Probablistic AI

1 Fundamentals

Useful PDFs:

Normal: $\frac{\exp(-\frac12(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu))}{\sqrt{(2\pi)^k\det(\Sigma)}}$

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

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

Properties of Expectation:
$\mathbb{E}[\mathbf{AX}+\mathbf{b}]=\mathbf{A}\mathbb{E}[\mathbf{X}]+\mathbf{b}$; $\mathbb{E}[\mathbf{X}+\mathbf{Y}]=\mathbb{E}[\mathbf{X}]+\mathbb{E}[\mathbf{Y}]$
$\mathbb{E}[\mathbf{XY}^\top]=\mathbb{E}[\mathbf{X}]\cdot\mathbb{E}[\mathbf{Y}]^\top$ (if independant)
$\mathbb{E}[\mathbf{g}(\mathbf{X})]=\int_{\mathbf{X}(\Omega)}\mathbf{g}(\mathbf{x})\cdot p(\mathbf{x})dx$ (if $\mathbf{g}$ nice and $\mathbf{X}$ cont.)
$\mathbb{E}_\mathbf{Y}[\mathbb{E}_\mathbf{X}[\mathbf{X}\mid\mathbf{Y}]]=\mathbb{E}[\mathbf{X}]$ (**Tower rule**)

Given two random vectors $\mathbf{X}$ in $\mathbb{R}^n$ and $\mathbf{Y}$ in $\mathbb{R}^m$, their **covariance** is defined as $\text{Cov}[\mathbf{X},\mathbf{Y}]=\mathbb{E}[(\mathbf{X}-\mathbb{E}[\mathbf{X}])(\mathbf{Y}-\mathbb{E}[\mathbf{Y}])^\top]$ We have: $\text{Cov}[\mathbf{AX}+\mathbf{c},\mathbf{BY}+\mathbf{d}]=\mathbf{A}\text{Cov}[\mathbf{X},\mathbf{Y}]\mathbf{B}^\top$
Uncorrelated if and only if $\text{Cov}[\mathbf{X},\mathbf{Y}]=\mathbf{0}$. The *correlation* of the random vectors $\mathbf{X}$ and $\mathbf{Y}$ is a normalized covariance:
$\text{Cor}[\mathbf{X},\mathbf{Y}](i,j)\doteq\frac{\text{Cov}[X_i,Y_j]}{\sqrt{\text{Var}[X_i]\text{Var}[Y_j]}}\in[-1,1]$

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

Properties of variance:
$\text{Var}[\mathbf{AX}+\mathbf{b}]=\mathbf{A}\text{Var}[\mathbf{X}]\mathbf{A}^\top$
$\text{Var}[\mathbf{X}+\mathbf{Y}]=\text{Var}[\mathbf{X}]+\text{Var}[\mathbf{Y}]+2\text{Cov}[\mathbf{X},\mathbf{Y}]$
$\text{Var}[\mathbf{X}+\mathbf{Y}]=\text{Var}[\mathbf{X}]+\text{Var}[\mathbf{Y}]$ (if $\mathbf{X}$, $\mathbf{Y}$ independant) $\text{Var}[\mathbf{X}]=\mathbb{E}_\mathbf{Y}[\text{Var}_\mathbf{X}[\mathbf{X}\mid\mathbf{Y}]]+\text{Var}_\mathbf{Y}[\mathbb{E}_\mathbf{X}[\mathbf{X}\mid\mathbf{Y}]]$ (**Law of total variance, LOTV**)

Change of variables formula Let $\mathbf{g}$ be differentiable and invertible. Then for $\mathbf{Y}=\mathbf{g}(\mathbf{X})$ we have $p_\mathbf{Y}(\mathbf{y})=p_\mathbf{X}(\mathbf{g}^{-1}(\mathbf{y}))\cdot|\det(\mathbf{D}\mathbf{g}^{-1}(\mathbf{y}))|$ where $\mathbf{Dg}^{-1}(\mathbf{y})$ is the Jacobian of $\mathbf{g}^{-1}$ evaluated at $\mathbf{y}$.

Bayes' rule: $p(\mathbf{x}\mid\mathbf{y})=\frac{p(\mathbf{y}\mid\mathbf{x})\cdot p(\mathbf{x})}{p(\mathbf{y})}$
Posterior $p(\mathbf{x}\mid\mathbf{y})$: updated belief about $\mathbf{x}$ after observing $\mathbf{y}$.
Prior $p(\mathbf{x})$: initial belief about $\mathbf{x}$.
Conditional likelihood $p(\mathbf{y}\mid\mathbf{x})$: how likely the observations $\mathbf{y}$ are under a given value $\mathbf{x}$.
Joint likelihood $p(\mathbf{x},\mathbf{y})=p(\mathbf{y}\mid\mathbf{x})p(\mathbf{x})$
Marginal likelihood $p(\mathbf{y})$: how likely the observations $\mathbf{y}$ are across all values of $\mathbf{x}$.
Marginal likelihood $p(\mathbf{y})=\int_{\mathbf{X}(\Omega)}p(\mathbf{y}\mid\mathbf{x})\cdot p(\mathbf{x})d\mathbf{x}$.

If prior $p(\mathbf{x})$ and posterior $p(\mathbf{x}\mid\mathbf{y})$ from same family of distributions, the prior is a **conjugate prior** to the likelihood $p(\mathbf{y}\mid\mathbf{x})$. The beta distribution is a conjugate prior to a binomial likelihood.

Marginal and conditional of Normal
Let $\mathbf{X}$ be Gaussian and index sets $A,B\subseteq[n]$. For any such **marginal distribution** $\mathbf{X}_A\sim\mathcal{N}(\mu_A,\Sigma_{AA})$ and that for any such **conditional distribution:**
$\mathbf{X}_A\mid\mathbf{X}_B=\mathbf{x}_B\sim\mathcal{N}(\mu_{A|B},\Sigma_{A|B})$ where:
$\mu_{A|B}\doteq\mu_A+\Sigma_{AB}\Sigma_{BB}^{-1}(\mathbf{x}_B-\mu_B)$ and
$\Sigma_{A|B}\doteq\Sigma_{AA}-\Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA}$.

Maximum likelihood estimate (MLE): $\hat\theta_{\text{MLE}}\doteq\underset{\theta\in\Theta}{\text{argmax}}\,p(y_{1:n}\mid\mathbf{x}_{1:n},\theta)=\underset{\theta\in\Theta}{\text{argmax}}\underbrace{\sum_{i=1}^n\log p(y_i\mid\mathbf{x}_i,\theta)}_{\text{log-likelihood}}$

We will denote the **negative log-likelihood** by $\ell_{\text{nll}}(\theta;\mathcal{D}_n)$. The MLE is **consistent** and **asymptotically normal** if: $\hat\theta_{\text{MLE}}\xrightarrow{\mathbb{P}}\theta^\star$ $\hat\theta_{\text{MLE}}\xrightarrow{\mathcal{D}}\mathcal{N}(\theta^\star,\mathbf{S}_n)$ as $n\to\infty$.

