OIT 611 Lecture Notes

Drift Method
from Stochastic Networks to Machine Learning

(Draft, October 18, 2021)
## Contents

1 Introduction 3
   1.1 Motivation for the Drift Method 3
   1.2 Variations on a theme of Lyapunov 4
   1.3 Organization of the course 6
   1.4 Notation 6

2 Probability Preliminaries 7
   2.1 Markov Process 7
   2.2 Poisson Process 8
   2.3 Recurrence of Markov Processes 8
   2.4 Martingales 9
   2.5 Notes 10

3 Introduction to Stochastic Processing Networks 11
   3.1 The M/M/1 Queue 11
   3.2 Stability and Steady-State Queue Lengths 12
   3.3 Stability Gets Tricky in Networks 13
   3.4 What’s Next? 16
   3.5 Notes 16

4 Maximum Stability in Stochastic Networks 17
   4.1 Single-Hop Stochastic Processing Network 17
   4.2 Main Theorem: Maximum Stability of the Max-Weight Policy 18
   4.3 Proof of Main Theorem and the Foster-Lyapunov Stability Criterion 19
   4.4 Multi-Hop Networks and the Back-pressure Algorithm 22
   4.5 Notes 22

5 More on the Foster-Lyapunov Criterion 23
   5.1 Proof of the Foster-Lyapunov Criterion 23
   5.2 Steady-State Drift via Foster-Lyapunov 26
   5.3 Bounding Steady-State Queue Lengths with Drift 27
   5.4 Notes 27

6 Drift in Complex Systems via Fluid Limits 29
   6.1 From Attractiveness to Stability 30
CONTENTS

6.2 Deploying the Fluid Limit Approach ........................................... 32
6.3 Why Fluid Limit ................................................................. 33
6.4 Other Types of Fluid Limits ..................................................... 34
6.5 Notes ................................................................................. 34

7 State Space Collapse and Optimality of Drift Methods .................. 35
7.1 Baby-Step SSC: An Example with Two Queues .......................... 35
7.2 Two ways lead to SSC ............................................................ 38
7.3 Preliminaries ........................................................................ 38
7.4 Statement of Main Result ........................................................ 40
7.5 Main Steps of the Proof ........................................................... 41
7.6 Notes ................................................................................. 45

8 Double Drift: Stationary Lyapunov Analysis with SSC ................. 47
8.1 A Parallel Load Balancing Model ............................................. 47
8.2 Main Result: Performance of the JSQ Policy ............................. 49
8.3 New Technique: Drift Conservation ........................................ 50
8.4 Drift Conservation for the Lower Bound .................................. 50
8.5 Upper Bound: Drift Conservation with State Space Collapse ........ 52
8.6 Discussion ........................................................................... 58
8.7 Notes ................................................................................. 58

9 Potential Functions in Online Learning ...................................... 59
9.1 A Marching Call: From Stochastic Networks to Online Learning .... 59
9.2 Prediction with Expert Advice ................................................ 60
9.3 Weighted Average Forecasters ................................................ 61
9.4 Drift Analysis of Weighted Average Forecasters ....................... 62
9.5 Blackwell Approachability ...................................................... 67
9.6 A Unified View through Approachability .................................. 72
9.7 Notes ................................................................................. 80

10 Drift Analysis in Bayesian Bandits with Information Ratio .......... 81
10.1 Bayesian Stochastic Multi-arm Bandit .................................... 81
10.2 Thompson Sampling ............................................................. 83
10.3 Regret Upperbound under Thompson Sampling ....................... 83
10.4 Coupling to an Information Piggy Bank ................................. 84
10.5 Drift Accounting via Information Ratio ................................. 85
10.6 Entropy as Information Measure ............................................ 88
10.7 Improved Distribution-Free Regret Bound ............................... 92
10.8 Notes ................................................................................. 93

Bibliography .............................................................................. 95

Index ....................................................................................... 99
Chapter 1

Introduction

This course is an exploration of the drift method: a family of simple, yet surprisingly powerful, meta-algorithm that in each step the greedily and incrementally minimizes a certain potential function. Manifested in different forms, such as the MaxWeight algorithm, the $c\mu$ rule, EXP3, policy gradient, to name a few, the drift method powers some of the most popular algorithmic paradigms in queueing networks, optimization and machine learning.

The name “drift method” is somewhat of a lazy catch-all, and the method is so old and broadly used and it’s hardly possible to call it a distinctive field of study. Therefore, our main interest here is not debate who invented the “drift method”, if there is such a person or group of people, but rather to investigate, by comparing seemingly different applications of such methods, questions such as

1. Historically, what kinds of problems motivated the use of “drift” as an analytical methodology. Then, projecting forward, what problems in the current areas of research could benefit from these ways of thinking?

2. How does the drift method take on different shades and flavors in different (but related) fields? What can we learn from these distinctions?

3. What are some of the more obvious versus more nuanced features and consequences of drift methods? Are there areas where their uses and successes are surprising?

1.1 Motivation for the Drift Method

To understand what we mean by “drift”, it is perhaps best to see a concrete example. One of the earliest application of the drift method dates back to the work of Alexander Lyapunov [Lyapunov, 1892] on the question of stability in dynamical systems. Consider a dynamical system defined as the solution $x(\cdot)$ to the ordinary differential equation:

$$\frac{dx(t)}{dt} = f(x(t)), \quad x \in \mathbb{R}^d, \quad (1.1)$$

and let us suppose that $x(t) = 0$ is a valid solution to this system. We’d like to know whether as $t \to \infty$, $x(t)$ will explode to infinity, or say, converge to 0. A direct solution to this problem, of course, is to solve the equation, write down a nice and clean equation for $x(t)$, and stare at it very hard until the answer becomes obvious. If only life were so simple... As we know, this is often impossible unless the dynamic $f(\cdot)$ has some very nice form. So, what can we do if $f$ is somewhat complicated?
Lyapunov came up the following idea. Suppose that we can demonstrate that there exists a Lyapunov function, \( V : \mathbb{R}^d \to \mathbb{R}^+ \), which satisfies the following properties:

1. \( V(x) = 0 \) if and only if \( x = 0 \).
2. \( \frac{d}{dt} V(x(t)) = \nabla V(x(t)) \cdot f(x(t)) < 0 \) whenever \( x(t) \neq 0 \).

