Probablistic AI

0 Fundamentals

Useful PDFs:

Normal: $\frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right)}{\sqrt{(2\pi)^k\det(\Sigma)}}$

Beta: $\mathrm{Beta}(\theta;\alpha,\beta)\propto\theta^{\alpha-1}(1-\theta)^{\beta-1}$

Laplace: $\frac{1}{2l}\exp\left(-\frac{|x-\mu|}{l}\right)$

Properties of Expectation:
$\mathbb{E}[\mathbf{g}(\mathbf{X})]=\int_{\mathbf{X}(\Omega)}\mathbf{g}(\mathbf{x})\cdot p(\mathbf{x})d\mathbf{x}$ (if $\mathbf{g}$ nice and $\mathbf{X}$ cont.) (**LOTUS**)
$\mathbb{E}_{\mathbf{Y}}[\mathbb{E}_{\mathbf{X}}[\mathbf{X}\,|\,\mathbf{Y}]]=\mathbb{E}[\mathbf{X}]$ (**Tower rule**)

Covariance:
$\mathrm{Cov}[\mathbf{X},\mathbf{Y}]\doteq\mathbb{E}\left[(\mathbf{X}-\mathbb{E}[\mathbf{X}])(\mathbf{Y}-\mathbb{E}[\mathbf{Y}])^\top\right]$

Correlation: $\mathrm{Cor}[\mathbf{X},\mathbf{Y}](i,j)\doteq\frac{\mathrm{Cov}[X_i,Y_j]}{\sqrt{\mathrm{Var}[X_i]\mathrm{Var}[Y_j]}}$

Variance: $\mathrm{Var}[\mathbf{X}]\doteq\mathrm{Cov}[\mathbf{X},\mathbf{X}]$

Properties of variance:
$\mathrm{Var}[\mathbf{AX}+\mathbf{b}]=\mathbf{A}\mathrm{Var}[\mathbf{X}]\mathbf{A}^\top$
$\mathrm{Var}[\mathbf{X}+\mathbf{Y}]=\mathrm{Var}[\mathbf{X}]+\mathrm{Var}[\mathbf{Y}]+2\mathrm{Cov}[\mathbf{X},\mathbf{Y}]$
$\mathrm{Var}[\mathbf{X}]=\mathbb{E}_{\mathbf{Y}}[\mathrm{Var}_{\mathbf{X}}[\mathbf{X}\,|\,\mathbf{Y}]]+\mathrm{Var}_{\mathbf{Y}}[\mathbb{E}_{\mathbf{X}}[\mathbf{X}\,|\,\mathbf{Y}]]$ (**LOTV**)

Jensen: Given g convex: $g(\mathbb{E}[X])\leq\mathbb{E}[g(X)]$

Change of variables formula $\mathbf{Y}=g(\mathbf{X})\implies p_{\mathbf{Y}}(\mathbf{y})=p_{\mathbf{X}}(\mathbf{g}^{-1}(\mathbf{y}))\cdot\left|\det\left(\mathbf{Dg}^{-1}(\mathbf{y})\right)\right|$

Bayes' rule: $p(\mathbf{x}\,|\,\mathbf{y})=\frac{p(\mathbf{y}|\mathbf{x})\cdot p(\mathbf{x})}{p(\mathbf{y})}$

Posterior $p(\mathbf{x}\,|\,\mathbf{y})$,
Prior $p(\mathbf{x})$, (Conditional) likelihood $p(\mathbf{y}\,|\,\mathbf{x})$, Joint likelihood $p(\mathbf{x},\mathbf{y})$, Marginal likelihood $p(\mathbf{y})$.

Marginal and conditional of Gaussians:
Given $A,B\subseteq\{1,...,n\}$: $\mathbf{X}_A\sim\mathcal{N}(\mu_A,\mathbf{\Sigma}_A)$ and $\mathbf{X}_A\,|\,\mathbf{x}_B\sim\mathcal{N}(\mu_A+\mathbf{\Sigma}_{AB}\mathbf{\Sigma}_B^{-1}(\mathbf{x}_B-\mu_B),\mathbf{\Sigma}_A-\mathbf{\Sigma}_{AB}\mathbf{\Sigma}_B^{-1}\mathbf{\Sigma}_{BA})$

conjugate iff prior
and posterior from same family of distributions.

MLE: $\hat{\theta}_{\mathrm{MLE}}\doteq\underset{\theta\in\Theta}{\operatorname{argmax}}\,p(y_{1:n}\,|\,\mathbf{x}_{1:n},\theta)$

MAP
estimate: $\hat{\theta}_{\mathrm{MAP}}\doteq\operatorname{argmax}_{\theta\in\Theta}p(\theta\,|\,\mathbf{x}_{1:n},y_{1:n})$

RM conditions Given a function $M(\theta)$ and random variables $N(\theta)$ with $\mathbb{E}[N(\theta)]=M(\theta)$ $\theta_{n+1}\leftarrow\theta_n-a_n(N(\theta_n)-\alpha)$ converges to $M(\theta_\star)=\alpha$ if
$a_t\geq0$, $\sum_{t=0}^\infty a_t=\infty$, $\sum_{t=0}^\infty a_t^2<\infty$. + some niceness conditions

Woodbury: $(\mathbf{A}+\mathbf{UCV})^{-1}=$
$\mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{U}(\mathbf{C}^{-1}+\mathbf{VA}^{-1}\mathbf{U})^{-1}\mathbf{VA}^{-1}$

1 Bayesian Linear Regression

Setting: $\mathbf{y}=\mathbf{Xw}+\epsilon$, $\epsilon\sim\mathcal{N}(\mathbf{0},\sigma_{\mathrm{n}}^2\mathbf{I})$

Prior: $\mathbf{w}\sim\mathcal{N}(\mathbf{0},\sigma_{\mathrm{p}}^2\mathbf{I})$

Posterior: $\mathbf{w}\,|\,\mathcal{D}\sim\mathcal{N}(\mu,\mathbf{\Sigma})$, with
$\mathbf{\Sigma}\doteq\left(\sigma_{\mathrm{n}}^{-2}\mathbf{X}^\top\mathbf{X}+\sigma_{\mathrm{p}}^{-2}\mathbf{I}\right)^{-1}$ and $\mu\doteq\sigma_{\mathrm{n}}^{-2}\mathbf{\Sigma}\mathbf{X}^\top\mathbf{y}$

MAP: $\hat{\mathbf{w}}_{\mathrm{MAP}}=\operatorname{argmin}_{\mathbf{w}}\|\mathbf{y}-\mathbf{Xw}\|_2^2+\frac{\sigma_{\mathrm{n}}^2}{\sigma_{\mathrm{p}}^2}\|\mathbf{w}\|_2^2$,
identical to ridge regression with $\lambda\doteq\sigma_{\mathrm{n}}^2/\sigma_{\mathrm{p}}^2$.

A **Laplace prior** on the weights is equivalent to **lasso regression** with decay $\lambda\doteq\sigma_{\mathrm{n}}^2/\ell$.

Inference:$y^\star\,|\,\mathbf{x}^\star,\mathcal{D}\sim\mathcal{N}(\mu^\top\mathbf{x}^\star,\mathbf{x}^{\star\top}\mathbf{\Sigma}\mathbf{x}^\star+\sigma_{\mathrm{n}}^2)$.
$\mathrm{Var}[y^\star\,|\,\mathbf{x}^\star]=$
$\underbrace{\mathbb{E}_\theta\left[\mathrm{Var}_{y^\star}[y^\star\,|\,\mathbf{x}^\star,\theta]\right]}_{\text{aleatoric uncertainty}}+\underbrace{\mathrm{Var}_\theta\left[\mathbb{E}_{y^\star}[y^\star\,|\,\mathbf{x}^\star,\theta]\right]}_{\text{epistemic uncertainty}}$.

