Projection: $\mathrm{proj}_{\boldsymbol{b}}(\boldsymbol{a}) = \frac{\boldsymbol{a}^\top\boldsymbol{b}}{\|\boldsymbol{b}\|^2}\boldsymbol{b}$.

Bayes. has 3 steps: (1) def. prior, (2) def. likelihood, and (3) Bayes rule to compute posterior.

Cholesky decomp: If $\mathbf{A}$ is PD, then $\exists\mathbf{L}$ s.t. $\mathbf{A} = \mathbf{L}^\top\mathbf{L}$.

Geometric series: $\sum_{k=0}^\infty ar^k = {}^a/_{1-r}$ if $|r| < 1$.

$1 - x \le \exp(-x) \implies (1-\epsilon)^n \le \exp(-n\epsilon)$.

Information theory:

$$\begin{aligned} H(p) &= \mathbb{E}_p[-\log p(X)] \\ D_{\mathrm{KL}}(p \parallel q) &= \mathbb{E}_p[\log {}^{p(X)}/_{q(X)}] \\ H(p,q) &= \mathbb{E}_p[-\log q(X)] = H(p) + D_{\mathrm{KL}}(p \parallel q) \\ \mathrm{I}(X;Y) &= \mathbb{E}[\log {}^{p(X,Y)}/_{p(X)p(Y)}] = H(X) - H(X \mid Y). \end{aligned}$$

Gaussian:
$(2\pi)^{-n/2}|\boldsymbol{\Sigma}|^{-1/2}\exp\Big(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\Big)$.

Conditional: If $\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{bmatrix} \sim \mathcal{N}\Big(\begin{bmatrix}\boldsymbol{\mu}_1\\ \boldsymbol{\mu}_2\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12}\\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22}\end{bmatrix}\Big)$. Then: $\boldsymbol{x}_2 \mid \boldsymbol{x}_1 = \boldsymbol{z} \sim \mathcal{N}(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}})$, where $\bar{\boldsymbol{\mu}} = \boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}(\boldsymbol{z}-\boldsymbol{\mu}_1)$ and $\bar{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}$.

Information theory:

$$D_{\mathrm{KL}} = \frac{1}{2}\left[\log\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|} - d + \mathrm{tr}(\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1) + (\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)^\top\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)\right]$$

$$H = \frac{d}{2}\log(2\pi e) + \frac{1}{2}\log|\boldsymbol{\Sigma}|.$$

Paradigms of data science

Frequentism (optimize likelihood, MLE): $\boldsymbol{\theta}^\star \in \mathrm{argmax}_{\boldsymbol{\theta}\in\Theta}\sum_{i=1}^n \log p(\boldsymbol{x}_i \mid \boldsymbol{\theta})$.

Bayesianism (optimize posterior, MAP): $\boldsymbol{\theta}^\star \in \mathrm{argmax}_{\boldsymbol{\theta}\in\Theta}\log p(\boldsymbol{\theta}) + \sum_{i=1}^n \log p(\boldsymbol{x}_i \mid \boldsymbol{\theta})$.

Statistical learning (optimize risk):
$f^\star \in \mathrm{argmax}_{f\in\mathcal{F}}\,\mathcal{R}(f) \doteq \mathbb{E}_{X,Y}[\ell(Y, f(X))]$
$\hat{f}_n \in \mathrm{argmax}_{f\in\mathcal{F}}\,\hat{\mathcal{R}}_n(f) \doteq \frac{1}{n}\sum_{i=1}^n \ell(y_i, f(x_i))$.

Anomaly detection

Objects $\mathcal{X} \subseteq \mathbb{R}^d$ with normal class $\mathcal{N} \subseteq \mathcal{X}$. Construct $\phi : \mathcal{X} \to \{0,1\}$ such that $\phi(\boldsymbol{x}) = \mathbb{1}\{\boldsymbol{x} \notin \mathcal{N}\}$. Anomaly is an "unlikely event" ⇒ Fit distribution to $\mathcal{X}$ and score according to $p(\boldsymbol{x})$.

PCA: Proj. $\mathcal{X}$ to low-dim. ⇒ $\Pi(\mathcal{N})$ is simpler.

Linearly project $\mathbb{R}^d$ to $\mathbb{R}^{d^-}$ such that maximum variance is preserved. Base case $d^- = 1$: Find $\boldsymbol{u}$ with $\|\boldsymbol{u}\| = 1$ s.t. $\boldsymbol{x} \mapsto \boldsymbol{u}^\top\boldsymbol{x}$. Sample mean and variance of reduced dataset: $\mathbb{E}[\boldsymbol{u}^\top\boldsymbol{x}] = \boldsymbol{u}^\top\mathbb{E}[\boldsymbol{x}]$ and $\mathbb{V}[\boldsymbol{u}^\top\boldsymbol{x}] = \boldsymbol{u}^\top\mathrm{Cov}(\boldsymbol{x})\boldsymbol{u}$. We want maximum variance so we have: $\boldsymbol{u}^\star \in \mathrm{argmax}_{\|\boldsymbol{u}\|=1}\boldsymbol{u}^\top\mathrm{Cov}(\boldsymbol{x})\boldsymbol{u}$. Solvable by vanishing Lag. grad. Easy to find that $\boldsymbol{u}^\star$ is eigenvector with maximum eigenvalue. Then project it out ($\mathcal{X}_1 = \{\boldsymbol{x} - \mathrm{proj}_{\boldsymbol{u}_1}(\boldsymbol{x})\} = \{\boldsymbol{x} - \boldsymbol{u}_1^\top\boldsymbol{x}\cdot\boldsymbol{u}_1\}$) and do the same for next dimension

GMM: Lin. proj. onto low-dim. spaces resemble Gaussian dist. ⇒ Fit GMM to $\Pi(\mathcal{X})$.