Then, we would know that if \( x(0) \) is initialized to be somewhere “close” to 0, then \( x(t) \) will eventually converge to 0.

The key idea here can seem misleadingly simple in hindsight, and goes something like this:

1. We start by wanting to understand the dynamics of \( x(t) \), a process taking place in a high-dimensional space \( \mathbb{R}^d \).
2. When this proves too difficult, we try to “cheat” by converting the problem into a much simpler, one-dimensional one. We do so by applying the Lyapunov function \( V \), as a “test”. To know where \( x(t) \) is, we simply look at the output \( V(x(t)) \): if it’s equal to 0, then we know that \( x(t) = 0 \); otherwise, we know it is somewhere else in \( \mathbb{R}^d \). This is essentially the point of property #1.
3. However, if this is all we are doing, we have merely shifted the difficulty of staring at \( x(t) \) to staring at \( V(x(t)) \): we still have no clue what \( V(x(t)) \) should be! Thus comes the crucial property #2, the “drift” part of the method. We might not know exactly where \( x(t) \) will be, but we know that whenever \( x(t) \neq 0 \), it will lead to an instantaneous decrease in \( V(x(t)) \).
4. At this point, we have put \( x(t) \) into a bind. Because \( V \) is non-negative, and whenever \( x(t) > 0 \) it will be pushed down by a negative drift, we know that eventually \( V(x(t)) \) will have nowhere to go but to stay at 0. Then, by property #1 we know that this means that the original process \( x(t) \) itself also has nowhere to go but to stay at 0. Voila!

Let us step back a bit. Where in these four steps did we make the problem “simpler” than the original? The key is in step #3. Notice that to obtain the drift of \( V(x(t)) \), we don’t have to tackle the full might of \( f(x(t)) \), but rather only its projection onto the gradient of \( V \). As we will see in many examples during the rest of this course, whereas \( f(x(t)) \) can be often unruly, the projection \( \nabla V(x(t)) \cdot f(x(t)) \) may be a lot more tractable.

### 1.2 Variations on a theme of Lyapunov

The four-step procedure outlined above for the original control problem that Lyapunov set out to solve serves as a pretty good starting point for us when it comes to understanding how the drift method will be used in the complex dynamical systems we will encounter. More generally, we would start with a stochastic process \( \{X(t)\} \) in \( \mathbb{R}^d \), whose limiting dynamics we wishes to understand. A typical and important question that we would ask is whether \( X(t) \) will eventually approach some “nice” set, e.g., one with a small norm. We then identify a non-negative Lyapunov (a.k.a. potential) function, \( V(\cdot) \), which admits the property that \( V(x) \) is small if and only if \( x \) in a nice set. Finally, we establish what whenever \( X(t) \) is “not nice”, \( V(X(t)) \) will exhibit some appropriate notion of negative drift.

Sounds simple enough, right?

As we start looking at some variations on this basic program, the power of the drift method will be increasingly evident, and sometimes surprisingly so. They are surprising in many ways, from the sheer range of applicability of this technique, to the strength and granularity of the results we are able
1.2. VARIATIONS ON A THEME OF LYAPUNOV

to obtain with it, to deeper more fundamental connections to the nature of the dynamics itself. Here is an incomplete list of some of these interesting topics we will come across as we go down this path:

1. **A stochastic Lyapunov theory.** What happens when $X(t)$ is stochastic? Our first stop is the stochastic generalization of the Lyapunov theory to the study of Markov chains. Whereas in the previous example, the dynamic system is that of a deterministic differential equation, we can generalize the theory to the setting where the process $X$ is a Markov chain with potentially very complicated dynamics. The analogous theory here is often referred to as the Foster-Lyapunov theory [Foster et al., 1953], where the notion of “converging to 0” for a deterministic process will be replaced by that of (positive) recurrence. This theory would allow us to analyze the dynamics induced by fairly complex adaptive resource allocation algorithms, such as the Max-Weight algorithm.

2. **From the qualitative to the quantitative.** Once we have the machinery to apply the drift method to studying Markov chains, we will see that the method not only let us answer qualitative questions of the type “will $X(t)$ visit a nice set”, but also quantitative questions such as “how often/frequent/quickly will $X(t)$ visit a nice set?” This turns out to be a pretty big deal, as it will allow us to obtain quantitative estimates of importance statistics of $X(\cdot)$ in steady state, such as its mean, higher moments, or moments of functions of $X$, etc.

3. **From “a” method to “the” method.** So far, the drift method may appear to be somewhat ad-hoc. Even if the method gives some kind of performance guarantees, the whole program hinges upon some arbitrary Lyapunov function $V(\cdot)$ that we happen to have picked. We will see that in certain cases, we can prove that not only algorithms or analysis based on the drift method “works”, they are in some sense the best algorithm or analysis possible. This type of optimalities don’t often come by, and when they do, they often give us deep insights into the nature of the system.

4. **Drifting in unexpected places.** You might say: okay, but the idea of using drift methods makes sense in an ordinary differential equation or Markov chain makes some intuitive sense. After all, there is a clear notion of time, and the original process itself already seems to possess some sense of “drift”, as per the differential operator or the generator of a Markov chain. However, some of the other effective, and arguably very interesting, applications of the drift method arise in domains where the idea of “drift” might not appear obvious at all. Of course, what constitute as “not obvious” is in eye of the beholder, but I would argue that here are some examples where it seems quite surprising to me to see drift method popping up:

a) Using drift methods in stochastic approximation and in conjunction with the Stein’s method to derive strong quantitative bounds on how closely one distribution (think Gaussian) approximates a not-so-nice one (e.g., your actual queueing delay).

b) The Langevin dynamics is well-studied family of continuous-time Markov processes in physics and simulation, and recently found quite some celebrity in machine learning [Welling and Teh, 2011, Cheng et al., 2018]. It turns out that the process can be viewed from a distributional perspective, and the dynamics in the distributional space can be well-explained as a drift under a certain potential function.

c) What can drift methods say about regrets in multi-arm bandits or performance of reinforcement learning algorithms, such as policy gradient?
1.3 Organization of the course

The course will be roughly divided into three parts.