The **maximum a posteriori estimate (or MAP estimate):**
$\hat\theta_{\text{MAP}}=\text{argmax}_{\theta\in\Theta}p(\theta\mid\mathbf{x}_{1:n},y_{1:n})=\text{argmin}_{\theta\in\Theta}\underbrace{-\log p(\theta)}_{\text{regularization}}+\underbrace{\ell_{\text{nll}}(\theta;\mathcal{D}_n)}_{\text{quality of fit}}$
Here, the **log-prior** $\log p(\theta)$ acts as a regularizer.

Common regularizers:
$p(\theta)=\mathcal{N}(\theta;0,\lambda\mathbf{I})\to-\log p(\theta)=\frac{\lambda}{2}\|\theta\|_2^2+\text{const}$
$p(\theta)=\text{Laplace}(\theta;\mathbf{0},\lambda)\to-\log p(\theta)=\lambda\|\theta\|_1+\text{const}$
uniform prior $\to$ const (no regularization)

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

2 Bayesian Linear Regression

Closed form solutions: $\hat{\mathbf{w}}_{\text{ls}}=(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$ $\hat{\mathbf{w}}_{\text{ridge}}=(\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$

Notable Results: $\text{Var}[\hat{\mathbf{w}}_{\text{ls}}\mid\mathbf{X}]=\sigma_n^2(\mathbf{X}^\top\mathbf{X})^{-1}$
A **Gaussian prior on the weights** $\mathbf{w}\sim\mathcal{N}(\mathbf{0},\sigma_p^2\mathbf{I})$, yields the posterior distribution: $\log p(\mathbf{w}\mid\mathbf{x}_{1:n},y_{1:n})=-\frac12[\mathbf{w}^\top\Sigma^{-1}\mathbf{w}-2\mu]+\text{const}$, with $\Sigma\doteq\left(\sigma_n^{-2}\mathbf{X}^\top\mathbf{X}+\sigma_p^{-2}\mathbf{I}\right)^{-1}$ and $\mu=\sigma_n^{-2}\Sigma\mathbf{X}^\top\mathbf{y}$. We have $\mathbf{w}\mid\mathbf{x}_{1:n},y_{1:n}\sim\mathcal{N}(\mu,\Sigma)$: *Gaussian's with known variance and linear likelihood* are self-conjugate.
MAP: $\hat{\mathbf{w}}_{\text{MAP}}=\text{argmin}_\mathbf{w}\|\mathbf{y}-\mathbf{Xw}\|_2^2+\frac{\sigma_n^2}{\sigma_p^2}\|\mathbf{w}\|_2^2$, *identical to ridge regression* with $\lambda\doteq\sigma_n^2/\sigma_p^2$.
A **Laplace prior** on the weights is equivalent to **lasso regression** with decay $\lambda\doteq\sigma_n^2/\ell$.

Bayesian inference: Distribution for a test point $\mathbf{x}^\star$ is: $y^\star\mid\mathbf{x}^\star,\mathbf{x}_{1:n},y_{1:n}\sim\mathcal{N}(\mu^\top\mathbf{x}^\star,\mathbf{x}^{\star\top}\Sigma\mathbf{x}^\star+\sigma_n^2)$.
$\text{Var}[y^\star\mid\mathbf{x}^\star]=\underbrace{\mathbb{E}_\theta[\text{Var}_{y^\star}[y^\star\mid\mathbf{x}^\star,\theta]]}_{\text{aleatoric uncertainty}}+\underbrace{\text{Var}_\theta[\mathbb{E}_{y^\star}[y^\star\mid\mathbf{x}^\star,\theta]]}_{\text{epistemic uncertainty}}$.
Aleatoric $\to$ noise in data;
Epistemic $\to$ noise in model. Applying linear regression to non linear functions: apply a non linear transformation ϕ to $\mathbf{X}$. Define $\Phi=\phi(\mathbf{X})$, so-called **Kernel**. With a gaussian prior we get: $\mathbf{f}\mid\mathbf{X}\sim\mathcal{N}(\Phi\mathbb{E}[\mathbf{w}],\Phi\text{Var}[\mathbf{w}]\Phi^\top)=\mathcal{N}(\mathbf{0},\mathbf{K})$, with $\mathbf{K}=\sigma_p^2\Phi\Phi^\top$. We define the **Kernel-function:** $k(\mathbf{x},\mathbf{x}')\doteq\sigma_p^2\cdot\phi(\mathbf{x})^\top\phi(\mathbf{x}')=\text{Cov}[f(\mathbf{x}),f(\mathbf{x}')]$.

Linear Kernel: $k(\mathbf{x},\mathbf{x}')=l\mathbf{x}^\top\mathbf{x}'$
RBF/Gaussian: $k(\mathbf{x},\mathbf{x}')=\exp-\frac{(\mathbf{x}-\mathbf{x}')^2}{2\sigma_p^2}$
Polynomial Kernel $k(\mathbf{x},\mathbf{x}')=(1+\mathbf{x}^\top\mathbf{x}')^d$
Laplacian kernel: $k(\mathbf{x},\mathbf{x}')=\exp(-\alpha\|\mathbf{x}-\mathbf{x}'\|)$

Properties of Kernels:
Symmetry: $k(\mathbf{x},\mathbf{x}')=k(\mathbf{x}',\mathbf{x})$ and $\mathbf{K}_{AA}$ is p.s.d. Kernels can be **composed** in the following ways to obtain a new kernel: addition, multiplication, positive scalar multiplication and composition with a funtion f if f is polynomial with positive coefficients or exp.

Stationary if there exists a $\tilde k$ s.t. $\tilde k(\mathbf{x}-\mathbf{x}')=k(\mathbf{x},\mathbf{x}')$, and **Isotropic** if there exists a $\tilde k$ s.t. $\tilde k(\|\mathbf{x}-\mathbf{x}'\|_2)=k(\mathbf{x},\mathbf{x}')$.

