Contents

Chapter 1

Introduction

Chapter 2

Multi-armed bandits

2.1 Problem formulation

In the multi-armed bandit problem, there are k distributions ('arms'), $\{P_1, P_2, \ldots, P_k\}$ with unknown means $\{\mu_1, \mu_2, \ldots, \mu_k\}$. For a given number of turns T, the goal is to maximise the expected reward by iteratively selecting a distribution to sample from. In particular, the goal is to minimise the regret defined as

$$R(T) = \sum_{t=1}^{T} \mu^* - \mu_t, \tag{2.1}$$

where μ^* is the mean of the distribution with the highest mean, and μ_t is the mean of the distribution selected at time t. The number of turns T is often referred to as the horizon and can be assumed to be greater than k. This poses a constant struggle between exploration and exploitation, where exploration is the process of trying out new distributions, and exploitation is the process of using the distribution with the highest mean.

Almost always, assumptions are made about the distributions. Otherwise, composing algorithm with any sort of optimality guarantee would be futile. A common assumption, for example, is that the distributions are Bernoulli. Often they are assumed to be Gaussian with unknown mean and maybe some restrictions on the variance. If no assumptions are made to the type of distributions, there are likely to be assumptions made to the variance or support of the distributions.

Table 2.1: Comparison of strategies.

Strategy	Regret	Tuning
Random	Linear	NA
Greedy	Linear	NA
Epsilon-greedy	Linear	Difficult
Epsilon-decay	Logarithmic	Difficult
UCB	Logarithmic	Optional
Thompson	Logarithmic	Priors

2.1.1 Best-arm identification

An alternative problem is to find the best arm with as few turns as possible. In this version, a δ is given, and the goal is to find the best arm with probability at least $1 - \delta$.

2.1.2 Bandit generalisations

There are many generalisations to the multi-armed bandit problem. For instance, the distributions may not be stationary, but instead change throughout the game. Alternatively, with contextual bandits, information about a context is given before each turn, which must then be taken into account when selecting an arm. Adversarial bandits complicates matters further, where the rewards are not stochastic from some distribution, but are instead selected by an adversary.

2.2 Strategies

2.2.1 Explore-only

Pure exploration is obviously a suboptimal strategy, but it is a good baseline to compare against. It can be implemented by selecting an arm uniformly or in order, but it will perform poorly either way. The arm-selection procedure is described by algorithm 2.

Algorithm 1 Random arm selection

```
procedure SelectArm(t)
    Sample i from {1, ..., k} uniformly
    return i
```

It is easy that the expected regret is

$$R(T) = T\left(\mu^* - \frac{1}{k}\sum_{i=1}^{k}\mu_i\right), \tag{2.2}$$

which is $\Theta(T)$. This motivates the search for an algorithm with sublinear regret.

2.2.2 Greedy

A simple algorithm and a good baseline is the greedy algorithm. Here, all arms are tried N initial times, and the empirical means are used to select the best arm. Afterwards, the arm with the highest empirical mean is selected for all remaining turns. The arm-selection procedure is listed in algorithm 2, where $\hat{\mu}_i$ is the empirical mean of arm i.

Algorithm 2 Greedy arm selection

```
procedure SelectArm(t)
    if t ≤ Nk then
        return (t mod k) + 1
    else
        return argmax_{i=1,...,k} μ̂_i
```

With greedy selection, the expected regret is

$$R(T) = \sum_{t=1}^{T}\mu^* - \hat{\mu}_t, \tag{2.3}$$

2.2.3 Epsilon-greedy

The problem with the greedy algorithm is that it may be unlucky and not discover the best arm in the initial exploration phase. To mitigate this, the epsilon-greedy algorithm may be used. In this algorithm, the estimated arm is pulled with probability $1 - \epsilon$ and a random arm is pulled with probability ϵ. This ensures convergence to correct exploitation as the horizon increases, and it will generally reduce the regret. Still, with a constant ϵ, a constant proportion of the turns will be spent exploring, keeping the regret linear in the horizon. Choosing ϵ is a trade-off between exploration and exploitation and can significantly affect the regret.

Epsilon-decay

If one allows the ϵ to decay over time, the regret can be reduced even further. This makes sense, as exploration is less worthwhile as estimates become ever more accurate. A common choice is to decay ϵ as $\epsilon_t = \epsilon_0/t$.

2.2.4 UCB

The upper confidence bound (UCB) algorithm is a more sophisticated algorithm based on estimating an upper bound for the mean of each arm. One always chooses the arm whose upper confidence bound is highest, a principle known as 'optimism in the face of uncertainty'.

In the original formulation, where support on only $[0, 1]$ is assumed, the upper confidence bound is given by

$$\mathrm{UCB}_t(a) = \hat{\mu}_t(a) + \sqrt{\frac{2 \ln t}{N_t(a)}}, \tag{2.4}$$

where $\hat{\mu}_t(a)$ is the empirical mean of arm a at time t, $N_t(a)$ is the number of times arm a has been pulled at time t. It generally achieves logarithmic regret.