Part I - Drift Method in Dynamic Resource Allocation. We will look at some of the (by now classic) uses of the drift method in the analysis of Markov chains, and in particular in the context of queueing networks and dynamic resource allocation. We begin by looking at problems where the Lyapunov functions are directly applied to the original Markov process, and we will use them to understand the stability and distributional properties of these systems in equilibrium. In this part, we will focus on building a solid foundation and understanding some of the basic implications of the Foster-Lyapunov theorem.

Part II - Stochastic Approximation. As we will realize quickly, sometimes the original Markov process, even with the aid of a Lyapunov function, can prove to too complex or messy. This naturally lead us to use idea from stochastic approximation, where we would apply limit theorems to argue that under some appropriate scaling, the limiting process can be approximated by some simpler process, upon which we will apply the drift analysis. The idea of applying the drift analysis upon the limiting, as opposed to the original, process turns out to be incredibly useful, and often leads to both sharper results and deeper insights. We will encounter a few more advanced concepts in this part of the course, such as state-space collapse, diffusion approximation, and the Stein’s method.

Part III - Drift Method in Machine Learning. In this last part of the course, we will turn our attention to machine learning. Compared to queueing networks, it appears that there is a less of a coherent body of theory concerning the drift method in machine learning. Nevertheless, similar ideas are still often evoked, and there have been many interesting applications of the drift method, sometimes quite surprising. Our journey here will be more of an exploratory nature, though I am sure there will not be a shortage of delightful finds.

1.4 Notation

Some common notation:

1. \( (x)^+ := \max\{x, 0\} \).

2. \( f(x) \ll g(x) := \lim_{x \to \infty} \frac{f(x)}{g(x)} = 0 \).
Chapter 2

Probability Preliminaries

We will cover in this chapter some basic concepts in probability theory which we will be using throughout. We will somewhat narrowly focus on the background immediately relevant for the main topics of the course, which means that you may find it handy to have a reference text on probability at hand for other fundamental concepts, such as probability space, conditional expectation, etc.

2.1 Markov Process

Discrete-time We say that \( X = \{X(t) \}_{t \geq 0} \) is a process if past and future are mutually independent conditional on the value of the process at a fixed time. More formally, we have

\[
P( X(t+1) = x_{t+1} \mid X(0) = x_0, X(1) = x_1, \ldots, X(t) = x_t ) = P( X(t+1) = x_{t+1} \mid X(t) = x_t )
\]

(2.1)

We call \( P(t)_{x,x'} = P( X(t+1) = x' \mid X(t) = x ) \) the transition probability of \( X \). We say that \( X \) is time homogeneous if its transition probabilities are identical over time. \( X \) is stationary if \( \pi(t) \) is invariant with respect to \( t \). The transition probabilities in a continuous-time Markov process are specified by the set

\[
\{p_{ij}(s,t)\}_{i,j \in S, s < t}
\]

where \( p_{ij}(s,t) \) is the probability that the process starting in state \( i \) at time \( s \) will be in state \( j \) at time \( t \). We say that the process is time homogeneous if \( p_{ij}(s,t) \) depends on \( s, t \) only through the different \( t - s \), in which case we will simply write \( p_{ij}(t - s) \).

A class of continuous-time Markov processes that is particularly useful for us is the pure-jump Markov process. As the name suggests, a pure-jump Markov process would linger in a state for a
random amount of time before jumping to a different state, and so on and so forth. Formally, we say a function \( f : \mathbb{R}_+ \to S \) is an \( S \)-valued pure-jump function if there exists a sequence of times \( 0 = t_0 < t_1, \ldots \) and states \( s_0, s_1, \ldots \) such that \( f(t) = s_i \) when \( t_i \leq t < t_{i+1} \). Then, a pure-jump Markov process \( X \) is one where \( X \) is a pure-jump function with probability 1.

To specify a time-homogeneous pure-jump Markov process, it would make sense to describe (a) the rate at which the process jumps from the current state and (b) the probabilities with which it chooses the next state. This is mediated by the notion of a (infinitesimal) generator. Let \( Q = \{ q_{ij} : i, j \in S \} \) be a set where \( q_{ij} \geq 0 \) for \( i, j \in S \) and \( q_{ii} = -\sum_{j \in S, j \neq i} q_{ij} \).

\[
(2.4)
\]

Then we say that \( Q \) is the generator for a time-homogeneous Markov process \( X \) if the transition probabilities satisfy:

\[
\lim_{\epsilon \downarrow 0} \frac{p_{ij}(\epsilon) - \mathbb{I}(i = j)}{\epsilon} = q_{ij}, \quad i, j \in S.
\]

\[
(2.5)
\]

Intuitively, this simply means that if \( X(t) = i \), then over a very small time interval \( \epsilon \), \( X \) would jump to state \( j \) with probability approximately \( \epsilon q_{ij} \). The following result ensures that the generator is indeed a good instrument in specifying a pure-jump Markov process:

**Proposition 2.1.1.** Fix \( Q \) that satisfies (2.4) and a probability distribution \( \pi(0) \) over \( S \). Then, there exists pure-jump, time-homogeneous Markov process with generator \( Q \) and initial distribution \( \pi(0) \), whose finite-order distributions are uniquely determined by \( \pi(0) \) and \( Q \).