Fit $p(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{j=1}^k \pi_j\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)$ to data with EM algorithm. Can derive $\log p(\mathbf{X};\boldsymbol{\theta}) = M(q,\boldsymbol{\theta}) + \mathbb{E}(q,\boldsymbol{\theta})$, where $M(q,\boldsymbol{\theta}) \doteq \mathbb{E}_q[\log {}^{p(\mathbf{X},z;\boldsymbol{\theta})}/_{q(z)}]$ and $E(q,\boldsymbol{\theta}) \doteq \mathbb{E}_q[\log {}^{q(z)}/_{p(z|\mathbf{X};\boldsymbol{\theta})}]$. Properties: $\log p(\mathbf{X};\boldsymbol{\theta}) \ge M(q,\boldsymbol{\theta})$ and $\log p(\mathbf{X};\boldsymbol{\theta}) = M(q^\star,\boldsymbol{\theta})$ where $q^\star = p(\cdot \mid \mathbf{X};\boldsymbol{\theta})$. Alg.: Iteratively $q^\star \in \mathrm{argmin}_q E(q,\boldsymbol{\theta}_{t-1})$ and $\boldsymbol{\theta}_t \in \mathrm{argmax}_{\boldsymbol{\theta}} M(q^\star,\boldsymbol{\theta})$. These can be done in closed form for GMM.

Density estimation

MLE properties: (1) *Consistency*: $\lim_{n\to\infty}\hat{\theta}_n^{\mathrm{MLE}} = \boldsymbol{\theta}$; (2) *Equivariance*: If $\hat{\boldsymbol{\theta}}$ is the MLE of $\boldsymbol{\theta}$, then $g(\hat{\boldsymbol{\theta}})$ is the MLE of $g(\boldsymbol{\theta})$; (3) *Asymptotically normal*: In the limit of n, ${}^{\hat{\boldsymbol{\theta}}-\boldsymbol{\theta}}/_{\sqrt{n}}$ converges to $\mathcal{N}(\mathbf{0},\mathcal{I}(\boldsymbol{\theta})^{-1})$; (4) *Asymptotically efficient*: In the limit of n, MLE has smallest variance among unbiased estimators.

Rao-Cramér bound: For any unbiased estimator:

$$\mathbb{V}[\hat{\theta}(\boldsymbol{y})] \ge \frac{(\frac{\partial}{\partial\theta}b_{\hat{\theta}}+1)^2}{\mathcal{I}_n(\theta)} + b_{\hat{\theta}}^2,$$

where $\mathcal{I}_n(\theta) \doteq \mathbb{E}_{\boldsymbol{y}|\theta}[(\frac{\partial}{\partial\theta}\log p(\boldsymbol{y} \mid \theta))^2]$ and $b_{\hat{\theta}} \doteq \mathbb{E}_{\boldsymbol{y}|\theta}[\hat{\theta}(\boldsymbol{y})] - \theta$. If unbiased: $\mathbb{V}[\hat{\theta}(\boldsymbol{y})] \ge {}^1/_{\mathcal{I}_n(\theta)}$. And MLE: $\lim_{n\to\infty}\mathbb{V}[\hat{\theta}^{\mathrm{MLE}}(\boldsymbol{y})] = {}^1/_{\mathcal{I}_n(\theta)}$.

Regression

Minimize loss: $\ell(f) = \frac{1}{n}\sum_{i=1}^n(f(\boldsymbol{x}_i) - y_i)^2$.

Linear regression: Assume $Y \mid \boldsymbol{X} = \boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\beta}_\star^\top\boldsymbol{x}, \sigma^2)$. We parameterize $f(\boldsymbol{x};\boldsymbol{\beta}) = \boldsymbol{\beta}^\top\boldsymbol{x}$.

OLSE: $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\boldsymbol{y}$ s.t. $\mathbf{X} \in \mathbb{R}^{n\times d}, \boldsymbol{y} \in \mathbb{R}^n$.

Potential problems:

1. Remove outliers, because linear models are heavily influenced by them;
2. Standardize data, because features on different scales result in unstable matrix inversion;
3. "Curse of dimensionality": In high dimensionality, logistic regression outputs overconfident outputs, due to overestimation of weights;
4. Collinear features result in unstable matrix inversion due to small eigenvalues.

Risk decomposition:
$\mathbb{E}[(\hat{f}(X) - Y)^2] = (\mathbb{E}[\hat{f}(X)] - \mathbb{E}[y])^2 + \mathbb{V}[\hat{f}(X)] + \mathbb{V}[y]$.
Proof: Use $Y = f(\boldsymbol{X}) + \epsilon$ and show $\mathbb{E}[\epsilon^2] = \mathbb{V}[Y]$ where $\mathbb{E}[\epsilon] = 0$. Then $\pm\mathbb{E}[\hat{f}(\boldsymbol{X})]$ and finalize.

Gauss-Markov: $\mathbb{V}[\boldsymbol{a}^\top\hat{\boldsymbol{\beta}}] \le \mathbb{V}[\boldsymbol{a}^\top\tilde{\boldsymbol{\beta}}]$ for any $\boldsymbol{a} \in \mathbb{R}^d$ and $\tilde{\boldsymbol{\beta}} = \mathbf{C}\boldsymbol{y}$ for $\mathbf{C} \in \mathbb{R}^{d\times n}$. (OLSE $\hat{\boldsymbol{\beta}}$ is unique min.-var. unbiased linear estimator.) This does not mean it is best, because adding some bias may decrease variance considerably.