$\mathbf{f}\,|\,\mathbf{X}\sim\mathcal{N}(\mathbf{\Phi}\mathbb{E}[\mathbf{w}],\mathbf{\Phi}\mathrm{Var}[\mathbf{w}]\mathbf{\Phi}^\top)=\mathcal{N}(\mathbf{0},\mathbf{K})$,
with $\mathbf{K}=\sigma_{\mathrm{p}}^2\mathbf{\Phi}\mathbf{\Phi}^\top$

Kernel-function:
$k(\mathbf{x},\mathbf{x}')\doteq\sigma_{\mathrm{p}}^2\cdot\phi(\mathbf{x})^\top\phi(\mathbf{x}')=\mathrm{Cov}[f(\mathbf{x}),f(\mathbf{x}')]$.

Linear: $k(\mathbf{x},\mathbf{x}')=l\mathbf{x}^\top\mathbf{x}'$

RBF/Gaussian: $k(\mathbf{x},\mathbf{x}')=\exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2\sigma_p^2}\right)$

Polynomial: $k(\mathbf{x},\mathbf{x}')=(1+\mathbf{x}^\top\mathbf{x}')^d$
Laplacian: $k(\mathbf{x},\mathbf{x}')=\exp(-\alpha\|\mathbf{x}-\mathbf{x}'\|)$
Matern:
$\frac{2^{1-\nu}}{\Gamma(\nu)}\left(\sqrt{2\nu}\frac{\|\mathbf{x}-\mathbf{x}'\|_2}{\sigma_p}\right)^\nu K_\nu\left(\sqrt{2\nu}\frac{\|\mathbf{x}-\mathbf{x}'\|_2}{\sigma_p}\right)$

Stationary: $k(\mathbf{x},\mathbf{x}')=k(\mathbf{x}-\mathbf{x}')$

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

Properties of Kernels:
$\mathbf{K}_{AA}$ is **symmetric** and **p.s.d.**
Composition: addition, multiplication, and composition with a function f with positive coefficients in Taylor expansion. **Bochner's Theorem:** A continuous kernel on $\mathbb{R}^d$ is p.s.d iff its Fourier transform $p(\omega)$ is non-negative.

Cost: $\mathcal{O}\left(d^3+nd^2\right)$, can be
performed online with cost $\mathcal{O}\left(d^2\right)$ per iteration.

2 Filtering

Kalman filter: $X_0\sim\mathcal{N}(\mu,\mathbf{\Sigma})$
$X_{t+1}=FX_t+\varepsilon_t$, $\varepsilon_t\sim\mathcal{N}(\mathbf{0},\Sigma_x)$
$Y_t=HX_t+\eta_t$, $\eta_t\sim\mathcal{N}(\mathbf{0},\Sigma_y)$

Conditioning:
compute $p(x_t|y_{1:t})$ from observing y_t

Prediction: compute $p(x_{t+1}|y_{1:t})$
$X_{t+1}|y_{1:t+1}\sim\mathcal{N}(\mu_{t+1},\mathbf{\Sigma}_{t+1},)$,
with $\mu_{t+1}=F_t\mu_t+\mathbf{K}_{t+1}(y_{t+1}-HF\mu_t)$
and $\mathbf{\Sigma}_{t+1}=(\mathbf{I}-\mathbf{K}_{t+1}H)(F_t\mathbf{\Sigma}_tF_t^T+\mathbf{\Sigma}_x)$

Kalman gain: $\mathbf{K}_{t+1}\doteq(F_t\mathbf{\Sigma}_tF_t^T+\mathbf{\Sigma}_x)H^T(H(F_t\mathbf{\Sigma}_tF_t^T+\mathbf{\Sigma}_x)H^T+\mathbf{\Sigma}_y)^{-1}$

3 Gaussian Processes

Gaussian process = infinite set of random variables s.t. any finite number of them are jointly Gaussian.
$f\sim\mathcal{GP}(\mu,k)$ with $\mu:\mathcal{X}\to\mathbb{R}$, $k:\mathcal{X}\times\mathcal{X}\to\mathbb{R}$
$\forall A\doteq\{\mathbf{x}_1,...,\mathbf{x}_m\}\subseteq\mathcal{X}:\mathbf{f}_A\sim\mathcal{N}(\mu_A,\mathbf{K}_{AA})$

Posterior: $f\,|\,\mathcal{D}\sim\mathcal{GP}(\mu',k')$,
$\mu'(\mathbf{x})\doteq\mu(\mathbf{x})+\mathbf{k}_{\mathbf{x},A}^\top(\mathbf{K}_{AA}+\sigma_{\mathrm{n}}^2\mathbf{I})^{-1}(\mathbf{y}_A-\mu_A)$,
$k'(\mathbf{x},\mathbf{x}')\doteq k(\mathbf{x},\mathbf{x}')-\mathbf{k}_{\mathbf{x},A}^\top(\mathbf{K}_{AA}+\sigma_{\mathrm{n}}^2\mathbf{I})^{-1}\mathbf{k}_{\mathbf{x}',A}$

Cost: $\mathcal{O}\left(n^3\right)$

Maximize Marginal Likelihood:
$\hat{\theta}_{\mathrm{MLE}}\doteq\operatorname{argmax}_\theta p(y_{1:n}\,|\,\mathbf{x}_{1:n},\theta)$
$=\operatorname{argmax}_\theta\int p(y_{1:n}\,|\,\mathbf{x}_{1:n},f,\theta)p(f\,|\,\theta)df$.
Write $\mathbf{K}_{\mathbf{y},\theta}\doteq\mathbf{K}_{f,\theta}+\sigma_{\mathrm{n}}^2\mathbf{I}$, and obtain:
$\hat{\theta}_{\mathrm{MLE}}=\operatorname{argmin}_\theta\frac{1}{2}\mathbf{y}^\top\mathbf{K}_{\mathbf{y},\theta}^{-1}\mathbf{y}+\frac{1}{2}\log\det\left(\mathbf{K}_{\mathbf{y},\theta}\right)$.

$\frac{\partial}{\partial\theta_j}\log p(y\,|$
$X,\theta)=\frac{1}{2}\mathrm{tr}\left(\left(\alpha\alpha^T-\mathbf{K}_{\mathbf{y},\theta}^{-1}\right)\frac{\partial\mathbf{K}_{\mathbf{y},\theta}}{\partial\theta_j}\right),\alpha=\mathbf{K}_{\mathbf{y},\theta}^{-1}y$

Local methods:
Only condition on $\mathbf{x}'$ where $|k(\mathbf{x},\mathbf{x}')|\geq\tau$.