### 2.2 Poisson Process

Since we will be dealing with resource allocation systems where demands can arrive randomly over time, the notion of counting process is useful and allows to keep track of the arrival patterns. A counting process \( X \) is a continuous-time stochastic process where, with probability one, \( X \) has a sample path that is non-decreasing, right continuous and integer valued. The value \( X(t) \) can be interpreted as the “count” up to time \( t \), and the difference \( X(t) - X(s) \) will be referred to as the increment, corresponding to the counts that occurred during the interval \( (s, t] \).

A very popular counting process is the Poisson process,

**Definition 2.2.1.** Fix \( \lambda > 0 \). We say that \( X \) is a Poisson process, if it is a counting process and

1. \( X \) has independent increments.
2. For any \( t \geq s \geq 0 \), the increment \( X(t) - X(s) \) follows a Poisson distribution with mean \( \lambda(t - s) \).

### 2.3 Recurrence of Markov Processes

The concept of recurrence, or lack of, in a Markov process will be essential in our investigation, so let us lay down a formal definition.

**Definition 2.3.1 (Recurrence).** Let \( S \) be a countable set, and \( X \) a Markov process taking values in \( S \). For a state \( i \in S \), define

\[
\tau_i := \min\{t \geq 1 : X(t) = i\}
\]

as the first passage time of the Markov process to state \( i \). Let

\[
M_i = \mathbb{E}[\tau_i | X(0) = i].
\]

The state \( i \) is said to be
1. transient, if $\mathbb{P}(\tau_i < \infty \mid X(0) = i) < 1$;
2. null recurrent, if $\mathbb{P}(\tau_i < \infty \mid X(0) = i) = 1$ and $M_i = \infty$;
3. positive recurrent, if $\mathbb{P}(\tau_i < \infty \mid X(0) = i) = 1$ and $M_i < \infty$;

Below are some useful properties related to the recurrence.

**Proposition 2.3.1.** Let $X$ be an irreducible and aperiodic Markov process. Then,

1. All states are transient, or all null recurrent, or all positive recurrent.
2. For any initial distribution $\pi_i(0)$, we have that $\lim_{t \to \infty} \pi(t) = 1/M_i$. If $M_i = \infty$, then this limit is understood to be equal to zero.
3. $X$ admits an equilibrium distribution, $\pi$, if and only if all states are positive recurrent. In this case, the equilibrium distribution is also unique, with $\pi_i = 1/M_i$.

## 2.4 Martingales

Let $S$ be a state space, and $X = \{X(t)\}_{t \geq 0}$ a discrete-time stochastic process taking value in $S$. A filtration $\mathcal{F} = \{\mathcal{F}_t\}$ is a sequence of nested $\sigma$-algebras where $\mathcal{F}_t \subset \mathcal{F}_{t+1}$. We say that $X$ is adapted to a filtration $\mathcal{F}$ if $X(t)$ is $\mathcal{F}_t$-measurable for each $t$.

**Definition 2.4.1.** Fix a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and a filtration $\mathcal{F} = \{\mathcal{F}_t\}_{t \geq 0}$. Let $X = \{X(t)\}_{t \geq 0}$ be a sequence of random variables that is adapted to $\mathcal{F}$. We say that $X$ is a martingale with respect to $\mathcal{F}$ if the following holds for all $t \geq 0$:

(i) $\mathbb{E} |X(t)| < \infty$;
(ii) $\mathbb{E} [X(t+1) \mid \mathcal{F}_t] = X_t$ almost surely.

We say that $X$ is a submartingale with respect to $\mathcal{F}$ if we replace the equality in (ii) with $\mathbb{E} [X(t+1) \mid \mathcal{F}_t] \geq X_t$, and a supermartingale, if we instead have $\mathbb{E} [X(t+1) \mid \mathcal{F}_t] \leq X_t$.

**Theorem 2.4.1** (Martingale Convergence Theorem). Let $X$ be a submartingale, with $\sup_{t \geq 0} \mathbb{E}((X(t))^+) < \infty$. Then $X(t)$ converges a.s. to a limit $X_\infty$ as $t \to \infty$, where $\mathbb{E}(|X|) < \infty$.

An important corollary of the above theorem is the following convergence result for non-negative supermartingales. The result follows by noticing that if $X$ is a supermartingale then $-X$ is a submartingale.

**Corollary 2.4.2.** Suppose $X$ is a non-negative supermartingale, then $X(t)$ converges a.s. to a limit $X$ and $\mathbb{E}(X_\infty) \leq \mathbb{E}(X(0))$.

Here’s an intuitive (though non-rigorous) way to interpret Corollary 2.4.2. Visualize the sequence of distributions $X(t)$. Since it is a supermartingale, you’d expect the means of these distributions to trend downward. However, we also know that $X(t)$ is non-negative, so these distributions sort of have nowhere to escape but get “sandwiched” between $X(0)$ and 0, leading to convergence. A proof of the martingale convergence theorem can be found in most standard texts (cf. [Durrett, 2019]).

The martingale convergence theorem is extremely powerful partially due to its generality: it is often possible to show that some process is a (sub/super)martingale even when the dynamics of the process are quite complicated or non-Markovian. We will encounter precisely such a scenario when analyzing the process associated with a Lyapunov function.
2.5 Notes

[Hajek, 2015] is an excellent reference text on probability and we use some of its notation and development in this chapter.
Chapter 3

Introduction to Stochastic Processing Networks

We will now venture into our first application area of the drift method: stochastic and queueing networks. Instead of diving straight into the main theorems, it would be useful to first have an understanding of the backdrop to the development and what eventually prompted the use of the drift method in the first place.

_Stochastic networks_ refer to a broad range of models and system that deal with the service of dynamically arriving jobs or tasks using constrained resources. A stochastic network typically has the following main components

1. A stream of tasks or jobs _arrivals_ to the system. Example include the arrivals of patients to a hospital, requests to a data center server, or manufacturing orders.

2. _Servers_ that process the jobs. Think of human experts, call-center agents, CPUs, data center servers, etc.

3. _Queues_ that hold these jobs while they await service, such as the waiting room in a hospital or memory buffers in a data center server.