Regularization: Ridge: Gaussian prior $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0},\lambda\mathbf{I})$. LASSO: Laplacian Gaussian $\boldsymbol{\beta} \sim \mathrm{Lap}(\mathbf{0},\lambda\mathbf{I})$. ℓ_1 results in sparse weights (better interpretation) and the sign of features remain.

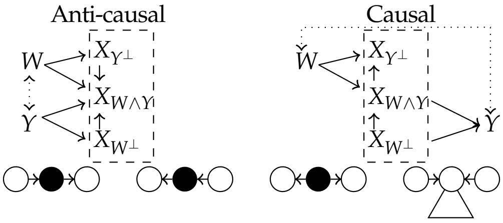
Polynomial regression: Feature map with all polynomials $\phi(\boldsymbol{x})$ and perform lin. reg. in this space: $\psi(\boldsymbol{x};\boldsymbol{\beta}) = \boldsymbol{\beta}^\top\phi(\boldsymbol{x})$. Problem: Infinitely dimensional ⇒ Ill-defined inner product. Solution: Fix by data-dependent scalar, specifically: $\phi(\boldsymbol{x}) = \exp(-\|\boldsymbol{x}\|^2/2)\left[\prod_{i=1}^d x_i^{\alpha_i}/\sqrt{\prod_{i=1}^d \alpha_i!}\right]_{\boldsymbol{\alpha}\in\mathbb{N}^d}$. ⇒ RBF kernel: $\langle\phi(\boldsymbol{x}),\phi(\boldsymbol{x}')\rangle = \exp(-\|\boldsymbol{x}-\boldsymbol{x}'\|^2/2)$. Now compute OLSE in this space: $\hat{\boldsymbol{\beta}} = (\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\boldsymbol{y}, \quad \boldsymbol{\Phi} \in \mathbb{R}^{n\times\infty}$. Problem: Cannot compute $\boldsymbol{\Phi}^\top\boldsymbol{\Phi} \in \mathbb{R}^{\infty\times\infty}$. Solution: Rewrite OLSE: $\hat{\boldsymbol{\beta}} = \boldsymbol{\Phi}^\top(\boldsymbol{\Phi}\boldsymbol{\Phi}^\top)^{-1}\boldsymbol{y}$. Prediction only contains kernel evaluations: $\psi(\boldsymbol{x}) = \boldsymbol{k}(\boldsymbol{x})^\top\mathbf{K}^{-1}\boldsymbol{y}$. Problem: $\mathcal{O}(n^3)$ runtime.

Causality

Causal fallacies where one might conclude X causes Y: (1) Reverse causality: Y causes X; (2) Third-cause fallacy: Z causes X and Y; (3) Bidirectional causation: X causes Y and Y causes X.

Domain shift: Test samples are drawn from different distribution than training set.

Shortcut learning: Spurious correlation between causal and non-causal features in the training depend on environment.



Necessary conditions for counterfactual invariance: Anti-causal: $f(\mathbf{X}) \perp \mathbf{W} \mid Y$. Causal without selection (but possibly confounded): $f(\mathbf{X}) \perp \mathbf{W}$. Causal without confounded (but possibly selection): $f(\mathbf{X}) \perp \mathbf{W} \mid Y$ as long as $\mathbf{X} \perp Y \mid \mathbf{X}_{\mathbf{W}^\perp}, \mathbf{W}$.

Gaussian processes

Outputs are modeled as $\boldsymbol{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$. Thus: $\boldsymbol{y} \mid \mathbf{X},\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta},\sigma^2\mathbf{I})$. BLR extends linear reg. with prior on $\boldsymbol{\beta}$: $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0},\boldsymbol{\Lambda}^{-1})$. Posterior: $\boldsymbol{\beta} \mid \mathbf{X},\boldsymbol{y} \sim \mathcal{N}(\tilde{\boldsymbol{\mu}},\tilde{\boldsymbol{\Sigma}})$, where

$$\tilde{\boldsymbol{\mu}} = -\frac{1}{\sigma^2}\tilde{\boldsymbol{\Sigma}}\mathbf{X}^\top\boldsymbol{y}, \quad \tilde{\boldsymbol{\Sigma}} = \sigma^2\Big(\mathbf{X}^\top\mathbf{X} + \sigma^2\boldsymbol{\Lambda}\Big)^{-1}.$$

Joint distribution over outputs (using prior): $\boldsymbol{y} \mid \mathbf{X} \sim \mathcal{N}\Big(\mathbf{0},\mathbf{X}\boldsymbol{\Lambda}^{-1}\mathbf{X}^\top + \sigma^2\mathbf{I}\Big)$. Prediction: $y^\star \mid \boldsymbol{x}^\star,\mathbf{X},\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}^\star,\boldsymbol{\Sigma}^\star)$, where $\boldsymbol{\mu}^\star = \boldsymbol{k}^\top(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\boldsymbol{y}$ and $\boldsymbol{\Sigma}^\star = k - \boldsymbol{k}^\top(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\boldsymbol{k}$. Problem: $\mathcal{O}(n^3)$ runtime.

Kernels: Kernel k must satisfy symmetry and PSD:

$$\int\int f(\boldsymbol{x})k(\boldsymbol{x},\boldsymbol{x}')f(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}' \ge 0, \quad \forall f \in L_2.$$

Or there exists ϕ s.t. $k(\boldsymbol{x},\boldsymbol{x}') = \phi(\boldsymbol{x})^\top\phi(\boldsymbol{x}')$.

