Q. Inf. Science 3 (8.372) — Fall 2024

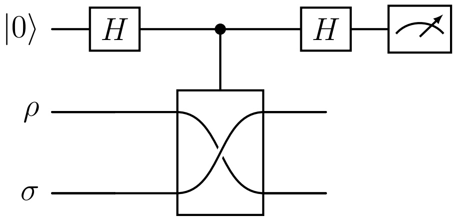
Assignment 7

Due: **Tuesday**, *Oct 29, 2024 at* ***9pm***

Turning in your solutions: Upload a single pdf file to gradescope.

1. **The SWAP test**

   Given two quantum states, are they equal or far apart? One method to test this is known as the "SWAP test".

   Let p denote the probability that this circuit outputs 0.

   (a) Relate p to $\operatorname{tr}\rho\sigma$.

   (b) Let us now explore how useful this is. Define $T = \frac{1}{2}\|\rho - \sigma\|_1$. How are p and T related when ρ and σ are pure states? Give an example of d-dimensional mixed states ρ, σ where $T \leq 1/2$ but $p \leq \frac{1}{2} + O\big(\frac{1}{d}\big)$.

   (c) While the SWAP test is not always an efficient way to estimate trace distance (and it turns out the d-dependence here cannot be removed), it can be used to estimate $\|\rho - \sigma\|_2^2$. Explain how to do this using $O(1)$ copies of ρ and σ. Use Cauchy-Schwarz to find the best constants $b \geq a > 0$ such that

   $$a\|\rho - \sigma\|_2 \leq \|\rho - \sigma\|_1 \leq b\|\rho - \sigma\|_2 \tag{1}$$

   (d) The purity of a density matrix is defined to be $\operatorname{tr}\rho^2$ and can also be estimated using the SWAP test. Likewise we can estimate the entanglement of $|\psi\rangle_{AB}$. Explain how the SWAP test can be applied to two copies of $|\psi\rangle$ to estimate $S_2(\psi_A)$.

   (e) In the above entanglement test, we use two copies of $|\psi\rangle$; say that these live on subsystems A_1B_1 and A_2B_2. However, we only apply the SWAP test to A_1A_2. We could equivalently have applied it to B_1B_2. In each case we just discard the remaining systems without looking at them. What happens if we apply a SWAP test to *both* A_1A_2 and B_1B_2? Does that yield any more information beyond measuring only A_1A_2?

2. **Tomography of pure states**

Suppose we have n copies of an unknown pure state $|\phi\rangle \in \mathbb{C}^d$. Our goal is to measure $|\phi\rangle^{\otimes n}$ and output an estimate $\hat{\phi}$. The most general strategy here consists of choosing a measurement operator $M_{\hat{\phi}} \geq 0$ for each possible estimate $\hat{\phi}$. Since these have continuous degrees of freedom the usual normalization condition now involves an integral:

$$M_{\text{fail}} + \int \mathrm{d}\hat{\phi}\, M_{\hat{\phi}} = I_{d^n} \tag{2}$$

Here we also include the possibility of a "failure" outcome M_{fail}. We define the integration measure so that $\int \mathrm{d}\hat{\phi}\, 1 = 1$. Equivalently we can write $\int \mathrm{d}\hat{\phi} = \mathbb{E}_{\hat{\phi}}$, where the expectation/integral is taken over all unit vectors.

(a) We will choose $M_{\hat{\phi}} = c\hat{\phi}^{\otimes n}$ for some constant $c > 0$, and $M_{\text{fail}} = I - \Pi_{\text{sym}}^{(d,n)}$. Show that this yields a valid measurement for the right choice of c. What is c?

(b) Let $F := F(\phi, \hat{\phi})$. Using this measurement, compute $\mathbb{E}[F^{2k}]$ for k a positive integer.

(c) *Average fidelity.* How large does n need to be in order to achieve $\mathbb{E}[F] \geq 1 - \epsilon$?

(d) *Large-deviation bounds.* Suppose we use an inadequate number of copies, say $n = d/10$. Show that $\Pr[F \geq 1/2]$ is exponentially small in d. As a hint, you might apply Markov's inequality to F^{2k} for $k = d/10$.

(More generally the constants 1/10 and 1/2 here are somewhat, but not completely, arbitrary.)