Kernel
Approximation: Construct a low dimensional feature map $\phi:\mathbb{R}^d\to\mathbb{R}^m$ that approximates the kernel: $k(\mathbf{x},\mathbf{x}')\approx\phi(\mathbf{x})^\top\phi(\mathbf{x}')$, then apply BLR

Random Fourier features: given k stationary,
$p(\omega)=\int_{\mathbb{R}^d}k(\xi)e^{-i\xi^\top\omega}d\xi$
$k(\mathbf{x}-\mathbf{x}')=\int_{\mathbb{R}^d}p(\omega)e^{i\omega^\top(\mathbf{x}-\mathbf{x}')}d\omega=$
$\mathbb{E}_{\omega\sim p,b\sim\mathcal{U}([0,2\pi])}[2\cos(\omega^Tx+b)\cos(\omega^Ty+b)]$.
$\phi(x)=\frac{1}{\sqrt{m}}[z_{\omega_1,b_1}(x),...,z_{\omega_m,b_m}(x)]^T$
with $z_{\omega,b}(x)=\sqrt{2}\cos(\omega^Tx+b)$

Inducing points: Use a subset k training points as inducing points and approximate the kernel matrix with a low rank approximation. **SoR:** assume 0 covariance
FITC: assume diagonal covariance
Runtime: $\mathcal{O}(nk^2)$

4 Variational Inference

Approximate $p(\theta\,|\,\mathbf{x}_{1:n},y_{1:n})$ with $q_\lambda(\theta)\in\mathcal{Q}$

Laplace Approximation:
$q(\theta)\doteq\mathcal{N}(\theta;\hat{\theta},\mathbf{\Lambda}^{-1})\propto\exp(\hat{\psi}(\theta))$, with $\hat{\theta}$ the mode and $\mathbf{\Lambda}\doteq-\mathbf{H}_\psi(\hat{\theta})=-\mathbf{H}_\theta\log p(\theta\,|\,\mathcal{D})\big|_{\theta=\hat{\theta}}$.
Inference: $p(y^\star\,|\,\mathbf{x}^\star,\mathcal{D})\approx\int p(y^\star\,|\,\mathbf{x}^\star,\theta)q_\lambda(\theta)d\theta$.

Suprise: $\mathrm{S}[u]\doteq-\log u$ (convex).
Entropy:$\mathrm{H}[p]\doteq\mathbb{E}_{x\sim p}[\mathrm{S}[p(x)]]$.
Cross-entropy: $\mathrm{H}[p\|q]\doteq\mathbb{E}_{x\sim p}[\mathrm{S}[q(x)]]$.
KL divergence: $\mathrm{KL}(p\|q)\doteq\mathrm{H}[p\|q]-\mathrm{H}[p]=\mathbb{E}_{x\sim p}\left[\log\frac{p(x)}{q(x)}\right]$.
Gaussian: $\mathrm{H}[\mathcal{N}(\mu,\mathbf{\Sigma})]=\frac{1}{2}\log\left((2\pi e)^d\det(\mathbf{\Sigma})\right)$.

$\mathrm{KL}(p\|q)\geq0$ (Gibbs); $\mathrm{KL}(p\|q)=0$ iff $p=q$ almost everywhere. $\mathcal{N}(\mu,\mathbf{\Sigma})$ has the **highest entropy** among all distributions mean μ and variance $\mathbf{\Sigma}$.
$\mathrm{KL}(\mathrm{Bern}(p)\|\mathrm{Bern}(q))=p\log\frac{p}{q}+(1-p)\log\frac{(1-p)}{(1-q)}$

Gaussian KL: $p\doteq\mathcal{N}(\mu_p,\mathbf{\Sigma}_p)$, $q\doteq\mathcal{N}(\mu_q,\mathbf{\Sigma}_q)$:
$\mathrm{KL}(p\|q)=\frac{1}{2}(\mathrm{tr}(\mathbf{\Sigma}_q^{-1}\mathbf{\Sigma}_p)$
$+(\mu_p-\mu_q)^\top\mathbf{\Sigma}_q^{-1}(\mu_p-\mu_q)\quad-d+\log\frac{\det(\mathbf{\Sigma}_q)}{\det(\mathbf{\Sigma}_p)})$.

Forward KL: $q_1^\star\doteq\operatorname{argmin}_{q\in\mathcal{Q}}\mathrm{KL}(p\|q)$;
minimize forward KL by **moment matching**
Reverse KL: $q_2^\star\doteq\operatorname{argmin}_{q\in\mathcal{Q}}\mathrm{KL}(q\|p)$.
Reverse KL tends to greedily
select the mode and underestimate the variance.

Evidence lower bound (ELBO)
$L(q,p;\mathcal{D})=p(\mathcal{D})-\mathrm{KL}(q\|p(\cdot|\mathcal{D}))$
$=\mathbb{E}_{\theta\sim q}[p(\mathcal{D}|\theta)]-\mathrm{KL}(q\|p)$

Reparametrization
trick: For $\epsilon\sim\phi$ independent of λ s.t. $\theta=\mathbf{g}(\epsilon;\lambda)$, then: $\mathbb{E}_{\theta\sim q_\lambda}[\mathbf{f}(\theta)]=\mathbb{E}_{\epsilon\sim\phi}[\mathbf{f}(\mathbf{g}(\epsilon;\lambda))]$. For ELBO: $\mathbf{\nabla}_\lambda\mathbb{E}_{\theta\sim q_\lambda}[\mathbf{f}(\theta)]=\mathbb{E}_{\epsilon\sim\phi}[\mathbf{\nabla}_\lambda\mathbf{f}(\mathbf{g}(\epsilon;\lambda))]$.
Gaussian: $q_\lambda(\theta)\doteq\mathcal{N}(\theta;\mu,\mathbf{\Sigma})$; $\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})$, set: $\theta=\mathbf{g}(\epsilon;\lambda)\doteq\mathbf{\Sigma}^{1/2}\epsilon+\mu$, then: $\phi(\epsilon)=q_\lambda(\theta)\cdot\left|\det\left(\mathbf{\Sigma}^{1/2}\right)\right|$ and $\epsilon=\mathbf{g}^{-1}(\theta;\lambda)=\mathbf{\Sigma}^{-1/2}(\theta-\mu)$

5 Markov Chains

A **Markov Chain** over $S\doteq\{0,...,n-1\}$, is a sequence $(X_t)_{t\in\mathbb{N}_0}\in S$, such that the **Markov property**: $X_{t+1}\perp X_{0:t-1}\,|\,X_t$ is satisfied.

Time-homogeneous if there exists $p(x'\,|\,x)\doteq\mathbb{P}(X_{t+1}=x'\,|\,X_t=x)$, with transition matrix $P_{ij}=p(x_j\,|\,x_i)$. Each row sums up to 1.
The state of a MC
at t is a probability distribution q_t: $q_{t+1}=q_t\mathbf{P}$.

A distribution π is **stationary** iff $\pi=\pi\mathbf{P}$.

Irreducible: $\forall x,x'\in S\exists t\in\mathbb{N}:p^{(t)}(x'\,|\,x)>0$.
Aperiodic: $\forall x\in S\exists t_0\in\mathbb{N}\forall t\geq t_0:p^{(t)}(x\,|\,x)>0$.
Ergodic: $\exists t\in\mathbb{N}:\forall x,x'\in S:p^{(t)}(x'\,|\,x)>0$.

Irreducible MC → ergodic MC use: $\mathbf{P}'=\frac{1}{2}\mathbf{P}+\frac{1}{2}\mathbf{I}$

An ergodic MC has a unique stat. dist. π (with full support) and $\lim_{t\to\infty}q_t=\pi$, independently of q_0.

Detailed balance equation:
$\forall x,x'\in S:\pi(x)p(x'\,|\,x)=\pi(x')p(x\,|\,x')\implies$ MC is **reversible** w.r.t. $\pi\implies\pi$ is **stationary**.

Ergodic theorem For an ergodic MC and a stat. dist. π as well as $f:S\to\mathbb{R}$: $\frac{1}{n}\sum_{i=1}^nf(x_i)\overset{a.s.}{\to}\sum_{x\in S}\pi(x)f(x)=\mathbb{E}_{x\sim\pi}[f(x)]$, for $n\to\infty$ where $x_i\sim X_i\,|\,x_{i-1}$.

