Background on Bayesian machine learning, inference and control

Tom Jackson

August 15, 2022

Abstract

This is a (disorganized) summary writeup on with pointers to literature I've compiled for my own use.

Chapter 1

Probability

1 Gaussians

1.1 Gaussian integrals

$$\int dx\, e^{-x^2} = \sqrt{\pi}.$$

Gaussian integral via completing the square:

$$\int d^d\mathbf{x}\, \exp\left[-\tfrac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{J}\mathbf{x}\right] = \sqrt{\frac{(2\pi)^d}{\det\mathbf{A}}}\, \exp\left[-\tfrac{1}{2}\mathbf{J}^T\mathbf{A}^{-1}\mathbf{J}\right].$$

1.2 Normal distributions

Ref: wiki.

Denote $X \sim \mathcal{N}(\mu, \mathbf{\Sigma})$ for a normally distributed random variable X:

$$p(X = \mathbf{x}) = \frac{1}{\sqrt{(2\pi)^d \det\mathbf{\Sigma}}} \exp\left[-\tfrac{1}{2}(\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right],$$

with $\mathbf{\Sigma}$ positive definite.

If $X \sim \mathcal{N}(\mu, \mathbf{\Sigma})$, an affine transformation $Y = \mathbf{H}X + \mathbf{c}$ is distributed as $Y \sim \mathcal{N}(\mathbf{H}\mu + \mathbf{c}, \mathbf{H}\mathbf{\Sigma}\mathbf{H}^T)$. Implies marginal distribution just drops relevant entries from $\mu, \mathbf{\Sigma}$.

Note that if X_1, X_2 are normal, their joint distribution need not be, even if they're uncorrelated [$\mathrm{cov}(X_1, X_2) = 0$].

Conditioning a Gaussian: Take X_1, X_2 joint normal, with $\mathbf{\Sigma}$ having a 2×2 block structure. Conditional distribution $p(X_1|X_2 = \mathbf{x}_2) = p(X_1, X_2)/p(X_2)$ can be read off using the Schur identity

$$\begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}^{-1} = \begin{bmatrix} S & -S\Sigma_{12}\Sigma_{22}^{-1} \\ -\Sigma_{22}^{-1}\Sigma_{21}S & \Sigma_{22}^{-1} + \Sigma_{22}^{-1}\Sigma_{21}S\Sigma_{12}\Sigma_{22}^{-1} \end{bmatrix}$$

with

$$S \equiv \left(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right)^{-1}.$$

Marginal $1/p(X_2)$ only cancels constant term. Reading off new mean and covariance from linear and quadratic X_1 dependence, we get

$$X_1|X_2 = \mathbf{x}_2 \sim \mathcal{N}\left(\mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{x}_2 - \mu_2), S^{-1}\right).$$

Note that conditioning on X_2 shifts the mean for X_1, which wouldn't have happened if we just took the marginal $p(X_1)$.

2 Information theory

2.1 Definitions

2.1.1 Entropy

$$H(X) = \int dx\, p(x) \log p(x) \geq 0.$$

2.1.2 Kullback-Leibler divergence

$$D_{KL}[P|Q] = \int dx\, p(x) \log \frac{p(x)}{q(x)} \geq 0,$$

with equality only for $P = Q$. Convex.

Also referred to as relative entropy. Asymmetric between P and Q, to reflect Bayesian inference: $D_{KL}[P|Q]$ is information gain from revising prior Q to posterior P.

2.1.3 Evidence lower bound

Fact that D_{KL} is always positive yields a relevant bound for variational inference. Let X be a latent variable and Z an observation. Then

$$D_{KL}[Q(X)|P(X|Z)] = \left\langle \log \frac{Q(X)}{P(X|Z)} \right\rangle_{X\sim Q} = \left\langle \log P(Z) - \log \frac{P(X,Z)}{Q(X)} \right\rangle_{X\sim Q};$$

$\log P(Z)$ is independent of X, so it can be taken outside the expectation, and the LHS is ≥ 0, so

$$\log P(Z) \geq \left\langle \log \frac{P(X,Z)}{Q(X)} \right\rangle_{X\sim Q}.$$

The RHS is the EBLO objective. Finding the arg-max of this objective over all distributions $Q(X)$ drives $Q(X)$ to the posterior $P(X|Z)$, so this can be taken as the starting point for variational Bayesian inference.

2.1.4 Fisher information

KL divergence is asymmetric, hence not a metric (some parallels to squared distance). "Infinitesimal form," in the following sense, is a valid metric. Assume P, Q depend on parameters θ, and let P be a small perturbation of Q: $P_\theta = Q_{\theta_0+\delta\theta}$. Then the hessian

$$g_{ij}(\theta_0) = \frac{\partial^2}{\partial\theta_i\, \partial\theta_j} D_{KL}\left[P_\theta | Q_{\theta_0}\right]$$

is a positive semidefinite Riemannian metric on the space of parameters θ.

2.1.5 Mutual information

$$I(X;Y) = D_{KL}[p(x,y)|p(x)p(y)] = \int dx\, dy\, p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \geq 0.$$

[...]

2.2 Recent work on mutual information estimators.

2.2.1 Mutual Information Neural Estimation

[BBR+18]; no first-party code but third-party implementations; e.g. [1], [2].

Application of neural network methods to estimate mutual information from a set of samples. Wide range of nonparametric estimators in prior literature (reviewed in [Pan03]), but argue they don't scale to high data dimensionality (also gradients can be ill-defined or expensive).

General principle used often in this and following sections: derive a functional bound for the true MI. Use this as a loss function and implement the function as a neural network, "trained" by standard parameter optimization. The best value of the loss function is then the estimator's estimate.

Here the bound used is

$$D_{KL}[P|Q] = \sup_f \left[\langle f \rangle_P - \log\langle e^f \rangle_Q \right],$$

for arbitrary scalar function f having the same domain as P, Q; this is used to get a variational lower bound for MI. Naive estimation of the gradient is biased, so an exponential weighted moving average is used [???]. For MI, minibatch sampling is applied; $P = p(x,y)$ is the empirical distribution of the batch, and $Q = p(x) \otimes p(y)$ is generated by sampling from the marginals, or by permuting one member of the (x,y) pairs.

2.2.2 Related work and follow-ups

Related work on MINE-style estimation:

- [NWJ10]: Earlier estimator (NWJ) using similar but looser bound.
- [LMGW20] comments; code. Learn a network that discriminates whether samples came from the joint or product-marginal distribution. Doesn't outperform state of the art, and reviewer points out it's a specific case of ideas in NWJ.
- [vdOLV19]: Another estimator (InfoNCE), based on contrast predictive coding.
- [LSN+19]; comments. Propose modification of MINE training to make it more data-efficient; unclear as to how — purely by separating data into train/test sets?
- [CAH+19]; code. Also addresses efficiency of MINE; claim improvement by estimating entropies as an intermediate step.
- [WZH+20]; comments propose a nontrivial estimator for the gradient of MI directly that may be more relevant when it's used in a loss function.

More general discussion:

- [MS20]; comments. Argue that there are fundamental limitations to the entire variational lower bound program.

- [PO19]: Compare variational methods in existing literature. Nothing yields good estimates for practical batch sizes, perhaps due to logic in previous ref. Propose new estimator I_α interpolating between InfoNCE and NWJ.
- [SE20]; comments. Elaborate on bias/variance tradeoff; variance in estimates from MINE et al. can blow up due to instability. Code for an improved estimator (SMILE).
- [CL20]; comments (code in supplementary material). Propose to regularize variance blow-ups in MINE by adding a term $-d(\log\langle e^f\rangle_Q, C)$ where d is a 1d distance function and C is an arbitrary constant; use $\lambda(x-C)^2$.

Remark 1. *Would appear that best current estimator is either [CAH+19] or [CL20]; former suffers from not providing comparison with other methods.*

2.2.3 Contrastive Log-ratio Upper Bound

[CHD+20]; code.

Prior work focusing on *upper* bounds on MI (needed for MI minimization objectives) required knowledge of $p(y|x)$, e.g. the approximation to the marginal in InfoBot (**1.5.1**). If $p(y|x)$ is known, use

$$\begin{aligned} I(X;Y) &\leq \langle \log p(y|x)\rangle_{p(x,y)} - \langle \log p(y|x)\rangle_{p(x)\otimes p(y)}, \\ &= \frac{1}{N^2}\sum_{i,j}^{N} \log(y_i|x_i) - \log(y_j|x_i); \end{aligned}$$

If $p(y|x)$ isn't known, represent it as a NN. The double sum can be improved to a single loop over the data by using a random sample of the $\{y_j\}$, similar to MINE.

3 Probabilistic graphical models

3.1 Markov chains.

Generic model of a memoryless process. Take a discretized state space $S \in \mathcal{S}$ in discrete time and describe stochastic time evolution through matrix of conditional transition probabilities

$$T_{ij} = \Pr(S_{t+1} = s_i | S_t = s_j).$$

($S_{t+1}, S_t \ldots$ are discrete random variables, while s_i, s_j label state space elements.)

T is a "right stochastic matrix": all entries are non-negative and rows sum to 1. (continuous version: kernel; volume preserving diffeos). Stationary distributions π are left eigenvectors: $\pi T = \pi$. *If* T is irreducible (one connected component) and aperiodic (aka *ergodic*), stationary distribution is unique Perron-Frobenius eignevector of eigenvalue 1.

Is there a model of chaotic dynamics that's discrete in both space and time? Possible to define attractor for cellular automata?

Detailed balance condition (aka "reversibility") is that there exists a π such that:

$$\pi_i T_{ij} = \pi_j T_{ji}.$$

If π exists, it's a steady-state distribution. For any T, π and matrix norm, can find a reversible T^* closest to T preserving π through quadratic convex optimization. Doubly-stochastic matrices have a neat combinatorical description (Birkhoff polytope).

3.2 Hidden Markov models.

Add unobserved, discrete latent variables (observed variables may be discrete or continuous.)

3.2.1 Graphical models.

Depiction of the factorization of a general distribution. Graph nodes are individual random variables (or components), with an arrow $v \to v'$ if the distribution factorizes as $p(S_{v'}|\cdots, S_v, \cdots)$. Markov chain is just a linear chain, while hidden Markov model has a comb structure: latent variables $\{Z_t\}$ by themselves are a Markov chain, while observable S_t depends on latent z_t at that time only. When limited to observables, S_t is not conditionally independent of any other S in its past.

At fixed t, can view as a mixture of distribution model (with mixture components labeled by values of z_{t-1}. Can view as an example of independent component analysis (with the hidden variables labeling the components.)

[...]

3.3 Markov Decision Processes

3.3.1 Terminology

Pick up from discussion of Markov chains; remain in discrete setting. Augment state with choice of action A_t taken at time t, and reward R_t. Transition probability now takes the form $T(s_+, r|s, a)$, where we abbreviate $s_+ = s_{t+1}$; note that this form allows for stochasticity in actions and rewards, since deterministic case is a special case of this. Can show that optimal policy for an MDP where everything is known is always deterministic, but stochastic policies can be optimal for POMDPs.

Also retain critical feature that s_+ "depends only on s," which is reflected in ansatz for stochastic policy function $\pi(a|s)$ — in particular, A isn't merely enlarging state space. Stated problem is to select π to maximize discounted total reward $\sum_t \gamma^t R(s_t, a_t)$. At this point we only introduce discounting factor $0 < \gamma < 1$ for convergence's sake.

Martingale? Is this what's meant by a "separator"?

Note this is a bit orthogonal to traditional ML: "semi-supervised" learning in that we learn reward, but it's up to the agent to determine how to relate that to policy. Also learning problems in general either omit feedback (offline) or tend to leave it implicit in the online case.

4 Monte Carlo

4.1 Intro

Ref: [Mac03] ch. 29.

Recall that Monte Carlo integration, at its simplest, estimates a high-dimensional integral by randomly sampling positions in the domain of integration and taking the average of values of the integrand evaluated at those points. This is wasteful if we draw lots of samples from areas where the integrand is small.

Or quasi-randomly sampling; see low-discrepancy sequences.

A special case of integration is computing expectation values of arbitrary quantities f with respect to a high-dimensional distribution P, $\langle f \rangle = \int P(x) d^d x\, f(x)$. This is related to the problem of simply generating samples from P, because if we can do that we can estimate $\langle f \rangle$ by evaluating f there.