4. Dispatching and scheduling policies that determine in real time which jobs should be served by what server, or in which queues they should be waiting.

3.1 The M/M/1 Queue

A simplest example of a stochastic processing network is a single queue with memoryless arrivals and services, a continuous-time version of which is known as the $M/M/1$ queue.

![Illustration of an M/M/1 queue](image)

Figure 3.1: Illustration of an $M/M/1$ queue
1. New jobs arrive to the system according to a Poisson process with rate $\lambda$. That is, during any unit time interval, we expect the number of arrivals to be a Poisson random variable with mean $\lambda$.

2. All jobs wait in a single queue, and the number of jobs in the queue at time $t$ is denoted by $Q(t)$. The scheduling policy is called first-come-first-serve (FCFS) or first-in-first-out (FIFO): the server will always work on the oldest job in the queue. If there isn’t any job, then the server just idles.

3. The server is modeled by a potential departure process, $D(t)$, which is a Poisson process with rate $\mu$. When there is a jump in $D(t)$, and the system is not empty, then we let one job depart from the system. We call $\mu$ the service rate. There are two ways to interpret this:

   a) Service tokens: think of the arrivals are people lining up to buy apples (let’s say the fruit, not the phone), and when a customer receives an apple (or service token) s/he then leaves the queue. Here, $D(t)$ can be thought of as the generating process for these apples.

   b) Exponential job sizes: in this interpretation, each arrival is associated with a certain size, the amount of work that requires to be performed before the job can be declared “finished”. If we have a server that works at unit speed, then $D(t)$ would correspond to the departures of jobs (when $Q(t) > 0$).

Whichever interpretation we go with, it’s clear that no departure can occur when the queue is empty, hence the word “potential” in “potential departures”.

3.2 Stability and Steady-State Queue Lengths

Nobody likes to wait in long queues, and that pretty much sums up what one would care about in a stochastic processing network. More formally, we will be mainly concerned with two types of performance measures

1. **Stability**: Do all jobs that arrive in the system eventually get processed? Or, will the system be so overwhelmed that the queues will blow up to infinity? Whether the system is stable is what we call a “zeroth-order” performance criterion.

2. **Steady-State queue length**: If stability is ensured, then we may expect that as $t \to \infty$, the queue lengths $Q(t)$ will converge to some kind of equilibrium distribution, $Q(\infty)$, which we call the steady-state distribution. What is $E(Q(\infty))$? Does it have an exponential tail?
Example 3.2.1 (Stability and Steady-State of an $M/M/1$ Queue). Let’s see how our simple $M/M/1$ queue performs on these two metrics. The queue length process in an $M/M/1$ queue evolves as a continuous-time homogeneous Markov process, with the transition rates given in Figure 3.2, also known as a birth-death process. In each state $i > 0$, the rate of transition to the state $i - 1$ is $\mu$, corresponding to the departure of a job, and to the state $i + 1$ is $\lambda$, corresponding to an arrival. In state 0, the only transition would be that of an arrival. For stability, it is not difficult to see that when $\mu > \lambda$, i.e., when service is faster than arrivals, the queue is more likely to go down than up, and the system is indeed stable. It is also easy to verify that the Markov chain admits a unique steady-state distribution:

$$P(Q(\infty) = k) = (1 - \rho)^k \rho^k, \quad k \in \mathbb{Z}_+,$$

where

$$\rho = \frac{\lambda}{\mu}$$

is known as the traffic intensity. The expression in (3.1) thus allows us to obtain the expected queue length in steady-state

$$\mathbb{E}(Q(\infty)) = \sum_{k=0}^{\infty} k(1 - \rho)\rho^k = \frac{\rho}{1 - \rho},$$

as well as other useful statistics such as the variance, higher moments, quantiles, etc. An example of an simulated $M/M/1$ is given in Figure 3.3.

3.3 Stability Gets Tricky in Networks

We have seen that the questions of stability and steady-state distribution are pretty straightforward with the $M/M/1$ queue (one server, one queue, exponential service times and a FCFS scheduling policy). As it turns out, that’s rather a special case and not the norm. As we make one or more of these dimension more complex, the system dynamics quickly become very non-trivial. In fact, even the seemingly simple question of “when is a queueing system stable?” can be very difficult to answer, and that this is the case is was itself an interesting discovery.

Staring at one queue and one server all day is perhaps not so interesting, and besides, most real-world system often have multiple queues and servers interacting with one another, forming some sort of a network. For example, in manufacturing, a single piece of electronic circuit will go through multiple machines to have different components installed, and sometimes one machine multiple times for different stages of the processing. Similar structure also appear in data center, where servers are dedicated to processing different types of requests, and a single request may require the attention of multiple servers during its entire lifecycle.

If we are to gather together a bunch of servers to form a queueing network, it would probably look something like the one in Figure 3.4. Here we have three servers, each have a dedicated buffer. We will call the collection of a server and its buffer a station a terminology that traces its origin to manufacturing (i.e., a station on the factory floor). Stations 1 and 2 both receive external arrivals, while station 3 only process jobs that have already been through stations 1 and 2. Furthermore, some of the departing jobs from station 3 might require a bit more reworking and are sent back to station 2, while others leave the system.

Queueing networks of this kind has a long history. For instance, the theory of Jackson networks [Jackson, 1963] roughly states that if the arrivals to the queues are Poisson, the service times are exponential and the scheduling policy in each station is FCFS, then the queue lengths admit a nice product-form steady-state distribution. More importantly, checking stability in a Jackson network is
fairly straightforward: it suffices to verify that the total inflow rate of jobs into a station is less than the maximum service rate of that server. If this is true for every station, then the network would be stable.

This however would be far from the end of story for queueing networks. Around the end of 1980s’, a number of counter examples were independently discovered by several researchers that point to the deeper subtitles and richness of the network dynamics ([Kumar and Seidman, 1989, Lu and Kumar, 1991] and [Rybko and Stolyar, 1992]).