3. **Relative entropy and the matrix multiplicative weight method.**
In this problem we will see how quantum relative entropy can be a useful tool in classical optimization algorithms that have applications in machine learning. For convenience, take log to be base-e in this problem. Some formulas that may be helpful:

$$\ln(A + B) = \ln(A) + \int_0^\infty \mathrm{d}z\, (A + zI)^{-1} B (A + B + zI)^{-1} \tag{3}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} e^{A(t)} = \int_0^1 \mathrm{d}s\, e^{sA} \frac{\mathrm{d}A}{\mathrm{d}t} e^{(1-s)A} \tag{4}$$

(a) *Variants of gradient descent.* Consider the problem of minimizing a function $f : \mathbb{R}^d \to \mathbb{R}$. We will discuss three algorithms for this problem.

i. *Proximal gradient descent.* The idea of gradient descent is to start with a point x_0 and then in the t^{th} step, move from x_t in the direction of $-\boldsymbol{\nabla} f(x_t)$, i.e. -1 times the gradient of f evaluated at x_t. At the same time, we don't want to move too far from x_t. These goals (moving in the direction of $-\boldsymbol{\nabla} f$ but not too far from x_t) compete and we choose x_{t+1} according to

$$x_{t+1} = \arg\min_{x_{t+1}} \eta \langle x_{t+1} - x_t, \boldsymbol{\nabla} f(x_t)\rangle + \frac{1}{2}\|x_{t+1} - x_t\|_2^2, \tag{5}$$

for some parameter $\eta > 0$. Solve for x_{t+1} in terms of x_t, η, and f. Does this correspond to a step in the direction of $-\boldsymbol{\nabla} f$?

ii. *Mirror descent on probabilities.* Let Δ_d be the set of probability distributions on d items, i.e. $\Delta_d = \{x \in \mathbb{R}^d : \forall i\, x(i) \geq 0, \sum_{i=1}^d x(i) = 1\}$. For probability distributions it is more natural to use the relative entropy as a distance measure instead of the ℓ_2 norm. Thus the *mirror descent* algorithm chooses x_{t+1} according to

$$x_{t+1} = \arg\min_{x_{t+1}} \eta \langle x_{t+1} - x_t, \boldsymbol{\nabla} f(x_t) \rangle + D(x_{t+1} \| x_t). \tag{6}$$

(The terminology "mirror descent" comes from a generalization using something known as as "mirror map" which we will not use in this pset.) Solve for x_{t+1} in terms of x_t, η, and f. As a hint, the update rule you find is called the "multiplicative weights" update rule.

iii. *Mirror descent on density matrices.* Now let $\mathcal{D}_d$ denote $d \times d$ density matrices and define $f : \mathcal{D}_d \mapsto \mathbb{R}$. Note that $\boldsymbol{\nabla} f$ is now a matrix, and for matrices A, B, we define $\langle A, B \rangle := \operatorname{tr}\left[A^\dagger B\right]$. Mirror descent here corresponds to the update rule

$$\rho_{t+1} = \arg\min_{\rho_{t+1}} \eta \langle \rho_{t+1} - \rho_t, \boldsymbol{\nabla} f(\rho_t) \rangle + D(\rho_{t+1} \| \rho_t). \tag{7}$$

Solve for ρ_{t+1} as a function of ρ_t, η and f, assuming for simplicity that ρ_t is full rank. As a hint, you may find the solution of problem 2(c) on pset 3 helpful.

iv. *Continuous-time matrix mirror descent.* It is sometimes more convenient to work with a continuous-time version of the map in eq. (7). Let $\rho(t)$ be a function of t and for $t \geq 0$ let

$$\rho(t + \mathrm{d}t) = \arg\min_{\rho(t+\mathrm{d}t)} \eta\, \mathrm{d}t\, \langle \rho(t + \mathrm{d}t) - \rho(t), \boldsymbol{\nabla} f(\rho(t)) \rangle + D(\rho(t + \mathrm{d}t) \| \rho(t)). \tag{8}$$

Write down a differential equation for $\ln \rho(t)$. [Hint: instead of solving eq. (8) directly, guess the form of the answer by analogy with your answer from part iii.]

(b) *Convergence of matrix multiplicative weights.* Let ρ_* be an arbitrary density matrix (which we will later take to be the minimizer of f).

i. *Progress.* Using the above differential equation show that

$$\frac{\mathrm{d}}{\mathrm{d}t} D(\rho_* \| \rho(t)) = \langle \eta \boldsymbol{\nabla} f(\rho(t)), \rho_* - \rho(t) \rangle \tag{9}$$

ii. *Convexity.* Suppose that f is convex. Prove that

$$f(\sigma_1) - f(\sigma_2) \leq \langle \boldsymbol{\nabla} f(\sigma_1), \sigma_1 - \sigma_2 \rangle, \tag{10}$$

for any density matrices σ_1, σ_2.

iii. *Convergence.* Let $\rho(0) = I/d$ and let $\rho_* = \arg\min f(\rho_*)$.
Show that $D(\rho_* \| \rho(0)) \leq \log(d)$. How large should T be to guarantee that $f(\rho(T)) \leq f(\rho_*) + \epsilon$? You may find it helpful to show that $\frac{df(\rho(t))}{dt} \leq 0$, which can be done either using the fact that $\rho(t)$ optimizes eq. (8) or with direct calculation.
Observe that relative entropy is used in the analysis but the final bound is only in terms of f and the update rule you derived can also be expressed without referencing the relative entropy. Your solution turns out to slightly overstate the power of this algorithm since actual computers need to work in discrete time and this introduces some additional difficulties. Still the mirror descent algorithm is a very powerful tool because of its favorable dependence on d.