3 Gaussian Processes

A **Gaussian process** is an infinite set of random variables such that any finite number of them are jointly Gaussian. We use a set $\mathcal{X}$ to index the collection of random variables. A Gaussian process is characterized by a **mean function** $\mu:\mathcal{X}\to\mathbb{R}$ and a **covariance function** (or **kernel function**) $k:\mathcal{X}\times\mathcal{X}\to\mathbb{R}$ such that for any $A\doteq\{\mathbf{x}_1,\dots,\mathbf{x}_m\}\subseteq\mathcal{X}$, we have $\mathbf{f}_A\doteq[f_{\mathbf{x}_1}\cdots f_{\mathbf{x}_m}]^\top\sim\mathcal{N}(\mu_A,\mathbf{K}_{AA})$ We write $f\sim\mathcal{GP}(\mu,k)$. In particular, given a mean function, covariance function, and using the homoscedastic noise assumption, $y^\star\mid\mathbf{x}^\star,\mu,k\sim\mathcal{N}(\mu(\mathbf{x}^\star),k(\mathbf{x}^\star,\mathbf{x}^\star)+\sigma_n^2)$

Maximize Marginal Likelihood:
$\hat\theta_{\text{MLE}}\doteq\text{argmax}_\theta p(y_{1:n}\mid\mathbf{x}_{1:n},\theta)=\text{argmax}_\theta\int p(y_{1:n}\mid\mathbf{x}_{1:n},f,\theta)p(f\mid\theta)df$.

Update: Joint distribution of the observations $y_{1:n}$ and the noise-free prediction $f^\star$ at a test point $\mathbf{x}^\star$ as $\begin{bmatrix}\mathbf{y}\\f^\star\end{bmatrix}\mid\mathbf{x}^\star,\mathbf{x}_{1:n}\sim\mathcal{N}(\tilde\mu,\tilde{\mathbf{K}})$
$\tilde\mu\doteq\begin{bmatrix}\mu_A\\\mu(x^\star)\end{bmatrix}$, $\tilde K\doteq\begin{bmatrix}K_{AA}+\sigma_n^2I & k_{x^\star,A}\\k_{x^\star,A}^\top & k(x^\star,x^\star)\end{bmatrix}$, $k_{x,A}\doteq\begin{bmatrix}k(x,x_1)\\\vdots\\k(x,x_n)\end{bmatrix}$

GP posterior: $f\mid\mathbf{x}_{1:n},y_{1:n}\sim\mathcal{GP}(\mu',k')$ where $\mu'(\mathbf{x})\doteq\mu(\mathbf{x})+\mathbf{k}_{\mathbf{x},A}^\top(\mathbf{K}_{AA}+\sigma_n^2\mathbf{I})^{-1}(\mathbf{y}_A-\mu_A)$ and $k'(\mathbf{x},\mathbf{x}')\doteq k(\mathbf{x},\mathbf{x}')-\mathbf{k}_{\mathbf{x},A}^\top(\mathbf{K}_{AA}+\sigma_n^2\mathbf{I})^{-1}\mathbf{k}_{\mathbf{x}',A}$ For GP-Regression $(y_{1:n}\mid\mathbf{x}_{1:n},\theta\sim\mathcal{N}(\mathbf{0},\mathbf{K}_{f,\theta}+\sigma_n^2\mathbf{I}))$, write $\mathbf{K}_{\mathbf{y},\theta}\doteq\mathbf{K}_{f,\theta}+\sigma_n^2\mathbf{I}$, and obtain:
$\hat\theta_{\text{MLE}}=\text{argmin}_\theta\frac12\mathbf{y}^\top\mathbf{K}_{\mathbf{y},\theta}^{-1}\mathbf{y}+\frac12\log\det(\mathbf{K}_{\mathbf{y},\theta})$.
Also: $\frac{\partial}{\partial\theta_j}\log p(y_{1:n}\mid\mathbf{x}_{1:n},\theta)=\frac12\text{tr}\left((\alpha\alpha^\top-\mathbf{K}_{\mathbf{y},\theta}^{-1})\frac{\partial\mathbf{K}_{\mathbf{y},\theta}}{\partial\theta_j}\right)$.

Approximations: Gaussian process need to invert Matrices $\to$ computational cost of $\mathcal{O}(n^3)$.
Local method: When sampling at $\mathbf{x}$ only condition on the samples $\mathbf{x}'$, that are close, i.e. where $|k(\mathbf{x},\mathbf{x}')|\geq\tau$ for some $\tau>0$, instead of all samples. **Problem:** τ has to be chosen carefully: if τ is chosen too large, samples become essentially independent.
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 Bayesian linear regression $\to$ time complexity of $\mathcal{O}(nm^2+m^3)$. This can be done with **Random Fourier features:** a *stationary* kernel k can be interpreted as a function in one variable, and has an associated Fourier transform which we denote by $p(\omega)$: $k(\mathbf{x}-\mathbf{x}')=\int_{\mathbb{R}^d}p(\omega)e^{i\omega^\top(\mathbf{x}-\mathbf{x}')}d\omega$.

Bochner's Theorem A continuous Kernel on $\mathbb{R}^d$ is p.s.d iff its Fourier transform $p(\omega)$ is non-negative.
$\implies$ If continuous and stationary kernel is p.s.d. and scaled correctly then $p(\omega)$ is a probability distribution named **spectral density** of k. The spectral density can be computed by: $p(\omega)=\int_{\mathbb{R}^d}k(\omega)e^{-i2\pi\xi^\top\omega}d\omega$.
Now write the kernel as an expectation: $k(\mathbf{x}-\mathbf{x}')=\int_{\mathbb{R}^d}p(\omega)e^{i\omega^\top(\mathbf{x}-\mathbf{x}')}d\omega=\mathbb{E}_{\omega\sim p}\left[e^{i\omega^\top(\mathbf{x}-\mathbf{x}')}\right]=\mathbf{z}(\mathbf{x})^\top\mathbf{z}(\mathbf{x}')$,
where $z_{\omega,b}(\mathbf{x})\doteq\sqrt2\cos(\omega^\top\mathbf{x}+b)$, and $\mathbf{z}(\mathbf{x})\doteq\frac{1}{\sqrt m}[z_{\omega^{(1)},b^{(1)}}(\mathbf{x}),\dots,z_{\omega^{(m)},b^{(m)}}(\mathbf{x})]^\top$ is a randomized feature map of Fourier transforms $\omega^{(i)}\overset{\text{iid}}{\sim}p$ and $b^{(i)}\overset{\text{iid}}{\sim}\text{Unif}([0,2\pi])$.
The error probability decays exponentially in ϵ.

4 Variational Inference

Idea: approximate the true posterior distribution with a simpler posterior that is easy to sample: $p(\theta\mid\mathbf{x}_{1:n},y_{1:n})=\frac1Z p(\theta,y_{1:n}\mid\mathbf{x}_{1:n})\approx q(\theta\mid\lambda)\doteq q_\lambda(\theta)$, where λ represents the parameters of the **variational posterior** q_λ.

Laplace Approximation: Idea: find a Gaussian approximation (i.e. second-order Taylor) of the posterior around its mode: $q(\theta)\doteq\mathcal{N}(\theta;\hat\theta,\Lambda^{-1})\propto\exp(\hat\psi(\theta))$, with $\hat\theta$ the mode (i.e. MAP estimate) and with $\mathbf{H}$ the Hessian: $\Lambda\doteq-\mathbf{H}_\psi(\hat\theta)=-\mathbf{H}_\theta\log p(\theta\mid\mathbf{x}_{1:n},y_{1:n})|_{\theta=\hat\theta}$.
Perform inference using the variations approximation: $p(y^\star\mid\mathbf{x}^\star,\mathbf{x}_{1:n},y_{1:n})\approx\int p(y^\star\mid\mathbf{x}^\star,\theta)q_\lambda(\theta)d\theta$.