Linear kernel: $k(\boldsymbol{x},\boldsymbol{x}') = \boldsymbol{x}^\top\boldsymbol{x}'$; *Polynomial kernel*: $k(\boldsymbol{x},\boldsymbol{x}') = (\boldsymbol{x}^\top\boldsymbol{x}'+1)^p$; *RBF kernel*: $k(\boldsymbol{x},\boldsymbol{x}') = \exp(-\|\boldsymbol{x}-\boldsymbol{x}'\|^2/\ell^2)$; *Sigmoid kernel*: $\tanh(\kappa\boldsymbol{x}^\top\boldsymbol{x}') - b$.

If k_1 and k_2 are valid kernels and $c > 0$, then the following are: $k_1 + k_2, k_1 \cdot k_2, c \cdot k_1$, and $\exp \circ k_1$.

Uncertainty quantification

Statistical model validation: Methods to evaluate $\hat{f}$ (or algorithm $\mathcal{A}$) that is trained on data $\mathcal{Z}$: *Cross-validation*: Partition $\mathcal{Z} = \bigcup_{k=1}^K \mathcal{Z}_k$ and produce K estimators $\hat{f}^{-k}$ from $\mathcal{Z}\backslash\mathcal{Z}_k$. Then estimate risk by $\mathcal{R}^{\mathrm{CV}}(\mathcal{A}) = \frac{1}{n}\sum_{i=1}^n \ell(y_i\hat{f}^{-k(i)}(\boldsymbol{x}_i))$, where k maps i to the partition such that $\boldsymbol{x}_i \in \mathcal{Z}_{k(i)}$.

Bootstrap: Used for measuring dist. over stat. params. Draw B bootstrap samples ⇒ Compute parameter for each ⇒ Compute statistics. Can also use for empirical risk: $\hat{\mathcal{R}}^{\mathrm{BS}}(\mathcal{A}) \doteq \frac{1}{n\cdot B}\sum_{b=1}^B\sum_{i=1}^n \ell(y_i,\hat{f}^{*b}(\boldsymbol{x}_i))$. Problem: Overly optimistic. Solution: $\mathcal{R}^{\mathrm{BS}}(\mathcal{A}) \doteq \frac{1}{n}\sum_{i=1}^n\frac{1}{|\mathcal{C}^{-i}|}\sum_{b\in\mathcal{C}^{-i}}\ell(y_i,\hat{f}^{*b}(\boldsymbol{x}_i))$.

Correct for optimism of $\hat{\mathcal{R}}^{\mathrm{BS}}$ by combining with $\mathcal{R}^{\mathrm{BS}}$: $\mathcal{R}^{(0.632)} = 0.368\hat{\mathcal{R}}^{\mathrm{BS}} + 0.632\mathcal{R}^{\mathrm{BS}}$. 0.632 is the prob. that a sample appears at least once in a bootstrap sample of size n.

Uncertainty in linear models: OLSE has distribution over estimators: $\hat{\boldsymbol{\beta}} \sim \mathcal{N}(\boldsymbol{\beta}^\star,\sigma^2(\mathbf{X}^\top\mathbf{X})^{-1})$. Unbiased estimator of σ^2: $\hat{\sigma}^2 = \frac{1}{n-d}\sum_{i=1}^n(\hat{\boldsymbol{\beta}}^\top\boldsymbol{x}_i - y_i)$. Then we have $1-\alpha$ confidence interval for $\beta_j^\star$: $\hat{\beta}_j \pm z_{\alpha/2}\epsilon(\hat{\beta}_j)$, where $z_{\alpha/2} = \Phi^{-1}(\alpha/2)$, Φ is standard Gaussian CDF, and $\epsilon(\hat{\beta}_j) = \hat{\sigma}^2(\mathbf{X}^\top\mathbf{X})_{jj}^{-1}$.

Statistical testing: Null hypothesis: $H_0 : \theta^\star \in \Theta$. Alternative hypothesis $H_1 : \theta^\star \in \Theta$. We are given n samples $x_1,\ldots,x_n \sim p(\cdot \mid \theta^\star)$ and a test statistic $t : \mathcal{X}^n \to \mathbb{R}$. The goal is to find a critical value $c \in \mathbb{R}$ such that $\mathbb{P}(t(X_1,\ldots,X_n) \ge c \mid \theta)$ is low when $\theta \in \Theta_0$ and high when $\theta \in \Theta_1$.

We want to minimize the prob. of choosing H_1 when H_0 holds (worst possible situation). We quantify this notion of risk as $\alpha_c \doteq \sup_{\theta\in\Theta_0}\mathbb{P}(t(x_1,\ldots,x_n) \ge c \mid \theta)$. Problem: $\alpha_c \to 0$ as $c \to \infty$, so $c^\star \to \infty$ minimizes the risk, but then we never accept H_1. Solution: Run test on realization $t(x_1,\ldots,x_n)$ and compute risk of least risky critical value that would incorrectly reject H_0: $p = \inf_{c\in\mathbb{R}}\{\alpha_c \mid t(x_1,\ldots,x_n) \ge c\}$. This is the p-value: $p \doteq \sup_{\theta\in\Theta_0}\mathbb{P}(t(X_1,\ldots,X_n) \ge t(x_1,\ldots,x_n) \mid \theta)$. Intuition: Inverse prob. of $x_{1:n}$ being an outlier.

Wald: $W = {}^{(\hat{\theta}-\theta_0)^2}/_{\hat{\sigma}^2}$; $H_0 : \theta = \theta_0$, $H_1 : \theta \neq \theta_0$.

