookrightarrow \hat{\beta} = \arg\min_{\beta\in\mathbb{R}^\infty}\sum_{i=1}^n (y_i - \beta^T\varphi(x_i))^2 = (\Phi^T\Phi)^{-1}\Phi^T Y \Rightarrow \Phi\in\mathbb{R}^{n\times\infty}$, $\Phi^T\Phi\in\mathbb{R}^{\infty\times\infty}$ ↯

$\Rightarrow \hat{\beta} = \Phi^T(\Phi\Phi^T)^{-1}Y$, $A = \Phi\Phi^T\in\mathbb{R}^{n\times n}$, $(\Phi^T\Phi)^{-1}\Phi^T = \Phi^T(\Phi\Phi^T)^{-1}$, $A_{ij} = \varphi(x_i)^T\varphi(x_j) = A_{ij} = K(x_i,x_j)$

Problem: $\Phi^T\in\mathbb{R}^{\infty\times n} \Rightarrow$ no problem: $\hat{f}(x) = \varphi(x)\hat{\beta} = \varphi(x)\Phi^T A^{-1}Y$

$k_i(x) = \varphi(x)^T\varphi(x_i) \Rightarrow \hat{\varphi}(x) = k(x)K^{-1}Y$ $\mathcal{O}(n^3)$ time, $=k(x)\in\mathbb{R}^m$

Regularization: $\hat{f}_\lambda(x) = k(\hat{x})(K + \lambda\mathbb{1})^{-1}Y$

4.4 Causality:

Shortcut learning: occurs when there is a spurious correlation btw causal & non-causal features in training. This usually occurs when there is a domain shift. i.e. test samples are drawn from different distr. than the training samples. Shortcut learning happens when the resulting estimator (ab)uses the non-causal features to generalize in the testing dataset. → Nachtrag next page

Counterfactual Invariance:

Let: X: features, Y: target var., W: non-causal feature vec. (influences X but does not influence Y)

Let f be fct used to estim Y based on X. $\forall w,w'\in W$

f is counterfactually invariant if $f(X(w)) = f(X(w'))$

Causal Scenario: Features in X are causes of Y

Anti-Causal Scenario: Features in X are caused by Y

Confounding: ∃ hidden RV that causes/influences X & W

Selection: ∃ hidden RV S that filters $\mathcal{D}$ based on X & W

Thm: If f is a counterfactually invariant predictor:

1) For anti-causal scenario: $f(X)\perp W|Y$, 2) For causal scenario without selection (but possibly confounded): $f(X)\perp W$

3) For causal scenario without confoundness (but possibly with selection: $Y\perp X|W, X_w^\perp$ & $f(X)\perp W|Y$

d-separation: A path (from X_i to X_j) is d-separated by S if all paths between X_i & X_j are blocked by S. A path is blocked if either 1) non-collider $\in S$, 2) collider & its descendants $\notin S$

Rem: d-sep ⇒ indep., not d-sep ⇏ dep

5. Gaussian Processes:

5.1 Bayesian Linear Regression:

$X = (X_1,\dots,X_d)^T$, $\beta\in\mathbb{R}^d$, $Y = X^T\beta + \varepsilon$, $\varepsilon\sim\mathcal{N}(\varepsilon|0,\sigma^2)$

$\hookrightarrow p(Y|X,\beta,\sigma) = \mathcal{N}(Y|X^T\beta,\sigma^2) \propto \exp\left(-\frac{1}{2}(Y - X^T\beta)^2\right)$

$\hookrightarrow$ prior: $p(\beta) = \mathcal{N}(\beta|0,\Lambda^{-1}) \propto \exp\left(-\frac{1}{2}\beta^T\Lambda\beta\right)$

$\hookrightarrow$ posterior (observed $X\in\mathbb{R}^{n\times d}$ & $y\in\mathbb{R}^n$): $p(\beta|X,y) = \mathcal{N}(\mu_\beta,\Sigma_\beta)$ with $\Sigma_\beta = \sigma^2(X^T X + \sigma^2\Lambda)^{-1}$, $\mu_\beta = (X^T X + \sigma^2\Lambda)^{-1}X^T y$

Rem: $\Lambda^{-1} = \alpha^{-1}\mathbb{1}$: $\alpha\to 0$ (max uncertainty): $\mu_\beta\to\mu_{MLE}$, $N=0$: $\mu_\beta\to\mu_0$ (prior)

5.2 Gaussian Processes:

GPs model the distribution over functions.

Def: A GP is a (potentially ∞) collection of RVs s.t. every finite subset of RVs is jointly Gaussian.

We write: $f\sim GP(\mu,K)$, $\mu(x)$ mean fct, $K(x,x')$ kernel

$Y = f(X) + \varepsilon \to Y\sim\mathcal{N}(y|0, K + \sigma^2\mathbb{1})$, $\varepsilon\sim\mathcal{N}(0,\sigma^2\mathbb{1})$

5.3 Predictions using GPs:

$\tilde{\mu} = \begin{pmatrix}\mu_A\\ \mu(x')\end{pmatrix}$, $\tilde{K} = \begin{bmatrix}K_{AA} + \sigma^2\mathbb{1} & k_{x'A}\\ k_{x'A}^T & k(x',x')\end{bmatrix}$

Goal: $p(y_{n+1}|x_{n+1}, X, y)$

Approach: $p\left(\begin{bmatrix}y\\ y_{n+1}\end{bmatrix}\middle| x_{n+1}, X\right) = \mathcal{N}\left(\begin{bmatrix}y\\ y_{n+1}\end{bmatrix}\middle| 0, \begin{bmatrix}K + \sigma^2\mathbb{1} & k\\ k^T & c\end{bmatrix}\right)$

where: $K_{ij} = k(x_i,x_j)$, $k = k(x_{n+1},X)$, $c = k(x_{n+1},x_{n+1}) + \sigma^2$

Thm: (Cond. Gaussians): $\begin{bmatrix}a_1\\ a_2\end{bmatrix}\sim\mathcal{N}\left(\begin{bmatrix}u_1\\ u_2\end{bmatrix}, \begin{bmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{21} & \Sigma_{22}\end{bmatrix}\right)$

$\hookrightarrow p(a_2|a_1 = z) = \mathcal{N}(u_2 + \Sigma_{21}\Sigma_{11}^{-1}(z - u_1), \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12})$

$\Rightarrow p(y_{n+1}|x_{n+1}, y, X) = \mathcal{N}(y_{n+1}|\mu_{n+1},\sigma_{n+1}^2)$ where:

$\mu_{n+1} = k^T(K + \sigma^2\mathbb{1})^{-1}y$, $\sigma_{n+1}^2 = c - k^T(K + \sigma^2\mathbb{1})^{-1}k$, $\varepsilon_i\sim\mathcal{N}(0,\sigma_n^2)$

Given $A = \{x_1,\dots,x_n\}$ & $y_i = f(x_i) + \varepsilon_i$, prior: $f\sim GP(0,K)$,

⇒ Posterior: $f\sim GP(\mu',K')$, $\mu'(x) = \mu(x) + k_{xA}^T(K_{AA} + \sigma_n^2\mathbb{1})^{-1}(y_A - \mu_A)$

5.4 Kernels:

$k(x,x') = Cov(f(x),f(x'))$ computes the similarity between two data points. $k'(x,x') = k(x,x') - k_{xA}^T(K_{AA} + \sigma_n^2\mathbb{1})^{-1}k_{x'A}$

Def: A fct $k: \mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}$ is a kernel fct iff it is an inner product $\langle\phi(x),\phi(x')\rangle$ for some mapping (∞ dim) $\phi(x)$

Properties of Kernels: valid kernels must satisfy: 1) Symmetry: $k(x,x') = k(x',x)$, 2) PSD: $\int_\Omega k(x,x')f(x)f(x')dxdx' \geq 0$

Equiv for 2): $\forall n$: $S = \{x_1,\dots,x_n\}$: GramMat: K: $K_{ij} = K(x_i,x_j)$ must be psd: $\forall a\in\mathbb{R}^n$: $a^T K a \geq 0$

Important kernels: RBF: $k(x,x') = \sigma^2\exp\left(-\frac{\|x - x'\|^2}{2l^2}\right)$

Periodic: $k(x,x') = \sigma^2\exp\left(-\frac{2}{l^2}\sin^2\left(\frac{\pi}{p}\|x - x'\|\right)\right)$

Laplace: $k(x,x') = \sigma^2\exp\left(-\frac{1}{l}|x - x'|\right)$

Kernel Composition:

Let: k_1 & k_2 be kernels, $c\in\mathbb{R}_{>0}$, $p: \mathbb{R}\to\mathbb{R}$ polynomial with positive coeff, $f: \mathcal{X}\to\mathbb{R}$ real valued fct, A a psd Mat.

Valid kernels are: 1) $k_1(x,x') + k_2(x,x')$, 2) $c\cdot k(x,x')$

3) $k_1(x,x')\cdot k_2(x,x')$, 4) $p(k_1(x,x'))$, 5) $\exp(k_1(x,x'))$

6) $f(x)k(x,x')f(x')$, 7) $x^T A x'$

Rem: Bayesian LinReg = GPR with $k(x,x') = \lambda x^T x'$

Stationary: $k(x,x') = \tilde{k}(x - x')$, Isotropic: $k(x,x') = \tilde{k}(\|x - x'\|_2)$

6. Model Valid. & Uncert. Quant

6.1 Statistical Learning & Model Val.

Goal: minimize expected risk $R(f) = \mathbb{E}[\mathbb{1}\{f(X)\neq Y\}]$

Given samples $\mathcal{Z} = \{(x_i,y_i)\}_{i=1}^{n}$: empirical risk: $\hat{R}(f) = \frac{1}{n}\sum_{i=1}^n \mathcal{L}(y_i, f(x_i)) \Rightarrow \hat{f} = \arg\min_{f\in\mathcal{H}}\hat{R}(f)$

Model Validation: def $R(\mathcal{A}) = \mathbb{E}_{\mathcal{Z}}[R(\mathcal{A}(\mathcal{Z}))]$ for algorithm $\mathcal{A}$. How to estim $\hat{R}(\mathcal{A})$:

Cross-Validation: 1) Partition $\mathcal{Z} = \mathcal{Z}_1\sqcup\dots\sqcup\mathcal{Z}_K$

2) $\forall\mathcal{Z}_v$ use Algo. $\mathcal{A}$ to train model $\hat{f}^{-v}$ on $\mathcal{Z}\setminus\mathcal{Z}_v$

3) $\hat{R}(\mathcal{A}) = \frac{1}{n}\sum_{i=1}^n \mathcal{L}(y_i, \hat{f}^{-k(i)}(x_i))$ where k maps i to the partition $\mathcal{Z}_{k(i)}$ which contains x_i

6.2 Bootstrap:

Measuring the distr. of params.