An essential point of MC is that the error in doing so goes as $\mathrm{var}(f)/N$, and is *independent* of the dimensionality d of x.

4.1.1 Importance sampling

If we have an approximation $Q(x)$ to $P(x)$ that's cheaper to evaluate, we can simply draw samples from Q and weight them according to $w_i = P(x_i)/Q(x_i)$. Problems are 1) this is essentially uncontrolled without some guarantee on how close Q is to P, and 2) suffers in high dimensions: weights become dominated by large values.

4.1.2 Rejection sampling

Similar, but now assume we know a constant c such that $cQ(x) > P(x)$ for all x. For each sample, generate a uniform variate u_i from 0 to $cQ(x_i)$ and only keep the sample if $u_i < P(x)$. Retained samples are independent from P(x). Problem is finding c so that rejection isn't too frequent, also harder in high dimensions.

4.2 Markov chain Monte Carlo

Ref: [Mac03] ch. 29.

Coming up with a single approximate $Q(x)$ is too hard, so instead make it dependent on the value of the last sample: $Q(x|x_t)$ (now using t to index steps). This means the sampling process is a stateful Markov chain. In general, method involves designing a Markov chain in x-space such that $P(x)$ is the unique invariant measure (uniqueness requires irreducibility and ergodicity). *Detailed balance* is another term for reversibility,

$$T(x, x')P(x') = T(x', x)P(x)$$

for all x, x'. Implies P is invariant, but not essential!

Transition matrices need to preserve P, i.e. $P(x') = \int dx\, T(x', x)P(x)$. Can build from "base transitions" by taking convex linear combos or by convolution.

Potential disadvantages of MC approach:

1. Samples are no longer independent: $\Pr(x_t) \sim P(x)$ for large t, but hard to tell how many steps are needed for convergence. Including dependent samples in average *doesn't* bias estimates [proof?], but doesn't help, and makes error estimates harder.
2. Doesn't allow direct access to normalization factor/partition function Z, although ratios possible.
3. Non-Bayesian. Properly Bayesian MC would give distribution for our knowledge of the estimator which would only depend on the evaluations at the same points, not on the initial configuration of the MC or other implementation details; see [GR03].

Matter of art whether to use one long chain (better convergence) or multiple short chains, restarted from different points (lower correlations; better chance to explore state space if P is multimodal).

4.2.1 Metropolis-Hastings

Generate new sample x' from $Q(x'|x_t)$ and compute weight

$$a = \frac{P(x')}{P(x_t)} \frac{Q(x_t|x')}{Q(x'|x_t)};$$

if $a \geq 1$, *accept* the new x' as x_{t+1}. If $a < 1$, accept with probability a and reject with $1 - a$, in which case keep $x_{t+1} = x_t$. More concretely: let Q be gaussian centered on x_t, then we're effectively doing a random walk with step size of order σ_Q. (If Q symmetric in arguments, second ratio is 1.)

Small steps (σ_Q) means slow convergence, while large steps (relative to scale of peaks/features of P) also means slow convergence due to frequent rejection. *But* these considerations don't get worse with high dimension (intrinsically), unlike importance and rejection sampling. "Efficient"/practical MC methods basically deal with reducing the problems arising from random walk behavior in vanilla Metropolis-Hastings.

What about fractal, multimodal P?

4.2.2 Gibbs sampling

The special case of the above when Q is P conditioned on all other components of the sample: $Q(x|\text{state}) = P(x_{(i)}|x_{(j)\neq i})$. Individual components of x are updated in deterministic order:

$$\begin{aligned} x_{t+1,(1)} &\sim P(x_{(1)}|x_{t,(2)}, x_{t,(3)}, \ldots) \\ x_{t+1,(2)} &\sim P(x_{(2)}|x_{t+1,(1)}, x_{t,(3)}, \ldots) \\ x_{t+1,(3)} &\sim P(x_{(3)}|x_{t+1,(1)}, x_{t+1,(2)}, \ldots) \end{aligned}$$

This is Glauber dynamics (single spin flips) in condensed matter. Motivated as a "quick-and-dirty" method.

4.3 Efficient MCMC

Ref: [Mac03] ch. 30.

4.3.1 Hamiltonian Monte Carlo

Improve random walk convergence with gradient information: adding drift to random walk gives linear instead of square-root convergence. Assumes $P(x) \sim \exp -\beta E(x)$, and that gradients of E are cheap. Do this by adding momentum term $p^2/2$ to $E(x)$ but retaining only the x coordinates of generated samples.

What if we can only afford sampling $b \cdot \partial_x E$ along a few vectors b?

Two-step Gibbs sampling scheme: First sample p (always accepted), then update x and then p according to

$$\dot{x} = p; \qquad \dot{p} = -\partial_x E(x).$$

If numerics exact, this step should also always be accepted under Metropoplis, since $p^2/2 + E(x)$ a constant of motion. On the other hand, dynamics need to be *exactly* reversible: state space volume must be conserved, and if $(x, p) \to (x', p')$ is generated as a deterministic update, we must also generate $(x, -p)$ starting from $(x', -p')$.

What about assigning fancier dynamics, like Nambu? Fictitious dynamics chosen so integrals of motion give efficient sampling.

4.3.2 Simulated annealing

Introduce fictitious temperature β conjugate to $E(x)$ and tune $\beta \searrow 1$. Can only couple β to "messy" terms in E to interpolate between distributions. Simulated tempering [MP92] fixes biases from getting trapped in individual minima by making β a dynamically updated variable, see also annealed importance sampling [Nea01].

4.3.3 MC as a communication channel

Ref: [Mac03] 30.5.

Sampling an x from $P(x)$ consumes at least $\log 1/P(x)$ random bits (cf. arithmetic coding).

Unfortunately sketchy. Anyone followed up?

Ignore Q updates, then all information about true P communicated by sequence of binary accept/reject Metropolis moves. So rule of thumb: maximize "information learned" about P by tuning acceptance probability to be about $1/2$ (max entropy). Efficient methods try to pick Q to beat this "one bit per trial" bound. Note that even importance sampling potentially gives us more than one bit to work with (the full ratio P/cQ.)

Evolutionary analogy: acceptance is mutated genome replacing old one. Not clear that genetic methods actually do this, though.

4.3.4 Exact sampling

Ref: [Mac03] ch. 32, [PW96].

Addresses questions of Markov chain convergence to target distribution. "Three ideas," following MacKay:

1. Markov chain coalescence: Due to finite memory, if two runs of a chain from different initial conditions but using the same random number generator hit the same value, all subsequent values will be identical: chain has *forgotten* the difference in initial conditions.

   Randomness as a channel. Is there a way to phrase this that's realization-independent?

2. Bounds on coalescence: Impractical to restart chain for all possible initial conditions. For practical use, look for a *partial order* on configurations that's invariant under MC dynamics. Extrema under this order provide a bound: if they've coalesced, know all conditions "between" them have as well.

   Relax this to partial domains of attraction? Live with probabilistic estimate that chains have converged?

   "Summary states" for non-attractive distributions (Huber 1998), (Harvey and Neal 2000): bounds don't have to be tight, or even physical trajectories of system. Example given uses *partial* configurations.
3. Coupling from the past: Coalescence is a distinguished event (depends on details of how the chain is designed), so coalescence doesn't immediately imply convergence. Instead start run at time T_0 in past and run up to present; if coalescence hasn't happened, increase T_0. If it has, unique configuration at present is an exact sample. All runs are made with identical realization of random numbers at each time.

   Cf. random dynamical systems.

Other applications of coupled Markov chains — gets into interacting particle systems, right?

4.3.5 Extreme values and large deviations

Remark 2. *Above assumes that we're taking expectation of things that are smooth. What if we're interested in extreme values instead?*

[...]

5 Kernel methods

5.1 Kernel trick

Want to apply linear techniques in nonlinear situations. Assume we can do so by mapping configurations $\mathbf{x}$ to a higher (possibly infinite)-dimensional feature space $\varphi(\mathbf{x})$; φ will be nonlinear.

Because we want to use linear methods in the feature space, it will usually be the case that φ will only enter in the form of a kernel, or Gram matrix of known samples:

$$K(\mathbf{x}, \mathbf{y}) \equiv \langle \varphi(\mathbf{x}), \varphi(\mathbf{x}) \rangle; \qquad K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j).$$

The "kernel trick" is that by working in terms of positive semi-definite $K(\mathbf{x}, \mathbf{y})$, we don't need to explicitly design, compute or store the feature map $\varphi(\mathbf{x})$.

Example: classification; want a hyperplane separating two classes in feature space, which corresponds to a curved (but still sharp) class boundary in configuration space.

5.2 Kernel mean embedding of distributions

5.2.1 Reproducing kernel Hilbert spaces

"Reproducing" here refers to the evaluation map ("evaluate f at x"); one-to-one with existence of kernel via $\langle f, K(\mathbf{x}, \cdot) \rangle = f(\mathbf{x})$. Hilbert spaces that aren't RKHS are somewhat contrived.

Representer theorems state that RKHS representation is useful for generalized forms of ridge regression. Min-error f takes the form of a linear combination of $K(x_i, \cdot)$.

5.2.2 Kernel embedding

Closely analogous. For a random variable $X \sim P$ and a given kernel K, the mean embedding is

$$\mu_X = \mathbb{E}\left[K(X, \cdot)\right] = \int dx\, p(x) \varphi(x).$$

We then get the expectation of any function via $\mathbb{E}\left[f(X)\right] = \langle f, \mu_X \rangle$. Hope that embedding $\varphi(X)$ captures sufficient statistics about X.

Empirical/sample estimator just $\widehat{\mu}_X = \frac{1}{N} \sum \varphi(x_i)$. Can show that convergence is $\sim 1/\sqrt{N}$, independent of the feature space dimensionality, so in this sense these methods get around the curse of dimensionality.

Joint distribution of X, Y as $C_{XY} = \mathbb{E}\left[\varphi(X) \otimes \varphi(Y)\right]$.

Conditional distributions: let $C_{Y|X} = C_{YX} C_{XX}^{-1}$. Assuming existence, conditional embedding is a series of feature vectors indexed by x:

$$\mu_{Y|X=x} = C_{Y|X} \varphi(x) = \int dy\, p(y|x) \varphi(y).$$

5.2.3 Induced properties

Norm on RKHS gives us a notion of "distance" between distributions, the maximum mean discrepancy

$$\mathrm{MMD}(P_X, P_Y) = ||\mu_X - \mu_Y||^2 = \sup_{||f|| \leq 1} \mathbb{E}\left[f(X)\right] - \mathbb{E}\left[f(Y)\right].$$

Asymmetry of D_{KL} means probably impossible to choose φ to reproduce info-th metrics, right?

Likewise, measure of dependence/correlation

$$\mathrm{HSIC}(X, Y) = ||C_{XY} - \mu_X \otimes \mu_Y||^2.$$

Remark 3. *How does this improve over Monte Carlo? Presumably on strength of kernel embedding. To what extent do we need to know φ explicitly for encoding/decoding, though?*

6 Kernel approximations

Computationally tractable implementations of kernel methods basically boil down to dense matrix operations; for large data applications these are too slow, and we instead want to use reduced-rank approximations.

6.1 Random features

"Random" in the sense that we regard "feature space" as being indexed by a random variable θ;

$$K(x, x') = \int d\theta\, p(\theta) \varphi_\theta(x) \varphi_\theta(x').$$

In computations we use a Monte Carlo approximation of this integral by M samples of θ.

Question then arises of how to choose φ_θ to reproduce known kernel K. Partial answer for translation-invariant K from

6.1.1 Random Fourier features

Question then arises of how to choose φ_θ to reproduce known kernel K. Partial answer for translation-invariant K from Bochner's theorem: take $\varphi_{a,b}(x) \propto \cos(a^T x + b)$, with b uniform on $[0, 2\pi)$ and a taken from an N-dimensional distribution.

$a \sim$ Gaussian reproduces the RBF kernel; $a \sim t$-distribution reproduces the Matern(1/2) kernel (in limit $M \to \infty$).

6.2 Gamblets

6.3 Neural networks

Chapter 2

The linear-Gaussian world

1 Regression

2 Kalman filtering

Ref: mostly wiki, with notation changes.