Bayesian neural networks: (S)GD only yields single point estimate of weights ⇒ Define prior $\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0},\sigma^2\mathbf{I})$ and likelihood $p(\mathcal{Z} \mid \boldsymbol{\theta}) = \prod_{\boldsymbol{x},y\in\mathcal{Z}} p(y \mid \boldsymbol{x},\boldsymbol{\theta})$ ⇒ Posterior with Bayes rule. Problem: $p(\mathcal{Z})$ is intractable. Solution: Variational inference with isotropic Gaussians and find

$$\begin{aligned} q^\star &\in \operatorname*{argmin}_{\boldsymbol{\mu},\sigma>0} D_{\mathrm{KL}}(\mathcal{N}(\boldsymbol{\mu},\sigma^2\mathbf{I}) \parallel p(\boldsymbol{\theta} \mid \mathcal{Z})) \\ &= \operatorname*{argmin}_{\boldsymbol{\mu},\sigma>0} \mathbb{E}_{\boldsymbol{\theta}\sim\mathcal{N}}[F(\boldsymbol{\mu},\sigma,\boldsymbol{\theta})], \end{aligned}$$

where $F(\boldsymbol{\mu},\sigma,\boldsymbol{\theta}) = \log\mathcal{N}(\boldsymbol{\theta};\boldsymbol{\mu},\sigma^2\mathbf{I}) - \log p(\mathcal{Z} \mid$

$\boldsymbol{\theta}) - \log p(\boldsymbol{\theta})$. Then, we can apply SGD with the following gradients:

$$\nabla_{\boldsymbol{\mu}} = \mathbb{E}_{\boldsymbol{\epsilon}}[\nabla_{\boldsymbol{\theta}} F(\boldsymbol{\mu},\sigma,\boldsymbol{\theta}) + \nabla_{\boldsymbol{\mu}} F(\boldsymbol{\mu},\sigma,\boldsymbol{\theta})]$$

$$\nabla_{\sigma} = \mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{\epsilon}^\top \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\mu},\sigma,\boldsymbol{\theta})] + \nabla_{\sigma} F(\boldsymbol{\mu},\sigma,\boldsymbol{\theta})],$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$ and $\boldsymbol{\theta} = \boldsymbol{\mu} + \sigma\boldsymbol{\epsilon}$.

Information-based transductive learning: We are given domain $\mathcal{X}$ that contains safe area $\mathcal{S} \subseteq \mathcal{X}$ and area of interest $\mathcal{A} \subseteq \mathcal{X}$. We have an unknown $f^\star$ that we want to explore within $\mathcal{A}$, but we can only query (noisy) observations in $\mathcal{S}$:

$$y_{\boldsymbol{x}} = f^\star(\boldsymbol{x}) + \epsilon_{\boldsymbol{x}}, \quad \mathbb{E}_{\epsilon_{\boldsymbol{x}}} = 0.$$

We are given a history of points $\mathcal{D}_{n-1}$ and need to compute which point will give the most additional information. ITL selects the next point as:

$$\boldsymbol{x}_n \in \operatorname*{argmax}_{\boldsymbol{x}\in\mathcal{S}} \mathrm{I}(\boldsymbol{f}_{\mathcal{A}}; y_{\boldsymbol{x}} \mid \mathcal{D}_{n-1}).$$

If $f \sim \mathrm{GP}(\mu,k)$, then

$$\mathrm{I}(\boldsymbol{f}_{\mathcal{A}}; y_{\boldsymbol{x}} \mid \mathcal{D}_{n-1}) = \frac{1}{2}\log\frac{\mathbb{V}[y_{\boldsymbol{x}} \mid \mathcal{D}_{n-1}]}{\mathbb{V}[y_{\boldsymbol{x}} \mid \boldsymbol{f}_{\mathcal{A}}, \mathcal{D}_{n-1}]}.$$

Proof: Use entropy of Gaussian.

Convex optimization and SVMs

$$\min \; f(\boldsymbol{x}), \quad \text{s.t.} \; g_i(\boldsymbol{x}) = 0, h_j(\boldsymbol{x}) \le 0.$$

where f and h_j are convex and g_i are affine.

Lagrangian: $\mathcal{L}(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\alpha}) \doteq f(\boldsymbol{x}) + \sum_{i=1}^n \lambda_i g_i(\boldsymbol{x}) + \sum_{j=1}^m \alpha_j h_j(\boldsymbol{x})$. Lagrange dual function: $\theta(\boldsymbol{\lambda},\boldsymbol{\alpha}) \doteq \inf_{\boldsymbol{x}\in\mathcal{X}} \mathcal{L}(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\alpha})$.

Weak duality: Let $\boldsymbol{x} \in \mathcal{C}, \boldsymbol{\alpha} \ge \mathbf{0}$, then $\theta(\boldsymbol{\lambda},\boldsymbol{\alpha}) \le f(\boldsymbol{x})$. Thus: $\max_{\boldsymbol{\lambda},\boldsymbol{\alpha}\ge\mathbf{0}} \theta(\boldsymbol{\lambda},\boldsymbol{\alpha}) \le \min_{\boldsymbol{x}\in\mathcal{C}} f(\boldsymbol{x})$.

If there is a Slater point (exists $\boldsymbol{x} \in \mathcal{C}$ such that $h_j(\boldsymbol{x}) < 0$ for all j) then strong duality: $\max_{\boldsymbol{\lambda},\boldsymbol{\alpha}\ge\mathbf{0}} \theta(\boldsymbol{\lambda},\boldsymbol{\alpha}) = \min_{\boldsymbol{x}\in\mathcal{C}} f(\boldsymbol{x})$.

If all g_i and h_j are differentiable, KKT conditions provide necessary (and sufficient for convex programming) conditions for strong duality:

$$\alpha_j^\star h_j(\boldsymbol{x}^\star) = 0, \quad \nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}^\star,\boldsymbol{\lambda}^\star,\boldsymbol{\alpha}^\star) = \mathbf{0}.$$

Or, condition 2: $\boldsymbol{x}^\star \in \operatorname{argmin}_{\boldsymbol{x}\in\mathcal{X}} \mathcal{L}(\boldsymbol{x},\boldsymbol{\lambda}^\star,\boldsymbol{\alpha}^\star)$.