Suprise of an event with probability u: $\text{S}[u]=-\log u$.
The **entropy** of a distribution p is the average surprise of samples from p: $\text{H}[p]\doteq\mathbb{E}_{x\sim p}[\text{S}[p(x)]]=\mathbb{E}_{x\sim p}[-\log p(x)]$.
Gaussian: $\text{H}[\mathcal{N}(\mu,\Sigma)]=\frac12\log((2\pi e)^d\det(\Sigma))$.
Jensen's Inequality: Given a convex function g, we have: $g(\mathbb{E}[X])\leq\mathbb{E}[g(X)]$ and if h is concave: $h(\mathbb{E}[X])\geq\mathbb{E}[h(X)]$
Observe that the surprise $\text{S}[u]$ is convex in u.
The **cross-entropy** of q relative to p is: $\text{H}[p\|q]\doteq\mathbb{E}_{x\sim p}[\text{S}[q(x)]]=\mathbb{E}_{x\sim p}[-\log q(x)]$.
Kullback-Leibler divergence, KL-divergence: $\text{KL}(p\|q)\doteq\text{H}[p\|q]-\text{H}[p]=\mathbb{E}_{\theta\sim p}\left[\log\frac{p(\theta)}{q(\theta)}\right]$
It measures the additional expected surprise when observing samples from p that is due to assuming the (wrong) distribution q.
Properties of KL: $\text{KL}(p\|q)\geq0$ (Gibbs) $\text{KL}(p\|q)=0$ if and only if $p=q$ almost surely and there exist distributions p and q such that $\text{KL}(p\|q)\neq\text{KL}(q\|p)$.
Note that: $\text{H}[p\|q]=\text{H}[p]+\text{KL}(p\|q)\geq\text{H}[p]$.
Normal dist. has the **highest entropy** among all distributions on $\mathbb{R}$ with fixed mean and variance.
$\text{KL}(\text{Bern}(p)\|\text{Bern}(q))=p\log\frac pq+(1-p)\log\frac{(1-p)}{(1-q)}$
Gaussians; $p\doteq\mathcal{N}(\mu_p,\Sigma_p)$ and $q\doteq\mathcal{N}(\mu_q,\Sigma_q)$: $\text{KL}(p\|q)=\frac12(\text{tr}(\Sigma_q^{-1}\Sigma_p)+(\mu_p-\mu_q)^\top\Sigma_q^{-1}(\mu_p-\mu_q)-d+\log\frac{\det(\Sigma_q)}{\det(\Sigma_p)})$.
Forward KL: $q_1^\star\doteq\text{argmin}_{q\in\mathcal{Q}}\text{KL}(p\|q)$;
Reverse KL: $q_2^\star\doteq\text{argmin}_{q\in\mathcal{Q}}\text{KL}(q\|p)$.
Reverse KL tends to greedily select the mode and underestimate the variance.

Evidence lower bound (ELBO), for given data $\mathcal{D}_n$: $L(q,p;\mathcal{D}_n)=\underbrace{\log p(y_{1:n}\mid\mathbf{x}_{1:n})}_{\text{const}}-\text{KL}(q\|p(\cdot\mid\mathbf{x}_{1:n},y_{1:n}))=\underbrace{\mathbb{E}_{\theta\sim q}[\log p(y_{1:n}\mid\mathbf{x}_{1:n},\theta)]}_{\text{log-likelihood}}\underbrace{-\text{KL}(q\|p(\cdot))}_{\text{proximity to prior}}$
The gradient of ELBO is generally intractable. We use the **reparametrization trick**: For $\epsilon\sim\phi$ which is independent of λ) and given a differentiable and invertible function $\mathbf{g}:\mathbb{R}^d\to\mathbb{R}^d$. Let $\theta\doteq\mathbf{g}(\epsilon;\lambda)$: $q_\lambda(\theta)=\phi(\epsilon)\cdot|\det(\mathbf{D}_\epsilon\mathbf{g}(\epsilon;\lambda))|^{-1}$, which yields: $\mathbb{E}_{\theta\sim q_\lambda}[\mathbf{f}(\theta)]=\mathbb{E}_{\epsilon\sim\phi}[\mathbf{f}(\mathbf{g}(\epsilon;\lambda))]$, for a *nice* $\mathbf{f}$.
For ELBO: $\nabla_\lambda\mathbb{E}_{\theta\sim q_\lambda}[\mathbf{f}(\theta)]=\mathbb{E}_{\epsilon\sim\phi}[\nabla_\lambda\mathbf{f}(\mathbf{g}(\epsilon;\lambda))]$. If we can find $\mathbf{g}$ and a suitable reference density ϕ which is independent of λ, we say q_λ is **reparametrizable**.
Gaussian: $q_\lambda(\theta)\doteq\mathcal{N}(\theta;\mu,\Sigma)$; $\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})$, set: $\theta=\mathbf{g}(\epsilon;\lambda)\doteq\Sigma^{1/2}\epsilon+\mu$, then: $\phi(\epsilon)=q_\lambda(\theta)\cdot|\det(\Sigma^{1/2})|$ and $\epsilon=\mathbf{g}^{-1}(\theta;\lambda)=\Sigma^{-1/2}(\theta-\mu)$

5 Markov Chains

A **Markov Chain** over $S=\{0,\dots,n-1\}$, is a sequence $(X_t)_{t\in\mathbb{N}_0}\in S$, such that the **Markov property**: $X_{t+1}\perp X_{0:t-1}\mid X_t$ is satisfied.
It is **time-homogeneous** if there is a **transition function**: $p(x'\mid x)\doteq\mathbb{P}(X_{t+1}=x'\mid X_t=x)$, with **transition matrix** as $(x_j\mid x_i)_{i,j=1}^n$. Each row sums up to 1
The state of a MC at t is a probability distribution $\mathbf{q}_t\in\in\mathbb{R}^{1\times|S|}$. We can write: $\mathbf{q}_{t+k}=\mathbf{q}_t\mathbf{P}^k$.
A distribution π is **stationary** iff $\pi(x)=\sum_{x'\in S}p(x\mid x')\pi(x')$ equivalently $\pi=\pi\mathbf{P}$
A MC is **irreducible** if every state is reachable from any state with positive probability.
A MC is **ergodic** iff there exists a $t\in\mathbb{N}_0$ such that for any $x,x'\in S$ we have: $p^{(t)}(x'\mid x)>0$ Equivalently: for some $t\in\mathbb{N}_0$ all entries of $\mathbf{P}^t$ are strictly positive
or that the MC is irreducible and aperiodic.
Irreducible MC $\to$ ergodic MC use: $\mathbf{P}'=\frac12\mathbf{P}+\frac12\mathbf{I}$
An ergodic MC has a unique stat. dist. π (with full support) and $\lim_{t\to\infty}q_t=\pi$, independently of q_0.
A MC satisfies the **detailed balance equation** w.r.t. π iff $\pi(x)p(x'\mid x)=\pi(x')p(x\mid x')$, for any $x,x'\in S$. We call such a MC **reversible** w.r.t. π.
If MC is reversible w.r.t. π, then π is a stat. dist.
Ergodic theorem For an ergodic MC and a stat. dist. π as well as $f:S\to\mathbb{R}$: $\frac1n\sum_{i=1}^n f(x_i)\xrightarrow{a.s.}\sum_{x\in S}\pi(x)f(x)=\mathbb{E}_{x\sim\pi}[f(x)]$, for $n\to\infty$ where $x_i\sim X_i\mid x_{i-1}$.