Example 3.3.1 (A Counterexample, Figure 3.5). The following counterexample is due to [Lu and Kumar, 1991]. The system has two servers with unit service rate. A single stream of jobs enter from the left at rate \( \lambda = 1 \), and for simplicity we can assume that they arrive uniformly over time. The job is to be served in an order following the blue arrow. That is, when a job in queue 1 completes service, it will join queue 1, and so on and so forth. The job sizes are deterministic and only depend on which queue the job is in. Let \( \tau_i \) denote the job’s size when in queue \( i \). Let \( \tau_1 = \tau_3 = 0 \), and \( \tau_2 = \tau_4 = 2/3 \). Finally, the servers \( A \) and \( B \) use a priority service rule: at server \( A \) priority is given to queue 4 and at server \( B \) priority is given to queue 2. Note that the average rate of inflow of workload to both stations is
3.3. STABILITY GETS TRICKY IN NETWORKS

\[
\frac{2/3 + 0}{2} = 1/3, \text{ which is well below the service rate 1. Plus, the scheduling policy is clearly work-conserving. One might thus expect that the system should be stable.}
\]

Let’s see what could go wrong in this example.

(a) Suppose initially \( Q_1(0) = n \), and all other queues are empty. Because \( \tau_1 = 0 \) all \( n \) jobs immediately get processed by server A and transferred to queue 2. Server B immediately starts to work on these jobs.

(b) By time \( t = 2n^- \), queue 2 will have been empty again for the first time: during the first \( 2n \) units of time, server B will have just completed the \( n \) initial jobs, as well as the \( 2n \) new jobs that have been added to the system during this time.

(c) At this point (\( t = 2n^+ \)), server B instantly completes all \( 3n \) jobs in queue 3 and sends them to queue 4.

(d) Server A now spends \( 2n \) units of time to complete these \( 3n \) jobs in queue 4, bringing the total time lapsed to \( t = 2n + 2n = 4n \).

Figure 3.4: Illustration of a network of queues.

(2/3 + 0)/2 = 1/3, which is well below the service rate 1. Plus, the scheduling policy is clearly work-conserving. One might thus expect that the system should be stable.

Let’s see what could go wrong in this example.

(a) Suppose initially \( Q_1(0) = n \), and all other queues are empty. Because \( \tau_1 = 0 \) all \( n \) jobs immediately get processed by server A and transferred to queue 2. Server B immediately starts to work on these jobs.

(b) By time \( t = 2n^- \), queue 2 will have been empty again for the first time: during the first \( 2n \) units of time, server B will have just completed the \( n \) initial jobs, as well as the \( 2n \) new jobs that have been added to the system during this time.

(c) At this point (\( t = 2n^+ \)), server B instantly completes all \( 3n \) jobs in queue 3 and sends them to queue 4.

(d) Server A now spends \( 2n \) units of time to complete these \( 3n \) jobs in queue 4, bringing the total time lapsed to \( t = 2n + 2n = 4n \).

Figure 3.4: Illustration of a network of queues.

Figure 3.5: A counter example by Lu and Kumar. The blue arrow indicates the job flow, and the red > signs indicate priority, with greater value being higher priority.

Figure 3.5: A counter example by Lu and Kumar. The blue arrow indicates the job flow, and the red > signs indicate priority, with greater value being higher priority.
We are now at $t = 4n$, and notice that all queues except for queue 1 are again empty. However, due to the priority rule at server A, there are now $2n$ new jobs in queue 1, which arrived between $t = 2n$ and $t = 4n$. During this period, server A was busy processing jobs in queue 4. In other words, we are back to the system state at time $t = 0$, except for that instead of $n$ jobs, there are now $2n$ jobs in queue 1. Repeating this argument, we see that the size of queue 1 is bound to double at $t = 4n, 16n, 64n, \ldots$. This shows that the system cannot be stable.

In general, these counter-examples demonstrate that even when the servers are work-conserving (i.e., never idle when there are jobs to process) and the total arrival rate to a station is less than that of the service rate, the system can still become in-stable. The culprit is often the scheduling policies used at the server, where a certain priority rules that prefer one type of jobs than others could lead to unnecessary “blocking”, and the resulting inefficiently will eventually drive the queues to infinity. The key take-away here is that complex dynamics can arise when we connect multiple simple queueing modules that individually have seemingly reasonable operating characteristics.

**Question 3.3.1.** As a thought experiment, discuss how you would go about modeling the dynamics of the network in Example 3.3.1. To make things simpler, you can assume that the jobs sizes are all i.i.d. exponentially distributed (even with the same mean), and the arrival process is Poisson. How easy is it to write down a distribution for the queue lengths at time $t$? You may also try simulating it; what do you notice when running the simulation?

### 3.4 What’s Next?

The discoveries of these counter examples led to much excitement and confusion in the research community. If even pretty simple networks can misbehave, how can we be sure that a processing network is designed in such a way that the type of deadlock does not happen? While the question of stability remains an open question in certain complex networks today, we will see in the next chapter a potent tool that will allow us to establish universal stability in a quite impressive family of systems. This will also be the first time we encounter the power of the drift method, in the form of the Foster-Lyapunov stability criteria.

### 3.5 Notes

While the network in Example 3.3.1 uses jobs with zero size, the example given by [Rybko and Stolyar, 1992] does not, but requires two streams of arrivals. We will generally refer to the network of interconnected servers and queues as stochastic processing networks, instead of queueing network, which in principle could be an equally valid terminology. However, as pointed out in [Dai and Harrison, 2020], the latter tend to have a more precise, and narrower, meaning in the literature.
Chapter 4

Maximum Stability in Stochastic Networks

We have seen in the previous chapter that connecting together queuing with seemingly reasonably scheduling policies could lead to a network that is in-stable, even when each server remains well underloaded. We will see in this section a solution to this problem, even in the absence of conflicts and inferences between the servers.