Support vector machine: We want to linearly separate a dataset with maximum margin ⇒ Model as convex program with constraint for each data point: $f[\boldsymbol{w},b](\boldsymbol{x},y) = y(\boldsymbol{w}^\top\boldsymbol{x} + b) \ge \epsilon > 0$.

Margin ($\boldsymbol{x}^+$ and $\boldsymbol{x}^-$ are support vectors):

$$2\cdot m(\boldsymbol{w},b) = \|\mathrm{proj}_{\boldsymbol{w}}(\boldsymbol{x}^+) - \mathrm{proj}_{\boldsymbol{w}}(\boldsymbol{x}^-)\|$$
$$= |\bar{\boldsymbol{w}}^\top(\boldsymbol{x}^+ - \boldsymbol{x}^-)|.$$

Ill-posed problem because infinite number of solutions ⇒ Only one solution satisfies

$$\boldsymbol{w}^\top\boldsymbol{x}^+ + b = 1, \quad \boldsymbol{w}^\top\boldsymbol{x}^- + b = -1.$$

Then, $m(\boldsymbol{w},b) = {}^1/_{\|\boldsymbol{w}\|}$:

$$\min \; \frac{1}{2}\|\boldsymbol{w}\|^2, \quad \text{s.t.} \; 1 - y_i\left(\boldsymbol{w}^\top\boldsymbol{x}_i + b\right) \le 0.$$

$\boldsymbol{w}^\star = \sum_{i=1}^n \alpha_i^\star y_i \boldsymbol{x}_i$, $b^\star = -\frac{1}{2}\left(\boldsymbol{w}_\star^\top\boldsymbol{x}^+ + \boldsymbol{w}_\star^\top\boldsymbol{x}^-\right)$, where $\boldsymbol{\alpha}^\star$ is the dual solution.

SVM variations: Soft-margin introduces slackness in case data is not linearly separable:

$$\text{minimize} \quad \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^n \xi_i$$

$$\text{subject to} \quad y_i\left(\boldsymbol{w}^\top\boldsymbol{x}_i + b\right) \ge 1 - \xi_i, \quad \xi_i \ge 0.$$

Optimal slackness parameters:

$$\xi_i^\star = \max\left\{0, 1 - y_i\left(\boldsymbol{w}_\star^\top\boldsymbol{x}_i + b^\star\right)\right\}.$$

If data is not linearly separable, use features and their kernels: $\boldsymbol{w}_\star^\top\phi(\boldsymbol{x}) = \sum_{i=1}^n \alpha_i^\star y_i k(\boldsymbol{x}_i,\boldsymbol{x})$.

We can generalize the margin notion to multiclass by introducing weights $\boldsymbol{w}_z$ per class. The margin is defined as the maximum $m \in \mathbb{R}$ s.t.

$$m \le \left(\boldsymbol{w}_{z_i}^\top\boldsymbol{y}_i + b_{z_i}\right) - \max_{z\ne z_i}\left\{\boldsymbol{w}_z^\top\boldsymbol{y}_i + b_z\right\}.$$

New optimization problem:

$$\min \; \frac{1}{2}\|\boldsymbol{w}\|^2 = \frac{1}{2}\sum_{z=1}^k \|\boldsymbol{w}_z\|^2$$

$$\text{s.t.} \quad \left(\boldsymbol{w}_{z_i}^\top\boldsymbol{y}_i + b_{z_i}\right) - \max_{z\ne z_i}\left\{\boldsymbol{w}_z^\top\boldsymbol{y}_i + b_z\right\} \ge 1.$$

Structural SVMs can have infinitely many classes. So, we need to define a joint feature map ψ such that $f_{\boldsymbol{w}}(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{w}^\top\psi(\boldsymbol{x},\boldsymbol{y})$. This is used to perform classification: $c(\boldsymbol{x}) = \operatorname{argmax}_{\boldsymbol{y}\in\mathcal{Y}} f_{\boldsymbol{w}}(\boldsymbol{x},\boldsymbol{y})$.

We need to construct an algorithm to efficiently compute this argmax and an algorithm to compute the max in the below optimization problem. Some structures are closer than others ⇒ Introduce a loss function Δ:

$$\min \; \frac{1}{2}\|\boldsymbol{w}\|^2 \quad \text{s.t.} \quad \boldsymbol{w}^\top\psi(\boldsymbol{x}_i,\boldsymbol{y}_i)$$
$$- \max_{\boldsymbol{y}\ne\boldsymbol{y}_i}\left\{\boldsymbol{w}^\top\psi(\boldsymbol{x}_i,\boldsymbol{y}) + \Delta(\boldsymbol{y},\boldsymbol{y}_i)\right\} \ge 0.$$

Ensembles

Average B estimators into $\hat{f}$ ⇒ avg. bias and:

$$\mathbb{V}[\hat{f}] = \frac{1}{B^2}\sum_{b=1}^B \mathbb{V}[\hat{f}_b] + \frac{1}{B^2}\sum_{b=1}^B\sum_{b'\ne b}^B \mathrm{Cov}(\hat{f}_b,\hat{f}_{b'}).$$

If the covariances are low, the variance is significantly decreased while the bias remains the same.

Bagging: B times take a bootstrap sample and train a classifier. This works well because covariances are small due to using different subsets for training and the variances are similar because each subsample behaves similarly on average.

Random forests do this with (very deep) decision trees. Very deep because they have low bias and high variance, which is reduced by ensembling.