1) For $b = 1,\dots,B$ draw $\mathcal{Z}^{*b}$ of size n with replacement from $\mathcal{Z}$

2) Compute estimate (param of interest) $S(\mathcal{Z}^{*b})$ $\forall b = 1,\dots,B$

3) Calculate: $\bar{S} = \frac{1}{B}\sum_b S(\mathcal{Z}^{*b})$, $\sigma^2(S) = \frac{1}{B-1}\sum_b (S(\mathcal{Z}^{*b}) - \bar{S})^2$

Empirical Risk: $\hat{R}(\mathcal{A}) = \frac{1}{B}\sum_{b=1}^B \frac{1}{n}\sum_{i=1}^n \mathcal{L}(y_i, \hat{f}^{*b}(x_i))$ ← too optim!

$\hat{R}^{bs} = \frac{1}{n}\sum_{i=1}^n \frac{1}{|C^{-i}|}\sum_{b\in C^{-i}} \mathcal{L}(y_i, \hat{f}^{*b}(x_i))$, C^{-i} contains all BS indices b s.t. $\mathcal{Z}^{*b}$ does not contain obs. (x_i,y_i)

Rem: Prob of (x_i,y_i) appears at least in 1 BS sample of size n: $1 - \left(1 - \frac{1}{n}\right)^n \xrightarrow{n\to\infty} 1 - \frac{1}{e} \approx 0.632 \Rightarrow$ Def $\hat{R}^{0.632}$

$\hat{R}^{0.632} = \frac{1}{e}\hat{R}(\mathcal{A}(\mathcal{Z})) + \left(1 - \frac{1}{e}\right)\hat{R}^{bs}$

6.3 Uncertainty Quantification with Linear Methods

Properties of OLSE: $X\in\mathbb{R}^{n\times d}$, $Y\in\mathbb{R}^n$ RVs: $Y|X\sim\mathcal{N}(X\beta_*,\sigma^2\mathbb{1})$

$\Rightarrow \hat{\beta}|X\sim\mathcal{N}(\beta_*,\sigma^2(X^T X)^{-1}) \Rightarrow \hat{\sigma}^2 = \frac{1}{n-d}\sum_{i=1}^n (X\hat{\beta} - Y)^2$

$\Rightarrow 1-\alpha$ conf. interval for β_{*j} is: $\hat{\beta}_j \pm z_{\alpha/2}\hat{se}(\hat{\beta}_j)$ where: $z_{\alpha/2} = \Phi^{-1}\left(\frac{\alpha}{2}\right)$, $\hat{se}(\hat{\beta}_j)$ = j-th diag element of $\sigma^2(X^T X)^{-1}$

Statistical Testing: Hypothesis set: $\mathcal{H} = \{p(.|\theta) : \theta\in\Theta\}$

Let $\theta^*\in\Theta$ be true param., Null Hyp. H_0: $\theta^*\in\Theta_0$, alternative: H_1: $\theta^*\in\Theta_1$ for predef $\Theta_0,\Theta_1\subseteq\Theta$, Samples $X_1,\dots,X_n\sim p(.|\theta^*)$ & statistic $t(X_1,\dots,X_n)$. Goal: find critical value $c\in\mathbb{R}$

$\mathbb{P}(t(X_1,\dots,X_n)\geq c|\theta)$ low if $\theta\in\Theta_0$ & high if $\theta\in\Theta_1$

Accept H_0 if $t(X_1,\dots,X_n) < c$ & Reject H_0 if $t(X_1,\dots,X_n)\geq c$

p-value: $\sup_{\theta\in\Theta_0}\mathbb{P}(t(X_1,\dots,X_n)\geq t(x_1,\dots,x_n)|\theta)$. ← Risk of the least risky critical value that rejects H_0.

Rem: prob. of seeing a more extreme value than what I observed

Wald Test: Let $\hat{\theta}$ estim of θ with sd $\hat{se}$ that is asymptot. normal $((\hat{\theta} - \theta_0)/\hat{se}) \to\mathcal{N}(0,1)$ pointwise. Let H_0: $\theta = \theta_0$ & H_1: $\theta\neq\theta_0$

⇒ Wald test stat.: $W = \frac{\hat{\theta} - \theta_0}{\hat{se}} \sim\mathcal{N}(0,1)$ under H_0

6.4 Bayesian Neural Networks:

Goal: $\hat{w} = \arg\min_w \hat{R}(f(.|w))$ where $f(.|w)$ is a NN

Gradient Descent: 1) Init $w^{(0)}$, define LR fct $t\to\alpha_t$

2) for $t = 1,\dots,T$ do $w^{(t)} \leftarrow w^{(t-1)} - \alpha_t\nabla_w\hat{R}(f(.|w^{(t-1)}))$

Robinson-Monro Algo: Let $h: \mathbb{R}\to\mathbb{R}$ unknown fct. We have noisy obs: $\Delta(w) = h(w) + \varepsilon$ with $\mathbb{E}[\varepsilon] = 0$. Goal: find $w^*\in\mathbb{R}$ s.t. $\mathbb{E}[\Delta(w^*)] = 0$ ⇒ RM-Algo:

1) Choose LRs $\alpha_1,\alpha_2,\dots$, 2) make initial guess w_0

3) for $t = 0,1,\dots$ do $w_{t+1}\leftarrow w_t - \alpha_t\Delta(w_t)$

RM Cond: If $\sum_t\alpha_t = \infty$ & $\sum_t\alpha_t^2 < \infty$, then: $\lim_{t\to\infty}\mathbb{P}(|w_t - w^*| > \varepsilon) = 0$ $\forall\varepsilon > 0$ ⇒ e.g. $\alpha_t = \frac{1}{t}$

In practice: $\alpha_t = 0.001$ ⇒ more stable updates but sacrifice ability to exactly converge to optimal solution.

Stochastic GD: 1) Init $w^{(0)}$, def LR fct $t\to\alpha_t$

2) for $t = 1,\dots,T$ do: 2.1) pick a point (x,y) from $\mathcal{Z}$.

2.2) Update: $w^{(t)}\leftarrow w^{(t-1)} - \alpha_t\nabla\mathcal{L}(y, f(x|w^{(t-1)}))$

Bayesian Neural Networks: Let w be weights of NN

1) Def prior $p(w)$ over $w\in W$. Assume: $w\sim\mathcal{N}(0,\alpha^2\mathbb{1})$

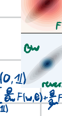
2) Likelihood fct: $w\to p(\mathcal{D}|w)$. If iid: $p(\mathcal{D}|w) = \prod_{(x_i,y_i)\in\mathcal{D}} p(y_i|x_i,w)$

3) Bayes Rule: $p(w|\mathcal{D}) \propto p(\mathcal{D}|w)p(w)$ ← intractable

Variational Inference: approximate posterior with a parametric distr. e.g. Gaussian: $q(w|\theta) = \mathcal{N}(\mu,\sigma^2\mathbb{1})$, $\theta = (\mu,\sigma^2)$

$\Rightarrow \theta^* = \arg\min_\theta KL(q(w|\theta)\|p(w|\mathcal{D}))$ (reverse KL)

Reverse KL: fitted around mode of $p(w|\mathcal{D})$



Equiv.: $\theta^* = \arg\min_\theta \mathbb{E}_{w\sim q(.|\theta)}[F(w,\theta)]$ where:

$F(w,\theta) = \log(q(w|\theta)) - \log(p(w)) - \log(p(\mathcal{D}|w))$

Reparam Trick: $w\sim\mathcal{N}(\mu,\sigma^2\mathbb{1}) \iff w = \mu + \sigma\varepsilon$, $\varepsilon\sim\mathcal{N}(0,\mathbb{1})$

⇒ find $\theta^* = (\mu^*,\sigma^{2*}) \Rightarrow \frac{\partial}{\partial\mu}\mathbb{E}_{w\sim q(.|\theta)}[F(w,\theta)] = \mathbb{E}_{\varepsilon\sim\mathcal{N}(0,\mathbb{1})}\left[\frac{\partial}{\partial w}F(w,\theta) + \frac{\partial}{\partial\mu}F(w,\theta)\right]$

$\Rightarrow \frac{\partial}{\partial\sigma}\mathbb{E}_{w\sim q(.|\theta)}[F(w,\theta)] = \mathbb{E}_{\varepsilon\sim\mathcal{N}(0,\mathbb{1})}\left[\varepsilon^T\frac{\partial}{\partial w}F(w,\theta) + \frac{\partial}{\partial\sigma}F(w,\theta)\right]$

Algo: GD for VI: 1) Init μ,σ & LR fct $t\to\alpha_t$. 2) for $t = 1,2,\dots$ do

2.1) Sample $\varepsilon\sim\mathcal{N}(0,\mathbb{1})$, 2.2) Compute $F(w,\theta)$, 2.3) update: $\mu^{(t+1)}\leftarrow\mu^{(t)} - \alpha_t\left(\frac{\partial}{\partial w}F(w,\theta) + \frac{\partial}{\partial\mu}F(w,\theta)\right)$, $\sigma^{(t+1)}\leftarrow\sigma^{(t)} - \alpha_t\left(\varepsilon^T\frac{\partial}{\partial w}F + \frac{\partial}{\partial\sigma}F\right)$

Properties of $\mathcal{N}$'s: $X\sim\mathcal{N}(\mu,\Sigma) \Rightarrow MX\sim\mathcal{N}(M\mu, M\Sigma M^T)$

$X'\sim\mathcal{N}(\mu',\Sigma') \Rightarrow X + X'\sim\mathcal{N}(\mu + \mu', \Sigma + \Sigma')$, $\varepsilon\sim\mathcal{N}(0,\mathbb{1}) \Rightarrow X = \Sigma^{1/2}\varepsilon + \mu$

6.5 Active Learning:

Domain $\mathcal{X}$, target space $\mathcal{A}\subseteq\mathcal{X}$, sample space $S\subseteq\mathcal{X}$. Observations $y_x = f_x + \varepsilon_x$ where ε_x noise with $\mathbb{E}[\varepsilon] = 0$ & assume $f\sim GP(\mu,k)$. Given $\mathcal{D}_{n-1} = \{(x_i,y_i)\}_{i=1}^{n-1}\subseteq S\times\mathbb{R}$ pick x_n & update μ & k.