2.1 Bayesian state estimation

2.1.1 Single Bayes update

Assume we have a Gaussian prior for an unchanging latent variable $X \sim \mathcal{N}(\mu_0, \mathbf{\Sigma}_0)$ updated with a noisy observation $Z = \mathbf{H}X + \eta$, with noise $\eta \sim \mathcal{N}(0, \mathbf{\Gamma})$. We want to update

$$P(X|Z = z) = \frac{P(Z = z|X)}{P(Z = z)} P(X).$$

From above, we have $Z|X \sim \mathcal{N}(\mathbf{H}X, \mathbf{\Gamma})$. Know posterior will also be normally distributed, so assume normalization works out; collect terms in $\mathbf{x}$ to read off parameters of $X|Z \sim \mathcal{N}(\mu, \mathbf{\Sigma})$. We have

$$\mathbf{\Sigma}^{-1} = \mathbf{\Sigma}_0^{-1} + \mathbf{H}^T \mathbf{\Gamma}^{-1} \mathbf{H}; \tag{2.1}$$

Apply Woodbury identity to invert

$$\mathbf{\Sigma} = \mathbf{\Sigma}_0 + \mathbf{\Sigma}_0 \mathbf{H}^T \mathbf{S}^{-1} \mathbf{H} \mathbf{\Sigma}_0,$$

abbreviating

$$\mathbf{S} \equiv \left(\mathbf{\Gamma} + \mathbf{H} \mathbf{\Sigma}_0 \mathbf{H}^T \right).$$

Then, collecting terms linear in $\mathbf{x}$ and matching against the desired term $-\mathbf{x}^T \mathbf{\Sigma}^{-1} \mu$,

$$\mu = \mathbf{\Sigma} \left(\mathbf{\Sigma}_0^{-1} \mu_0 + \mathbf{H}^T \mathbf{\Gamma}^{-1} \mathbf{z} \right). \tag{2.2}$$

Expanding,

$$\begin{aligned} \mu = \mu_0 - \mathbf{\Sigma}_0 \mathbf{H}^T \mathbf{S}^{-1} \mathbf{H} \mu_0 &+ \mathbf{\Sigma}_0 \mathbf{H}^T \left[1 - \mathbf{S}^{-1} \mathbf{H} \mathbf{\Sigma}_0 \mathbf{H}^T \right] \mathbf{\Gamma}^{-1} \mathbf{z} \\ &= \text{''} + \mathbf{\Sigma}_0 \mathbf{H}^T \left[\mathbf{S}^{-1} \mathbf{S} - \mathbf{S}^{-1} \left(\mathbf{S} - \mathbf{\Gamma} \right) \right] \mathbf{\Gamma}^{-1} \mathbf{z} \\ &= \text{''} + \mathbf{\Sigma}_0 \mathbf{H}^T \mathbf{S}^{-1} \mathbf{z}. \end{aligned}$$

2.1.2 More terminology

Re-express the above results by introducing new terms.

Innovation is residual before update:

$$\mathbf{y}_0 = \mathbf{z} - \mathbf{H}\mu_0,$$

satisfying $\mathbb{E}[\mathbf{y}_0] = 0$ and $\mathrm{cov}(\mathbf{y}_0) = \mathbf{S}$.

Kalman gain defined as how much we use $\mathbf{y}_0$ to correct μ_0;

$$\mu = \mu_0 + \mathbf{K}\mathbf{y}_0.$$

In the filtering problem, can define the optimal Kalman gain as the one that minimizes the posterior residual $\mathbf{y} = \mathbf{z} - \mathbf{H}\mu$. Unsurprisingly, the minimum mean-square error gain is what we got above via Bayes' rule, namely

$$\mathbf{K}^* = \mathbf{\Sigma}_0 \mathbf{H}^T \mathbf{S}^{-1}.$$

In these terms, the Bayes update is

$$\mu = \mu_0 + \mathbf{K}^*\mathbf{y}_0 = (1 - \mathbf{K}^*\mathbf{H})\mu_0 + \mathbf{K}^*\mathbf{z}; \qquad \mathbf{\Sigma} = (1 - \mathbf{K}^*\mathbf{H})\mathbf{\Sigma}_0. \tag{2.3}$$

2.1.3 Information filter

Jumping back and forth between $\mathbf{\Sigma}$ and $\mathbf{\Sigma}^{-1}$ is cumbersome. Instead, remain in inverse space, defining

$$\check{\mathbf{\Gamma}} \equiv \mathbf{H}^T\mathbf{\Gamma}^{-1}\mathbf{H}; \qquad \check{\mathbf{z}} \equiv \mathbf{H}^T\mathbf{\Gamma}^{-1}\mathbf{z}; \qquad \check{\mu} \equiv \mathbf{\Sigma}^{-1}\mu; \qquad \check{\mathbf{\Sigma}} \equiv \mathbf{\Sigma}^{-1}.$$

The update can be read off from (2.1) and (2.2):

$$\check{\mu}_n = \check{\mu}_0 + \sum_{j=1}^{n} \check{\mathbf{z}}_j; \qquad \check{\mathbf{\Sigma}}_n = \check{\mathbf{\Sigma}}_0 + n\check{\mathbf{\Gamma}}, \tag{2.4}$$

In these variables the update is simply additive, so we can write down the multiple-update solution.

2.2 Adding dynamics

Context is estimation of a hidden Markov model (**3.2**) for continuous quantities in discrete time. We have stochastic system state

$$\mathbf{x}_t = \mathbf{F}_t\mathbf{x}_{t-1} + \mathbf{B}\mathbf{u}_t + \xi_t,$$

with iid noise $\xi \sim \mathcal{N}(0, \mathbf{\Gamma}_\xi)$, possibly depending on control input $\mathbf{u}_t$.

[...]

3 Control

3.1 General background on control

Ref: [KR16].

Generically we have a system state $\mathbf{x}$, control signal $\mathbf{u}$ and known dynamics $\dot{\mathbf{x}} = f[t, \mathbf{x}(t), \mathbf{u}(\mathbf{x}, t)]$. The *representation problem* is that it's wasteful to compute optimal $\mathbf{u}$ for all possible $\mathbf{x}$.

If dynamics are deterministic, only need control input along optimal trajectory $\mathbf{u}^*(t) = \mathbf{u}(\mathbf{x}^*(t), t)$. Example of *open-loop* control, where we apply $\mathbf{u}(t)$ regardless of what the state actually is.

With noise (stochastic dynamics), this isn't sufficient: we need to know what input to apply if we're perturbed off the optimal trajectory, and hence $\mathbf{u}$ has to depend on $\mathbf{x}$ (*closed-loop* control). Can improve via a *linear feedback* controller: Taylor expand around $\mathbf{x}^*(t)$ to linear order in dynamics and quadratic in control cost. Then we can solve everything, obtaining a controller that stabilizes $\mathbf{x}^*(t)$ when weak noise is turned on. However, turning on noise perturbs the optimal trajectory from the noiseless result, so we can repeat the construction with this new $\mathbf{x}^*(t)$. This is "differential dynamic programming" or "iterative LQG."

Model predictive control addresses representation in a different way, by only computing $\mathbf{u}(\mathbf{x}, t)$ for states $\mathbf{x}$ as needed. At time t solve the finite-horizon control problem for the interval $[t, t+T]$, but only apply the control for a shorter time $[t, t+dt]$; then solve the finite-horizon problem again from the new state ("receding horizon.") Not globally optimal, but more robust.

Remark 4. *Versus directed polymer in random medium? At fixed* $\mathbf{u}(\mathbf{x}, t)$*, assume dynamics of* $\mathbf{x}$ *follow from least-action. Then functional average (path integral) over all* $\mathbf{u}$*, weighted by cost as* $e^{-\beta R(\mathbf{x},\mathbf{u})}$*, except we want quenched average.*

3.2 Linear-Quadratic regulator

Ref: wiki; this blog post; first chapter of Bertsekas.

3.2.1 Notation

Simplest model of a linear feedback controller. Turn off noise for this section; system state $\mathbf{x}_t$, control action $\mathbf{u}_t$, time t taken discrete. Start with finite-horizon; $t \in [0, T+1]$.

Take dynamics to be deterministic, linear and exactly known:

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t) = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t, \tag{3.1}$$

with boundary condition $\mathbf{x}_0$. We want to find the control trajectory $\mathbf{u}_t^*$ which is optimal in the sense of minimizing cost, assumed quadratic:

$$G = \sum_t g(\mathbf{x}_t, \mathbf{u}_t) = \tfrac{1}{2}\mathbf{x}_{T+1}^T \mathbf{\Psi} \mathbf{x}_{T+1} + \sum_{t=0}^{T} \tfrac{1}{2}\mathbf{x}_t^T \mathbf{Q} \mathbf{x}_t + \tfrac{1}{2}\mathbf{u}_t^T \mathbf{R} \mathbf{u}_t.$$

Drop possible $\mathbf{x}-\mathbf{u}$ cross term in above in the name of simplicity; for the same reason matrices could be t-dependent, but we suppress this. Crunch through solution first, then return to question of what conditions we need on all these matrices.

3.2.2 Via dynamic programming (Bellman)

"Bellman optimality principle," and starting point for applicability of dynamic programming, is that sub-trajectory of an optimal trajectory is itself optimal (assuming convexity, or more broadly absence of *topological* obstructions).

Define "cost-to-go" $J_t(\mathbf{x}_t)$ as cost of sub-trajectory on $[t, T+1]$, starting from initial condition $\mathbf{x}_t$. Careful on notation: J_t has time dependence beyond that from $\mathbf{x}_t$; also implicit dependence on rest of trajectory of $\mathbf{x}, \mathbf{u}$.

Optimal cost-to-go refers to optimizing with respect to $\mathbf{u}_t$. This is done *backwards* in time:

$$J_t^*(\mathbf{x}_t) = \min_{\mathbf{u}_t} \left[g(\mathbf{x}_t, \mathbf{u}_t) + J_{t+1}^*\left(f(\mathbf{x}_t, \mathbf{u}_t)\right)\right] \tag{3.2}$$

"Arrow of time" fixed by need to substitute $f(\mathbf{x}_t, \mathbf{u}_t)$ for $\mathbf{x}_{t+1}$. Boundary condition is set in future, at $T+1$, because no control action in last step.

So $J_t^*(\mathbf{x}_t)$ is like a time-dependent *potential*: need to maintain $\mathbf{x}_t$ as a free variable when solving. Need to retain full functional dependence essentially the main obstacle with reinforcement learning, and motivation for introducing approximation methods.

Boundary condition is $J_{T+1}^*(\mathbf{x}_{T+1}) = \frac{1}{2}\mathbf{x}_{T+1}^T \mathbf{\Psi} \mathbf{x}_{T+1}$; we take this as an ansatz and show that if we have $J_{t+1}^*(\mathbf{x}_{t+1}) = \frac{1}{2}\mathbf{x}_{t+1}^T \mathbf{M}_{t+1}\mathbf{x}_{t+1}$ for some matrix $\mathbf{M}_{t+1}$, this form will be preserved under the backwards recursion, i.e. it implies $J_t^*(\mathbf{x}_t) = \frac{1}{2}\mathbf{x}_t^T\mathbf{M}_t\mathbf{x}_t$ for some other $\mathbf{M}_t$.