AdaBoost: AdaBoost reduces cov. by using a different weighting for each estimator. The weights are determined by error of prev. classifiers.

$$w_i^{(b+1)} = w_i^{(b)}\exp(\alpha_b\mathbb{1}\{c_b(\boldsymbol{x}_i) \ne y_i\})$$
$$\alpha_b = \log({}^{1-\epsilon_b}/_{\epsilon_b})$$
$$\epsilon_b = \sum_{i=1}^n \frac{w_i^{(b)}}{\sum_{j=1}^n w_j^{(b)}}\mathbb{1}\{c_b(\boldsymbol{x}_i) \ne y_i\}.$$

Final classifier: $\hat{c}(\boldsymbol{x}) = \mathrm{sgn}(\sum_{b=1}^B \alpha_b c_b(\boldsymbol{x}))$.

It can be shown that AdaBoost fits an additive model in base learners optimizing the exponential loss $\mathbb{E}[\exp(-yf(\boldsymbol{x}))]$ via Newton-like updates.

Stable diffusion

Diffusion models: Iteratively denoise. Continuous:

$$\mathrm{d}\boldsymbol{x}_t^+ = \boldsymbol{\mu}(\boldsymbol{x}_t,t)\mathrm{d}t + \sigma(\boldsymbol{x}_t,t)\mathrm{d}\omega_t$$
$$\mathrm{d}\boldsymbol{x}_t^- = \left[\boldsymbol{\mu}(\boldsymbol{x}_t,t) - \sigma^2(\boldsymbol{x}_t,t)\nabla_x\log p_t(\boldsymbol{x}_t)\right]\mathrm{d}t + \sigma(\boldsymbol{x}_t,t)\mathrm{d}\bar{\omega}_t.$$

DDPM scheduler: $\boldsymbol{x}_{t+1} = \sqrt{1-\beta_t}\boldsymbol{x}_t + \sqrt{\beta_t}\boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$. Backward process: $\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t)\right) + \sqrt{\beta_t}\boldsymbol{z}$, where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{\tau=1}^t \alpha_t$.

Diffusion models are trained by sampling $\boldsymbol{x}_0 \sim p_0$, $t \sim \mathrm{Unif}(\{1,\ldots,T\})$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$ and performing gradient step on $\ell = \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t)\|^2$.

Non-parametric Bayesian methods

Beta distribution ($x \in [0,1], \alpha,\beta > 0$): $\mathrm{Beta}(x;\alpha,\beta) \propto x^{\alpha-1}(1-x)^{\beta-1}$. Dirichlet generalizes ($\boldsymbol{x} \in \Delta^{n-1}$): $\mathrm{Dir}(\boldsymbol{x};\boldsymbol{\alpha}) \propto \prod_{k=1}^n x_k^{\alpha_k-1}$.

Problem: Need to know K (#clusters) beforehand.

A Dirichlet process $\mathrm{DP}(\alpha,H)$ is a distribution over probability distributions on a space Θ, where α is a concentration parameter. A sample $G \sim \mathrm{DP}(\alpha,H)$ is a function $G : \Theta \to \mathbb{R}_{\ge 0}$ such that $\int_\Theta G(\theta)\mathrm{d}\theta = 1$. For every partition $(T_1,\ldots,T_k)$ of Θ and $G \sim \mathrm{DP}(\alpha,H)$, we have

$$(G(T_1),.,G(T_k)) \sim \mathrm{Dir}(\alpha H(T_1),.,\alpha H(T_k)).$$

Dir can be sampled recursively by stick-breaking:

$$\beta_i \sim \mathrm{beta}\left(\alpha_i, \prod_{k=i+1}^K \alpha_k\right), \quad \rho_i = \beta_i\prod_{j=1}^{i-1}(1-\beta_j)$$

$$(\rho_{i+1},\ldots,\rho_K) \sim \mathrm{Dir}(\alpha_{i+1},\ldots,\alpha_K).$$

Still limited to fixed K. GEM distribution fixes this by fixing α such that $\beta_i \sim \mathrm{Beta}(1,\alpha)$ for all i. Recursion: $\beta_i \sim \mathrm{Beta}(1,\alpha)$ and

$$\rho_i = \beta_i\prod_{j=1}^{i-1}(1-\beta_j), \quad \rho_K = \beta_k\left(1 - \sum_{i=1}^{K-1}\rho_i\right).$$

Keep sampling cluster probs until satisfied.

If $(\rho_1,\rho_2,\ldots) \sim \mathrm{GEM}(\alpha)$ and $\theta_k \sim H$, then this is sample from $\mathrm{DP}(\alpha,H)$: $G(\theta) = \sum_{k=1}^\infty \rho_k\delta_{\theta_k}(\theta)$.

Chinese restaurant process:

$$P(n+1 \text{ joins table } \theta \mid \mathcal{P}) = \begin{cases} \frac{|\theta|}{\alpha+n} & \theta \in P \\ \frac{\alpha}{\alpha+n} & \text{else.} \end{cases}$$

Probability of partition $\mathcal{P}$ can be written as

$$P(\mathcal{P}) = \alpha^{|\mathcal{P}|}\frac{\alpha!}{(N+\alpha)!}\prod_{\tau\in\mathcal{P}}(|\tau|-1)!.$$

Problem is exchangeable. $\mathbb{E}[|\mathcal{P}|] \in \mathcal{O}(\alpha\log N)$.