Information-based Transductive Learning:

$x_n = \arg\max_{x\in S} I(\{f_x\}_{x\in\mathcal{A}}; y_x|\mathcal{D}_{n-1})$ Maximize conditional mutual information restricted to $\mathcal{A}$. If $f\sim GP(\mu,k)$:

$I(f_{\mathcal{A}}; y_x|\mathcal{D}_{n-1}) = \frac{1}{2}\log\left(\frac{Var[y_x|\mathcal{D}_{n-1}]}{Var[y_x|f_{\mathcal{A}},\mathcal{D}_{n-1}]}\right)$ Proof: $H(a|B) = \dots = \frac{1}{2}\log(2\pi e\sigma_{a|B}^2)$

Safe Bayesian Optimization:

Let f be a stochastic process we try to maximize.

Let g be another stochastic process modeling the safe area. $S = \{x\in\mathcal{X} : g(x)\geq 0\}$. f, g & S are unknown & need to be learned.

Def: l_n^f & u_n^f s.t.: $\forall x\in\mathcal{X}$, $[l_n^f(x), u_n^f(x)]$ is the 95% CI of $\mathbb{E}[f(x)]$

Def: l_n^g & u_n^g analogously.

Def: $S_n = \{x : l_n^g(x)\geq 0\}$, $\hat{S}_n = \{x : u_n^g(x)\geq 0\}$

Set of estims: $\mathcal{A}_n = \{x\in\hat{S}_n : u_n^f(x)\geq\max_{x'\in S_n} l_n^f(x')\}$

⇒ Perform ITL using S_n as sample space & $\mathcal{A}_n$ as target space

6.6 Batch Active Learning:

Select a batch of points in each iteration.

Goal: find $L\subseteq\mathcal{X}$ s.t. $|L| = b$ & query oracle for: $\{f(x) : x\in L\}$.

Defs: · $\forall x\in\mathcal{X}$ & $\delta > 0$, let $B_\delta(x) = \{x'\in\mathcal{X} : \|x' - x\|\leq\delta\}$

$B_\delta(x)$ is pure if f is const all over $B_\delta(x)$

· For $L\subseteq\mathcal{X}$, let $C(L,\delta) = \bigcup_{x\in L} B_\delta(x)$. Note $C = C_r\cup C_w$ where $C_r = \{x\in C : \hat{f}(x) = f(x)\}$ & $C_w = \{x\in C : \hat{f}(x)\neq f(x)\}$

· Impurity of δ is $\pi(\delta) = \mathbb{P}(\{x\in\mathcal{X} : B_\delta(x)$ is not pure$\})$

· $L\subseteq\mathcal{X}$, known $Z = \{(x,f(x)) : x\in L\}$, $\hat{f}$: 1-NN Classifier

Rem: · $C_w\subseteq\{x : B_\delta(x)$ is not pure$\} \Rightarrow \mathbb{P}(C_w)\leq\pi(\delta)$

· $\{x : \hat{f}(x)\neq f(x)\}\subseteq C_r^c\cup C_w\subseteq C^c\cup\{x : B_\delta(x)$ is not pure$\}$

· ⇒ $R(\hat{f}) = \mathbb{P}(\hat{f}(x)\neq f(x))\leq 1 - \mathbb{P}(C(L,\delta)) + \pi(\delta)$

⇒ Choose L & δ that minimizes $R(\hat{f})$

⇒ $\arg\max_{L\subseteq\mathcal{X}, |L|=b} P\left(\bigcup_{x\in L} B_\delta(x)\right) \approx \arg\max_{L\subseteq\mathcal{X}, |L|=b} \frac{1}{|X|}|\{x' : \|x' - x\|\leq\delta$ for $x\in L\}|$

Probcover Algo:

Require $X\subseteq\mathcal{X}$, $b\in\mathbb{N}$, Ensure: $L\subseteq X$ of size b

1) $G = (X,E)$ where $E = \{(x,x') : \|x - x'\|\leq\delta\}$, 2) $L\leftarrow\emptyset$,

3) for $i = 1\dots b$ do: 3.1) $\hat{x}\leftarrow\arg\max_{x\in X}|\{x' : (x,x')\in E, x'\in X\}|$,

3.2) $L\leftarrow L\cup\{\hat{x}\}$, 3.3) $E\leftarrow E\setminus(\{\hat{x}\}\times(B_\delta(\hat{x})\cap X))$

7. Convex Optimization:

Def: A constrained optimization problem is a problem of the form: $\min_{w\in\mathbb{R}^d} f(w)$ s.t. $g_i(w) = 0$, $h_j(w)\leq 0$ $\forall i = 1,\dots,m$, $j = 1,\dots,n$

7.1 Convex Optimization with Lagrange Multiplier

Def: The Lagrangian of a constrained optim problem is: $\mathcal{L}(w,\lambda,\alpha) = f(w) + \sum_{i=1}^m\lambda_i g_i(w) + \sum_{j=1}^n\alpha_j h_j(w)$

Lemma: Any optimal solution w^* of a convex optim problem must fulfil: $\nabla_w\mathcal{L} = 0$, $g_i(w) = 0$ $\forall i$, $h_j(w)\leq 0$, $\alpha_j\geq 0$ $\forall j$

Dual Problem:

Lemma: Given $\mathcal{L}$ of convex optim problem, with optim w^* & $\alpha\geq 0$, then: $\mathcal{L} \overset{g(w^*)=0}{=} f(w) + \sum_{j=1}^n\alpha_j h_j(w) \overset{h\leq 0}{\leq} f(w)$

$\Rightarrow \inf_w\mathcal{L}\leq\mathcal{L}(w^*)\leq f(w^*)$

$\Rightarrow \max_{\lambda,\alpha\geq 0}\inf_w\mathcal{L}(\lambda,\alpha,w) =: \max_{\lambda,\alpha\geq 0}\theta(\lambda,\alpha)\leq f(w^*)$ (convex)

$\Rightarrow \theta(\lambda^*,\alpha^*)\leq f(w^*) \Rightarrow \lambda^*,\alpha^*$ sol to: $\max_{\lambda,\alpha\geq 0}\theta(\lambda,\alpha)$

Def: Maximization of $\theta(\lambda,\alpha)$ is called the dual problem

Strong Duality: Dual provides a lower bound

Def: A convex optim problem satisfies strong duality if $\theta(\lambda^*,\alpha^*) = f(w^*)$ where w^* sol of primal problem & (λ^*,α^*) is sol of dual problem

Slaters condition: A convex optim problem satisfies Slaters Cond. if ∃ w^0 feasible s.t. $h_j(w^0) < 0$ $\forall j$

Lemma: If a convex optim problem satisfies Slater's condition, then it satisfies strong duality

Rem: · Converse is not true.

In short $h_j(w^*) < 0$ $\forall j$ ⇒ sol of dual = sol of primal

Lemma: If a convex optim problem satisfies strong duality, then $\forall$ optim w^*: $f(w^*) = \mathcal{L}(\lambda^*,\alpha^*,w^*)$ where (λ^*,α^*) sol of dual. Complementary slackness: $\alpha_j h_j(w^*) = 0$ $\forall j$

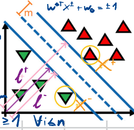
Solving convex optim problems: 2 strategies

1) Solve primal: 1) verify Slaters Cond., 2) if true, solve: $\nabla\mathcal{L} = 0$, $g_i = 0$, $h_j\leq 0$, $\alpha_j\geq 0$, $\alpha_j h_j = 0$ $\forall i,j$. If intractable:

2) Solve Dual $\quad$ * $w^* = \arg\min_w\mathcal{L}(w,\lambda^*,\alpha^*)$

Gamma Distr: $\gamma(x;\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}x^{\alpha-1}e^{-\beta x}$ ⇒ Mean: α/β, Variance: α/β^2

7.2 SVMs:



$\max_{w,w_0} 2m$ s.t. $y_i(w^T x_i + w_0)\geq m$

Reformulate the primal: $2m = \|l^+ - l^-\| = \frac{|w^T x^+ - w^T x^-|}{\|w\|}$

Rem: (w,w_0) optim → $(\gamma w,\gamma w_0)$ optim, $\gamma > 0$

$\Rightarrow \min_{w,w_0}\frac{1}{2}\|w\|_2^2$ s.t. $y_i(w^T x_i + w_0)\geq 1$ $\forall i\leq n$

Checking Slater's Condition: ✓

∃ (w,w_0): $y_i(w^T x_i + w_0)\geq 1$ $\forall i \Rightarrow y_i(\gamma w^T x_i + \gamma w_0) > 1$, $\gamma > 1$ ✓

Formulation of the dual:

$\max_\alpha\min_{w,w_0}\mathcal{L}(w,w_0,\alpha)$ s.t. $\alpha_i\geq 0$ $\forall i$

where: $\mathcal{L}(w,w_0,\alpha) = \frac{1}{2}\|w\|_2^2 + \sum_{i=1}^n\alpha_i(1 - y_i(w^T x_i + w_0))$

$\Rightarrow \nabla\mathcal{L} = 0 \Rightarrow w^* = \sum_{i=1}^n\alpha_i y_i x_i$, $\sum_{i=1}^n\alpha_i y_i = 0$

$\Rightarrow \mathcal{L}(w^*,w_0^*,\alpha) = \sum_{i=1}^n\alpha_i - \frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n\alpha_i\alpha_j y_i y_j x_i^T x_j$

$\Rightarrow$ Dual: $\max_\alpha\mathcal{L}(w^*,w_0^*,\alpha)$ s.t. $\begin{cases}\alpha_i\geq 0\\ \sum_i\alpha_i y_i = 0\end{cases}$

Retrieving the optimal solution:

Complementary Slackness: $\alpha_i^*(1 - y_i(w^{*T}x_i + w_0^*)) = 0$

⇒ either $\alpha_i^* = 0$ or $y_i(w^{*T}x_i + w_0^*) = 1 \iff x_i$ is support vect.

⇒ α_i^* only $\neq 0$ if x_i is support vector

⇒ w^* is sparse lin. comb of support vectors

⇒ $w_0^* = -\frac{1}{2}(w^{*T}x^+ + w^{*T}x^-)$

7.3 Soft Margin SVM:

Allow some points to be misclassified:

$\min_{w,w_0,\xi}\frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^n\xi_i$ s.t.: $y_i(w^T x_i + w_0)\geq 1 - \xi_i$ $\forall i$, $\xi_i\geq 0$ $\forall i$

C controls trade-off between the penalty & margin

$C\to\infty$ ⇒ hard SVM

Dual Problem:

$\mathcal{L}(w,w_0,\xi,\alpha,\lambda) = \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^n\xi_i + \sum_{i=1}^n\alpha_i(1 - \xi_i - y_i(w^T x_i + w_0)) - \sum_{i=1}^n\lambda_i\xi_i$

Rem: $\xi_i = 0$ $\forall x_i\notin$ margin, $\xi_i = 1$ $\forall x_i$ on boundary, $\xi_i > 1$ $\forall x_i$ misclassified.