Acceptance distribution (Metropolis-Hastings): $\mathrm{Bern}(\alpha(\mathbf{x}'\,|\,\mathbf{x}))$ where $\alpha(\mathbf{x}'\,|\,\mathbf{x})\doteq\min\left\{1,\frac{q(\mathbf{x}')r(\mathbf{x}|\mathbf{x}')}{q(\mathbf{x})r(\mathbf{x}'|\mathbf{x})}\right\}$ to decide whether to follow the proposal yields a Markov chain with stationary distribution $p(\mathbf{x})=\frac{1}{Z}q(\mathbf{x})$. **Gibbs distribution:** $p(\mathbf{x})=\frac{1}{Z}\exp(-f(\mathbf{x}))$, f is the **energy function**.
f convex $\implies p$ **log-concave**.
$\alpha(\mathbf{x}'\,|\,\mathbf{x})=\min\left\{1,\frac{r(\mathbf{x}|\mathbf{x}')}{r(\mathbf{x}'|\mathbf{x})}\exp(f(\mathbf{x})-f(\mathbf{x}'))\right\}$.
$\mathrm{S}[p(\mathbf{x})]=f(\mathbf{x})+\log Z$

MALA/LMC: Shift the proposal distribution perpendicularly to the gradient of the energy function: $r(\mathbf{x}'\mid\mathbf{x})=\mathcal{N}(\mathbf{x}';\mathbf{x}-\eta_t\mathbf{\nabla}f(\mathbf{x}),2\eta_t\mathbf{I})$.
ULA: Unadjusted Langevin Algorithm (MALA with $\alpha(\mathbf{x}'\,|\,\mathbf{x})=1$).
SGLD: approximate gradient of ULA with unbiased estimator
HMC: lift samples up to a higher dimension and use Hamiltonian dynamics to sample from the target distribution.

Diffusion: Simulate Gaussian noising process as MC and learn backward process.
$q(x_t\,|\,x_{t-1})=\mathcal{N}(x_t;\sqrt{1-\beta_t}x_{t-1},\beta_tI)$
$q(x_t\,|\,x_0)=\mathcal{N}(x_t;\sqrt{\bar{\alpha}_t}x_0,1-\bar{\alpha}_t),\bar{\alpha}_t=\prod_{j=1}^t(1-\beta_j)$
$q(x_{t-1}\,|\,x_t,x_0)=\mathcal{N}(x_{t-1};\mu_t'(x_t,x_0),\beta_t'I)$
$\mu_t'(x_t,x_0)=\frac{(1-\bar{\alpha}_{t-1})\sqrt{\alpha_t}}{1-\bar{\alpha}_t}x_t+\frac{1-\alpha_t\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_t}x_0$
$=\frac{1}{\sqrt{\alpha_t}}\left(x_t+\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\lambda(x_t,t)\right)$; $\beta_t'=\sigma_t^2$
$L_t=\frac{(1-\alpha_t)^2}{\sigma_t^2(1-\bar{\alpha}_t)\alpha_t}\|\epsilon-\epsilon_\lambda(x_t,t)\|^2$

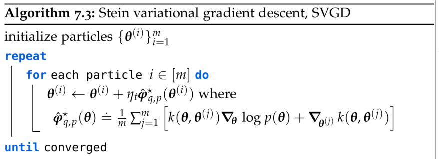
6 Bayesian Deep Learning

Bayesian neural networks: Gaussian prior on weights $\theta\sim\mathcal{N}(\mathbf{0},\sigma_{\mathrm{p}}^2\mathbf{I})$, and Gaussian likelihood to describe how well the data is described by the model:
$y\,|\,\mathbf{x},\theta\sim\mathcal{N}(f(\mathbf{x};\theta),\sigma_{\mathrm{n}}^2)$. The MAP estimate is:
$\hat{\theta}_{\mathrm{MAP}}=\operatorname{argmin}_\theta\frac{1}{2\sigma_{\mathrm{p}}^2}\|\theta\|_2^2+\frac{1}{2\sigma_{\mathrm{n}}^2}\sum_{i=1}^n(y_i-f(\mathbf{x}_i;\theta))^2$. Update rule: $\theta\leftarrow\theta(1-\frac{\eta_t}{\sigma_{\mathrm{p}}^2})+\eta_t\sum_{i=1}^n\mathbf{\nabla}\log p(y_i\,|\,\mathbf{x}_i,\theta)$

Heteroscedastic Noise:
Use a neural network with 2 outputs f_1,f_2, and define: $y\,|\,\mathbf{x},\theta\sim\mathcal{N}(\mu(\mathbf{x};\theta),\sigma^2(\mathbf{x};\theta))$ where $\mu(\mathbf{x};\theta)\doteq f_1(\mathbf{x};\theta)$ and $\sigma^2(\mathbf{x};\theta)\doteq\exp(f_2(\mathbf{x};\theta))$.

Approximate inference:
Variational inference: $p(y^\star\mid\mathbf{x}^\star,\mathcal{D})\approx\mathbb{E}_{\theta\sim q_\lambda}[p(y^\star\,|\,\mathbf{x}^\star,\theta)]\approx\frac{1}{m}\sum_{i=1}^mp(y^\star\mid\mathbf{x}^\star,\theta^{(i)})$.
MCMC/SWA: store T snapshots $\theta^{(1)},...,\theta^{(T)}$ and sample from Gaussian approximation: $\theta\sim\mathcal{N}(\mu,\mathbf{\Sigma})$ with $\mu=\frac{1}{T}\sum_{i=1}^T\theta^{(i)}$ and
$\mathbf{\Sigma}=\frac{1}{T-1}\sum_{i=1}^T(\theta^{(i)}-\mu)(\theta^{(i)}-\mu)^T$.
Probabilistic ensembles: run m models on m independently sampled datasets and average the predictions.

Dropout/Dropconnect: we also need to perform dropout/dropconnect during inference.


Algorithm 7.3: Stein variational gradient descent, SVGD

```
initialize particles $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^m$
repeat
    for each particle $i\in[m]$ do
        $\boldsymbol{\theta}^{(i)}\leftarrow\boldsymbol{\theta}^{(i)}+\eta_t\hat{\boldsymbol{\varphi}}_{q,p}^\star(\boldsymbol{\theta}^{(i)})$ where
        $\hat{\boldsymbol{\varphi}}_{q,p}^\star(\boldsymbol{\theta})\doteq\frac{1}{m}\sum_{j=1}^m\left[k(\boldsymbol{\theta},\boldsymbol{\theta}^{(j)})\nabla_{\boldsymbol{\theta}}\log p(\boldsymbol{\theta})+\nabla_{\boldsymbol{\theta}^{(j)}}k(\boldsymbol{\theta},\boldsymbol{\theta}^{(j)})\right]$
until converged
```

Expected calibration error: For m bins: $\ell_{\mathrm{ECE}}\doteq\sum_{m=1}^M\frac{|B_m|}{n}|\mathrm{freq}(B_m)-\mathrm{conf}(B_m)|$
Maximum Calibration Error: $\ell_{\mathrm{MCE}}\doteq\max_m|\mathrm{freq}(B_m)-\mathrm{conf}(B_m)|$

Histogram binning: calculate
$q_m=\mathrm{freq}(B_m)$ on validation set and return q_m when confidence is in B_m during inference.
Platt scaling: replace
logits z_i with $\sigma(az_i+b)$ and find the optimal a,b
Temperature
scaling: Platt scaling with $b=0$ and $a=\frac{1}{T}$.