DPMM: Assume $\Theta = \mathbb{R}$ with $\mu \in \Theta$ corresponding to $\mathcal{N}(\mu,\sigma)$ for fixed $\sigma > 0$ and $H = \mathcal{N}(\mu_0,\sigma_0)$ for fixed μ_0,σ_0. DPMM: Cluster probs are sampled from GEM: $(\rho_1,\rho_2,\ldots) \sim \mathrm{GEM}(\alpha)$. Cluster centers are sampled from base measure: $\mu_1,\mu_2,\ldots \sim \mathcal{N}(\mu_0,\sigma_0)$. Clusters are assigned: $z_i \sim \mathrm{Cat}(\rho_1,\rho_2,\ldots), \forall i \in [n]$. Data points are sampled: $x_i \sim \mathcal{N}(\mu_{z_i},\sigma), \forall i \in [n]$. This process is exchangeable. To fit a DPMM, we use a collapsed Gibbs sampling formulation: $p(z_i = k \mid \boldsymbol{z}^{-i},\boldsymbol{x},\alpha,\boldsymbol{\mu}) \propto p(z_i = k \mid \boldsymbol{z}^{-i},\alpha)p(x_i \mid \boldsymbol{x}^{-i},z_i = k,\boldsymbol{z}^{-i},\boldsymbol{\mu})$. Prior is as in CRP:

$$p(z_i = k \mid \boldsymbol{z}^{-i},\alpha) = \begin{cases} \frac{N_k^{-i}}{\alpha+N-1} & \text{existing } k \\ \frac{\alpha}{\alpha+N-1} & \text{else.} \end{cases}$$

Likelihood (right term) is cond. on cluster k:

$$\ell = \begin{cases} p(x_i \mid \boldsymbol{x}_k^{-i},\boldsymbol{\mu}) = \frac{p(x_i,\boldsymbol{x}_k^{-i}|\boldsymbol{\mu})}{p(\boldsymbol{x}_k^{-i}|\boldsymbol{\mu})} & \text{existing } k \\ p(x_i \mid \boldsymbol{\mu}) & \text{else.} \end{cases}$$

Statistical learning theory

PAC learning: *Definition*: A learning algorithm $\mathcal{A}$ can learn a concept $c \in \mathcal{C}$ if there exists $\mathrm{poly}(\cdot,\cdot,\cdot)$ such that for any distribution p on $\mathcal{X}$ and $\epsilon,\delta \in (0,{}^1/_2)$, if $\mathcal{A}$ receives a sample of size $n \ge \mathrm{poly}({}^1/_\epsilon,{}^1/_\delta,\mathrm{size}(c))$, then $\mathcal{A}$ outputs $\hat{c}$ such that $\mathbb{P}(\mathcal{R}(\hat{c}) \le \epsilon) \ge 1-\delta$. This probability is taken over the randomness of $\mathcal{Z}$ and $\mathcal{A}$. $\mathcal{C}$ is PAC learnable from $\mathcal{H}$ if there is an $\mathcal{A}$ that can learn any $c \in \mathcal{C}$.

If $\mathcal{A}$ runs polynomial in only ${}^1/_\delta$ and ${}^1/_\epsilon$, then $\mathcal{C}$ is efficiently PAC learnable.

In the stochastic setting, y is also random and not deterministically decided by a concept $c \in \mathcal{C}$. Now the criterium is $\mathbb{P}_{\mathcal{Z}\sim p}(\mathcal{R}(\hat{c}) - \inf_{c\in\mathcal{C}}\mathcal{R}(c) \le \epsilon) \ge 1-\delta$.

Vapnik-Chervonenkis: VC dimension is the cardinality of the largest set of points that $\mathcal{C}$ can shatter.

Vapnik and Chervonenkis: Assume a finite concept class and $\mathcal{R}(c^\star) = 0$ and define $c_n^\star \in \{c \in \mathcal{C} \mid \hat{\mathcal{R}}_n(c) = 0\}$. Then, for every $n \in \mathbb{N}$ and $\epsilon > 0$,

$$\mathbb{P}(\mathcal{R}(\hat{c}_n^\star) > \epsilon) \le |\mathcal{C}|\exp(-n\epsilon).$$

And: $\mathbb{E}[\mathcal{R}(\hat{c}_n^\star)] \le \frac{1+\log|\mathcal{C}|}{n}$.

VC inequality: $\mathbb{P}(\mathcal{R}(\hat{c}_n^\star) - \inf_{c\in\mathcal{C}}\mathcal{R}(c) > \epsilon) \le \mathbb{P}(\sup_{c\in\mathcal{C}}|\hat{\mathcal{R}}_n(c) - \mathcal{R}(c)| > \frac{\epsilon}{2})$.

Hoeffding: Let $X_i \in [a_i,b_i]$ be i.i.d. and $S_n = \sum_{i=1}^n X_i$, then for any $t > 0$,

$$\mathbb{P}(S_n - \mathbb{E}[S_n] \ge t) \le \exp\left(-\frac{2t^2}{\sum_{i=1}^n(b_i-a_i)^2}\right).$$

Same bound for $\le -t$.

As a result: $\mathbb{P}(\tilde{S}_n - \mathbb{E}[\tilde{S}_n] \ge \epsilon) \le \exp\left(-\frac{2n\epsilon^2}{\sum_{i=1}^n {}^{(b_i-a_i)^2}/_n}\right)$, where $\tilde{S}_n = {}^{S_n}/_n$.

Assume $|\mathcal{C}| \le N$, then for all $\epsilon > 0$,

$$\mathbb{P}(\sup_{c\in\mathcal{C}}|\hat{\mathcal{R}}_n(c) - \mathcal{R}(c)| > \epsilon) \le 2N\exp(-2n\epsilon^2).$$

We can deal with infinite $|\mathcal{C}|$ by representing hypotheses by the classifications that they yield. Or measuring the VC dimension of $\mathcal{C}$.