7.4 Kernelization:

Def Feature-Map: $\varphi(x)$

$\hookrightarrow\mathcal{L} = \sum_{i=1}^n\alpha_i - \frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n\alpha_i\alpha_j y_i y_j\underbrace{\varphi(x_i)^T\varphi(x_j)}_{=K(x_i,x_j)}$

$\hookrightarrow w^{*T}\varphi(x) = \sum_{i=1}^n\alpha_i^* y_i\underbrace{\varphi(x_i)^T\varphi(x)}_{K(x_i,x)}$

7.5 Inputs from IML:

Objective: $\max_{\|w\|_2^2=1}\min_{1\leq i\leq n} y_i\langle w,x_i\rangle$ (= margin)

equiv.: $\min_{w\in\mathbb{R}^d}\|w\|_2$ s.t. $y_i\langle w,x_i\rangle\geq 0$ $\forall i$

Soft-Margin SVM: $\min_{w\in\mathbb{R}^d,\xi\in\mathbb{R}^d}\left[\|w\|_2^2 + C\sum_{i=1}^n\max(0, 1 - y_i\cdot w^T x_i)\right]$ (= "Hinge Loss")

7.6 Inputs Buhmann:

Soft SVM: $\min\frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^n\xi_i$ s.t. $\forall i$: $z_i(w^T y_i + w_0)\geq 1 - \xi_i$, $\xi_i\geq 0$

Dual: $\max\sum_{i=1}^n\alpha_i - \frac{1}{2}\sum_i\sum_j z_i z_j\alpha_i\alpha_j x_i^T x_j$ s.t. $\sum_{j=1}^n z_j\alpha_j = 0$ & $\forall i$: $C\geq\alpha_i\geq 0$

Rem: Only difference to hard SVM: upper bound $\alpha_i\leq C$

7.7 Structural SVMs:

Multiclass SVMs: each class has own w:

$\min_{\{w_z\}_{z=1}^K}\frac{1}{2}\sum_{z=1}^K w_z^T w_z + C\cdot\sum_i\xi_i$ s.t.: (Soft SVM)

$(w_{z_i}^T y_i + w_{z_i,0}) - \max_{z\neq z_i}(w_z^T y_i + w_{z,0})\geq 1 - \xi_i$ $\forall y_i\in\mathcal{Y}$

Structural SVMs:

Predict whole complex state, not just parts of it.

Problems: 1) Compact representation of output space, 2) Efficient Prediction, 3) Prediction error, 4) Efficient Training.

Output space representation: SSVMs could be formulated as multiclass SVMs (one weight vector per class) however ⇒ blow up of #params! ↯ Solution:

1) Def: Joint Feature Map $\Psi(z,y)$ that combines properties of inputs & outputs.

2) Def: Scoring Fct.: $f_w(z,y) = w^T\Psi(z,y)$

$\hookrightarrow$ #features depends on dim of Ψ & indep of #classes

3) Perform Classification via: $\hat{z} := h(y) = \arg\max_{z\in K} f_w(z,y)$

Optim Problem:

$\min_w\frac{1}{2}w^T w$ s.t.: $w^T\Psi(z_i,y_i) - \max_{z\neq z_i} w^T\Psi(z,y_i)\geq 1$

Soft Optim Problem:

Def loss fct $\Delta: \mathbb{K}\times\mathbb{K}\to\mathbb{R}$, $\Delta(z',z)$ loss of pred z' if z correct

$\min_{w,\xi\geq 0}\frac{1}{2}w^T w + C\sum_{i=1}^n\xi_i$ s.t.

$w^T\Psi(z_i,y_i) - \max_{z\neq z_i}[\Delta(z,z_i) + w^T\Psi(z,y_i)]\geq -\xi_i$ $\forall i$

Algorithm Training SSVMs with margin-rescaling
Require: training data $((z_1,\mathbf{y}_1),\dots,(z_n,\mathbf{y}_n))$, tradeoff parameter C, precision ϵ
repeat
  for $i=1,\dots,n$ **do**
    $\tilde z \leftarrow \arg\max_{z'\in\mathbb{K}} (\Delta(z',z_i)+\mathbf{w}^\top\Psi(z',\mathbf{y}_i))$ // Loss augmented inference
    if $\mathbf{w}^\top[\Psi(z_i,\mathbf{y}_i)-\Psi(\tilde z,\mathbf{y}_i)]<\Delta(\tilde z,z_i)-\xi_i-\epsilon$ **then**
      $\mathcal{W}\leftarrow\mathcal{W}\cup\{\mathbf{w}^\top[\Psi(z_i,\mathbf{y}_i)-\Psi(\tilde z,\mathbf{y}_i)]\ge\Delta(\tilde z,z_i)-\xi_i\}$ // Add constraint to constraint set $\mathcal{W}$
      $(\mathbf{w},\xi)\leftarrow\arg\min_{\mathbf{w}',\xi'\ge0}\frac12\mathbf{w}'^\top\mathbf{w}'+C\sum_{i=1}^n\xi_i'$ s.t. $\mathcal{W}$
    end if
  end for
until $\mathcal{W}$ has not changed during iteration
return w // Return weights

Applying SSVMs to new tasks:
Need to design/implement: 1) Feature Map Ψ, 2) Loss Fct Δ
3) Loss Augmented Inference: $\tilde z=\arg\max_{z'\in K}(\Delta(z',z_i)+w^T\Psi(z',y_i))$
4) Prediction Rule: $\hat z=\arg\max_{z\in K} w^T\Psi(z,y)$

8 Ensemble Methods

Given simple learners $f_1,\dots,f_B$, consider: $f(x)=\sum_{i=1}^B f_i(x)$
Bias: $bias[\hat f]=E_D[\hat f(x)]-E[Y|X]=\frac1B\sum_{i=1}^B bias[f_i]$
↳ unbiased estim remains unbiased
Variance: $Var[\hat f]=E_D[(\hat f(x)-E_D[\hat f])^2]=E[(\frac1B\sum_{i=1}^B(\hat f_i-E_D[f_i]))^2]$
$=\frac1{B^2}\sum_{i=1}^B Var_D[\hat f_i(x)]+\frac1{B^2}\sum_{i=1}^B\sum_{j=1}^B Cov[\hat f_i(x),\hat f_j(x)]=*$
↳ If Cov vanishes/very small & $Var[f_i]=\sigma^2$
$\Rightarrow Var[\hat f(x)]\approx\frac{\sigma^2}{B}\Rightarrow$ Var reduced if $Cov\approx0$
$\Rightarrow$ more stable & generalizable models, that are less prone to overfitting
Exmpl: $c_1,\dots,c_B$ simple classifiers $\Rightarrow \hat c(x)=sgn(\sum_{i=1}^B\alpha_i c_i(x))$ (weights), $c\in\{\pm1\}$
Rem: to minimize, train c_i on subsets of $\mathcal{D}$, features, ...

8.1 Bagging: Var. red. technique
Bootstrapping reduces cov.

Algorithm 7 Bagging Classifiers
Require: $\mathcal{Z}=\{(x_1,y_1),\dots,(x_n,y_n)\}$
Ensure: $\hat c_{BAG}$ with majority vote of $\{c_1,c_2,\dots,c_B\}$
1: **for** $b=1$ to B **do**
2:   $\mathcal{Z}^{*b}\leftarrow b$-th bootstrap sample from $\mathcal{Z}$
3:   Construct classifier c_b based on $\mathcal{Z}^{*b}$
4: **end for**
5: **return** $\hat c_{BAG}(x)=\text{sgn}(\sum_{i=1}^B c_i(x))$ =0

Comparing 2 Classifiers:
Algorithm 8 First Compare, Then Bag
Require: $\mathcal{Z}=(x_1,y_1),\dots,(x_n,y_n)$
Ensure: Classifier $\hat c'_{BAG}(x)$ or $\hat c''_{BAG}(x)$
1: **for** $b=1$ to B **do**
2:   $\mathcal{Z}^{*b}\leftarrow b$-th bootstrap sample from $\mathcal{Z}$
3:   Construct classifiers c'_b, c''_b based on $\mathcal{Z}^{*b}$
4:   Calculate error difference $\Delta_b:=\hat R_n(c''_b)-\hat R_n(c'_b)$
5: **end for**
6: **if** median$(\Delta_1,\dots,\Delta_B)>0$ **then**
7:   **return** Ensemble classifier $\hat c'_{BAG}(x)=\text{sgn}(\sum_{i=1}^B c'_i(x))$
8: **else**
9:   **return** Ensemble classifier $\hat c''_{BAG}(x)=\text{sgn}(\sum_{i=1}^B c''_i(x))$

*: $Cov(X_i,X_j)=\rho(X_i,X_j)\cdot\sqrt{Var(X_i)Var(X_j)}$
↳ $=\frac1B\sigma^2+\frac{\sigma^2}{B^2}\sum_{i\ne j}\rho(X_i,X_j)$
$=\frac{\sigma^2}{B}+\sigma^2\rho\cdot B(B-1)/B^2$
$=\rho\sigma^2+\frac{\sigma^2}{B}(1-\rho)$

8.2 Random Forests
Decision Tree: Recursively split $\mathcal{D}$ into subsets based on feature values that best separate the target classes. At each node, Algorithm chooses split that maximizes some criteria, e.g. information gain. Stop when e.g. max depth or min nodesize reached.
Random Forests: Combination of multiple dec. trees with randomness: 1) Each tree trained on bootstrap sample, 2) At each split within a tree, Algo only considers random subset of features. ($m\approx\sqrt p$ features typically)

Algorithm 9 Random Forest Algorithm
Require: $\mathcal{Z}=\{(x_1,y_1),\dots,(x_n,y_n)\}$, $n_{\min}$
Ensure: Ensemble of trees $\{c_b(x)\}_{b=1}^B$
1: **for** $b=1$ to B **do**
2:   Draw bootstrap sample $\mathcal{Z}^\star=\{(x_1^\star,y_1^\star),\dots,(x_n^\star,y_n^\star)\}$ of size n from $\mathcal{Z}$
3:   **repeat** ▷ *generation of a random forest tree*
4:     Select m features at random from p features $m\approx\sqrt p$
5:     Pick the best feature/split-point among the m selected features
6:     Split the node into two daughter nodes
7:   **until** node size $n_{\min}$ is reached
8: **end for**
9: **return** Ensemble of trees $\{c_b(x)\}_{b=1}^B$

8.3 Boosting: Idea: train consecutive classifiers & upweigh samples that have been misclassified. ⇒ Bias reduction technique