Acceptance distribution (Metropolis-Hastings): $\text{Bern}(\alpha(\mathbf{x}'\mid\mathbf{x}))$ where $\alpha(\mathbf{x}'\mid\mathbf{x})\doteq\min\left\{1,\frac{q(\mathbf{x}')r(\mathbf{x}\mid\mathbf{x}')}{q(\mathbf{x})r(\mathbf{x}'\mid\mathbf{x})}\right\}$ to decide whether to follow the proposal yields a Markov chain with stationary distribution $p(\mathbf{x})=\frac1Z q(\mathbf{x})$

Algorithm 6.18: Metropolis-Hastings algorithm
initialize $x\in\mathbb{R}^n$
for $t=1$ **to** T **do**
  sample $x'\sim r(x'\mid x)$
  sample $u\sim\text{Unif}([0,1])$
  if $u\leq\alpha(x'\mid x)$ **then** update $x\leftarrow x'$
  else update $x\leftarrow x$

Algorithm 6.20: Gibbs sampling
initialize $x=[x_1,\dots,x_n]\in\mathbb{R}^n$
for $t=1$ **to** T **do**
  pick a variable i uniformly at random from $\{1,\dots,n\}$
  set $x_{-i}\doteq[x_1,\dots,x_{i-1},x_{i+1},\dots,x_n]$
  update x_i by sampling according to the posterior distribution $p(x_i\mid x_{-i})$

The stationary distribution of the simulated Markov chain is $p(\mathbf{x})$. A **Gibbs distribution** is a continuous distribution p whose PDF is of the form $p(\mathbf{x})=\frac1Z\exp(-f(\mathbf{x}))$. f is also called an **energy function**. When the energy function f is convex, its Gibbs distribution is called **log-concave distribution**. Can write: $\alpha(\mathbf{x}'\,|\,\mathbf{x}) = \min\left\{1, \frac{r(\mathbf{x}|\mathbf{x}')}{r(\mathbf{x}'|\mathbf{x})}\exp(f(\mathbf{x})-f(\mathbf{x}'))\right\}$.

For $p(\mathbf{x})\propto\exp(-f(\mathbf{x}))$: $S[p(\mathbf{x})]=f(\mathbf{x})+\log Z$

Langevin Dynamics: Shift the proposal distribution perpendicularly to the gradient of the energy function: $r(\mathbf{x}'\mid\mathbf{x}) = \mathcal{N}(\mathbf{x}';\mathbf{x}-\eta_t\nabla f(\mathbf{x}),2\eta_t\mathbf{I})$.

6 Bayesian Deep Learning

A deep **neural network** is a function: $\mathbf{f}(\mathbf{x};\theta) \doteq \varphi(\mathbf{W}_L\varphi(\mathbf{W}_{L-1}(\cdots\varphi(\mathbf{W}_1\mathbf{x}))))$, where $\theta \doteq [\mathbf{W}_1,\dots,\mathbf{W}_L]$ is a vector of **weights**, and $\varphi:\mathbb{R}\to\mathbb{R}$ is a component-wise nonlinear **activation function**. Examples of such:

Hyperbolic tangent: $\mathrm{Tanh}(z) \doteq \frac{\exp(z)-\exp(-z)}{\exp(z)+\exp(-z)}$

Rectified linear unit: $\mathrm{ReLU}(z) \doteq \max\{z,0\}\in[0,\infty)$.

Softmax: $\sigma_i(\mathbf{f}) \doteq \frac{\exp(f_i)}{\sum_{j=1}^c \exp(f_j)}$ (classification)

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

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 $\sigma^2(\mathbf{x};\theta)\doteq\exp(f_2(\mathbf{x};\theta))$. $\log p(y_i\,|\,\mathbf{x}_i,\theta) = \text{const} - \frac12\left[\log\sigma^2(\mathbf{x}_i;\theta)+\frac{(y_i-\mu(\mathbf{x}_i;\theta))^2}{\sigma^2(\mathbf{x}_i;\theta)}\right]$. Can approximate the predictive distribution by sampling from the variational posterior $p(y^\star\,|\,\mathbf{x}^\star,\mathbf{x}_{1:n},\mathbf{y}_{1:n})\approx\mathbb{E}_{\theta\sim q_\lambda}[p(y^\star\,|\,\mathbf{x}^\star,\theta)]\approx\frac1m\sum_{i=1}^m p(y^\star\,|\,\mathbf{x}^\star,\theta^{(i)})$.

7 Active Learning

Conditional Entropy: $\mathrm{H}[\mathbf{X}\,|\,\mathbf{Y}] \doteq \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y})}[\mathrm{H}[\mathbf{X}\,|\,\mathbf{Y}=\mathbf{y}]] = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p(\mathbf{x},\mathbf{y})}[-\log p(\mathbf{x}\,|\,\mathbf{y})]$

Joint entropy: $\mathrm{H}[\mathbf{X},\mathbf{Y}] \doteq \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p(\mathbf{x},\mathbf{y})}[-\log p(\mathbf{x},\mathbf{y})]$

Properties: $\mathrm{H}[\mathbf{X},\mathbf{Y}] = \mathrm{H}[\mathbf{Y}]+\mathrm{H}[\mathbf{X}|\mathbf{Y}] = \mathrm{H}[\mathbf{X}]+\mathrm{H}[\mathbf{Y}|\mathbf{X}]$

$\mathrm{H}[\mathbf{X}|\mathbf{Y}]=\mathrm{H}[\mathbf{Y}|\mathbf{X}]+\mathrm{H}[\mathbf{X}]-\mathrm{H}[\mathbf{Y}]$ (Bayes Rule)

$\mathrm{H}[\mathbf{X}|\mathbf{Y}]\le\mathrm{H}[\mathbf{X}]$ (Information never hurts)

Mutual Information: $\mathrm{I}(\mathbf{X};\mathbf{Y})\doteq\mathrm{H}[\mathbf{X}]+\mathrm{H}[\mathbf{Y}]-\mathrm{H}[\mathbf{X},\mathbf{Y}]$

Have: $\mathrm{I}(\mathbf{X};\mathbf{Y})=\mathbb{E}_{\mathbf{y}\sim p}[\mathrm{KL}(p(\mathbf{x}|\mathbf{y})\|p(\mathbf{x}))]$.