4.1 Single-Hop Stochastic Processing Network

We begin by formally define the single-hop stochastic processing network, also known as the generalized switch [Stolyar et al., 2004]. The model operates in discrete time, and consists of the following components:

1. There are $n$ queues. The queue length at time $t$ is denoted by $Q(t) \in \mathbb{Z}_n^+$. 

2. The arrivals to the system at time $t$ is denoted by $A(t) \in \mathbb{Z}_n^+$, where $A_i(t)$ is the number of jobs arriving to queue $i$. The $A(t)$'s are assumed to be i.i.d. We will call $\lambda = \mathbb{E}[A(0)]$ the arrival rate vector. The exact distribution of $A$ turns out to be not so essential when it comes to stability. To make our exposition simpler, we will assume that $A_i(0)$ follows a Bernoulli distribution with mean $\lambda_i$.

3. The schedule set $D = \{d^{(1)}, \ldots, d^{(|D|)}\}$ is a finite subset of $\mathbb{Z}_n^+$. Each element of $D$ is called a schedule, representing the expected number of jobs that are expected to depart, should a given scheduling action is chosen. In particular, if the decision maker chooses at $t$ a schedule $D(t) \in D$, then 

$$Q(t + 1) = (Q(t) + A(t) - D(t))^+.$$ 

(4.1)

Alternatively, we can write the above evolution as 

$$Q(t + 1) = Q(t) + A(t) - D(t) + Z(t),$$

(4.2)

where $Z(t) := [D(t) - (Q(t) + A(t))]^+$ is the amount of idling service, i.e., the portion of $D(t)$ that did not result in any departure due to the queue having already reached zero.

Throughout, we will assume that $D$ is fixed and known. Without loss of generality, we will require that each queue is served, to some degree, by at least one schedule:

$$\min_{i=1,\ldots,n} \max_{d \in D} d_i > 0.$$ 

(4.3)
CHAPTER 4. MAXIMUM STABILITY IN STOCHASTIC NETWORKS

It’s clear that if this condition does not hold, then jobs in a certain queue \( i \) will never be served, and the system cannot be stable whenever queue \( i \) receives any arrivals at all.

The job of a scheduling policy is thus to choose in every time slot a schedule from \( D \), while trying to keep the queues small. As mentioned in the Introduction, a basic criteria of desirable performance is that all jobs that arrive to the system should be processed within finite time. In other words, the queue size should remain stable over time. We now make this notion of stability precise.

**Definition 4.1.1 (Stationary Policy).** We say a policy \( \phi \) is stationary if the distribution of its action at time \( t \) only depends on the state of the system, \( Q(t) \).

For simplicity, we will define the notion of stability with respect to stationary policies. This is because if we apply a stationary scheduling policy, then the queue length process \( Q \) evolves as a time-homogeneous Markov process.\(^1\)

**Definition 4.1.2.** Fix a stationary scheduling policy \( \phi \) and arrival rate vector \( \lambda \). We say that \( \phi \) stabilizes the system if the resulting queue length process is positive recurrent.

Define the maximum stability set, \( \Pi \), to be:

\[
\Pi = \{ \lambda : s \in \text{Conv}(D), s \geq \lambda \}, \tag{4.4}
\]

where \( \text{Conv}(D) \) is the convex hull of \( D \), defined to be the set of all vectors that can be written as convex combinations of elements in \( D \). The set \( \Pi \) is often referred to as the capacity region. The name “maximum stability” comes from the fact that if \( \lambda \notin \Pi \), then it’s clear that there is no scheduling policy that could be stable under \( \lambda \), as given by the following lemma.

**Definition 4.1.3.** We say a stationary scheduling policy \( \phi \) is maximumly stable, if for any \( \rho \in (0,1) \), it is stable for any \( \lambda \in \rho \Pi \).

Note that the maximum stability criterion is quite a bit stronger than plain-vanilla stability: while stability requires \( \phi \) to stabilize one arrival rate vector, maximum stability mandates that \( \phi \) stabilize all arrival rate vectors in the interior of \( \Pi \).

### 4.2 Main Theorem: Maximum Stability of the Max-Weight Policy

**Definition 4.2.1 (Max-Weight Policy).** The Max-Weight policy selects a schedule based on the following rule:

\[
D(t) \in \arg \max_s \langle s, Q(t) \rangle. \tag{4.5}
\]

For each unit of potential departure from queue \( i \), the Max-Weight policy associated with it “weight” that is equal to the queue length, \( Q_i(t) \). It then chooses the schedule that has the maximum total weight. An intuitive justification for this policy is that, everything else being equal, we prefer to give priority to draining queues that are longer.

We’ve now arrived our main theorem:

**Theorem 4.2.1.** The Max-Weight policy is maximumly stable.

\(^1\)Note that the stationarity of a policy is not to be confused with the stationarity of a Markov process. E.g., \( Q \) under a stationary policy might not be stationary.
4.3 Proof of Main Theorem and the Foster-Lyapunov Stability Criterion

To prove Theorem 4.2.1, we will make use of the following theorem, known as the Foster-Lyapunov stability criterion. Let $X$ be an irreducible discrete-time Markov process defined on a countable state-space $S$, with transition probabilities $P$. We can apply $P$ upon a function $f : S \to \mathbb{R}_+$ as an operator in the following sense: $Pf$ is another function $S \to \mathbb{R}_+$, where

$$Pf(i) := \sum_{j \in S} P_{ij} f(j), \quad i \in S.$$  \hfill (4.6)

That is, $Pf(i) = \mathbb{E}[f(X(t + 1)) \mid X(t) = i]$.

**Theorem 4.3.1** (Foster-Lyapunov Criterion). Let $X$ be an irreducible discrete-time Markov process defined on a countable state-space $S$, with transition probabilities $P$. Fix $V : S \to \mathbb{R}_+$, and a finite set $C$ of $S$. Then,

1. If $\{i \in S : V(i) \leq K\}$ is finite for all $K$, and

   $$(PV - V)(i) \leq 0$$  \hfill (4.7)

   for all $i \in S \setminus C$, then $X$ is recurrent.