Algorithm 10 AdaBoost.M1
Require: $\mathcal{Z}=(x_1,y_1),\dots,(x_n,y_n)$
Ensure: Classifier $\hat c_{ADA}(x)$
1: **initialize** observation weights $w_i=1/n$
2: **for** $b=1$ to B **do**
3:   Fit a classifier $c_b(x)$ to the training data using weights w_i
4:   Compute $\epsilon_b\leftarrow\frac{\sum_{i=1}^n w_i^{(b)}\mathbb{I}\{c_b(x_i)\ne y_i\}}{\sum_{i=1}^n w_i^{(b)}}$ ▷ *error of* c_b
5:   Compute $\alpha_b\leftarrow\log\frac{1-\epsilon_b}{\epsilon_b}$ ▷ *final weight of* c_b
6:   Set $\forall i: w_i\leftarrow w_i\exp(\alpha_b\mathbb{I}\{c_b(x_i)\ne y_i\})$
7: **end for**
8: **return** $\hat c_{ADA}(x)=\text{sgn}(\sum_{b=1}^B\alpha_i c_i(x))$

Theoretical Results on AdaBoost: exp. loss ✓
AdaBoost equiv to stagewise logistic modeling using
In LogReg: $\log(P(y=1|x)/P(y=-1|x))=\sum_{b=1}^B c_b(x)=:F(x)$
$\Rightarrow P(y=1|x)=\frac{\exp(F(x))}{1+\exp(F(x))}=1-P(y=-1|x)$
⇒ AdaBoost behaves as an additive LogReg model
⇒ AdaBoost Objective: minimize $J(F)$ wrt $F(x)$: $J(F)=E[\exp(-yF(x))]$
Lemma: $E[e^{-yF(x)}]$ is minimized at $F(x)=\frac12\log(P(y=1|x)/P(y=-1|x))$
$\Rightarrow P(y=1|x)=e^{F(x)}/(e^{-F(x)}+e^{F(x)})=e^{2F(x)}/(1+e^{2F(x)})=1-P(y=-1|x)$
Thm: Discrete AdaBoost is Grad Desc. minimizing $E[e^{-yF(x)}]$
Proof: $J(F+\alpha c)=E[e^{-y(F+\alpha c)}]\approx E[e^{-yF}(1-y\alpha c(x)+\frac12\alpha^2 y^2c^2(x))]+O(\alpha^3)$ ($y^2c^2=1$)
↳ Pointwise minimization: Def: $E_w[g(x,y)|x]=\frac{E[w(x,y)g(x,y)|x]}{E[w(x,y)|x]}$
where $w(x,y)=e^{-yF(x)}$
⇒ optim classifier $c^*(x)=\arg\min_c J=\arg\max_c E_w[yc(x)|x]$
$\Rightarrow c(x)=\begin{cases}1 & \text{if } E_w[y|x]=P_w(y=1|x)-P_w(y=-1|x)>0\\-1 & \text{else}\end{cases}$

Rem: $P_w(y=1|x)=(P(y=1|x)\exp(-F(x)))/(P(y=1|x)e^{-F(x)}+P(y=-1|x)e^{F(x)})$
Rem: $E_w[(y-c(x))^2]/2-1=-E_w[y\cdot c(x)]$
$\Rightarrow\max_c E[yc(x)|x]=\min_c E[(y-c(x))^2]$
⇒ Newton-like step
Rem: $\alpha^*=\arg\min_\alpha E_w[\exp(-\alpha yc(x))]=\frac12\log(\frac{1-err}{err})$
$=\arg\min_\alpha\{e^{\alpha}E_w[\mathbb{1}\{y\ne c(x)\}]+e^{-\alpha}E_w[\mathbb{1}\{y=c(x)\}]\}$ (err, 1-err)
Result:
$F(x)\leftarrow F(x)+\alpha^*c(x)=F(x)+\frac12\log(\frac{1-err}{err})c(x)$
$w(x,y)\leftarrow w(x,y)\exp(-\alpha^*yc(x))=w(x,y)\exp(\log(\frac{1-err}{err})\mathbb{1}\{y\ne c(x)\})e^{-\alpha^*}$

9 Deep Learning:

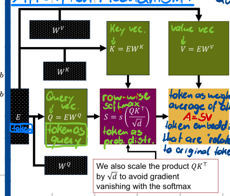
1) Fw Diff: $x(t+\Delta t)=x(t)+\sigma(t)\sqrt{\Delta t}\cdot\epsilon$
Rev Diff: $x(t+\Delta t)=x(t)+\sigma^2(T-t)s(x,t)\Delta t+\sigma(T-t)\sqrt{\Delta t}\cdot\epsilon$
Score fct $s(x,t)=\frac{d}{dx}\log(p(x,t))$ ⇒ Learn score fct

9.1 VAEs
Req for good Representations: 1) Informative: Given repr., should be easy to reconstruct input, 2) Disentangled: Every component in the representation is associated with distinct feature 3) Robust: Noisy Perturbations shouldn't affect representation
Infomax Principle: (Solves informativeness)
Let $\mathcal{H}=\{enc_\theta:\mathcal{X}\to\mathcal{Z}|\theta\in\Theta\}$ space of encoders
Infomax principle: $\hat\theta=\arg\max_\theta I(X;enc_\theta(X))$ for $X\in\mathcal{X}$
Approx: $\hat\theta=\arg\max_\theta\sum_{i=1}^n E_{z|x_i}[\log(p(x_i|z))]$
↳ Problem: not necessarily disentangled & robust!
VAEs:
Idea: Decoder: $z\xrightarrow{p(x|z)}X$ & Encoder: $z\xleftarrow[q_\phi(z|x)]{p(z|x)}X$ (prior $p(z)$, Likelihood $p(x|z)$, intractable posterior, approx posterior)
Scatch: prior: $\mathcal{N}(0,\sigma^2\mathbb{1})$
Decoder: $z\mapsto(\mu_z,\Sigma_z)\Rightarrow$ Obj. Space: $\mathcal{N}(\mu_z,\Sigma_z)$ (Likelihood)
Encoder: $x\mapsto(\mu_x,\sigma_x^2)\Rightarrow$ Repr. Space: $\mathcal{N}(\mu_x,\sigma_x^2\mathbb{1})$ $(q_\phi(z|x))$
Training: $\hat\theta,\hat\theta',\phi=\arg\max_{\theta,\theta',\phi}\sum_{i=1}^n\log(p_{\theta,\theta'}(x_i))$
$\log(p_{\theta,\theta'}(x_i))=E_{z\sim q_\phi}[\log(\frac{p_{\theta,\theta'}(x_i,z)}{q_\phi(z|x_i)})]+E[\log(\frac{q_\phi(z|x_i)}{p_{\theta,\theta'}(z|x_i)})]$
(ELBO$_{\theta,\theta',\phi}(x_i)$; $KL(q_\phi(.|x_i)\|p_{\theta,\theta'}(.|x_i))\ge0$)
$\Rightarrow\log(p_{\theta,\theta'}(x_i))\ge$ ELBO$_{\theta,\theta',\phi}(x_i)$
ELBO$_{\theta,\theta',\phi}(x_i)=E[\log(p_\theta(x_i|z))]+E[\log(\frac{p_{\theta'}(z)}{q_\phi(z|x_i)})]$
(Infomax (Cross-Entropy); $-KL(q_\phi(.|x_i)\|p_{\theta'}(.))$ "penalty term" ↳ regul.)
⇒ KL Regularization ensures disentanglement & robustness
Rem: Goal of VAE: $p(x)=\int p(z)p_\theta(x|z)dz\approx\int q_\phi(z|x)p_\theta(x|z)dz$ (decoder, encoder)

9.2 Diffusion Eq
Setup: (i-2, i-1, i, i+1, i+2; p, q) concentration c
$c_i(t+1)=qc_{i-1}(t)+pc_{i+1}(t)+(1-p-q)c_i(t)$
$c_i(t+1)-c_i(t)=\frac{p-q}{2}(c_{i+1}(t)-c_{i-1}(t))+\frac{p+q}{2}(c_{i+1}(t)-2c_i(t)+c_{i-1}(t))$
Master eq: $q'=q\frac{\Delta t}{\tau}$, $p'=p\frac{\Delta t}{\tau}\Rightarrow\tau\frac{d}{dt}c_i=\frac{p-q}{2}(c_{i+1}-c_{i-1})+\frac{q+p}{2}(c_{i+1}-2c_i+c_{i-1})$
τ characteristic time scale, δ = char. length scale
Diffusion without drift: $q'=q\cdot(\frac{\delta}{\Delta x})^2$, $p'=p(\frac{\delta}{\Delta x})^2$, $p=q=\tilde D$
↳ $\frac{\partial}{\partial t}c(x,t)=D\frac{\partial^2}{\partial x^2}c(x,t)$, $D=\delta^2/\tau\,\tilde D$
Solution: $c(x,t)=\frac{1}{\sqrt{4\pi Dt}}\exp(-\frac{(x-x_0)^2}{4Dt})$

2) Learning score fct: Objective: $J=E_{t\in(0,T),x_0\sim p_0(.)}[\|s(x_{noised},t)\sigma^2(t)+(x_{noised}-x_0)\|_2^2]$
3) $J=E_{t\in(0,T),\epsilon\sim\mathcal{N}}[\|s(x_0+\sigma(t)\epsilon,t)\sigma(t)+\epsilon\|_2^2]$

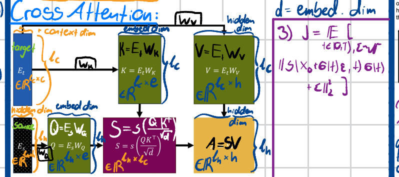
9.3 Transformers:
Tokenization: Wordpiece: 1) Identify Substring of fixed length that, when removed, shortens the sentence the most. 2) Iterate step 1. This leads to decomp. of text into S tokens. (S fixed)
Embedding: every token is associated with input embedding vector in $\mathbb{R}^d$. (d fixed)
⇒ Embedding matrix $E\in\mathbb{R}^{S\times d}$ that can be learned.
Self Attention: represent relations in a sentence with binary matrices. Matrices are Attention-Maps. Attention Matrices should not be symmetric.
Attention Mechanism: Query (Q): What you want to look up.
Keys (K): Keyword of values
Values (V): Values that you want to look up.
↳ Goal: learn the mapping from keys to values.