Variation with respect to $\mathbf{u}_t$ in (3.2) gives

$$0 = \mathbf{R}\mathbf{u}_t^* + \mathbf{B}^T\mathbf{M}_{t+1}\mathbf{A}\mathbf{x}_t + \mathbf{B}^T\mathbf{M}_{t+1}\mathbf{B}\mathbf{u}_t^*,$$

so

$$\mathbf{u}_t^* = -\left[\mathbf{R} + \mathbf{B}^T\mathbf{M}_{t+1}\mathbf{B}\right]^{-1}\mathbf{B}^T\mathbf{M}_{t+1}\mathbf{A}\mathbf{x}_t \equiv -\mathbf{K}_{t+1}\mathbf{x}_t.$$

This lets us eliminate $\mathbf{u}_t$ in favor of $\mathbf{x}_t$ on the RHS of (3.2), so we can see that the ansatz for the form of J_t is preserved. Explicitly,

$$\mathbf{M}_t = \mathbf{Q} + \mathbf{A}^T\mathbf{M}_{t+1}\mathbf{A} + \mathbf{K}_{t+1}^T\left[\mathbf{R} + \mathbf{B}^T\mathbf{M}_{t+1}\mathbf{B}\right]\mathbf{K}_{t+1} - \left(\mathbf{A}^T\mathbf{M}_{t+1}\mathbf{B}\mathbf{K}_{t+1} + \mathbf{K}_{t+1}^T\mathbf{B}^T\mathbf{M}_{t+1}\mathbf{A}\right),$$

but expanding the third term shows that it's equal to half the term in parentheses, so we obtain the discrete-time algebraic Riccati equation:

$$\mathbf{M}_t = \mathbf{Q} + \mathbf{A}^T\mathbf{M}_{t+1}\mathbf{A} - \mathbf{A}^T\mathbf{M}_{t+1}\mathbf{B}\left[\mathbf{R} + \mathbf{B}^T\mathbf{M}_{t+1}\mathbf{B}\right]^{-1}\mathbf{B}^T\mathbf{M}_{t+1}\mathbf{A}. \tag{3.3}$$

If we make the further assumption that $\mathbf{M}$ is invertible, we can apply the Woodbury identity

$$\left[\mathbf{A} + \mathbf{U}^T\mathbf{C}\mathbf{U}\right]^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}^T\left[\mathbf{C}^{-1} + \mathbf{U}\mathbf{A}^{-1}\mathbf{U}^T\right]^{-1}\mathbf{U}\mathbf{A}^{-1}$$

to obtain

$$\mathbf{M}_t = \mathbf{Q} + \mathbf{A}^T\left[\mathbf{M}_{t+1}^{-1} + \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\right]^{-1}\mathbf{A}. \tag{3.4}$$

3.2.3 Via Pontryagin maximum principle

Constrain dynamics to be physical though Lagrange multipliers ("costates" or "adjoints") λ_t: $L = G - \lambda_{t+1}^T(\mathbf{x}_{t+1} - \mathbf{A}\mathbf{x}_t - \mathbf{B}\mathbf{u}_t)$. Find optimality at extrema of Lagrangian; we have

cf. on-shell vs. off-shell when we move to path integral?

$$\text{Costate equation: } 0 = \frac{\delta L}{\delta \mathbf{x}_t} \Rightarrow \lambda_t = \mathbf{A}^T\lambda_{t+1} + \mathbf{Q}\mathbf{x}_t, \tag{3.5}$$

$$\text{Boundary condition: } 0 = \frac{\delta L}{\delta \mathbf{x}_{T+1}} \Rightarrow \lambda_{T+1} = \mathbf{\Psi}\mathbf{x}_{T+1}, \tag{3.6}$$

$$\text{Stationary equation: } 0 = \frac{\delta L}{\delta \mathbf{u}_t} \Rightarrow \mathbf{u}_t = -\mathbf{R}^{-1}\mathbf{B}^T\lambda_{t+1}, \tag{3.7}$$

$$\text{State equation: } 0 = \frac{\delta L}{\delta \lambda_t} \Rightarrow \mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t. \tag{3.8}$$

Similar to previous section, we find the costate equation describes evolution of λ_t backwards in time from a boundary condition at $T + 1$.

Stationary equation (3.7) can viewed as (related to?) a Legendre transform, exchanging $\mathbf{u}$ for λ as dependent variable. Eliminating $\mathbf{u}_t$ from (3.5), (3.8) gives joint recursion

$$\begin{pmatrix} \mathbf{x}_{t+1} \\ \lambda_t \end{pmatrix} = \begin{bmatrix} \mathbf{A} & -\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \\ \mathbf{Q} & \mathbf{A}^T \end{bmatrix} \begin{pmatrix} \mathbf{x}_t \\ \lambda_{t+1} \end{pmatrix}.$$

Solution proceeds similarly, taking boundary condition as ansatz, and showing that if $\lambda_{t+1} = \mathbf{M}_{t+1}\mathbf{x}_{t+1}$ for some $\mathbf{M}_{t+1}$, this dependence is preserved. Use notation $\mathbf{M}$ because we obtain same discrete-time Riccati equation as above.

Remark 5. *Is there something deeper going on here? A priori, could say that λ_t depends on future trajectory of $\mathbf{x}$s, while $\mathbf{x}_t$ depends on history of $\mathbf{x}$s, so "intersection" is to have λ_t and $\mathbf{x}_t$ depend on each other only at that value of t.*

Substituting in ansatz, we get

$$\mathbf{x}_{t+1} = \left[1 + \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{M}_{t+1}\right]^{-1} \mathbf{A}\mathbf{x}_t,$$

so

$$\lambda_t = \left\{\mathbf{Q} + \mathbf{A}^T\mathbf{M}_{t+1}\left[1 + \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{M}_{t+1}\right]^{-1} \mathbf{A}\right\} \mathbf{x}_t$$

Quantity in curly brackets is $\mathbf{M}_t$; inserting $1 = \mathbf{M}_{t+1}^{-1}\mathbf{M}_{t+1}$ gives us the discrete-time Riccati equation in the form (3.4).

Remark 6. *Relationship to Bellman? Write up relationship between Pontryagin and HJB. Is this a substantive change in what we're solving for — i.e. can we take λ_t to be a field, or is it indirect bookkeeping for the potential $J_t^*(\mathbf{x}_t)$ — seems like we need to retain $\mathbf{x}_t$-dependence, as with form of ansatz.*

Are we just doing $\lambda_t = \delta_{\mathbf{x}_t} J_t^*(\mathbf{x}_t)$?

3.2.4 Infinite-horizon problems

For $T \longrightarrow \infty$, physical solutions to the above are steady states, so fixed points of the Riccati equation $\mathbf{M}_{t+1} = \mathbf{M}_t = \mathbf{M}$.

3.3 Linear-Quadratic control with Gaussian noise

[...]

Linearity means zero-mean, IID noise may be absorbed into a change of variable: $\widetilde{x}_t = x_t - e_{t-1}$. Then $\langle \widetilde{x} Q \widetilde{x} \rangle_e = xQx + \langle eQe \rangle_e$. Maybe better to assume noiseless, then show adding noise doesn't create substantial problems: Achievable *reward* degrades, but the optimal control signal u is identical to the noiseless case.

3.4 Control vs. ML

Ref: [NES21].

(PO)MDPs are problem contexts, while RL refers to a family of algorithms for solving related problems.

- MDP assumes all dynamics is known to the agent at outset, and we only need to learn a policy. "For any MDP, there exists an optimal policy that is both memoryless and deterministic."

Even with stochastic dynamics?

- POMDP assumes only part of the MDP state is accessible to the agent, and is revealed through observations.
- RL assumes dynamics are knowable/fixed, but must be learned by the agent through exploration; it's the problem of learning a fixed MDP (either entirely online, or with offline data).

In addition, dynamics and observations may be deterministic or stochastic.

POMDP formally reducible to MDP: replace partial knowledge of real state with exact knowledge of "belief state," the probability of the real state given the full history of observations up to that point. This obviously invokes the curse of dimensionality.

Conversely, an MDP with uncertainty in its parameters can be modeled as a POMDP. "Bayesian RL" is RL done in the context of a belief distribution over the underlying MDP. Of course, this is intractable.

Harder or easier than general POMDP?

[GRK+21] frames poor generalization of RL from in-sample/offline training in terms of an "epistemic POMDP."

Sub-area	s' in dynamics?	s' in reward?	s' constant?	Policy inputs	RL objective	Domain shift?
Standard POMDP	Y	Y	N	O, A, R	Avg	N
Meta RL	∼N	Y	Y	O, A, R, d	Avg	N
Robust RL	∼Y	∼N	∼Y	O, A	Worst	N
Gen'lization in RL	∼Y	∼N	∼Y	O, A	Avg	∼Y

Table 2.1: From [NES21]. s' refers to the hidden POMDP state; O, A, R, d refer to the sequence of observations, actions, rewards, and done signals, respectively. $\sim$ means the categorization doesn't hold for all work.

Chapter 3

Continuum formulations

1 Probability

1.1 Gaussian Processes

[...]

1.2 Lagrangians

(ref for this (Strang)).
Three forms for all calculus of variations problems:

1. Variational (optimization; here least action principle):

$$\min_x \mathcal{S}[x] = \min_x \left\{ \int dt\, \mathcal{L}[t, x(t), \dot{x}(t)] \right\},$$

with Dirichlet BCs on endpoints $x(0), x(T)$.

2. Weak form (here "principle of virtual work"). Take any test function y, form $\mathcal{S}[x+y] - \mathcal{S}[x]$ and set term linear in y to zero to express optimality.

$$\delta\mathcal{S} = \int dt\, \left[\frac{\partial \mathcal{L}}{\partial x} \delta x + \frac{\partial \mathcal{L}}{\partial \dot{x}} \delta \dot{y} \right] = 0.$$

3. Strong form (here Euler-Lagrange equations): Integrate weak form by parts. Boundary conditions handle surface term.

$$\frac{\partial \mathcal{L}}{\partial x} - \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{x}} = 0.$$

Even simpler matrix case (to motivate KKT et al): $\min_u \frac{1}{2} u' A u - u' b$; then $v' A u = v' b$ for any v, or $Au = f$.

2 Control

2.1 Bellman 2

(ref for this, also wiki).

Switch notation. Control theory only fixes initial endpoint (just adding additional bookkeeping for optimization over final BC, right?). Also deals with differential constraints, so introduce u as a multiplier for $\dot{x}$ and you'd say

$$\begin{aligned} \min_u \mathcal{S}[u] &= \min_u \left\{ \Phi[x(T)] + \int_0^T dt\, \mathcal{F}[t, x(t), u(t)] \right\}, \\ \dot{x} &= g[t, x(t), u(t)]. \end{aligned}$$

where the second equation (first-order dynamic constraints) is called the "state equation" and Φ is the "endpoint cost." Can also have "path constraints" on x, u (no derivatives).

Pontryagin's Maximum Principle just buys us the ability to deal with discontinuity? Replace $\delta\mathcal{H} = 0$ condition with simple $\max \mathcal{H}$?

2.2 Path integral control

Ref: [TBS10].

Finite horizon stochastic control. Make assumption that dynamics *linear* in u_t, reward is *quadratic* in u_t:

$$\dot{x}_t = f(x_t, t) + g(x_t)[u_t + dw_t]; \qquad r_t = q(x_t, t) + \frac{1}{2} u_t^T R u_t,$$

where Gaussian noise dw has variance Σ_w. HJB equation for cost-to-go is then

$$-\partial_t J(x_t, t) = \min_u \left[r_t + (\partial_x J)^T (f_t + g_t u_t) + \frac{1}{2} \operatorname{Tr} g_t \Sigma_w g_t^T \right],$$

which is solved by

$$u^*(x_t) = -R^{-1} g_t^T \partial_x J.$$

Substituting this back into HJB gives a nonlinear PDE for J, which may be linearized by the change of variables $J = -\lambda \log \Psi(x_t, t)$.

How does this relate to Schrodinger/physical notions? λ like $\hbar$...

Crucial point for this method: We need to assume $\lambda R^{-1} = \Sigma_w$, which is expressing the observation that variance of the control input and cost of that input are inversely related. (This is dictated by the need for linearization and isn't a well-motivated assumption in general: it implies that noiseless variables can't be controlled [and vice versa?]).

Under this assumption we get the Chapman-Kolmogorov PDE

$$-\partial_t \Psi = -\frac{1}{\lambda} q_t \Psi + f_t^T \partial_x \Psi + \frac{1}{2} \operatorname{Tr} (\partial_x^2 \Psi) g_t \Sigma_w g_t^T,$$

with boundary condition $\Psi(t_F) = \exp[-\phi(t_F)/\lambda]$. This can be expressed as a path integral using the Feynman-Kac theorem:

$$\Psi(x_i, t_i) = \int d\xi \, \exp -\frac{1}{\lambda} \left(\phi(t_F) + \int_{t_i}^{t_F} dt\, q_t \right),$$

where the integration is over all trajectories ξ starting at $x_i(t_i)$.