7 Active Learning

$\mathrm{H}[\mathbf{X}\,|\,\mathbf{Y}]\doteq\mathbb{E}_{\mathbf{y}\sim p}[\mathrm{H}[\mathbf{X}\,|\,\mathbf{Y}=\mathbf{y}]]$
$=\mathbb{E}_{\mathbf{x},\mathbf{y}\sim p}[-\log p(\mathbf{x}\,|\,\mathbf{y})]$
$\mathrm{H}[\mathbf{X},\mathbf{Y}]\doteq\mathbb{E}_{\mathbf{x},\mathbf{y}\sim p}[-\log p(\mathbf{x},\mathbf{y})]$
$\mathrm{H}[\mathbf{X},\mathbf{Y}]=\mathrm{H}[\mathbf{Y}]+\mathrm{H}[\mathbf{X}\,|\,\mathbf{Y}]=\mathrm{H}[\mathbf{X}]+\mathrm{H}[\mathbf{Y}\,|\,\mathbf{X}]$
$\mathrm{H}[\mathbf{X}\,|\,\mathbf{Y}]=\mathrm{H}[\mathbf{Y}\,|\,\mathbf{X}]+\mathrm{H}[\mathbf{X}]-\mathrm{H}[\mathbf{Y}]$ (Bayes Rule)
$\mathrm{H}[\mathbf{X}\,|\,\mathbf{Y}]\leq\mathrm{H}[\mathbf{X}]$ (Information never hurts)
$\mathrm{I}(\mathbf{X};\mathbf{Y})\doteq\mathrm{H}[\mathbf{X}]+\mathrm{H}[\mathbf{Y}]-\mathrm{H}[\mathbf{X},\mathbf{Y}]=\mathrm{KL}(p(x,y)\|p(x)p(y))$
$\mathrm{I}(\mathbf{X};\mathbf{Y}\mid\mathbf{Z})=\mathrm{H}[\mathbf{X}\,|\,\mathbf{Z}]-\mathrm{H}[\mathbf{X}\,|\,\mathbf{Y},\mathbf{Z}]=\mathrm{KL}(p(x,y|z)\|p(x|z)p(y|z))$.
$\mathrm{I}(\mathbf{X};\mathbf{Y};\mathbf{Z})=\mathrm{I}(\mathbf{X};\mathbf{Z})+\mathrm{I}(\mathbf{X};\mathbf{Y}\mid\mathbf{Z})$

MI of dependent
Gaussians: given $X\sim\mathcal{N}(\mu,\mathbf{\Sigma})$ and $Y=X+\varepsilon$ with $\varepsilon\sim\mathcal{N}(\mathbf{0},\sigma^2\mathbf{I})$: $\mathrm{I}(X;Y)=\frac{1}{2}\log|\sigma^{-2}\Sigma+I|$
Given a (discrete) function $F:\mathcal{P}(\mathcal{X})\to\mathbb{R}$:
Marginal gain: $\Delta_F(\mathbf{x}\,|\,A)\doteq F(A\cup\{\mathbf{x}\})-F(A)$
Submodular:
$\forall\mathbf{x}\in\mathcal{X}\forall A\subseteq B\subseteq\mathcal{X}:\Delta_F(\mathbf{x}\,|\,A)\geq\Delta_F(\mathbf{x}\,|\,B)$.
Monotone: $\forall A\subseteq B:F(A)\leq F(B)$.
I is **monotone submodular**.
Uncertainty
sampling: $\mathbf{x}_{t+1}\doteq\operatorname{argmax}_{\mathbf{x}\in\mathcal{X}}\Delta_I(\mathbf{x}\,|\,S_t)$
$=\operatorname{argmax}_{\mathbf{x}\in\mathcal{X}}\mathrm{I}(f_{\mathbf{x}};y_{\mathbf{x}}\,|\,\mathbf{y}_{S_t})$
$=\operatorname{argmax}_{\mathbf{x}\in\mathcal{X}}\log\left(1+\frac{\sigma_t^2(\mathbf{x})}{\sigma_{\mathrm{n}}(\mathbf{x})^2}\right)$ **Greedy** maximization of I is a $(1-1/e)$-approximation of the optimum. Does not work with heteroscedastic noise: fails to distinguish between sources of uncertainty.

Bayesian active learning by disagreement (BALD): $\mathbf{x}_{t+1}\doteq\operatorname{argmax}_{\mathbf{x}\in\mathcal{X}}\mathrm{I}(\theta;y_{\mathbf{x}}\,|\,\mathbf{x}_{1:t},y_{1:t})=\operatorname{argmax}_{\mathbf{x}\in\mathcal{X}}\mathrm{H}[y_{\mathbf{x}}\,|\,\mathbf{x}_{1:t},y_{1:t}]-\mathbb{E}_{\theta|\mathbf{x}_{1:t},y_{1:t}}\mathrm{H}[y_{\mathbf{x}}\,|\,\theta]$

Transductive
learning: $x_{t+1}=\operatorname{argmax}_{x\in\mathcal{X}}\mathrm{I}(f_x^\star;y_x\,|\,\mathbf{x}_{1:t},y_{1:t})$

8 Bayesian Optimization

MAB: We are given a
set of k actions, and want to maximize reward.

The **Regret** for a time horizon T associated with choices $\{\mathbf{x}_t\}_{t=1}^T$ is defined as: $R_T\doteq\sum_{t=1}^T\underbrace{\left(\max_{\mathbf{x}}f^\star(\mathbf{x})-f^\star(\mathbf{x}_t)\right)}_{\textit{instantaneous regret}}$.

Goal: **sublinear regret:** $\lim_{T\to\infty}\frac{R_T}{T}=0$.
Acquisition function used to greedily pick the next point to sample based on the current model
Well-calibrated confidence intervals:
with probability $\geq1-\delta$: $\forall x\in\mathcal{X}:f^\star(x)\in\mathcal{C}_t(x)=[\mu_t(x)-\beta_t\sigma_t(x),\mu_t(x)+\beta_t\sigma_t(x)]$.
UCB: $\mathbf{x}_{t+1}\doteq\operatorname{argmax}_{\mathbf{x}\in\mathcal{X}}\mu_t(\mathbf{x})+\beta_{t+1}\sigma_t(\mathbf{x})$
Choosing $\beta_t(\delta)\in\mathcal{O}\left(\sqrt{\log(|\mathcal{X}|t/\delta)}\right)$

appropriately we get: $R_T = \mathcal{O}(\sqrt{T\gamma_T})$, where $\gamma_T \doteq \max_{\substack{S\subseteq\mathcal{X}\\|S|=T}} \mathrm{I}(\mathbf{f}_S;\mathbf{y}_S) = \max_{\substack{S\subseteq\mathcal{X}\\|S|=T}} \frac{1}{2}\log\det\left(\mathbf{I}+\sigma_n^{-2}\mathbf{K}_{SS}\right)$, is the maximum information gain after T rounds.

Information gain of some kernels:
Linear: $\gamma_T = \mathcal{O}(d\log T)$
Gaussian: $\gamma_T = \mathcal{O}\left((\log T)^{d+1}\right)$

Improvement: $I_t(x) \doteq \max\{f(x)-\hat{f}_t, 0\}$
PI: $x_{t+1} = \operatorname{argmax}_{x\in\mathcal{X}} \mathbb{P}(I_t(x)>0)$
EI: $x_{t+1} = \operatorname{argmax}_{x\in\mathcal{X}} \mathbb{E}[I_t(x)]$
Thompson Sampling: sample $\tilde{f}_{t+1} \sim p(\cdot \mid \mathbf{x}_{1:t}, y_{1:t})$ and select $\mathbf{x}_{t+1} \doteq \operatorname{argmax}_{\mathbf{x}\in\mathcal{X}} \tilde{f}_{t+1}(\mathbf{x})$.
Information ratio: $\Psi_t(x) \doteq \frac{\Delta(x)^2}{I_t(x)}$, with $\Delta(x) \doteq \max_{x'} f^\star(x') - f^\star(x)$
IDS: $x_{t+1} = \operatorname{argmin}_{x\in\mathcal{X}} \left\{\hat{\Psi}_t(x) = \frac{\hat{\Delta}_t(x)^2}{I_t(x)}\right\}$
with $\hat{\Delta}_t(x) \doteq \max_{x'} u_t(x') - l_t(x)$

9 Markov Decision Processes

A **(finite) Markov decision process** is specified by a (finite) set of **states** $X \doteq \{1, \dots, n\}$; a (finite) set of **actions** $A \doteq \{1, \dots, m\}$; **transition probabilities** $p(x' \mid x,a) \doteq \mathbb{P}(X_{t+1} = x' \mid X_t = x, A_t = a)$; a **reward function** $r : X \times A \to \mathbb{R}$ which maps the current state x and an action a to some **reward**.

r induces a sequence of rewards: $R_t \doteq r(X_t, A_t)$.