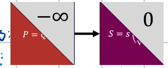
E, W^K, W^V, W^Q learnable

Multi-Headed Attention Mechanism: Calculate several relationship maps in parallel & aggregate in the end:
$[A_1=S_1V_1\ \ A_2=S_2V_2\ \dots]\times[W^A]=A$, W^A learnable. d = embed. dim

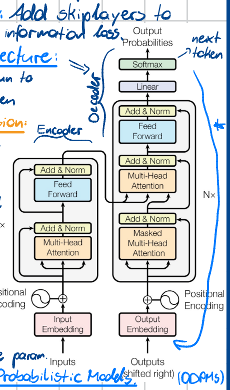
Cross Attention:



Masked Self-Attention:

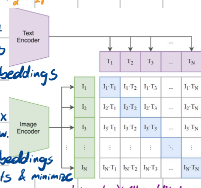


Adjust S like this:
Positional Encoding:
Goal: Create embedding that encodes the position of the position of the word in the sentence.
Binary Encoding: $BP_{ij}=(i\div2^j)\bmod2=\lfloor2^{-j}i\rfloor\bmod2$
↳ Gives position of word in binary encoding.
Continuous Encoding: $P_{i,2j}=\sin(\alpha^{-2j}\cdot i)$, $P_{i,2j+1}=\cos(\alpha^{2j}\cdot i)$
typically $\alpha=10^{4/d}$ where d = dim of positional embedding.
Implementation: Add this pos. enc. mat. $PE\in\mathbb{R}^{T\times d}$ (T = max nbr of tokens in one sentence, d = dim of Pos. Emb) to the word embeddings (instead of concat.)
Reasons: • Concat increases #params too much
• pos. emb. "statistically indep" from token embed. ⇒ both embeddings are "orthogonal" ⇒ Information is preserved
• Most operations linear ⇒ final embed. "addition" of result of the operations applied to both pos. & token embed.
Residual Blocks: Add skiplayers to address problem of information loss.
Transformer Architecture:
* Add to output & rerun to produce the next token

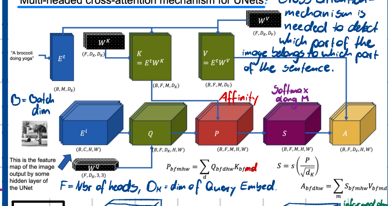


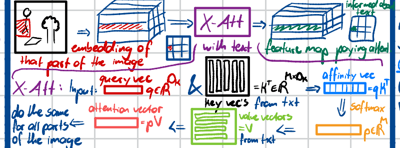
9.4 Stable Diffusion:
U-Net: NN Architecture that first compresses input image with convolutional & maxpool layers. Then, with up-conv. layers & skip-layers (copy & crop) try to reconstruct image or learn segmentation. Conv. filters are learnable param.
Denoising Diffusion Probabilistic Models (DDPMs)
Forward Path / Destruction: 1) In each step, add a little bit of gaussian noise to the image, until you only have noise. 2) $\forall t\le T$, chose $\beta_t>0$, let $\alpha_t=1-\beta_t$ & $\bar\alpha_t=\prod_{s\le t}\alpha_s$. Then:
$q(x_t|x_{t-1})=\mathcal{N}(x_t|\sqrt{1-\beta_t}x_t,\beta_t\mathbb{1})\Rightarrow q(x_t|x_0)=\mathcal{N}(x_t|\sqrt{\bar\alpha_t}x_t,(1-\bar\alpha_t)\mathbb{1})$
Backward Path/Construction/Generative Process:
$\forall t\le T$, the UNet tries to reconstruct all noise $\epsilon_\theta(x,t)$ that was added to x_t from x_0. Can reconstruct x_{t-1} as follows:
$x_{t-1}=\frac{1}{\sqrt{\alpha_t}}(x_t-\frac{1-\alpha_t}{\sqrt{1-\bar\alpha_t}}\epsilon_\theta(x_t,t))+\sigma_t z$, σ_t hyperparam & $z\sim\mathcal{N}(0,\mathbb{1})$
Rem: See tutorials for more detail
Conditional Diffusion Processes:
At each denoising step include the embedding of the word/sentence/number you are trying to generate.

Stable Diffusion:
CLIP: 1) Select an Image & text encoder, 2) Encode all images & texts, 3) Transform all embeddings to the same euclidian space
4) Compute a similarity matrix containing all similarities btw. each pair of text-image embeddings
5) Maximize diagonal elements & minimize non diagonal elements.



Integrate X-Att in UNet
Multi-headed cross-attention mechanism for UNets: Cross attention-mechanism is needed to detect which part of the image belongs to which part of the sentence.





Nachtrag Counterfactual Invariance:
$X_1,\dots,X_m\to Z(X_1),\dots,Z(X_m)$, $Y_1,\dots,Y_n\to Z(Y_1),\dots,Z(Y_n)$ } want these to look alike
↳ Needed metric $M(\tilde X,\tilde Y)\ge0$
$M(\tilde X,\tilde Y)=0$ iff $\tilde X,\tilde Y$ come from same distribution
Add M as penalty term in training of network
Setup: $X_1,\dots,X_m\sim p^*$, $Y_1,\dots,Y_n\sim q^*$ unknown ⇒ $p=q$?
↳ $p\ne q\Leftrightarrow\exists A\in\Sigma: p(A)\ne q(A)\Leftrightarrow\exists A\in\Sigma: E_{x\sim p}[\mathbb{1}_A(x)]\ne E_{x\sim q}[\mathbb{1}_A(x)]$
$\Leftrightarrow\exists f\in\mathcal{C}(\mathcal{X}): E_{x\sim p}[f(x)]\ne E_{x\sim q}[f(x)]$
Maximum Mean Discrepancy: $MMD[\mathcal{F},p,q]=\sup_{f\in\mathcal{F}}|E_{x\sim p}[f(x)]-E_{y\sim q}[f(y)]|$
Restrict ourselves to unit sphere in RKHS
Riesz Theorem: Table
↳ Let $\mathcal{H}=\{Polys\}$, $\tau\in\mathcal{H}$, $x\in\mathcal{X}$

Poly	Op	$\mathbb{R}$	$\mathbb{R}^\infty$
τ	eval@x	$\tau(x)$	$\beta_\tau^T\varphi(x)$
τ	$E_{x\sim p}$	$E_{x\sim p}[\tau(x)]$	$\beta_\tau^T\mu_p$
τ_x	eval@y	$\tau_x(y)$	$\varphi^T(x)\varphi(y)=K(x,y)$

$\Rightarrow MMD^2[\mathcal{F},p,q]$ ($\mathcal{F}$ = unit sphere) $=\sup_{f\in\mathcal{F}}(\beta_f^T\mu_p-\beta_f^T\mu_q)^2=\sup_{\beta:\|\beta\|=1}(\beta^T(\mu_p-\mu_q))^2=\|\mu_p-\mu_q\|^2$
$\Rightarrow MMD^2=\mu_p^T\mu_p-2\mu_p^T\mu_q+\mu_q^T\mu_q$
$\dots=E_{x,x'\sim p}[k(x,x')]-2E_{x\sim p,y\sim q}[k(x,y)]+E_{y,y'\sim q}[k(y,y')]$

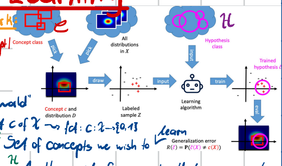
10. Clustering
Obj. fct: $\arg\min_{A,\mu}\sum_{k=1}^K\sum_{x\in A_k}\|x-\mu_k\|^2$
10.1 K-means:
given $\mathcal{X}=\{x_1,\dots,x_n\}$, $x_i\in\mathbb{R}^d$ $=\arg\min_c\sum_{x\in\mathcal{X}}\|x-\mu_{c(x)}\|_2^2$
Clustercenters: $\vec\mu_j\in\mathbb{R}^d$, $j=1,\dots,K$, Assignments: $z_i=\arg\min_j\|\vec x_i-\vec\mu_j\|_2$
Lloyd's Heuristic: Iteratively update z_i & $\mu_j=\frac{1}{n_j}\sum_{i:z_i=j}\vec x_i$, $n_j=\#\{z_i=j\}$
Def: $\vec z^{(t)}=(z_1^{(t)},\dots,z_n^{(t)})$, $\vec\mu^{(t)}=(\vec\mu_1^{(t)},\dots,\vec\mu_k^{(t)})\Rightarrow$ 1) Init $\mu^{(0)}$,
2) Repeat: $z_i^{(t)}=\arg\min_j\|\vec x_i-\vec\mu_j^{(t-1)}\|_2$ & $\mu_j^{(t)}=\frac{1}{n_j^{(t)}}\sum_{i:z_i=j}\vec x_i$ $\forall j=1,\dots,K$
10.2 GMMs:
$p(x|\vec\pi,\vec\mu,\vec\Sigma)=\sum_{i=1}^K\pi_i\mathcal{N}(x|\mu_i,\Sigma_i)$
$\Rightarrow p(\mathcal{X}|\pi,\theta)=\prod_{x\in\mathcal{X}}\sum_{i=1}^K\pi_i p(x|\theta_i)\Rightarrow\mathcal{L}=\sum_{x\in\mathcal{X}}\log(\sum_{i=1}^K\pi_i p(x|\theta_i))$
EM Algorithm: Repeat until convergence: 1) Calculate $Q(\theta;\theta^{(j)})=E_{\mathcal{X}_L}[\mathcal{L}(\mathcal{X},\mathcal{X}_L|\theta)|\mathcal{X},\theta^{(j)}]$, 2) $\theta^{(j+1)}\in\arg\max_\theta Q(\theta;\theta^j)$
Soft EM: E-Step: $\gamma_j^{(t)}(\vec x_i)=P(x_i|z_j,\theta^j)P(z_j|\theta^j)/P(x_i|\theta^j)$
$\gamma_j^{(t)}(\vec x_i)=\frac{\pi_j^{(t-1)}P(\vec x|\Sigma_j^{(t-1)},\mu_j^{(t-1)})}{\sum_{c=1}^K\pi_c^{(t-1)}P(\vec x|\Sigma_c^{(t-1)},\mu_c^{(t-1)})}$
M-Step: $\pi_j^{(t)}=\frac1n\sum_{i=1}^n\gamma_j^{(t)}(\vec x_i)$, $\mu_j^{(t)}=\sum_i\gamma_j(\vec x_i)\vec x_i/\sum_i\gamma_j(\vec x_i)$
$\Sigma_j^{(t)}=\sum_{i=1}^n\gamma_j^{(t)}(\vec x_i)(\vec x_i-\mu_j^{(t)})(\vec x_i-\mu_j^{(t)})^T/\sum_{i=1}^n\gamma_j^{(t)}(\vec x_i)$
Hard EM: E-Step: $z_i^{(t)}=\arg\max_z P(z|\vec x_i,\theta^{(t-1)})$ ⇒ Bayes Thm
M-Step: $\theta^{(t)}=\arg\max_\theta P(\mathcal{D}^{(t)}|\theta)\Rightarrow\gamma_j(\vec x_i)=\mathbb{1}\{z(\vec x_i)=j\}$, & Π_j,μ_j,Σ_j