Conditional mutual information: $\mathrm{I}(\mathbf{X};\mathbf{Y}\mid\mathbf{Z})=\mathrm{H}[\mathbf{X}|\mathbf{Z}]-\mathrm{H}[\mathbf{X}|\mathbf{Y},\mathbf{Z}]$.

Given a (discrete) function $F:\mathcal{P}(\mathcal{X})\to\mathbb{R}$, the **marginal gain** of $\mathbf{x}\in\mathcal{X}$ given $A\subseteq\mathcal{X}$ is defined as $\Delta_F(\mathbf{x}\,|\,A)\doteq F(A\cup\{\mathbf{x}\})-F(A)$.

The function is called **submodular** iff for any $\mathbf{x}\in\mathcal{X}$ and any $A\subseteq B\subseteq\mathcal{X}$ it satisfies $F(A\cup\{\mathbf{x}\})-F(A)\ge F(B\cup\{\mathbf{x}\})-F(B)$, it is called **monotone** it satisfies $F(A)\le F(B)$.

Maximization objective: monotone submodular function: $I(S)\doteq I(\mathbf{f}_S;\mathbf{y}_S)=\mathrm{H}[\mathbf{f}_S]-\mathrm{H}[\mathbf{f}_S\,|\,\mathbf{y}_S]$.

Greedy: Pick the locations $\mathbf{x}_1$ through $\mathbf{x}_n$ individually by greedily finding the location with the maximal mutual information, this provides a $(1-1/e)$-approximation of the optimum.

Uncertainty sampling: Have already picked $S_t=\{\mathbf{x}_1,\dots,\mathbf{x}_t\}$; Solve the following: $\mathbf{x}_{t+1}\doteq\operatorname{argmax}_{\mathbf{x}\in\mathcal{X}}\Delta_I(\mathbf{x}\,|\,S_t) = \operatorname{argmax}_{\mathbf{x}\in\mathcal{X}} \mathrm{I}(f_\mathbf{x};y_\mathbf{x}\,|\,\mathbf{y}_{S_t})$. Does not work with heteroscedastic noise: large aleatoric uncertainty may dominate the epistemic uncertainty.

In classification corresponds to selecting the label that maximizes the entropy of the predicted label: $\mathbf{x}_{t+1}\doteq\operatorname{argmax}_{\mathbf{x}\in\mathcal{X}}\mathrm{H}[y_\mathbf{x}\,|\,\mathbf{x}_{1:t},y_{1:t}]$.

Bayesian active learning by disagreement (BALD): This identifies those points $\mathbf{x}$ where the models *disagree* about the label $y_\mathbf{x}$ (that is, each model is "confident" but the models predict different labels): $\mathbf{x}_{t+1}\doteq\operatorname{argmax}_{\mathbf{x}\in\mathcal{X}}\mathrm{I}(\theta;y_\mathbf{x}\,|\,\mathbf{x}_{1:t},y_{1:t}) = \operatorname{argmax}_{\mathbf{x}\in\mathcal{X}}\mathrm{H}[y_\mathbf{x}\,|\,\mathbf{x}_{1:t},y_{1:t}]-\mathbb{E}_{\theta|\mathbf{x}_{1:t},y_{1:t}}\mathrm{H}[y_\mathbf{x}\,|\,\theta]$

8 Bayesian Optimization

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: Achieve sublinear regret: $\lim_{T\to\infty}\frac{R_T}{T}=0$.

Algorithm 9.3: Bayesian optimization (with GPs)

initialize $f\sim\mathcal{GP}(\mu_0,k_0)$
for $t=1$ **to** T **do**
  choose $x_t=\arg\max_{x\in\mathcal{X}}F(x;\mu_{t-1},k_{t-1})$
  observe $y_t=f(x_t)+\epsilon_t$
  perform a Bayesian update to obtain μ_t and k_t

It is common to use an **aquisition function** to greedily pick the next point to sample based on the current model.

Upper confidence bound: $\mathbf{x}_{t+1}\doteq\operatorname{argmax}_{\mathbf{x}\in\mathcal{X}}\mu_t(\mathbf{x})+\beta_{t+1}\sigma_t(\mathbf{x})$, where $\sigma_t(\mathbf{x})\doteq\sqrt{k_t(\mathbf{x},\mathbf{x})}$.

If $\beta_t=0$ then UCB is purely exploitative; if $\beta_t\to\infty$, UCB recovers uncertainty sampling.

Choosing β_t appropriately we get: $R_T=\mathcal{O}(\sqrt{T\gamma_T})$, where $\gamma_T\doteq\max_{S\subseteq\mathcal{X}, |S|=T}\mathrm{I}(\mathbf{f}_S;\mathbf{y}_S) = \max_{S\subseteq\mathcal{X},|S|=T}\frac12\log\det\left(\mathbf{I}+\sigma_n^{-2}\mathbf{K}_{SS}\right)$, is the maximum information gain after T rounds.

Information gain of some kernels: Linear: $\gamma_T=\mathcal{O}(d\log T)$

Gaussian: $\gamma_T=\mathcal{O}((\log T)^{d+1})$

Matérn for $\nu>\frac12$: $\gamma_T=\mathcal{O}\left(T^{\frac{d}{2\nu+d}}(\log T)^{\frac{2\nu}{2\nu+d}}\right)$

Thompson Sampling: At time $t+1$, we sample a function $\tilde f_{t+1}\sim p(\cdot\,|\,\mathbf{x}_{1:t},y_{1:t})$ from our posterior distribution. Then, we simply maximize $\tilde f_{t+1}$, $\mathbf{x}_{t+1}\doteq\operatorname{argmax}_{\mathbf{x}\in\mathcal{X}}\tilde f_{t+1}(\mathbf{x})$.

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** $p(x'\,|\,x,a)\doteq\mathbb{P}(X_{t+1}=x'\,|\,X_t=x,A_t=a)$; 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 $t>0$: $\pi(a\,|\,x)\doteq\mathbb{P}(A_t=a\,|\,X_t=x)$.

A policy induces a MC $(X_t^\pi)_{t\in\mathbb{N}_0}$: $p^\pi(x'\,|\,x)\doteq\mathbb{P}(X_{t+1}^\pi=x'\,|\,X_t^\pi=x)=\sum_{a\in A}\pi(a\,|\,x)p(x'\,|\,x,a)$.