A **policy** is a function that maps each state $x \in X$ to a probability distribution over the actions. That is, for any $t > 0$: $\pi(a \mid x) \doteq \mathbb{P}(A_t = a \mid X_t = x)$.

A policy induces a MC $(X_t^\pi)_{t\in\mathbb{N}_0}$
Discounted payoff: $G_t \doteq \sum_{m=0}^\infty \gamma^m R_{t+m}$, $\gamma \in [0,1)$ is the **discount factor**.

State value function: $v_t^\pi(x) \doteq \mathbb{E}_\pi[G_t \mid X_t = x]$
State-action value function (Q-function): $q_t^\pi(x,a) \doteq \mathbb{E}_\pi[G_t \mid X_t = x, A_t = a]$
Bellman Expectation Equation:
$v^\pi(x) = r(x,\pi(x)) + \gamma \mathbb{E}_{x'\mid x,\pi(x)}[v^\pi(x')]$
For stochastic policies: $v^\pi(x) = \mathbb{E}_{a\sim\pi(x)}[q^\pi(x,a)]$

Can be used to find v^π given policy π, by solving linear system of equations in $\mathcal{O}(n^3)$.
Fixed point iteration: $\mathbf{B}^\pi \mathbf{v} \doteq \mathbf{r}^\pi + \gamma \mathbf{P}^\pi \mathbf{v}$.
$\mathbf{B}^\pi$ is contraction with contraction factor $\gamma < 1 \implies$ unique optimal value function $v^\star$.

Greedy policy: $\pi(x) \doteq \operatorname{argmax}_{a\in A} q_t^\pi(x,a)$
Bellman's Theorem: A policy $\pi^\star$ is optimal iff it is greedy w.r.t. its own value function.

Bellman optimality equations:
$v^\star(x) = \max_{a\in A} q^\star(x,a)$
$q^\star(x,a) = r(x,a) + \gamma \mathbb{E}_{x'\mid x,a}[\max_{a'\in A} q^\star(x',a')]$

Algorithm 10.17: Policy iteration

```
initialize $\pi$ (arbitrarily)
repeat
    compute $v^\pi$
    compute $\pi_{v^\pi}$
    $\pi \leftarrow \pi_{v^\pi}$
until converged
```

For finite MDPs, policy iteration converges to an optimal policy (monotonic improvement).

Algorithm 10.20: Value iteration

```
initialize $v(x) \leftarrow \max_{a\in A} r(x,a)$ for each $x \in X$
for $t = 1$ to $\infty$ do
    $v(x) \leftarrow (\boldsymbol{B}^\star \boldsymbol{v})(x) = \max_{a\in A} q(x,a)$ for each $x \in X$
choose $\pi_v$
```

Value iteration to an ϵ-optimal policy in polynomial time, as $v^\star$ and $q^\star$ are a fixed-points of the Bellman update $\mathbf{B}^\star$. v_t corresponds the the optimal value function assuming only t steps are ever taken.

A **Partially observable Markov decision process (POMDP)** is a Markov process with **hidden states**, a set of supplementary **observations** Y, and **observation probabilities** $o(y \mid x) \doteq \mathbb{P}(Y_t = y \mid X_t = x)$. Given a POMDP, the corresponding **Belief-state Markov decision process** is a Markov decision process specified by the **belief space** $\mathcal{B} \doteq \Delta^X$; the set of **actions** A; **transition probabilities** $\tau(b' \mid b,a) \doteq \mathbb{P}(B_{t+1} = b' \mid B_t = b, A_t = a)$; and **rewards** $\rho(b,a) \doteq \mathbb{E}_{x\sim b}[r(x,a)] = \sum_{x\in X} b(x) r(x,a)$.
$b_{t+1}(x) = \mathbb{P}(X_{t+1} = x \mid y_{1:t+1}, a_{1:t})$ is deterministic.
$\mathbb{P}(y_{t+1} \mid b_t, a_t)$
$= \mathbb{E}_{x\sim b_t}\left[\mathbb{E}_{x'\mid x,a_t}[\mathbb{P}(y_{t+1} \mid X_{t+1} = x')]\right]$
$= \sum_{x\in X} b_t(x) \sum_{x'\in X} p(x' \mid x,a_t) \cdot o(y_{t+1} \mid x')$.

10 Tabular Reinforcement Learning

The reinforcement learning problem: probabilistic planning in unknown environments. A trajectory τ is a sequence: $\tau \doteq (\tau_0, \tau_1, \tau_2, \dots)$, with $\tau_i \doteq (x_i, a_i, r_i, x_{i+1})$.

On-policy(On): Agent chooses policy.
Off-policy(Off): No choice of policy. More sample efficient, less stable.
Model-based(MB): Learn underlying MDP. More sample efficient, allows for planning and transfers well to new tasks.
Model-free(MF): Learn value function directly. Simpler, doesn't suffer from model bias, tends to perform better.
Value estimation(VE): Learn value function given policy.
Control(C): Determine optimal policy

A sequence $(\pi_t)_{t\in\mathbb{N}_0}$ of policies is **greedy in the limit of infinite exploration (GLIE)** if: All pairs (x,a) are visited infinitely often. $\lim_{t\to\infty} \pi_t(a \mid x) = \mathbf{1}\{a = \operatorname{argmax}_{a'\in A} Q_t^\star(x,a')\}$, where $Q_t^\star$ is the optimal action-value function for the estimated MDP at time t.

Model-based MLE: $\hat{p}(x' \mid x,a) = \frac{N(x'|x,a)}{N(a|x)}$
$\hat{r}(x,a) = \frac{1}{N(a|x)} \sum_{t=0, x_t=x, a_t=a}^\infty r_t$
ε-greedy: With probability ε, choose a random action, otherwise choose the action with the highest value. $(\varepsilon_t)_{t\in\mathbb{N}_0}$ satisfies RM $\implies$ GLIE $\implies$ convergence. **softmax exploration**: $\pi(a \mid x) \propto \exp(Q(x,a)/\lambda)$ with temperature $\lambda > 0$.

Algorithm 11.6: $R_{\max}$ algorithm

```
add the fairy-tale state $x^\star$ to the Markov decision process
set $\hat{r}(x,a) = R_{\max}$ for all $x \in X$ and $a \in A$
set $\hat{p}(x^\star \mid x,a) = 1$ for all $x \in X$ and $a \in A$
compute the optimal policy $\hat{\pi}$ for $\hat{r}$ and $\hat{p}$
for $t = 0$ to $\infty$ do
    execute policy $\hat{\pi}$ (for some number of steps)
    for each visited state-action pair $(x,a)$, update $\hat{r}(x,a)$
    estimate transition probabilities $\hat{p}(x' \mid x,a)$
    after observing "enough" transitions and rewards, recompute the
      optimal policy $\hat{\pi}$ according the current model $\hat{p}$ and $\hat{r}$.
```

With probability at least $1-\delta$, $R_{\max}$ reaches an ϵ-optimal policy in a number of steps that is polynomial in $|X|$, $|A|$, T, $1/\epsilon$, $1/\delta$, and $R_{\max}$.