11. Nonparametric Bayesian Methods
Beta Distr.: $Beta(x|a,b)=\frac{1}{B(a,b)}\cdot x^{a-1}(1-x)^{b-1}$, $x\in[0,1]$ prob of observing a success. $B(a,b)=\Gamma(a)\Gamma(b)/\Gamma(a+b)$, $\Gamma(a)=\int_0^\infty e^{-x}x^{a-1}dx$
Dirichlet Distr.: Multivariate generalization of Beta distr.
$x=(x_1,\dots,x_n)$, $\alpha=(\alpha_1,\dots,\alpha_n)$, $x_i\in[0,1]$: $Dir(x|\alpha)=\frac{1}{B(\alpha)}\cdot\prod_{k=1}^n x_k^{\alpha_k-1}$,
$B(\alpha)=(\prod_{k=1}^n\Gamma(\alpha_k))/\Gamma(\sum_{k=1}^n\alpha_k)$
Choice of K: K = nbr of clusters. Selection of K somewhat arbitrary. For finite nbr of draws (Datapoints) N, we do not have to realize all K clusters ⇒ ∃ latent Clusters = Clusters from which we do not have any Samples.
Selecting K = ∞: "Nonparametric" Bayesian Method. "NP": we have ∞ many params & we can keep drawing new params here: params = cluster probabilities: $p_{1..k}\sim Dir(\alpha_{1..k})$
Dirichlet Process: A $DP(\alpha,H)$ is a distr. on a space Θ. Here: $\alpha\in\mathbb{R}_{>0}$ concentration param, H base measure on Θ. Hence, a sample $G\sim DP(\alpha,H)$ is a fct $G:\Theta\to\mathbb{R}_{\ge0}$ s.t. $\int_\Theta G(\theta)d\theta=1\Rightarrow G(\theta)$ is a pdf
$DP(\alpha,H)$ is characterized by: ∀ partition $(T_1,\dots,T_K)$ of Θ & $G\sim DP(\alpha,H)$: $(G(T_1),\dots,G(T_K))\sim Dir(\alpha H(T_1),\dots,\alpha H(T_K))$
Stick breaking Process: Sampling $(p_1,\dots,p_K)\sim Dir(\alpha_1,\dots,\alpha_K)$ is equivalent to Sampling $p_1\sim Beta(\alpha_1,\dots,\alpha_K)$ & $(p_2,\dots,p_K)\sim Dir(\alpha_2,\dots,\alpha_K)$
$\Rightarrow\beta_1\sim Beta(\alpha_1,\dots,\alpha_K)\Rightarrow p_1=\beta_1$, $\beta_2\sim Beta(\alpha_2,\dots,\alpha_K)\Rightarrow p_2=\beta_2(1-\beta_1)$
$\beta_3\sim Beta(\alpha_3,\dots,\alpha_K)\Rightarrow p_3=\beta_3(1-\beta_1)(1-\beta_2)$, ..., $p_K=1-\sum_{i=1}^{K-1}p_i$
⇒ Only works for finite K ↯ ⇒ Solution: fix α s.t. $\beta_i\sim Beta(1,\alpha)$
$\Rightarrow\beta_1\sim Beta(1,\alpha)\Rightarrow p_1=\beta_1$, $\beta_2\sim Beta(1,\alpha)\Rightarrow p_2=\beta_2(1-\beta_1)$, ...
$\beta_K\sim Beta(1,\alpha)\Rightarrow p_K=\beta_K(1-\sum_{i=1}^{K-1}p_i)$
This is called the GEM Distribution: $p\sim GEM(\alpha)$, $p=\{p_k\}_{k=1}^\infty$
Connection to DP: If $p\sim GEM(\alpha)$ & $\theta_k\sim H$ then: $G(\theta)=\sum_{k=1}^\infty p_k\delta_{\theta_k}(\theta)$ is a sample from $DP(\alpha,H)$
⇒ Samples from DP are almost surely discrete measures on Θ, supported on a countable set $\{\theta\}_{k=1}^\infty\subseteq\Theta$
Sample $\theta^{(1)},\theta^{(2)},\dots$ from G, $G\sim DP(\alpha,H)\Rightarrow\theta^{(i)}=\theta_{k_i}$
Chinese Restaurant process: Technique to draw from DP
Customers: Observations $\theta^{(i)}$, tables: Clusters θ_k
When a new customer arrives, join existing table with prob ∝ #people sitting there, start new table: $p\propto\alpha$

$P(\text{customer } n+1 \text{ joins table }\tau|\mathcal{P})=\begin{cases}\frac{|\tau|}{\alpha+n} & \text{if }\tau\in\mathcal{P}\\\frac{\alpha}{\alpha+n} & \text{otherwise}\end{cases}$ ($\mathcal{P}$ = Partition)
↳ $P(\mathcal{P})=\frac{\alpha^{|\mathcal{P}|}}{\alpha^{(n)}}\prod_{\tau\in\mathcal{P}}(|\tau|-1)!$ ⇒ Order & Label invariant
Rem: $E[\#tables]\sim O(\alpha\cdot\log(N))$ ⇒ Proof: $S(n)=\sum_k P(X_k\sim F_0)=\sum\frac{\alpha}{\alpha+k-1}$, $I(n)=\int_{x=1}^n\frac{\alpha}{\alpha+x-1}dx$, $I(n)\le S(n)\le I(n)+1$, $S(n)=E[\sum\mathbb{1}\{X_k\sim F_0\}]$
Exchangeability: A seq of RV is exchangeable if ∀ permutations $\pi:\mathbb{N}\to\mathbb{N}$ the random vecs. have the same distr.: $(X_1,X_2,\dots)\overset{D}{=}(X_{\pi(1)},X_{\pi(2)},\dots)$
De Finetti's Thm: Let $(X_1,X_2,\dots)$ be an ∞ exchangeable seq of RVs. Then: $\forall n: p(X_1,\dots,X_n)=\int(\prod_{i=1}^n p(x_i|G))dP(G)$ for some RV G.
Polya Urn: Setup: Urn with colored balls. Procedure: Pick a ball at random, put it back in & add a ball of the same color to urn
Hoppe Urn: Extend Polya Urn with a black ball, that if picked, adds ball with new color
DP Mixture Model: Let: Θ set that parametrizes a set of prob distr., H: measure on Θ. Example:
• $\Theta=\mathbb{R}$, $\mu\in\Theta$ corresponding to $\mathcal{N}(\mu,\sigma^2)$ for some fixed $\sigma>0$
• $H=\mathcal{N}(\mu_0,\sigma_0^2)$ for some fixed $\mu_0\in\mathbb{R}$ & $\sigma_0\in\mathbb{R}_{>0}$
⇒ prob. of Clusters ("mixture weights"): $p=(p_1,p_2,\dots)\sim GEM(\alpha)$
Centers: $\mu_k\sim\mathcal{N}(\mu_0,\sigma_0^2)$, Assignments: $z_i\sim Categorial(p)$
Coord. of Datapoints: $x_i\sim\mathcal{N}(\mu_{z_i},\sigma^2)$
Fitting: exchangeability ⇒ Any point considerable as "last arrived". ⇒ Gibbs sampling
$p(z_i=k|\underline z_{-i},\underline x,\alpha,\mu)\propto p(\underline x|z_i=k,\underline z_{-i},\mu)p(z_i=k|\underline z_{-i},\alpha)$
$\propto p(z_i=k|\underline z_{-i},\alpha)\cdot p(x_i|\underline x_{-i},z_i=k,\underline z_{-i},\mu)$ (prior; Likelihood)
z_i exchangeable & $x_i\perp\!\!\!\perp\underline z_i|\underline z_{-i}$ by d-sep
Prior: $p(z_i=k|\underline z_{-i},\alpha)=\begin{cases}\frac{N_{k,-i}}{\alpha+N-1} & \text{for existing }k\\\frac{\alpha}{\alpha+N-1} & \text{otherwise}\end{cases}$ $N_{k,-i}$ = nbr of elements @table k, excluding i
$\underline x_{-i,c}=\{x_j:z_j=c,j\ne i\}$
$\mathcal{L}$: $p(x_i|\underline x_{-i},z_i=k,\underline z_{-i},\mu)=\begin{cases}p(x_i|\underline x_{-i,k},\mu)=\frac{p(x_i,\underline x_{-i,k}|\mu)}{p(\underline x_{-i}|\mu)} & \text{for ex. }k\\p(x_i|\mu) & \text{otherwise}\end{cases}$
$\Rightarrow p(z_i=k|\underline z_{-i},\underline x,\alpha,\mu)\propto$ prior × $\mathcal{L}$
$\propto\begin{cases}\frac{N_{k,-i}}{\alpha+N-1}p(x_i|\underline x_{-i,k},\mu) & \text{for existing }k\\\frac{\alpha}{\alpha+N-1}p(x_i|\mu) & \text{otherwise}\end{cases}$

for $i=1$ to N in random order **do**
  Remove x_i's sufficient statistics from old cluster z_i;
  for $k=1$ to K **do**
    Compute $p_k(x_i)=p(x_i|\mathbf{x}_{-i,k},\boldsymbol\mu)$;
    Set $N_{k,-i}=|\mathbf{x}_{-i,k}|$;
    Compute $p(z_i=k|z_{-i},\mathbf{x},\boldsymbol\mu)=\frac{N_{k,-i}}{\alpha+N-1}p_k(x_i)$;
  end for
  Compute $p_*(x_i)=p(x_i|\boldsymbol\mu)$;
  Compute $p(z_i=\star|z_{-i},\mathbf{x},\boldsymbol\mu)$;
  Normalize $p(z_i|\cdot)$;
  Sample $z_i\sim p(z_i|\cdot)$;
  Add x_i's sufficient statistics to new cluster z_i;
  If any cluster is empty, remove it and decrease K;
end for

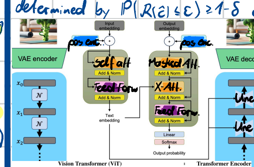
12. PAC Learning
12.1 Framework:
|Hyp Class| ≤ |Concept| (usually)