Theodorou et al. go on to generalize to the case where g_t is state-dependent and partitioned into controlled $[(c)]$ and non-controlled degrees of freedom. Define generalized cost

$$\begin{aligned} \widetilde{S}(\xi) &= S(\xi) + \frac{\lambda}{2} \int_{t_i}^{t_F} dt\, \log |H(t_j)|, \text{ where} \\ S(\xi) &= \phi(t_F) + \int_{t_i}^{t_F} dt\, \left(q_t + ||\dot{x}_t^{(c)} - f_t^{(c)}||^2_{H_t^{-1}} \right); \\ H_t &= g_t^{(c)} R^{-1} g_t^{(c)T}. \end{aligned}$$

$\widetilde{S}(\xi)/\lambda$, normalized by the associated partition function $\widetilde{Z}$, is the path integral probability measure for the path ξ. The optimal control can be written as an expectation with respect to it:

How does this compare to e.g. MSR for the uncontrolled Langevin dynamics? What would diagrams look like?

$$u^* = \frac{1}{\widetilde{Z}} \int d\xi \, e^{-\frac{\widetilde{S}}{\lambda}} R^{-1} g_t^{(c)T} H_t^{-1} \left[g_t^{(c)} dw_t - \frac{\lambda}{2} H_t \operatorname{Tr}\left(H_t^{-1} \partial_x H_t\right) \right].$$

Chapter 4

Variational/Bayesian autoencoders.

1 Information bottleneck

1.1 Overview

Ref: [TPB99]

Given a source X and target Y, we want a "compressed representation" T that preserves "only the information about X relevant for Y."

In more detail, assume $T \to X \to Y$ is Markov. In other words, we have a factorization assumption

$$p(X, T, Y) = p(T|X, Y)p(Y|X)p(X) = p(T|X)p(Y|X)p(X).$$

The fact that $p(T|X, Y) = p(T|X)$ means T can't "look directly at the labels" in Y. Then we want

- $\min I(T; X)$ to minimize complexity, and
- $\max I(T; Y)$ to maximize accuracy.

This motivates the info bottleneck Lagrangian:

$$\mathcal{L}_{IB} = I(X;T) - \beta I(Y;T).$$

[note that literature differs in minimization vs. maximization, and which term has the β.] This is implemented in terms of stochastic encoding and decoding functions, $p_{\text{enc}}(t|x)$ and $p_{\text{dec}}(y|t)$ respectively. The former is what's minimized when we minimize $\mathcal{L}_{IB}$; for the present case, the latter is fully defined in terms of it via the Markov/factorization assumption:

$$p_{\text{dec}}(y|t) = \sum_x p(x, y|t) = \sum_x p(y|x)p(x|t) = \sum_x p(y|x) \frac{p_{\text{enc}}(t|x)p(x)}{p_{\text{enc}}(t)}.$$

As formulated, $\mathcal{L}_{IB}$ is non-convex, making optimizing p_{enc} difficult.

1.2 Sufficient dimensionality reduction

Ref: [GT03]

Considers learning continuous *features*: a regression problem, rather clustering. Original IB formulated in terms of discrete variables, for which this distinction not really present; the fact that the IB objective involves mutual information only means that it's invariant under reparameterizations, but this is important in practice.

Formulate regression as a sufficient statistics problem; learning $y = f(x) = \langle x \rangle_{p(x|y)}$. "Feature extraction" as functions $\phi(x)$ of one variable which are maximally informative with respect to other variables. Let $\mathcal{P}(\phi)$ be space of joint distributions $\widetilde{p}(x,y)$ having same marginals and $\langle\phi(x)\rangle$ as the real $p(x,y)$. "Info in measurement ϕ" is

$$I_{\text{meas}}(\phi;p)=\min_{\mathcal{P}(\phi)} I(X;Y)=\max_{\mathcal{P}(\phi)} H(X,Y)+\text{const.},$$

Follows this uniquely achieved by the exponential $\widetilde{p}(x,y)\propto\exp\left[\lambda_X(x)+\lambda_Y(y)+\sum_i\lambda_i(y)\phi_i(x)\right]$, where $\{\lambda\}$s are Lagrange multipliers, all of which depend on the choice of $\{\phi\}$. Claim we find optimal $\widetilde{p}^*(x,y)$ from minimizing $D_{KL}[p|\widetilde{p}^*]$, restricted to this exponential form (recall that sufficient statistics exist iff distribution belongs to an exponential family; this is just trying to find the best-fit exponential for p.)

- Problem actually has a symmetry $X\leftrightarrow Y$, $\phi_i(x)\leftrightarrow\lambda_i(y)$.
- Information-geometric interpretation of all this.
- "Most informative features" $\phi^*(x)$ maximize $I_{\text{meas}}(\phi;p)$. How to find systematically?
- Can incorporate "side information" in the form of other variables that we want features to be *un*informative about [GCT12], by adding term to objective function with opposite sign.
- How does this differ from a vanilla variational autoencoder? [BKG18] seek to minimize objective
$$\langle\log q(x|t)+\log q(y|t)\rangle_{q(t|x)}-D_{KL}[q(t|x)|q(t)]$$
for NN encoder $q(t|x)$, decoder $q(x|t)$ and classifier $q(y|t)$.

[...]

1.3 Tishby NIPS 2011 tutorial.

Ref: video on youtube

1.3.1 Sufficient statistics

Bayesian hypothesis testing. Given samples $\mathbf{x}=\{x_i\}$, determine which distribution ω_j they came from (start with distinguishing between two distributions.) Write the information gain Δ about ω provided by $\mathbf{x}$ as

$$\begin{aligned}\Delta(\omega|\mathbf{x})=\frac{p(\omega|\mathbf{x})}{p(\omega)}&=\frac{p(\mathbf{x}|\omega)}{p(\mathbf{x})}=\sum_j p(\mathbf{x}|\omega_j);\\ \text{rewrite as }&=\left(1+\exp\sum_i\log\frac{p(\mathbf{x}|\omega_2)}{p(\mathbf{x}|\omega_1)}\right)^{-1}\\ &=-\log\left(1+\exp\sum_i\log\frac{p(\mathbf{x}|\omega_2)}{p(\mathbf{x}|\omega_1)}\right).\end{aligned}$$

In what follows, define $T(\mathbf{x})=\sum_i\log\frac{p(\mathbf{x}|\omega_2)}{p(\mathbf{x}|\omega_1)}$. This is an additive function of the samples $\{x\}$ only.

Fisher-Neyman factorization: can factorize the *joint* distribution as $p(\mathbf{x},\omega)=f(\mathbf{x})g(\omega,T(\mathbf{x}))$, meaning $T(\mathbf{x})$ is a sufficient statistic for ω. No matter how many samples we have, all that matters is the single number $T(\mathbf{x})$.

Since $T(\mathbf{x})$ is a sum of IID terms, the law of large numbers says

$$\frac{1}{N}\sum^{N}\log\frac{p(\mathbf{x}|\omega_2)}{p(\mathbf{x}|\omega_1)} \xrightarrow[N\to\infty]{} \left\langle \log\frac{p(\mathbf{x}|\omega_2)}{p(\mathbf{x}|\omega_1)}\right\rangle_{p(\mathbf{x},\omega)} .$$

through "typicality": asymptotics are dominated by the average logs. This is basically the KL divergence

$$0 \leq D[\omega_1|\omega_2] = \left\langle \log\frac{\omega_2(\mathbf{x})}{\omega_1(\mathbf{x})}\right\rangle_{\omega_2(\mathbf{x})} .$$

Expected info gain is just the mutual information: $\langle\Delta(\omega|\mathbf{x})\rangle_{p(\mathbf{x},\omega)} = I(X;\Omega) \geq 0$. Note that $T(\mathbf{x})$ grows linearly with N (extensive), while I is subextensive: $\sim \log N$ for a parametric distribution (Cramer-Rao bound on parameter estimation) or $\sim N^{\eta}$ for $0 < \eta < 1$ for a distribution that's nonparametric but still "learnable", cf. minimal description length.

To summarize, this illustrates connections between

- Learning with a binary hypothesis (need more than one $T(\mathbf{x})$ to distinguish between more than two distributions),
- Making an optimal binary decision,
- The optimal linear discriminator (claim here a perceptron on a simplex).

Minimal sufficient statistics are great but exist *only* for exponential families (= Gibbs states?). ML tries to be distribution independent. Recall a sufficient statistic T for a hypothesis θ satisfies $p(\mathbf{x}|T,\theta) = p(\mathbf{x}|T)$. A *minimal* sufficient statistic is an (algebraic?) function of any other sufficient statistic (or vice versa?): it's the coarsest possible "partition" of sample space.

Exponential families: Pitman, Koopman, Darmois.

$$p(x|\theta) = h(x)\exp\left[\sum_r \eta_r(\theta)A_r(x) - A_0(\theta)\right];$$

the maximum-entropy distribution subject to the constraints defined by the $\{A_r\}$. Then sufficient statistics are $T_r(\mathbf{x}) = \sum_i A_r(x_i)$; additive for IID samples.

1.3.2 Info bottleneck

Motivate info bottleneck as the appropriate generalization of sufficient statistics: for the case of exponential families, IB recovers sufficient statistics.

Claim we go beyond the formalism of exponential families and sufficient statistics with *mutual information*, as "the maximum number of independent bits about Y that can be given by measurement of X." Can define MI as the unique measure satisfying both:

1. Data processing inequality: if $X \to Y \to Z$ is Markov then $I(X;Z) \leq I(X;Y)$: a feedforward process can't increase information. Many measures other than I obey this, cf. Renyi entropy, Chisaeu (sp?) divergences.
2. Bregman divergences: averaging inequality.

Sufficiency and information. Bayesian approach: take θ random. Then T is sufficient for θ iff $I(T;\theta) = I(X,\theta)$. S is minimally sufficient if it retains the *least* MI: for any T, $I(S;X) \leq I(T;X)$.

1.3.3 Discrete case

Info bottleneck for clustering: original method proposed in [TPB99]. Self-consistent equations

$$\begin{aligned} p(t|x) &= \frac{p(t)}{Z} \exp -\beta D_{KL}[p(y|x)|p(y|t)], \\ p(t) &= \sum_x p(t|x)p(x), \\ p(y|t) &= \sum_x p(y|x)p(x|t). \end{aligned}$$

Propose solving this iteratively using the first equation to update $p_{\text{new}}(t|x)$, but this is nonconvex. Arimoto-Blahut algorithm: alternating "I-projection" on three convex sets (left-hand sides of above). Proved technical results on convergence from empirical samples, uniqueness, optimality.

1.3.4 Gaussian case

Ref: [CGTW05].

Recover canonical correlation analysis. Bottleneck T is a combination of CCA eignevectors: $T = AX$ where

$$A = [\alpha_1 \mathbf{u}_1, \dots, \alpha_n \mathbf{u}_n]; \qquad (\Sigma_{XY}\Sigma_{XX}^{-1})\mathbf{u}_k = \lambda_k \mathbf{u}_k,$$

and $\alpha_k^2 = \max[0, (\beta(1-\lambda_k)-1)/\lambda_k]$. The $\max[\dots]$ produces discrete structural transitions: β sets the rank of A. cf. Shannon "water filling" analogy. IB tradeoff curve can be obtained analytically in terms of the $\{\lambda_k\}$.

Remark that for self-similar data, the $\{\lambda_k\}$ satisfy a recursion and the IB curve is a power law. How deep is the recursion? How to determine empirically? What about multifractal processes??

1.3.5 Kernel IB

Jacoby and Tishby 2011. "When things aren't Gaussian, make them Gaussian": Embed data (nonlinearly) in a sufficiently high-dimensional space with the "kernel trick," then hope linear analysis on that works. Same method used in support vector machines, kernel PCA, kernel CCA. Means choice of embedding kernel is key.

Q: what if X, Y don't have finite second moments?

1.3.6 Predictive information and control

Estimation and control → compression and prediction. Not all info from the past is usable for predicting the future. Want to find this (cf. rate distortion coding) and perform a past-future IB. Past info as a "perception channel" and future state as a "prediction channel." "We see what we expect to see"— perception guided by prediction. Coarse grained variables are predictable further into the future.

Partially-observed Markov decision process: add a hidden Markov model to a MDP. Hidden state of world W, observed state M, observation channel O and action A. Claim that if observations reveal full state of the world, we're back to an MDP and memory isn't needed.