The **discounted payoff** from time t is: $G_t\doteq\sum_{m=0}^\infty\gamma^m R_{t+m}$, for $\gamma\in[0,1)$, the **discount factor**.

The **state value function**: $\mathbb{E}_\pi[\cdot]\doteq\mathbb{E}_{(X_t^\pi)_{t\in\mathbb{N}_0}}[\cdot]$ measures the average discounted payoff from time t starting from state $x\in X$.

The **state-action value function (Q-function)**: $q_t^\pi(x,a)\doteq\mathbb{E}_\pi[G_t\,|\,X_t=x,A_t=a] = r(x,a)+\gamma\sum_{x'\in X}p(x'\,|\,x,a)\cdot v_{t+1}^\pi(x')$ measures the average discounted payoff from time t starting from state $x\in X$ and with playing action $a\in A$.

Bellman Expectation Equation: $v^\pi(x) = r(x,\pi(x))+\gamma\mathbb{E}_{x'|x,\pi(x)}[v^\pi(x')]$, if stochastic policy: $v^\pi(x)=\mathbb{E}_{a\sim\pi(x)}[q^\pi(x,a)]$. Also get: $q^\pi(x,a)=r(x,a)+\gamma\mathbb{E}_{x'|x,a}\mathbb{E}_{a'\sim\pi(x')}[q^\pi(x',a')]$

For deterministic: $v^\pi(x)=q^\pi(x,\pi(x))$.

Can be used to find v^π given policy π, by solving linear system of equations in cubic time in the size of the state space. Can also be solved using fixed point iteration: $\mathbf{B}^\pi\mathbf{v}\doteq\mathbf{r}^\pi+\gamma\mathbf{P}^\pi\mathbf{v}$.

A **greedy policy** w.r.t. to a state-action value function q is $pi_q(x)\doteq\operatorname{argmax}_{a\in A}q(x,a)$; a **greedy policy** w.r.t. a state value function v is: $\pi_v(x)\doteq\operatorname{argmax}_{a\in A}r(x,a)+\gamma\sum_{x'\in X}p(x'\,|\,x,a)\cdot v(x')$.

Bellman's Theorem: A policy $\pi^\star$ is optimal iff it is greedy with respect to its own value function. In other words, $\pi^\star$ is optimal iff $\pi^\star(x)$ is a distribution over the set $\operatorname{argmax}_{a\in A}q^\star(x,a)$.

If or every state there is a unique action that maximizes the state-action value function, the policy $\pi^\star$ is deterministic and unique, $\pi^\star(x)=\operatorname{argmax}_{a\in A}q^\star(x,a)$.

Algorithm 10.17: Policy iteration

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

For finite Markov decision processes, policy iteration converges to an optimal policy.

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** ∞ **do**
  $v(x)\leftarrow(\mathbf{B}^\star v)(x)=\max_{a\in A}q(x,a)$ for each $x\in X$
choose π_v

Value iteration converges to an optimal policy, as $v^\star$ and $q^\star$ are a fixed-points of the Bellman update $\mathbf{B}^\star$.

A **Partially observable Markov decision process (POMDP)** is a Markov process, with a set of supplementary **observations** Y, and **observation probabilities** $o(y\,|\,x)\doteq\mathbb{P}(Y_t=y\,|\,X_t=x)$.

POMDP are hard to solve in general, but can be reduced to MDP with an enlarged state space. We consider MDP whose state are **beliefs**: $b_t(x)\doteq\mathbb{P}(X_t=x\,|\,y_{1:t},a_{1:t-1})$. Keeping track of how beliefs change over time is **Bayesian filtering**: Given a prior belief b_t, an action taken a_t, and a new observation y_{t+1}, the belief state can be updated as: $b_{t+1}(x)=\mathbb{P}(X_{t+1}=x\,|\,y_{1:t+1},a_{1:t}) = \frac1Z o(y_{t+1}\,|\,x)\sum_{x'\in X}p(x\,|\,x',a_t)b_t(x')$, where $Z\doteq\sum_{x\in X}o(y_{t+1}\,|\,x)\sum_{x'\in X}p(x\,|\,x',a_t)b_t(x')$.

The sequence of belief-states defines the sequence of random variables $(B_t)_{t\in\mathbb{N}_0}$: $B_t\doteq X_t\,|\,y_{1:t},a_{1:t-1}$, where the (state-)space of all beliefs is the (infinite) space of all probability distributions over X: $\mathcal{B}\doteq\Delta^X\doteq\left\{\mathbf{b}\in\mathbb{R}^{|X|}:\mathbf{b}\ge\mathbf{0},\sum_{i=1}^{|X|}\mathbf{b}(i)=1\right\}$.

Given a POMDP, the corresponding **Belief-state Markov decision process** is a Markov decision process specified by the **belief space** $\mathcal{B}\doteq\Delta^X$ depending on the **hidden states** X; the set of **actions** A; **transition probabilities** $\tau(b'\,|\,b,a)\doteq\mathbb{P}(B_{t+1}=b'\,|\,B_t=b,A_t=a)$; and **rewards** $\rho(b,a)\doteq\mathbb{E}_{x\sim b}[r(x,a)]=\sum_{x\in X}b(x)r(x,a)$.

Have: $\tau(b_{t+1}\,|\,b_t,a_t)=\mathbb{P}(b_{t+1}\,|\,b_t,a_t)=\sum_{y_{t+1}\in Y}\mathbb{P}(b_{t+1}\,|\,b_t,a_t,y_{t+1})\mathbb{P}(y_{t+1}\,|\,b_t,a_t)$. We also set: $\mathbb{P}(b_{t+1}\,|\,b_t,a_t,y_{t+1})=1$ iff b_{t+1} matches the belief update given b_t,a_t, and y_{t+1}, and 0 else. Finally the likelihood is: $\mathbb{P}(y_{t+1}\,|\,b_t,a_t)=\mathbb{E}_{x\sim b_t}[\mathbb{E}_{x'|x,a_t}[\mathbb{P}(y_{t+1}\,|\,X_{t+1}=x')]] = \sum_{x\in X}b_t(x)\sum_{x'\in X}p(x'\,|\,x,a_t)\cdot o(y_{t+1}\,|\,x')$.

10 Tabular Reinforcement Learning

A trajectory τ is a sequence: $\tau\doteq(\tau_0,\tau_1,\tau_2,\dots)$, with $\tau_i\doteq(x_i,a_i,r_i,x_{i+1})$. Agent can choose any policy → **on-policy** method. No choice of policy → **off-policy** method.

Model-based → Learn the underlying MDP

Model-free → Learn the value function directly.