Defs:
· Instance Space $\mathcal{X}$ ↳ obj. in "learners world"
· Concept: subset c of $\mathcal{X}$ ⇝ fct: $c:\mathcal{X}\to\{0,1\}$
· Concept Class $\mathcal{C}$: Set of concepts we wish to learn
· Hypothesis Class $\mathcal{H}$: Another set of concepts that we use to learn a target concept from the concept class.
· Learning Algo: Let $\mathcal{H}$ be a concept class & c a concept. A learning Algo. receives as inputs labeled samples $\mathcal{Z}=\{(x_i,y_i)\}_{i=1}^n$ $\forall i, y_i=c(x_i)$ & outputs Hypothesis $\hat c\in\mathcal{H}$.
Def PAC: A learning Algo. $\mathcal{A}$ can learn a concept c if ∃ polynomial fct poly(...) s.t. 1) ∀ distr. $\mathcal{D}$ on $\mathcal{X}$ & 2) $\forall0<\epsilon<\frac12$ & $0<\delta<\frac12$ ⇒ if $\mathcal{A}$ receives as input a sample $\mathcal{Z}$ of size $n\ge poly(\frac1\epsilon,\frac1\delta,size(c))$, then $\mathcal{A}$ outputs $\hat c$ s.t.: $P_{\mathcal{Z}\sim\mathcal{D}^n}(R(\hat c)\le\epsilon)\ge1-\delta$ ↳ size(c) = size of represent. of concept c
A concept class $\mathcal{C}$ is PAC Learnable from a hypothesis class $\mathcal{H}$ if ∃ algo $\mathcal{A}$ that can learn any concept in $\mathcal{C}$.
If $\mathcal{A}$ runs in time polynomial in $\frac1\epsilon$ & $\frac1\delta$ we say that $\mathcal{C}$ is efficiently PAC learnable.
12.2 General PAC Learning:
Def: A learning algorithm $\mathcal{A}$ can learn a concept class C from $\mathcal{H}$ if ∃ poly(...) s.t. 1) ∀ distr $\mathcal{D}$ on $\mathcal{X}\times[0,1]$ & 2) $\forall0<\epsilon<\frac12$ & $0<\delta<\frac12$ ⇒ if $\mathcal{A}$ receives as input a sample $\mathcal{Z}$ of size $n\ge poly(\frac1\epsilon,\frac1\delta,size(c))$, then $\mathcal{A}$ outputs $\hat c\in\mathcal{H}$ s.t. $P_{\mathcal{Z}\sim\mathcal{D}^n}(R(\hat c)-\inf_{c\in\mathcal{C}}R(c)\le\epsilon)\ge1-\delta$
Error Bounds:
Let $\epsilon>0$. For a sample $\mathcal{Z}=\{(x_1,y_1),\dots,(x_n,y_n)\}$, let: ($\mathcal{C}=\mathcal{H}$)
$\hat c_n^*=\arg\min_{c\in\mathcal{C}}\frac1n|\{(x_i,y_i):c(x_i)\ne y_i,i\le n\}|$. If $|\mathcal{C}|<\infty$:
$P(R(\hat c_n^*)-\inf_{c\in\mathcal{C}}R(c)>\epsilon)\le2|\mathcal{C}|\exp(-2n\epsilon^2)$ (entropy term; energy term)
For uncountably large $\mathcal{H}$: the VC-dim of a concept class $\mathcal{C}$ is a complexity measure for $\mathcal{C}$.
If $V_\mathcal{C}\ge2$: $P(R(\hat c_n^*)-\inf_{c\in\mathcal{C}}R(c)>\epsilon)\le9n^{V_\mathcal{C}}\exp(-\frac{n\epsilon^2}{32})$

Thm: Assume $|\mathcal{C}|<\infty$ & $R(c^*)=0$. Then for every n & ε>0 it holds:
1) $P(R(\hat c_n^*)>\epsilon)\le|\mathcal{C}|e^{-n\epsilon}$
2) $E[R(\hat c_n^*)]\le\frac{1+\log(|\mathcal{C}|)}{n}$
Proof: 1) $P(R(\hat c_n^*)>\epsilon)\le P(\max_{c\in\mathcal{C}:\hat R_n(c)=0}R(c)>\epsilon)$
$=E[\max_{c\in\mathcal{C}}\mathbb{1}\{\hat R_n(c)=0\}\mathbb{1}\{R(c)>\epsilon\}]$
$\le\sum_{c\in\mathcal{C}:R(c)>\epsilon}P(\hat R_n(c)=0)\le|\mathcal{C}|\cdot(1-\epsilon)^n\le|\mathcal{C}|e^{-n\epsilon}$
2) $E[X]=\int_0^\infty P(X>t)dt=\int_0^u\dots dt+\int_u^\infty\dots dt$ optimize for u
Thm: $R(\hat c_n^*)-\inf_{c\in\mathcal{C}}R(c)=R(\hat c_n^*)-\hat R_n(\hat c_n^*)+\hat R_n(\hat c_n^*)-\inf_{c\in\mathcal{C}}R(c)$
$\le R(\hat c_n^*)-\hat R_n(\hat c_n^*)+\hat R_n(c^*)-R(c^*)$
$\le\sup_{c\in\mathcal{C}}|\hat R_n(c)-R(c)|+\sup_{c\in\mathcal{C}}|\hat R_n(c)-R(c)|$ ("optim." uniform conv.; "prediction" pointwise conv.)
$\le2\cdot\sup_{c\in\mathcal{C}}|\hat R_n(c)-R(c)|$
$\Rightarrow P(R(\hat c_n^*)-\inf_{c\in\mathcal{C}}R(c)>\epsilon)\le P(\sup_{c\in\mathcal{C}}|\hat R_n(c)-R(c)|>\frac\epsilon2)$
Rem: Distinguish 2 cases:
1) Classifier c selected before seeing $\mathcal{D}_{train}$. Then the empirical classification risk of c conv. against its expected classification risk (LLN)
2) Classifier $\hat c_n$ selected according to an induction principle (ERM) after we have seen the data $\mathcal{Z}$ ↳ LLN does not apply since $\hat c_n=\hat c_n(\mathcal{Z}_n)$
Markov Ineq: $X\ge0$ RV ⇒ $P(X\ge\epsilon)\le\frac{E[X]}{\epsilon}$
Thm: X RV with $E[X]$ & $a\le X\le b$ ⇒ $\forall s>0$: $E[\exp(sX)]\le\exp(s^2(b-a)^2/8)$
Hoeffding's Thm: Let $X_1,\dots,X_n$ be indep bounded RVs s.t. $X_i\in[a_i,b_i]$ & $S_n=\sum_{i=1}^n X_i$. Then $\forall t>0$:
$P(S_n-E_X[S_n]\ge\pm t)\le\exp(-\frac{2t^2}{\sum_{i=1}^n(b_i-a_i)^2})$
Proof Idea: 1) Markov with $P(\exp(X)\ge\exp(\epsilon))$, 2) iid, 3) Thm, 4) $s=\frac{4t}{\sum_i(b_i-a_i)^2}$
Lemma: $\tilde S=\frac{S_n}{n}$, $t=n\epsilon\Rightarrow P(|\tilde S_n-E_X[\tilde S_n]|\ge\epsilon)\le2\exp(-\frac{2n\epsilon^2}{\sum_i(b_i-a_i)^2/n})$
Thm: Assume that the cardinality of $\mathcal{C}$ is bounded by N ⇒ $P(\sup_{c\in\mathcal{C}}|\hat R_n(c)-R(c)|>\epsilon)\le2N\exp(-2n\epsilon^2)$
↳ Def $\delta:=2N\exp(-2n\epsilon^2)\Rightarrow\epsilon=\sqrt{\frac{\ln(N)-\ln(\delta/2)}{2n}}$ "variance"
$\Rightarrow R(c)\le\hat R_n(c)+\epsilon$
Empirical Risk min. for hyperplanes: $a\in\mathbb{R}^{d+1}$: $\{x\in\mathbb{R}^d:\sum_{i=1}^d a_ix_i+a_0\}$ set of hyperplanes. Select $\binom nd$ hyperplanes & estim error: $\#\mathcal{H}=2\cdot\binom nd$
$\Rightarrow\hat c=\arg\min_{i=1,\dots,2\binom nd}\hat R_n(c_i)\Rightarrow\forall c\in\mathcal{C}:\hat R_n(c)\ge\hat R_n(\hat c)-\frac dn$
Thm: Assume X has a density. If $\hat c$ is found by emp. error minimization, then, ∀ distr of (X,Y), if $n\ge d$ & $2\frac dn\le\epsilon\le1$ we have: $P(R(\hat c)>\inf_c R(c)+\epsilon)\le e^{2d\epsilon}(2\binom nd+1)\exp(-\frac{n\epsilon^2}{2})$
Moreover: if $n\ge d$: $E[R(\hat c)-R]\le\sqrt{\frac2n(d+1)(\log(n)+2)}$
Rem: $\binom nd\sim O(n^d)$

13. Useful Remarks
$\mathbb{E}_{D,X,Y}[(\hat f(X)-Y)^2]=\underbrace{(\mathbb{E}_{D,X}[\hat f(X)]-\mathbb{E}[Y])^2}_{\text{bias}^2}+\underbrace{\mathbb{E}_D\left[\left(\mathbb{E}_{D,X}[\hat f(X)]-\hat f(X)\right)^2\right]}_{\text{variance}}+\underbrace{\mathbb{E}[(\mathbb{E}[Y]-Y)^2]}_{\text{noise}}$
$p(f_*|\mathbf{y})=\int p(f_*|\mathbf{f})p(\mathbf{f}|\mathbf{y})$. [Hint: $\int\mathcal{N}(a;Bc,D)\mathcal{N}(c;e,F)dc=\mathcal{N}(a;Be,D+BFB^T)$]
Hinge Loss: $L_H=\max\{0,1-y\cdot z\}$, $z\in\{\pm1\}$
Newton update: $w^{t+1}\leftarrow w^t-H_w^{-1}(\nabla_w\mathcal{L})$
Sylvester's Criterion: M PSD ⟺ all upper left k×k submatrices have positive det. (k ≤ n)
Time Complexity of $\frac{\partial\mathcal{L}}{\partial\theta_j}$ for $\mathcal{L}=\log(\mathcal{N}(0,K_\theta+\sigma^2\mathbb{1}))$ is $O(mn^2)$
Rem PAC: Let $\mathcal{C}$ be a finite concept class & assume that $\mathcal{H}=\mathcal{C}$. Let $\mathcal{A}$ be an algorithm that returns consistent Hypothesis ($\forall n<\infty:\hat R_n(\hat c)=0$) for any target concept $c\in\mathcal{C}$ & any iid sample $\mathcal{Z}$. For any ε,δ>0, if $n\ge\frac1\epsilon(\log(|\mathcal{H}|)+\log(\frac1\delta))$, then success prob. is determined by $P(R(\hat c)\le\epsilon)\ge1-\delta$ or $P(R(\hat c)>\epsilon)\le\delta$



Rem DL: Forward Diff process $dx=\sigma^t dw$, $p(x(t)|x(0))=\mathcal{N}(x(0),\sigma^2(t))$, $\sigma^2(t)=\frac{\sigma^{2t}-1}{2\log(\sigma)}$