TD learning: On/MF/VE
$V(x) \leftarrow V(x) + \alpha_t(r + \gamma V(x') - V(x))$

SARSA: On/MF/VE
$Q(x,a) \leftarrow Q(x,a) + \alpha_t(r + \gamma Q(x',a') - Q(x,a))$
Off-policy version (expected SARSA): $Q(x,a) \leftarrow Q(x,a) + \alpha_t\left(r + \gamma \mathbb{E}_{a'\sim\pi(x')}[Q(x',a')] - Q(x,a)\right)$

Q learning: Off/MF/C $Q(x,a) \leftarrow (1-\alpha_t)Q(x,a) + \alpha_t\left(r + \gamma \max_{a'\in A} Q(x',a')\right)$

$(\alpha_t)_{t\in\mathbb{N}_0}$ satisfies RM + GLIE $\implies$ convergence for TD, SARSA and Q learning. All 3 methods can be initialized arbitrarily.

Algorithm 11.14: Optimistic Q-learning

```
initialize $Q^\star(x,a) = V_{\max} \prod_{t=1}^{T_{\text{init}}} (1-\alpha_t)^{-1}$
for $t = 0$ to $\infty$ do
    pick action $a_t = \arg\max_{a\in A} Q^\star(x,a)$ and observe the transition
      $(x, a_t, r, x')$
    $Q^\star(x,a_t) \leftarrow (1-\alpha_t)Q^\star(x,a_t) + \alpha_t(r + \gamma \max_{a'\in A} Q^\star(x',a'))$
      // (11.27)
```

With probability at least $1-\delta$, Q learning converges to an ϵ-optimal policy in a number of steps that is polynomial in $|X|$, $|A|$, T, $1/\epsilon$, $1/\delta$, and $R_{\max}$.

11 Model-free Reinforcement Learning

Parametric value function approximation: learn approximation $V(\mathbf{x};\theta)$ or $Q(\mathbf{x},\mathbf{a};\theta)$ parametrized by θ. Can view TD-learning as SGD on the squared loss $\ell(\theta;x,r,x') \doteq \frac{1}{2}\left(r + \gamma\theta^{\text{old}}(x') - \theta(x)\right)^2$.

Q-learning with function approximation: scaling to large state spaces (Off/MF/C)
Bellman error:
$\delta_{\text{B}} \doteq r + \gamma \max_{\mathbf{a}'\in\mathcal{A}} Q^\star(\mathbf{x}',\mathbf{a}';\theta^{\text{old}}) - Q^\star(\mathbf{x},\mathbf{a};\theta)$.
Update:$\theta \leftarrow \theta + \alpha_t \delta_{\text{B}} \boldsymbol{\nabla}_\theta Q^\star(\mathbf{x},\mathbf{a})$ with $\theta^{\text{old}} = \theta$ being treated as **constant** w.r.t. θ.

DQN: stabilizing targets
Train 2 separate networks: target network and online network. $\ell_{\text{DQN}} \doteq \frac{1}{2}\left(r + \gamma \max_{\mathbf{a}'\in\mathcal{A}} Q^\star(\mathbf{x}',\mathbf{a}';\theta^{\text{old}}) - Q^\star(\mathbf{x},\mathbf{a};\theta)\right)^2$.
Update target network with hard updates or Polyak averaging: $\theta^{\text{old}} \leftarrow \alpha\theta + (1-\alpha)\theta^{\text{old}}$
DDQN: avoiding maximization bias
Choose maximum action from online network and evaluate it with target network.

Policy optimization/Policy gradient methods: directly optimize policy π_φ instead of value function.
Trajectory distribution:
$\Pi_\varphi(\tau) \doteq p(x_0) \prod_{t=0}^{T-1} p(x_{t+1} \mid x_t, a_t) \pi_\varphi(a_t \mid x_t)$

Policy value function: $j(\varphi) = j(\pi_\varphi) \doteq \mathbb{E}_{\pi_\varphi}[G_0] = \mathbb{E}_{\pi_\varphi}\left[\sum_{t=0}^\infty \gamma^t R_t\right]$
Bounded variant: $j_T(\pi) \doteq \mathbb{E}_\pi[G_{0:T}]$

Score function trick:
$\boldsymbol{\nabla}_\varphi \mathbb{E}_{\tau\sim\Pi_\varphi}[G_0] = \mathbb{E}_{\tau\sim\Pi_\varphi}[G_0 \boldsymbol{\nabla}_\varphi \log\Pi_\varphi(\tau)] = \mathbb{E}_{\tau\sim\Pi_\varphi}\left[G_0 \sum_{t=0}^{T-1} \boldsymbol{\nabla}_\varphi \log\pi_\varphi(a_t|x_t)\right]$.
Has **high variance** unlike the reparametrization trick.
Baseline:
$\mathbb{E}_{\tau\sim\Pi_\varphi}[G_0 \boldsymbol{\nabla}_\varphi \log\Pi_\varphi(\tau)] = \mathbb{E}_{\tau\sim\Pi_\varphi}\left[\sum_{t=0}^{T-1} (G_0 - b(\tau_{0:t-1})) \boldsymbol{\nabla}_\varphi \log\pi_\varphi(a_t|x_t)\right]$

REINFORCE (On/MF/C): Select baseline $b_t = g_{0:t-1}$: $\boldsymbol{\nabla}_\varphi\, j_T(\varphi) = \mathbb{E}_{\tau\sim\Pi_\varphi}\left[\sum_{t=0}^{T-1} \gamma^t g_{t:T} \boldsymbol{\nabla}_\varphi \log\pi_\varphi(a_t|x_t)\right]$

Policy gradient theorem:
$\boldsymbol{\nabla} j(\varphi) = \sum_{t=0}^\infty \mathbb{E}_{x_t,a_t}\left[\gamma^t q^{\pi_\varphi}(x_t,a_t) \nabla_\varphi \log\pi_\varphi(a_t|x_t)\right]$
$\propto \mathbb{E}_{x\sim\rho_\varphi^\infty} \mathbb{E}_{a\sim\pi_\varphi(a|x)}[q^{\pi_\varphi}(x,a) \nabla_\varphi \log\pi_\varphi(a|x)]$

Actor-Critic methods: scaling to large action spaces
Parameterized policy $\pi(\mathbf{a}|\mathbf{x};\varphi) \doteq \pi_\varphi$ (actor)
Value function approximation $q^{\pi_\varphi}(\mathbf{x}, \mathbf{a}) \approx Q^{\pi_\varphi}(\mathbf{x},\mathbf{a};\theta)$ (critic).

On-policy AC: learn critic through SARSA and actor through policy gradient methods

Algorithm 12.10: Online actor-critic

```
initialize parameters $\boldsymbol{\varphi}$ and $\boldsymbol{\theta}$
repeat
    use $\pi_{\boldsymbol{\varphi}}$ to obtain transition $(\boldsymbol{x}, \boldsymbol{a}, r, \boldsymbol{x}')$
    $\delta = r + \gamma Q(\boldsymbol{x}', \pi_{\boldsymbol{\varphi}}(\boldsymbol{x}'); \boldsymbol{\theta}) - Q(\boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta})$
    // actor update
    $\boldsymbol{\varphi} \leftarrow \boldsymbol{\varphi} + \eta\gamma^t Q(\boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta}) \boldsymbol{\nabla}_{\boldsymbol{\varphi}} \log\pi_{\boldsymbol{\varphi}}(\boldsymbol{a} \mid \boldsymbol{x})$
    // critic update
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \eta\delta \boldsymbol{\nabla}_{\boldsymbol{\theta}} Q(\boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta})$
until converged
```

Advantage: $a^\pi(\mathbf{x},\mathbf{a}) \doteq q^\pi(\mathbf{x},\mathbf{a}) - v^\pi(\mathbf{x})$
π is optimal $\iff \forall \mathbf{x}\in\mathcal{X}, \mathbf{a}\in\mathcal{A} : a^\pi(\mathbf{x},\mathbf{a}) \leq 0$
Advantage actor-critic (A2C):
replace Q with advantage function A (predicting sign is easier than predicting absolute quantity).
Advantage isn't directly parametrized: we parametrize V^π and approximate Q with $\sum_{t=k}^T \gamma^{t-k} r_t + \gamma^{T-k} V^\pi(x_{T+1})$.
When compared to REINFORCE, actor-critic methods have **lower variance** and **higher bias**.