Reinforcement learning: assign reward to each transition in world $W_t \to W_{t+1}$. Also introduce an "intrinsic reward" for uncovering more info from observations $M_t \to M_{t+1}$. Switch to discrete setting for tractability: stochastic MDP defines states s, actions a, transitions $p(s'|s,a)$. Stochasticity because we can't be certain what state we're in.

Planning problem: want optimal policy $\pi(a|s)$ maximizing expected future reward. Bellman optimality: see Emo Todorov, Bert Kappen, Karl Friston.

1.4 Recent work on info bottleneck.

1.4.1 Related work and follow-ups

[GP20] is a recent review of many of the refs below.

- [SS17]; code. Proposes to replace the “soft”/stochastic cluster assignments generated by IB with “hard”/deterministic ones through the use of $\mathcal{L}_{DIB} = H(T) - \beta I(Y;T)$, where minimization still done over cluster assignments $p(t|x)$. $\mathcal{L}_{DIB} - \mathcal{L}_{IB} = H(T|X)$, so IB encourages stochasticity in its assignments. Claim DIB solution performs similar to IB solution in terms of IB loss, while being a significant improvement in DIB terms, while converging faster (for an Arimoto-Blahut-type algorithm.)
- [KTK18]; comments, code. For the case where Y is a deterministic function of X, the MI tradeoff curve can’t be explored by varying β (because it’s piecewise-linear, not concave) and for all β find trivial solutions obtained by probabilistically “forgetting” a portion of X. Propose to fix this via modifying $\mathcal{L}'_{IB} = I(X;T)^2 - \beta I(Y;T)$. Same problems arise in DIB, and are fixed with $\mathcal{L}'_{DIB} = H(T)^2 - \beta I(Y;T)$.

“Units”?

- [RGTS20]; code clarify results of the above: I^2 can be replaced with any convex function, and this can be used to relate β to the achieved compression rate.
- [NS21]; comments. “Perturbation theory” in that perturbation is done around the nonzero threshold β_c below which the representation T is uninformative ($I(T;Y) = I(T;X) = 0$; [WFCT19] for more on this phenomenon). The perturbation is done around an uninformative $p_{\text{enc}}(t|x) = p(t)$. Nice but not usable for applications.
- [HG21]; code. Show that convergence can be guaranteed with ADMM if the state space is augmented with the marginal $p(t)$. Legit?

1.4.2 Deep variational IB

Ref: [AFDM16]; code.

Makes IB implementable using a neural network for encoding/decoding. [CMT16] does the same with kernels; claim this is more efficient.

Derivation of the variational bound: let $q(Y|T)$ be an approximation to the true decoder. Then $D_{KL}[p(Y|T), q(Y|T)] \geq 0$ implies

$$\begin{aligned} I(T;Y) &\geq \sum_{y,t} p(y,t) \log \frac{q(y|t)}{p(y)} = H(Y) + \sum_{y,t} p(y,t) \log q(y|t), \\ &\geq \sum_{x,y,t} p(x)p(y|x)p(t|x) \log q(y|t). \end{aligned}$$

In the last line we inserted the Markov factorization and dropped $H(Y)$ since it’s independent of T. For a bound on the other term $I(X;T)$, we likewise need an approximation $q(t)$ to the true marginal $p(t)$. Similar considerations give

Should be able to do better: take follow-up papers to MINE estimator or [PO19].

$$I(T;X) \leq \sum_{x,t} p(x)p(t|x) \log \frac{p(t|x)}{q(t)}.$$

Combining these yields an upper bound on $\mathcal{L}_{IB}$.

Propose to actually evaluate this by plugging in the empirical distribution for $p(x, y)$ and using the "reparameterization trick": write $t = f(x, \epsilon)$ as a deterministic function of x and a Gaussian random variable ϵ. Then $p(t|x)dt = p(\epsilon)d\epsilon$. Propose to do this by using a neural net to represent mean and covariance of T??

Could regularize $p(x, y)$ and do better?

Note that variational formulation breaks reparameterization (copula) invariance present in real IB [WWMR18, WR20].

1.4.3 Variational Predictive IB

Ref: [Ale20].

Specialize above to the past-future IB case, where X is the observable past of a timeseries and Y is its future. Need modification because we haven't observed the future; use Markov property that T and Y are conditionally independent given X. This means $I(T;Y) = I(T;X) - I(T;X|Y)$: The conditioned MI term avoids the need to know the future: it measures the inefficiency of T, as measured after we know the future. Our objective is

$$\min_{p(t|x)} I(T;X|Y) - \beta I(T;X).$$

Assuming the posterior $q(X|T)$ factorizes (claim this isn't necessary and can be replaced by a better approximation), this is

$$\min_{p(t|x)} \left\langle \log \frac{p(t|x)}{q(t)} - \beta \sum_x \log q(x|t) \right\rangle.$$

Refer to [AF18] (comments) for more refined approximations than used here.

1.4.4 Conditional entropy bottleneck

Ref: [Fis20], comments; [FA20], comments.

Proposes to address non-informative encodings by attempting to reach the "minimum necessary information" point, at which $I(X;Y) = I(X;T) = I(Y;T)$; this is not always achievable. Proposes

$$\min_T I(X;T|Y) - \gamma I(Y;T);\ \text{minimized when}$$
$$\min_T -H(T|X) + H(T|Y) + \gamma H(Y|T)\ \text{is.}$$

For deterministic $X \longrightarrow Y$, achieve MNI at $\gamma = 1$. Equivalent to IB at $\gamma = \beta - 1$ under Markov assumption, since then $I(X;T|Y) = I(X;T) - I(Y;T)$; not identical because we dropped $H(Y)$ in second line.

Variational implementation via learning *three* functions (similar to VIB): $q_{\text{enc}}(t|x)$, such that joint $p(x, y, t) = p(x, y)q_{\text{enc}}(t|x)$; "classifier" $q_{\text{dec}}(y|t)$ and "backward encoder" $q_{\text{dec}}(t|y)$ instead of VIB's marginal $q(t)$. Argue this gives a tighter bound than VIB (not necessarily; [GF20]):

$$\min_{\text{all } q\text{s}} \left\langle \log \frac{q_{\text{enc}}(t|x)}{q_{\text{dec}}(t|y)} - \gamma \log q_{\text{dec}}(y|t) \right\rangle_{p(x,y)q_{\text{enc}}(t|x)}.$$

Do we need constraints to keep these three functions consistent?

Discusses several extensions. One is to hierarchical models $Y \leftrightarrow X = T_0 \longrightarrow T_1 \longrightarrow T_2 \longrightarrow \cdots$:

$$\min_{\{T_i\}} \sum_i -H(T_i|T_{i-1}) + H(T_i|Y) + H(Y|T_i).$$

Another is the predictive IB setting. Can simply plug in $X = X_<; Y = X_\geq$ above. Can also work in the “bidirectional” context, where we learn two representations, $T_<$ and $T_\geq$:

$$\min_{T_<, T_\geq} [-H(T_<|X_<) + H(T_<|X_\geq) + \gamma H(X_\geq|T_<)] + [(< t) \leftrightarrow (\geq t)].$$

These are tied together by using the same encoder and backwards encoder. Introducing a fourth “decoder” distribution $q_{\text{enc}}(x|t)$,

$$\min_{\{q\}} \left\langle \log \frac{q(t_<|x_<)q(t_\geq|x_\geq)}{q(t_<|x_\geq)q(t_\geq|x_<)} - \gamma \log q(x_\geq|t_<)q(x_<|t_\geq) \right\rangle_{p(x,y)q(t_<|x_<)q(t_\geq|x_\geq)} .$$

Propose to address multi-scale time series analysis by combining these objectives: each level of the hierarchy of T_is would correspond to greater smoothings, conditioned on the set of T_{i-1}s. Reference WaveNet [vdODZ+16, vdOLB+17] as an example of a multi-scale neural architecture.

1.5 IB applications to RL/control.

1.5.1 InfoBot

Ref: [GIS+19].

Apply IB for regularization in RL (for increased generalization, avoiding overfitting). Specifically, want to minimize policy dependence on the goal as measured by $I(A; G|S)$. “Goal” G seems to refer to variable but undesirable details of training data, like the location of the goal in a maze. This is equivalent to a KL regularization term where we penalize deviations of the policy from a “default” policy that integrates out dependence on G:

$$\pi_0 = \sum_g p(g)\pi(A|S, g).$$

Refer to system states S where we it’s worth deviating from this default as “decision states”; reward exploration by incentivizing agent to seek these out.

Why do they think these are isolated?

Adds an extra variable to the Markov structure of IB: $S, G \longrightarrow T \longrightarrow A$, but in addition $S \longrightarrow A$ (formally distinguishing the roles of S and G). Build policy from IB encoder/decoder:

$$\pi(A|S, G) = \sum_t p_{\text{dec}}(A|S, t)p_{\text{enc}}(t|S, G).$$

Cost-to-go is $J(\pi) = \langle r \rangle_\pi - \beta I(A; G|S)$; we bound the MI with $I(T; G|S)$. This requires the marginal $p(T|S) = \sum_g p(g)p_{\text{enc}}(T|S, g)$, which is difficult (goal G plays role of the “future”; we probably have poor knowledge about the out-of-sample $p(G)$ the agent will encounter.) Replace it with a variational approximation $q(T|S)$ to get the lower bound used in practice:

$$J(\pi) \geq \widetilde{J}(\pi) = \langle r - \beta D_{KL}[p_{\text{enc}}(T|S, G)|q(T|S)]\rangle_\pi ,$$

with parameter update rule (under the “Reinforce” algorithm; Monte Carlo policy gradient)

$$\nabla_\theta \widetilde{J}(\pi)\Big|_t = \left(\sum_{t'=t}^{T} \gamma^{t'-t}\widetilde{r}_{t'}\right) \log \pi(a_t|s_t, g_t) - \beta\nabla_\theta D_{KL}[p_{\text{enc}}(T|s_t, g_t)|q(T|s_t)],$$

where the modified reward

$$\widetilde{r}_t = r_t + \beta D_{KL}[p_{\text{enc}}(T|s_t, g_t)|q(T|s_t)].$$

Again, still seeking to maximize reward, rather than an info-theoretic quantity, which distinguishes this from IB. Correspondence with variational IB is recovered if we were to replace $\langle r \rangle$ with $I(A^*; A|S)$, where A^* is the true optimal action — this gives the VIB objective between G and A^*, conditioned on S.

1.5.2 Variational Bandwidth bottleneck

Ref: [GBBL19].

One problem with variational IB is that encoder needs access to full input training, so itself can be overfitted, in the sense that it might not compress new inputs. Fix by defining two classes of input: "standard" S and "privileged" G (assumed independent here). These deliberately map onto the state/goal representations in InfoBot. We want to avoid using G, either because we want to generalize with respect to it, or because it's intrinsically expensive to obtain/calculate. Now we minimize conditional MI between T and G given S; the algorithm makes decisions on whether to invoke G before looking at it (InfoBot always accesses G.)

For example, it could be model output...

With variables and Markov dependencies as in InfoBot, the bound used is

$$I(T;G|S) \leq \sum_{s,g} p(s)p(g)D_{KL}[p_{\text{enc}}(T|s,g)|q(T)].$$

How do we decide when to use G? Refer to a "budget" (channel capacity d_c, taken between 0 and 1). Could just stochastically choose between a deterministic encoder $f(S,G)$, and the "prior" $q(T)$. This binary choice is non-differentiable, though. Instead define a function $d_c = B(S)$, to be parameterized by a NN.

$$D_{KL}[p_{\text{enc}}(T|s,g)|q(T)] = -d_c \log d_c + (1-d_c)\left\{\log p(f(s,g)) - \log[d_c p(f(s,g)) + (1-d_c)]\right\}.$$

In here as in InfoBot, assume $q(T)$ is a unit Gaussian. Why? Not sensitive to details?

Need to see an implementation for this to make sense.

1.5.3 Predictive info soft actor-critic

Ref: [LFL+20]; code.

[...]

1.5.4 Robust predictable control

Ref: [ESL21]; code.

[...]

2 Variational autoencoders

"GANs bypass any inference of latent variables, and auto-regressive models abstain from using latent variables. VAEs jointly learn an inference model and a generative model, allowing them to infer latent variables from observed data."