Many variants of the UCB algorithm exist. Different assumptions about the distributions changes the confidence bounds.

2.2.5 Bayesian: Thompson sampling

Thompson sampling is a Bayesian approach to the multi-armed bandit problem, originally described in 1933 as a way to handle the exploration-exploitation dilemma in the context of medical trials. The idea is to sample from the posterior distribution of the means of the arms and pull the arm with the highest sample.

2.3 Simulations

Chapter 3

Quantum computing

Chapter 4

Quantum algorithms

4.1 Grover's algorithm

The quantum search algorithm of Grover [**grover1996**] is a quantum algorithm that finds an element in an unstructured list with high probability. While such a problem necessarily requires $O(N)$ time in a classical setting, needing on average $N/2$ steps to find the element and in the worst case N, Grover's algorithm finds the element in $O(\sqrt{N})$ steps. This is a quadratic speed-up.

Grover's algorithm is provably optimal; no quantum algorithm can perform such general searches faster. This should not be surprising. If an exponential speed-up were possible, it could be used to find the solution to NP-hard problems fast.

For Grover's algorithm to work, assume there is a function $f : \{0, \ldots, N-1\} \to \{0, 1\}$ that maps the index of an element to 1 if it is the one desired and 0 otherwise. Then, one assumes access to a quantum oracle, $\mathcal{O}_f$ (effectively a black box subroutine) that implements f thus:

$$\mathcal{O}_f \left|x\right\rangle = (-1)^{f(x)} \left|x\right\rangle . \tag{4.1}$$

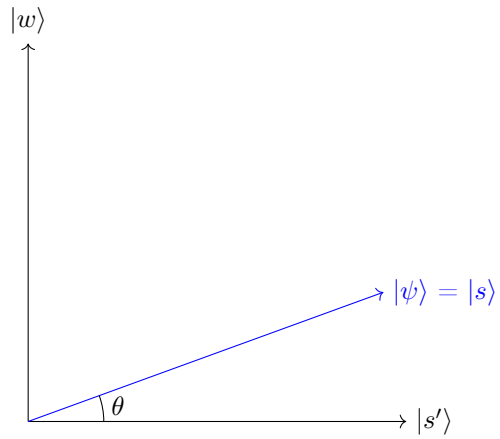
A single application of this oracle is not enough to find the desired element, as the square of the amplitude of the desired element remains unchanged. Central to Grover's algorithm is the idea of amplifying the amplitude of the desired element. This is done by applying a sequence of operations that is repeated until the amplitude of the desired element is large enough for it is most likely to be measured, while the amplitudes of the other elements are reduced.

Let the state $|w\rangle$ which be the winner state, a state with amplitude 1 for the desired element and 0 for all others. Then consider the state $|s\rangle$, which is a uniform superposition state, a state with equal amplitudes for all elements.

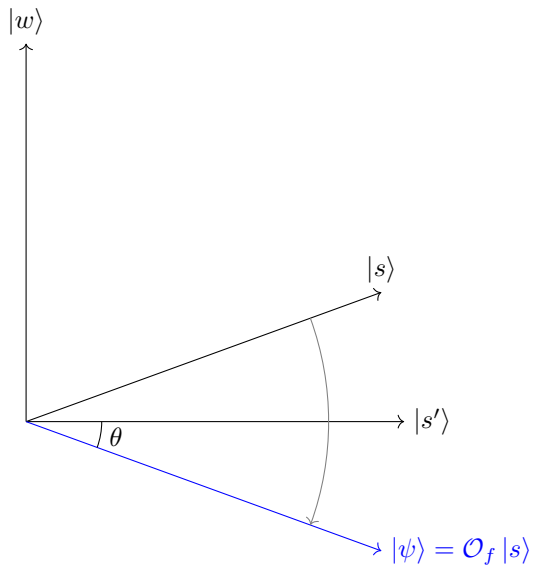
Define the state $|s'\rangle$ by subtracting the projection of $|w\rangle$ onto $|s\rangle$ from $|s\rangle$:

$$|s'\rangle = |s\rangle - \langle w|s\rangle\,|w\rangle\,. \tag{4.2}$$

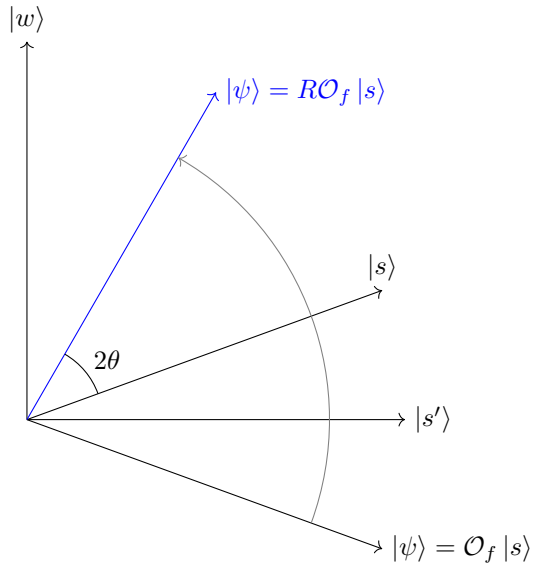
These two orthogonal states from a basis of a two-dimensional subspace of the greater Hilbert space. The uniform superposition state $|s\rangle$ serves as a starting point for the algorithm, and is achieved by applying Hadamard gates to all qubits. It is expressible as $|s\rangle = \cos(\theta)\,|s'\rangle + \sin(\theta)\,|w\rangle$, where $= \theta = \arcsin\left(1/\sqrt{N}\right)$, and it can be seen as a point in the s-w-plane, thus:



Applying the oracle on $|s\rangle$ leaves its $|s'\rangle$ component unchanged, but flips the sign of the $|w\rangle$ component. This results in the state $|\psi\rangle = \cos(-\theta)\,|s'\rangle + \sin(-\theta)\,|s'\rangle$, which can be seen as reflection of $|s\rangle$ in the $|s'\rangle$ direction.



Next, the state $|\psi\rangle$ is reflected about the initial $|s\rangle$ state, resulting in the state $|\psi'\rangle = \cos(3\theta)|s'\rangle + \sin(3\theta)|w\rangle$. Reflection thus is achieved by the diffusion operator $R = H^{\otimes n} S_0 (H^{\otimes n})^{-1} = H^{\otimes n} S_0 H^{\otimes n}$, where $S_0 = 2|0\rangle\langle 0| - I$ is the reflection operator about the $|0\rangle$ state, that is an operator that flips the sign of all but the $|0\rangle$ component.



The product of the oracle and the diffusion operator defines the Grover operator, which is simply applied until the amplitude of the $|w\rangle$ is sufficiently amplified. After k iterations, the state is $|\psi_k\rangle = \cos((2k+1)\theta)\,|s'\rangle + \sin((2k+1)\theta)\,|w\rangle$. Measuring the correct state has probability $\sin^2((2k+1)\theta)$. Therefore, $k \approx \pi/4\theta$ iterations should be completed. Assuming large N, for a short list would not warrant the use of Grover's algorithm, $\theta = \arcsin(1/\sqrt{N}) \approx 1/\sqrt{N}$, and so $k \approx \pi\sqrt{N}/4$.

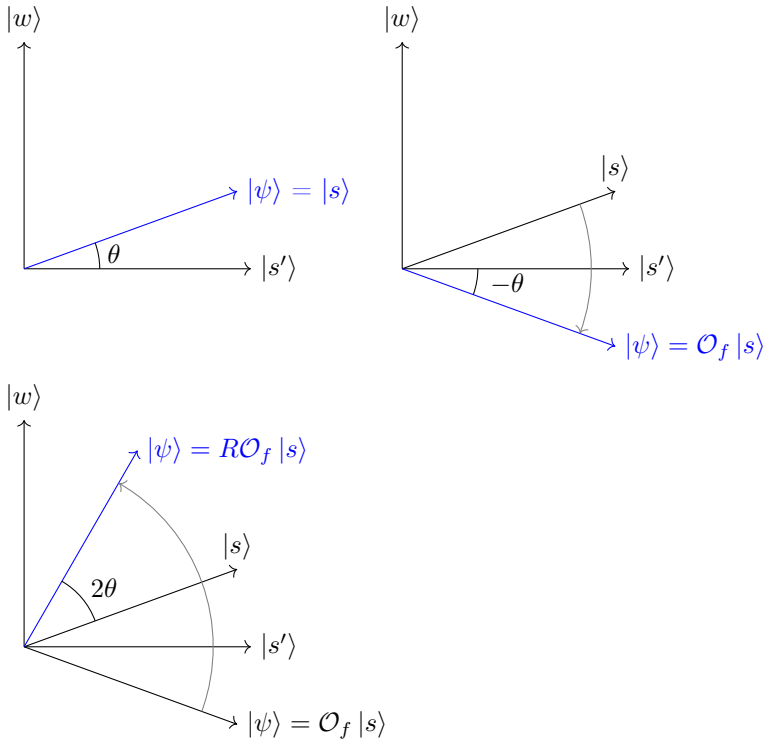


Figure 4.1: Grover's algorithm.

Chapter 5

Quantum bandits

Several formulations of the multi-armed bandit problem have been made for a quantum computing setting. As the central issue in bandit problems lie in sample efficiency rather than computational difficulties, quantum computers offer little advantage assuming classical bandits. However, by allowing bandits to be queried in superposition, major speed-ups can be achieved. For such bandits, regret minimisation is no longer a valid objective, and instead the problem is to find a strategy that maximises the probability of finding the optimal arm with as few queries as possible.

5.1 Casalé

In [**casale2020**], an algorithm based on amplitude amplification is proposed and is shown to find the optimal arm with quadratically fewer queries than the best classical algorithm for classical bandits — albeit with a significant drawback: the probability of the correct arm being suggested can not be set arbitrarily high, but is instead given by the ratio of the best arm's mean to the sum of the means of all arms. This