TRPO: improving sample efficiency in on-policy AC (On/MF/C)
$\varphi_{k+1} \leftarrow \operatorname{argmax}_\varphi J(\varphi)$ subject to $\mathrm{KL}(\pi_{\varphi_k}(\cdot|x) \| \pi_\varphi(\cdot|x)) \leq \delta$
$J(\varphi) \doteq \mathbb{E}_{x\sim\rho_{\varphi_k}^\infty, a\sim\pi_{\varphi_k}(\cdot|x)}[w_k(\varphi;x,a) A^{\pi_{\varphi_k}}(x,a)]$
$w_k(\varphi;x,a) \doteq \frac{\pi_\varphi(a|x)}{\pi_{\varphi_k}(a|x)}$ are the importance sampling weights.
PPO: uncontrained objective $\underset{\varphi}{\operatorname{argmax}} J(\varphi) - \lambda \mathbb{E}_{x\sim\rho_{\varphi_k}^\infty} \mathrm{KL}(\pi_{\varphi_k}(\cdot|x) \| \pi_\varphi(\cdot|x))$

GRPO: improving compute efficiency
PPO with heuristic approximation of advantage
$\hat{A}_{t,i} = \frac{g_{t:T}^{(i)} - \text{mean}(g_{t:T})}{\text{std}(g_{t:T})}$

Off-policy AC: Parametrize maximum over actions with π_φ.
Objective (**deterministic policy gradients/hill-climbing**): $\varphi^\star = \operatorname{argmax}_\varphi J_\mu(\varphi) = \operatorname{argmax}_\varphi \mathbb{E}_{x\sim\mu}[Q^\star(x,\pi_\varphi(x);\theta)]$
Learn critic through **bootstrapped** Q-learning
The **exploration distribution** $\mu(x)$ is typically selected as uniform sampling from replay buffer. Exploration can be achieved through **Gaussian dithering** as in **DDPG**.
TD3 = DDPG with 2 critic networks for evaluating policy and calculating maximum.

For randomized policies like **SVG**, we get: $J_\mu(\varphi) = \mathbb{E}_{x\sim\mu} \mathbb{E}_{a\sim\pi_\varphi(\cdot|x)}[Q^\star(x,a;\theta)]$
Reparametrize to get the gradient: $\boldsymbol{\nabla}_\varphi J_\mu(\varphi) = \mathbb{E}_{x\sim\mu} \mathbb{E}_{\varepsilon\sim\phi}\left[D_a Q^\star(x,a)\big|_{a=g(\varepsilon;\varphi)} D_\varphi g(\varepsilon;\varphi)\right]$
MERL: encourage exploration through new objective: $j_\lambda(\varphi) = j(\varphi) + \lambda \mathrm{H}[\Pi_\varphi] = \sum_{t=0}^\infty \mathbb{E}_{(x_t,a_t)\sim\Pi_\varphi}[r(x_t,a_t) + \lambda \mathrm{H}[\pi_\varphi(\cdot|x_t)]]$
Assuming HMM defined by $p(\mathcal{O}_t|x_t,a_t) \propto \exp\left(\frac{1}{\lambda} r(x_t,a_t)\right)$, we get:
$\Pi_\star(\tau) = p(\tau|\mathcal{O}_{1:T}) \propto \left[p(x_1) \prod_{t=1}^{T-1} p(x_{t+1}|x_t,a_t)\right] \exp\left(\frac{1}{\lambda} \sum_{t=1}^T r(x_t,a_t)\right)$
The objective $\mathrm{KL}(\Pi_\star \| \Pi_\varphi)$ is equivalent to the MERL objective.
Soft-Actor-Critic and **MAP optimization** are off-policy actor-critic methods with entropy regularization
Soft-value function: $q^\star(x,a) = \frac{1}{\lambda} r(x,a) + \mathbb{E}_{x'\sim x,a}\left[\log\int_{\mathcal{A}} \exp(q^\star(x',a'))da'\right]$
Changes the value function, not just the objective

Finetuning LLMs:
Bradley-Terry model: $p(y_A \succ y_B|x,r) = \sigma(r(y_A|x) - r(y_B|x))$
RLHF: 1. calculate reward with MLE $(\theta = \operatorname{argmax}_\theta p(\mathcal{D}|r_\theta))$ 2. calculate policy with PPO
Optimal policy: $\Pi_\star \propto \Pi_{\text{init}} \exp\left(\frac{1}{\lambda} r(y|x)\right)$
DPO: optimize over $r_\varphi(y|x) = \lambda\log\frac{\Pi_\varphi}{\Pi_{\text{init}}} + \text{const.}$

12 Model-based Reinforcement Learning

Strict generalization of model-free RL.

Algorithm 13.1: Model-based reinforcement learning (outline)

```
start with an initial policy $\pi$ and no (or some) initial data $\mathcal{D}$
for several episodes do
    roll out policy $\pi$ to collect data
    learn a model of the dynamics $f$ and rewards $r$ from data
    plan a new policy $\pi$ based on the estimated models
```

Assuming the dynamics are known and deterministic $(x_{t+1} = f(x_t,a_t))$:
MPC: plan over finite time horizon H
To solve the problem of **sparse rewards**, add $\gamma^H V(x_H)$ to the reward.
Target shooting:
generate random sequences of actions and choose the best one. (primitive **tree search** method)
Trajectory sampling = MPC with stochastic dynamics
closed-loop control: planning done online
open-loop control: policy precomputed offline
$\implies$ apply model-free techniques to the new objective
Learning dynamics: $x_{t+1} \sim f(x_t,a_t;\psi)$
(approximate) greedy opimization:
PILCO for GPs, **PETS** for neural networks

Thompson sampling: sample f from posterior and maximize
Optimistic exploration: optimize over set $\mathcal{M}(\mathcal{D})$ of "plausible" models
H-UCRL:
$\pi_{t+1} = \operatorname{argmax}_\pi \max_{\eta(\cdot)\in[-1,1]^d} J_H(\pi;\hat{f}_t)$ with $\hat{f}_t(x,a) = \mu_t(x,a) + \beta_t \eta(x,a) \sigma_t(x,a)$

Constrained optimization: $\max_\pi J_\mu(\pi;f)$ subject to $J_\mu^c(\pi;f) = \mathbb{E}_{x\sim\mu, x_{1:\infty}\sim\pi,f}\left[\sum_{t=0}^\infty \gamma^t c(x_t)\right] \leq \delta$
Assuming a family of plausible models $\mathcal{M}(\mathcal{D})$, we can be **optimistic** w.r.t. rewards and **pessimistic** w.r.t. constraints.
$\max_\pi \max_{f\in\mathcal{M}(\mathcal{D})} J_\mu(\pi;f)$
subject to $\max_{f\in\mathcal{M}(\mathcal{D})} J_\mu^c(\pi;f) \leq \delta$

By Leo Schmidt-Traub
– based off of Nils Jensen's notes.