Chapter 5

Inital write-up of handwritten notes

Keyword: *approximate Bayesian computation*, when observations/likelihood evaluations are expensive. *Active learning/experimental design*, in that we want to plan likelihood evaluations to maximize information gained from each.

1 First batch

1.1 Huan RL for experimental design

"Passive POMDP": Want actions to only change knowledge, not the observable.

World dynamics $p(W_t|W_{t-1})$, t-indep. Agent *memory* $q_t(M_t|M_{t-1}, O_t)$. Noisy observations $\sigma(O_t|W_t)$. Cost to be minimized: $\langle \frac{1}{N}\sum_t d(W_t, M_t)\rangle$.

Unbounded agent: let M_t be belief state (about W, given *all* $\{O_t\}$, i.e. $M_t = B_t(W_t|\{O_t\})$. Bayes update is

$$B_t(W_t|\{O_t\}) \propto \sum_{W_{t-1}} B_{t-1}(W_{t-1}|\{O_{t-1}\})p(W_t|W_{t-1})\sigma(O_t|W_t).$$

(so Huan turns off dynamics for $W_t = \theta$; $\sigma(\cdot)$ is $y = G(\theta) + \eta$).

Versus Huan's terms: experiment index $k = 1 \dots N$, belief state X_k^B, physical state X_k. Design d_k is parameters for kth experiment, or policy $\pi_k(X_k) = d_k$. Observations y_k taken to be $G_k(\theta, d_k) + \eta$, with static parameters θ. Let $z_k = (y_k, d_k)$, and Z_k be the full history of zs up to experiment k.

Can write Bellman for this. Reward/cost $g_k(x, y, d)$ with terminal $g_N(x_N)$. Dynamical update $X_{k+1} = F_k(x, y, d)$.

Bayes for θ:

$$p(\theta|Z_k) = \frac{p(y_k|\theta, d_k, Z_{k-1})}{p(y_k|d_k, Z_{k-1})}p(\theta|Z_{k-1}).$$

Set terminal reward to info we've gained from all experiments:

$$g_N = D_{KL}[p(\theta|Z_N)|p(\theta|\emptyset)],$$

claim this means Bellman is non-convex? Each g_k is the cost (in terms of resources) to run each experiment.

Then Bellman

$$J_k(x_k) = \max_{d_k}\langle g_k + J_{k+1}(F)\rangle$$

$$\pi_k(x_k) = \arg\max_{d_k} Q_k(x, d).$$

(Cost-to-go J_k is the "critic"; π is the "actor.") "Model free" since can go $Q \longrightarrow \pi$ without explicit knowledge of F.

Remark 7. *Remarks:*

1. *Bring in Tishby stuff for dimensionality, instead of directly discretizing pdfs.*
2. *How's D_{KL} deal with singular models?*
3. *In 2021 paper, get exploration via noise (sec. 3.2.4). Want to do this via entropy-like term in g_k.*
4. *MINE-type estimators for D_{KL}.*
5. *2021 paper uses vanilla actor-critic; bring in soft actor-critic.*
6. *Setting is for experimental design. What about optimization, i.e. "find θ in best agreement with summary statistics"? Just tack on summary statistic and comparison into G? Could plug into CES: $\{y_k\}$ are the summary statistics, $\{d_k\}$ are the model parameters for each run.*

1.2 CES again

Remark 8.

- *Issue with CES is that the stages don't "know about" each other, but do target max posterior density. Minimize cost of emulator evaluations for MCMC — e.g. with random features. Dimensional reduction 1: only stuff for* functions *of y: suffice to do $p(y|\theta)$? Or could compress $\theta \longrightarrow y$ itself.*
- *"C" then separate "S" feels redundant.*
- *Emulator serves a necessary function of smoothing data (purely from η in this model, right? Spectrum?) but done ad hoc. Does enKF break on multimodal distributions?*
- *Calibrate $\longrightarrow$ enKF $\longrightarrow$ PI control a la Kappen? Or "smoothing" = Sample? Want RL agent to plan model evaluations; smoothing then done offline.*
- *Can we do anything clever with FDT or differentiation with respect to θs to estimate $\mathrm{cov}(\widehat{y}, \theta)$? Or θ dependence given GP kernel?*

2 Second batch

3 Third batch

3.1 Normalizing flows for inverse inference

3.2 Liu-Wang '16

3.3 Marzouk ATM algorithm

3.4 SNPLA

3.5 SNPE-C

Ref: Greenberg '19

Problem: after first $\{x, \theta\}$ samples, want to update our proposal, *but* if we don't use prior, we don't learn the real posterior.

3.6 Contrastive learning

Ref: Durkan '20; nothing new on SNPE-C per se.

3.7 SNVI

Better likelihood?

3.8 Active learning for SNPE-C/SNVI

3.9 Blau '22

Chapter 6

Applications

1 Data assimilation

2 Portfolio theory

2.1 Motivation: betting strategies.

2.1.1 Kelly criterion

[CT91] chapter 6; Kelly 1956.

Let I be a discrete random variable representing the winner of a horse race. Horse i has probability p_i to win; assume a bookmaker offers odds $1/q_i$ on this horse. For now assume the odds are "fair" in that $\sum_i q_i = 1$; more realistic is "subfair" odds $\sum_i q_i > 1$, where the house always takes a cut. A gambler bets a fraction f_i of their bankroll on horse i; $\sum f_i = 1$. The relative payoff is f_i/q_i if i wins, otherwise 0.

For a single race, we might want to put everything on the bet with the largest expected payoff $\langle 1/q_i \rangle_p$, although this carries a large chance of being wiped out completely. To quantify risk, we instead consider the setting of maximizing terminal wealth after a series of repeated races. Treating the wealth V_n after n races as a stochastic variable,

$$V_n(I_1, I_2, \ldots, I_n) = V_0 \prod_{t=1}^{n} \frac{f_{i,t}}{q_{i,t}};$$

(assume probabilities and odds remain constant). We've assumed race outcomes $\{I_1, I_2, \ldots, I_n\}$ are iid $\sim p(x)$. The product structure, which arises simply due to the fact that we can reinvest our previous winnings, means that $\{\log f_{i,t}/q_{i,t}\}_t$ are also statistically independent. Then invoke the weak law of large numbers [i.e., for large n outcome i will happen np_i times] to deduce that

Understand terms of this statement better: "convergence in probability."

$$\frac{1}{n}\log V_n \xrightarrow[p]{} \left\langle \log \frac{f_i}{q_i} \right\rangle_p = \left\langle \log \frac{f_i}{p_i}\frac{p_i}{q_i} \right\rangle_p = D_{KL}(p|q) - D_{KL}(p|f),$$

expressed in terms of KL divergences. Terminal wealth grows exponentially with n as $\langle \log f_i/q_i \rangle_p$, and this latter quantity is what we want to maximize. Include a lagrange multiplier λ enforcing $\sum_i f_i = 1$, then

$$\arg\max_f \left[\langle \log f_i/q_i \rangle_p + \lambda \sum_i f_i \right] \Rightarrow f_i^* = -\frac{p_i}{\lambda^*} = p_i,$$

with the optimal growth rate

$$W^*[I] = \left\langle \log \frac{f_i^*}{q_i} \right\rangle_p = -\langle \log q_i \rangle_p - H[I].$$

This is Kelly's result ("proportional gambling.") Returning to the KL divergence interpretation, we can only make money (again, in the large-*n* sense) if our estimate of the true distribution p, as expressed through our bets f, is closer in the KL sense than the bookmaker's estimate q.

2.1.2 Side information in the horse race.

Same assumptions as above, but we want to condition our strategy on an arbitrary stochastic variable Y.

Formally, need only replace $p_i \to p(i, y)$ and $f_i \to f(i|y)$ in the derivation above; now have conditioned growth rate

$$W^*[I|Y] = \sum_{i,y} p(i, y) \log \frac{f^*(i|y)}{q_i} = -\langle \log q_i \rangle_p - H[I|Y].$$

This is still optimized by $f^*(i|y) = p(i|y) = p(i, y)/p(y)$.

We see that $W^*[I|Y] - W^*[I] = I(I; Y)$: the increase in growth rate provided by knowing y is equal to the mutual information Y provides about the race outcome I. In a more realistic portfolio scenario, this is only an upper bound, obtained when the market takes the horse racing form.

2.1.3 Optimal gambling is optimal coding/compression

Both concern themselves with estimation of the distribution p underlying an iid process $\{I_1, I_2, \ldots, I_n\}$. Made explicit in [CT91] 6.5.

[...]

2.1.4 Observations on Kelly

- The above criteria don't tell us how to find an edge, only how to best exploit it once it's identified. If the bookmaker offers perfect odds $q(x) = p(x)$, the optimal solution is not to bet at all [after first extending the scenario to include this option; optimal f^* in this case now depends on the offered odds $1/q(x)$].
- We assumed stationarity across different races, duh.
- It may take an infeasibly long time to reach the asymptotic growth regime. You might not be able to find enough opportunities with edge to make it into this regime.
- *Opportunity costs*: the Kelly fraction for an asset can only be calculated with reference to the entire universe of opportunities. "Common error," since it leads to overestimating the fraction, and Kelly is generally the most concentrated it makes sense to be [Thorp 08].

2.2 Optimal portfolios.

2.2.1 Kelly criterion in the portfolio context

The horse race may be viewed as a specialized case of the general portfolio allocation problem (in which "returns" are only accumulated for one period with a winner-take-all structure.)

[...]

The above considerations suggest maximizing $\langle \log r \rangle$. This is at odds with classical portfolio theory due to Markowitz, which regards the allocation question as a tradeoff between risk and expected return.

BUT log-optimal reduces to mean-variance in a limit [which?]

Properties of the log-optimal portfolio [in MacLean et al 11]:

- Log utility maximizes asymptotic growth; it's *myopic*/greedy in the sense that maximization period-by-period (independent of history) yields the global maximum [Kelly 56].
- The ratio of a log-optimal portfolio to any "essentially different" strategy (must differ from it an extensive fraction of the time, but this is not sufficient; this is subtle. See [Thorp 08].) grows asymptotically without bound. The log-optimal portfolio minimizes the expected time to reach a value threshold. [Breiman 61].
- For arbitrary return distribution, log-optimality maximizes asymptotic growth rate [Algolet and Cover 88].
- The log-optimal portfolio is always on the geometric mean-variance frontier (geometric rather than arithmetic means are appropriate for the multi-period setting). Arithmetic MV-efficient porfolios are generally not geometrically efficient, and vice versa [Thorp 71].
- Log utility is myopic for general return distributions [Hakansson 71].

2.2.2 Growth-optimal portfolios

[CT91] ch. 16; [CL06] ch. 10.

In an iterated context, the principle underlying Cover's universal portfolio is *volatility harvesting*, an essentially mean-reverting strategy (if only because volatility, by itself, has no drift by definition). As a minimal example, consider two assets, one of which remains constant in value and the other of which oscillates at a fixed frequency. If our portfolio is *constantly rebalanced* to maintain a constant ratio of value between the two assets, we sell the oscillating asset at its highs and buy it at its lows.

This is independent of a choice of the stochastic process generating the returns.

This is infeasible in practice, because "volatility" is a high-frequency phenomenon, so this would require high volatility and infeasibly frequent rebalancing. It would fall short in

What about multiply-leveraged ETFs?

[...]

[what's the relationship between Cover and Kelly?]

Claim for the Cover portfolio is that asymptotic performance is equal to the best constant [wealth fraction] rebalanced portfolio, chosen in hindsight (i.e., with access to future price series.)

2.3 Control/RL for the trading problem.

Trading as the full problem, including execution/timing, as opposed to portfolio optimization.

As described above, impact of Kelly is just to use $\langle \log r \rangle$ as the objective in the control problem. Non-linear reward means Bellman recursion is non-linear; rules out some algorithms.

2.3.1 Framing the problem.

Ideally we'd like a hierarchy of approximations, which could be expressed perturbatively in respective algorithms.

1. Roughest to assume zero impact: this is would be MDP where state is {prices, holdings}, and control inputs (policy actions; trades) only affect the latter.