For model based approaches MLE yields: $\hat p(x'\,|\,x,a)=\frac{N(x'|x,a)}{N(a|x)}$ and $\hat r(x,a)=\frac{1}{N(a|x)}\sum_{t=0,x_t=x,a_t=a}^\infty r_t$

Algorithm 11.2: ϵ-greedy

for $t=0$ **to** ∞ **do**
  sample $u\in\mathrm{Unif}([0,1])$
  if $u\le\epsilon_t$ **then** pick action uniformly at random among all actio
  else pick best action under the current model

Will converge but will take time.

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 $\hat p(x^\star\,|\,x,a)=1$ 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** ∞ **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 $\hat p(x'\,|\,x,a)$
  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 $R_{\max}$.

Algorithm 11.9: Temporal-difference (TD) learning

initialize V^π arbitrarily (e.g., as 0)
for $t=0$ **to** ∞ **do**
  follow policy π to obtain the transition (x,a,r,x')
  $V^\pi(x)\leftarrow(1-\alpha_t)V^\pi(x)+\alpha_t(r+\gamma V^\pi(x'))$ // (1

This is an on-policy method.

Algorithm 11.12: Q-learning

initialize $Q^\star(x,a)$ arbitrarily (e.g., as 0)
for $t=0$ **to** ∞ **do**
  observe the transition (x,a,r,x')
  $Q^\star(x,a)\leftarrow(1-\alpha_t)Q^\star(x,a)+\alpha_t(r+\gamma\max_{a'\in A}Q^\star(x',a'))$

The Monte Carlo approximation does not depend on the policy → algorithm is off-policy. The update rule can also be expressed as: $Q^\star(x,a)\leftarrow Q^\star(x,a)+\alpha_t(r+\gamma\max_{a'\in A}Q^\star(x',a')-Q^\star(x,a))$.

RM conditions: $\alpha_t\ge0$, $\sum_{t=0}^\infty\alpha_t=\infty$, $\sum_{t=0}^\infty\alpha_t^2<\infty$.

If α_t satisfy RM conditions and every state-action pair is visited infinitely often, then we have convergence for both algorithms.

Algorithm 12.11: REINFORCE algorithm

initialize policy weights φ
repeat
  generate an episode (i.e., rollout) to obtain trajectory τ
  for $t=0$ **to** $T-1$ **do**
    set $g_{t:T}$ to the downstream return from time t
    $\varphi\leftarrow\varphi+\eta\gamma^t g_{t:T}\nabla_\varphi\log\pi_\varphi(a_t\,|\,x_t)$ // (12.45)
until converged

11 Model-free Reinforcement Learning

Can view TD-learning as SGD on the squared loss $\ell(\theta;x,r,x')\doteq\frac12\left(r+\gamma\theta^{\mathrm{old}}(x')-\theta(x)\right)^2$.

Parametric value function approximation: To scale to large state spaces, learn approximation of (action) value function $V(\mathbf{x};\theta)$ or $Q(\mathbf{x},\mathbf{a};\theta)$. For e.g. the parameters θ of a neural network.

Q-learning with function approximation: In state $\mathbf{x}$, pick action a; Observe $\mathbf{x}'$, reward r. Update $\theta\leftarrow\theta+\alpha_t\delta_B\phi(\mathbf{x},\mathbf{a})$, where $\delta_B\doteq r+\gamma\max_{a'}Q^\star(\mathbf{x}',\mathbf{a}';\theta^{\mathrm{old}})-Q^\star(\mathbf{x},\mathbf{a};\theta)$.

Given a policy π, the **advantage function** is $a^\pi(\mathbf{x},\mathbf{a})\doteq q^\pi(\mathbf{x},\mathbf{a})-v^\pi(\mathbf{x}) = q^\pi(\mathbf{x},\mathbf{a})-\mathbb{E}_{a'\sim\pi(\mathbf{x})}[q^\pi(\mathbf{x},\mathbf{a}')]$

π is optimal $\iff\forall\mathbf{x}\in\mathcal{X},\mathbf{a}\in\mathcal{A}: a^\pi(\mathbf{x},\mathbf{a})\le0$

The **policy value function** measures the discounted payoff of policy π: $j(\pi)\doteq\mathbb{E}_\pi[G_0]=\mathbb{E}_\pi\left[\sum_{t=0}^\infty\gamma^tR_t\right]$, and the bounded variant: $j_T(\pi)\doteq\mathbb{E}_\pi[G_{0:T}]=\mathbb{E}_\pi\left[\sum_{t=0}^{T-1}\gamma^tR_t\right]$. Abbreviate $j(\varphi)\doteq j(\pi_\varphi)$.

Actor-Critic methods consist of two components: a parameterized policy, $\pi(\mathbf{a}\,|\,\mathbf{x};\varphi)\doteq\pi_\varphi$ which is called **actor**; and a value function approximation, $q^{\pi_\varphi}(\mathbf{x},\mathbf{a})\approx Q^{\pi_\varphi}(\mathbf{x},\mathbf{a};\theta)$, which is called**critic**. In the following, we will abbreviate Q^{π_φ} by Q.

Use gradient approximation: $\nabla_\varphi J(\varphi)\approx\sum_{t=0}^\infty\mathbb{E}_{(\mathbf{x}_t,\mathbf{a}_t)\sim\pi_\varphi}\left[\gamma^tQ(\mathbf{x}_t,\mathbf{a}_t;\theta)\nabla_\varphi\log\pi_\varphi(\mathbf{a}_t\,|\,\mathbf{x}_t)\right]$

Algorithm 12.16: Online actor-critic

initialize parameters φ and θ
repeat
  use π_φ to obtain transition $(\mathbf{x},\mathbf{a},r,\mathbf{x}')$
  $\delta=r+\gamma Q(\mathbf{x}',\pi_\varphi(\mathbf{x}');\theta)-Q(\mathbf{x},\mathbf{a};\theta)$
  // actor update
  $\varphi\leftarrow\varphi+\eta\gamma^tQ(\mathbf{x},\mathbf{a};\theta)\nabla_\varphi\log\pi_\varphi(\mathbf{a}\,|\,\mathbf{x})$ //
  // critic update
  $\theta\leftarrow\theta+\eta\delta\nabla_\theta Q(\mathbf{x},\mathbf{a};\theta)$ //
until converged

12 Model-based Reinforcement Learning

Algorithm 13.1: Model-based reinforcement learning (outline)

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

Algorithm 13.2: Model predictive control, MPC

for $t=0$ **to** ∞ **do**
  observe x_t
  plan over a finite horizon H,
    $\max_{a_{t:t+H-1}}\sum_{\tau=t}^{t+H-1}\gamma^{\tau-t}r(x_\tau,a_\tau)$ such that $x_{\tau+1}=f(x_\tau,a_\tau)$ (13
  carry out action a_t

By Nils Jensen — nils.jensen@inf.ethz.ch — HS2023