2. Could then add a rough slippage/impact model to let trades affect prices directly.
3. Next step would be to add estimation of the parameters of the stochastic process underlying prices to control.
4. Ditto for parameters of slippage model.

Remark that vanilla RL addresses the case where the reward function is unknown and can only be determined through active exploration. Neither of these assumptions apply to trading in the regimes we'd want to be active in:

- Prices are transparent and we can mark-to-market at any time. This breaks down when assets are hard to value (through not only illiquidity but non-standardization: say real estate or residential MBSes).
- Again due to price transparency, we can backtesting a hypothetical strategy straightforwardly. This breaks down in situations of high slippage/impact (illiquidity), closer to a multi-armed bandit problem in which we can only use the information of market impact from our previous trades.

See [BBD+17] for implementation of the MDP problem.
Refer to sec. **3.4**.

- POMDP → belief MDP. Kelly (or rather, its Bayesian extension) is a deterministic mapping from beliefs to bet sizes (actions).
-

References

[AF18] A. A. Alemi and I. Fischer. TherML: Thermodynamics of Machine Learning. *arXiv:1807.04162 [cond-mat, stat]*, October 2018. arXiv:1807.04162. Cited on p. 27.

[AFDM16] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy. Deep Variational Information Bottleneck. November 2016. Cited on p. 26.

[Ale20] A. A. Alemi. Variational Predictive Information Bottleneck. In *Proceedings of The 2nd Symposium on Advances in Approximate Bayesian Inference*, pages 1–6. PMLR, February 2020. Cited on p. 27.

[BBD+17] S. Boyd, E. Busseti, S. Diamond, R. N. Kahn, K. Koh, P. Nystrup, and J. Speth. Multi-Period Trading via Convex Optimization. *arXiv:1705.00109 [math, q-fin]*, April 2017. arXiv:1705.00109.

[BBR+18] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm. Mutual Information Neural Estimation. In *International Conference on Machine Learning*, pages 531–540. PMLR, July 2018. Cited on p. 5.

[BKG18] E. Banijamali, A.-H. Karimi, and A. Ghodsi. Deep Variational Sufficient Dimensionality Reduction. *arXiv:1812.07641 [cs, stat]*, December 2018. arXiv:1812.07641. Cited on p. 23.

[CAH+19] C. Chan, A. Al-Bashabsheh, H. P. Huang, M. Lim, D. S. H. Tam, and C. Zhao. Neural Entropic Estimation: A faster path to mutual information estimation. May 2019. Cited on pp. 5, 6.

[CGTW05] G. Chechik, A. Globerson, N. Tishby, and Y. Weiss. Information Bottleneck for Gaussian Variables. *Journal of Machine Learning Research*, 6(Jan):165–188, February 2005. Cited on p. 25.

[CHD+20] P. Cheng, W. Hao, S. Dai, J. Liu, Z. Gan, and L. Carin. CLUB: A Contrastive Log-ratio Upper Bound of Mutual Information. *arXiv:2006.12013 [cs, stat]*, July 2020. arXiv:2006.12013. Cited on p. 6.

[CL06] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, March 2006.

[CL20] K. Choi and S. Lee. Regularized Mutual Information Neural Estimation. November 2020. arXiv:2011.07932. Cited on p. 6.

[CMT16] M. Chalk, O. Marre, and G. Tkacik. Relevant sparse codes with variational information bottleneck. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1957–1965. Curran Associates, Inc., 2016. Cited on p. 26.

[CT91] T. M. Cover and J. A. Thomas. *ELEMENTS OF INFORMATION THEORY*. 1991.

[ESL21] B. Eysenbach, R. Salakhutdinov, and S. Levine. Robust Predictable Control. *arXiv:2109.03214 [cs]*, September 2021. arXiv:2109.03214. Cited on p. 29.

[FA20] I. Fischer and A. A. Alemi. CEB Improves Model Robustness. *Entropy*, 22(10):1081, September 2020. arXiv:2002.05380, doi:10/gk77v6. Cited on p. 27.

[Fis20] I. Fischer. The Conditional Entropy Bottleneck. *Entropy*, 22(9):999, September 2020. arXiv:2002.05379, doi:10.3390/e22090999. Cited on p. 27.

[GBBL19] A. Goyal, Y. Bengio, M. Botvinick, and S. Levine. The Variational Bandwidth Bottleneck: Stochastic Evaluation on an Information Budget. September 2019. Cited on p. 29.

[GCT12] A. Globerson, G. Chechik, and N. Tishby. Sufficient Dimensionality Reduction with Irrelevant Statistics. *arXiv:1212.2483 [cs, stat]*, October 2012. arXiv:1212.2483. Cited on p. 23.

[GF20] B. C. Geiger and I. S. Fischer. A Comparison of Variational Bounds for the Information Bottleneck Functional. *Entropy*, 22(11):1229, November 2020. doi:10/gn2zfz. Cited on p. 27.

[GIS+19] A. Goyal, R. Islam, D. Strouse, Z. Ahmed, M. Botvinick, H. Larochelle, Y. Bengio, and S. Levine. InfoBot: Transfer and Exploration via the Information Bottleneck. *arXiv:1901.10902 [cs, stat]*, April 2019. arXiv:1901.10902. Cited on p. 28.

[GP20] Z. Goldfeld and Y. Polyanskiy. The Information Bottleneck Problem and its Applications in Machine Learning. *IEEE Journal on Selected Areas in Information Theory*, 1(1):19–38, May 2020. doi:10.1109/JSAIT.2020.2991561. Cited on p. 26.

[GR03] Z. Ghahramani and C. Rasmussen. Bayesian Monte Carlo. In *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2003. Cited on p. 8.

[GRK+21] D. Ghosh, J. Rahme, A. Kumar, A. Zhang, R. P. Adams, and S. Levine. Why Generalization in RL is Difficult: Epistemic POMDPs and Implicit Partial Observability. *arXiv:2107.06277 [cs, stat]*, July 2021. arXiv:2107.06277. Cited on p. 18.

[GT03] A. Globerson and N. Tishby. Sufficient dimensionality reduction. *jmlr.org*, 3:1307–1331, February 2003. Cited on p. 22.

[HG21] T.-H. Huang and A. E. Gamal. A Provably Convergent Information Bottleneck Solution via ADMM. *arXiv:2102.04729 [cs, math]*, May 2021. arXiv:2102.04729. Cited on p. 26.

[KR16] H. J. Kappen and H. C. Ruiz. Adaptive Importance Sampling for Control and Inference. *J Stat Phys*, 162(5):1244–1266, March 2016. doi:10.1007/s10955-016-1446-7. Cited on p. 14.

[KTK18] A. Kolchinsky, B. D. Tracey, and S. V. Kuyk. Caveats for information bottleneck in deterministic scenarios. In *International Conference on Learning Representations*, September 2018. arXiv:1808.07593. Cited on p. 26.

[LFL+20] K.-H. Lee, I. Fischer, A. Liu, Y. Guo, H. Lee, J. Canny, and S. Guadarrama. Predictive Information Accelerates Learning in RL. *arXiv:2007.12401 [cs, math, stat]*, October 2020. arXiv:2007.12401. Cited on p. 29.

[LMGW20] R. Liao, D. Moyer, P. Golland, and W. M. Wells. DEMI: Discriminative Estimator of Mutual Information. *arXiv:2010.01766 [cs, stat]*, November 2020. arXiv:2010.01766. Cited on p. 5.

[LSN+19] X. Lin, I. Sur, S. A. Nastase, A. Divakaran, U. Hasson, and M. R. Amer. Data-Efficient Mutual Information Neural Estimator. May 2019. arXiv:1905.03319. Cited on p. 5.

[Mac03] D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003. Cited on pp. 7, 8, 9, and 10.

[MP92] E. Marinari and G. Parisi. Simulated Tempering: A New Monte Carlo Scheme. *EPL*, 19(6):451–458, July 1992. doi:10/bvn22s. Cited on p. 10.

[MS20] D. McAllester and K. Stratos. Formal Limitations on the Measurement of Mutual Information. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, pages 875–884. PMLR, June 2020. Cited on p. 5.

[Nea01] R. M. Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, April 2001. doi:10/cgjxp4. Cited on p. 10.

[NES21] T. Ni, B. Eysenbach, and R. Salakhutdinov. Recurrent Model-Free RL is a Strong Baseline for Many POMDPs. *arXiv:2110.05038 [cs]*, October 2021. arXiv:2110.05038. Cited on p. 18.

[NS21] V. Ngampruetikorn and D. J. Schwab. Perturbation Theory for the Information Bottleneck. *arXiv:2105.13977 [cond-mat, physics:physics]*, October 2021. arXiv:2105.13977. Cited on p. 26.

[NWJ10] X. Nguyen, M. J. Wainwright, and M. I. Jordan. Estimating Divergence Functionals and the Likelihood Ratio by Convex Risk Minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, November 2010. doi:10.1109/TIT.2010.2068870. Cited on p. 5.

[Pan03] L. Paninski. Estimation of entropy and mutual information. *Neural Comput.*, 15(6):1191–1253, June 2003. doi:10.1162/089976603321780272. Cited on p. 5.

[PO19] B. Poole and S. Ozair. On variational lower bounds of mutual information. page 9, 2019. Cited on pp. 6, 26.

[PW96] J. G. Propp and D. B. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures & Algorithms*, 9(1-2):223–252, 1996. doi:10/c9pks9. Cited on p. 10.

[RGTS20] B. Rodríguez Gálvez, R. Thobaben, and M. Skoglund. The Convex Information Bottleneck Lagrangian. *Entropy*, 22(1):98, January 2020. doi:10.3390/e22010098. Cited on p. 26.

[SE20] J. Song and S. Ermon. Understanding the Limitations of Variational Mutual Information Estimators. *arXiv:1910.06222 [cs, math, stat]*, March 2020. arXiv:1910.06222. Cited on p. 6.

[SS17] D. Strouse and D. J. Schwab. The Deterministic Information Bottleneck. *Neural Computation*, 29(6):1611–1630, June 2017. doi:10/gbgzhn. Cited on p. 26.

[TBS10] E. Theodorou, J. Buchli, and S. Schaal. A Generalized Path Integral Control Approach to Reinforcement Learning. *J. Mach. Learn. Res.*, 11:3137–3181, December 2010. Cited on p. 20.

[TPB99] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. In *The 37th Allerton Conference on Communications, Control and Computing*, pages 368–377, Urbana, September 1999. Univ. of Illinois. Cited on pp. 22, 25.

[vdODZ+16] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. WaveNet: A Generative Model for Raw Audio. *arXiv:1609.03499 [cs]*, September 2016. arXiv:1609.03499. Cited on p. 28.

[vdOLB+17] A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van den Driessche, E. Lockhart, L. C. Cobo, F. Stimberg, N. Casagrande, D. Grewe, S. Noury, S. Dieleman, E. Elsen, N. Kalchbrenner, H. Zen, A. Graves, H. King, T. Walters, D. Belov, and D. Hassabis. Parallel WaveNet: Fast High-Fidelity Speech Synthesis. *arXiv:1711.10433 [cs]*, November 2017. arXiv:1711.10433. Cited on p. 28.

[vdOLV19] A. van den Oord, Y. Li, and O. Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv:1807.03748 [cs, stat]*, January 2019. arXiv:1807.03748. Cited on p. 5.

[WFCT19] T. Wu, I. Fischer, I. L. Chuang, and M. Tegmark. Learnability for the Information Bottleneck. *Entropy*, 21(10):924, September 2019. arXiv:1907.07331, doi:10.3390/e21100924. Cited on p. 26.

[WR20] A. Wieczorek and V. Roth. On the Difference between the Information Bottleneck and the Deep Information Bottleneck. *Entropy*, 22(2):131, February 2020. doi:10.3390/e22020131. Cited on p. 27.

[WWMR18] A. Wieczorek, M. Wieser, D. Murezzan, and V. Roth. Learning Sparse Latent Representations with the Deep Copula Information Bottleneck. In *International Conference on Learning Representations*, February 2018. Cited on p. 27.

[WZH+20] L. Wen, Y. Zhou, L. He, M. Zhou, and Z. Xu. Mutual Information Gradient Estimation for Representation Learning. In *arXiv:2005.01123 [Cs, Stat]*, May 2020. arXiv:2005.01123. Cited on p. 5.