2. Suppose there exist constants $\epsilon$ and $b > 0$, such that

   $$(PV - V)(i) \leq -\epsilon + bI(i \in C), \quad \forall i \in S.$$  \hfill (4.8)

   Then, $X$ is positive recurrent.

The function $V$ here is commonly known as the Lyapunov function. There are two key conditions for achieving positive recurrence (Figure 4.1.1). First, if $X(t)$ is not inside the finite set $C$, we must observe a negative drift in the Lyapunov function of magnitude at least $\epsilon$. Second, if $X(t)$ is in $C$, we should...
then the Lyapunov function is allowed to have a positive drift, but the magnitude of such a drift must be uniformly bounded from above by some constant $b$.

**Proof of Theorem 4.2.1.** We first state a useful lemma concerning the geometry of the set $\Pi$.

**Lemma 4.3.2.** Fix any $x \in \mathbb{R}_+$, and $\epsilon \in (0, 1)$. Let $y \in (1 - \epsilon)\Pi$. We have

$$
\langle x, y \rangle \leq (1 - \epsilon) \max_{d \in \mathcal{D}} \langle x, d \rangle.
$$

(4.9)

**Proof.** Let $y' := y(1 - \epsilon)^{-1}$. By definition, $y' \in \Pi$, which further implies that there exists a probability vector $\{p_i : i = 1, 2, \ldots, |\mathcal{D}|\}$, such that $y' \leq \sum_{i=1}^{\mathcal{D}} p_i d_i$. We thus have

$$
\langle x, y' \rangle \leq \left( x, \sum_i p_i d_i \right) = \sum_i p_i \langle x, d_i \rangle \leq \sum_i p_i \max_{d \in \mathcal{D}} \langle x, d \rangle = \max_{d \in \mathcal{D}} \langle x, d \rangle,
$$

(4.10)

where in the last step we used the fact that $\{p_i\}$ is a probability vector and $\sum_i p_i = 1$. Substituting $y = y'(1 - \epsilon)$ into the above inequality proves the lemma. $\square$

![Figure 4.2: A visual illustration for Lemma 4.3.2.](image)

Note that the queue length process $Q$ is an irreducible Markov chain under the Max-Weight policy, taking values in the countable state space $S = \mathbb{Z}_n^+$. We will use the following quadratic Lyapunov function:

$$
V(x) = \sum_{i=1}^{\mathcal{D}} (x_i)^2 = \|x\|_2^2,
$$

(4.11)

where $\|\cdot\|_2$ is the $L_2$ norm. Fix $M > 0$, and let $\mathcal{C}_M$ be defined by

$$
\mathcal{C}_M = \{x \in \mathbb{Z}_n^+ : \max_{d \in \mathcal{D}} \langle d, x \rangle \leq M\}.
$$

(4.12)

We can verify that $\mathcal{C}_M$ has a finite cardinality as long as the condition in (4.3) holds.
Define $\Delta(t) := A(t) - D(t)$. Let $c_1$ be a constant such that $c_1 > |A_i| + |D_i|$ with probability one. Fix $x \in S$. We have that

$$(PV - V)(x) \leq E \left[ V(Q(t + 1)) - V(Q(t)) \mid Q(t) = x \right]$$

$$= E \left[ \|x + A(t) - D(t) + Z(t)\|^2_2 - \|x\|^2_2 \mid Q(t) = x \right]$$

$$\leq E \left[ \|x + A(t) - D(t)\|^2_2 - \|x\|^2_2 \mid Q(t) = x \right]$$

$$= \sum_{i=1}^n E \left[ (x_i + \Delta_i(t))^2 - x_i^2 \mid Q(t) = x \right]$$

$$= \sum_{i=1}^n E \left[ 2x_i\Delta_i(t) + \Delta_i(t)^2 \mid Q(t) = x \right]$$

$$\leq c_1^2 n + 2 \sum_{i=1}^n E \left[ x_i\Delta_i(t) \mid Q(t) = x \right]$$

$$= c_1^2 n + 2 \langle x, \ E \left[ \Delta(t) \mid Q(t) = x \right] \rangle \tag{4.13}$$

where step (a) follows from the fact that whenever $Z_i(t) \neq 0$, we have $Q_i(t + 1) = 0$, and therefore setting $Z_i(t)$ to 0 cannot decrease the term $\|x + A(t) - D(t) + Z(t)\|^2_2$. Step (b) follows from the definition of $c_1$.

We now bound the term $\langle x, \ E \left[ \Delta(t) \mid Q(t) = x \right] \rangle$, and this is where the Max-Weight policy comes in. Recall that the Max-Weight policy chooses in each time slot the schedule $d \in D$ that maximizes the inner product $\langle d, Q(t) \rangle$. We thus have

$$\langle x, \ E \left[ \Delta(t) \mid Q(t) = x \right] \rangle = \langle x, \ E \left[ A(t) - D(t) \mid Q(t) = x \right] \rangle$$

$$= \langle x, \ E \left[ A(t) \mid Q(t) = x \right] \rangle - \langle x, \ E \left[ D(t) \mid Q(t) = x \right] \rangle$$

$$\overset{(a)}{=} \langle x, \lambda \rangle - \langle x, \ E \left[ D(t) \mid Q(t) = x \right] \rangle$$

$$\overset{(b)}{=} \langle x, \lambda \rangle - \max_{d \in D} \langle x, d \rangle . \tag{4.14}$$

where step (a) is due to the fact that the arrivals are independent of the queue lengths, and (b) follows from the definition of the Max-Weight policy. We next evoke Lemma 4.3.2, with $\epsilon = 1 - \rho$. Recall that we have assumed $\lambda \in \rho \Pi$, which implies that

$$\langle x, \lambda \rangle \leq \rho \max_{d \in D} \langle x, d \rangle . \tag{4.15}$$

Combining the above two inequalities, we obtain that

$$\langle x, \ E \left[ \Delta(t) \mid Q(t) = x \right] \rangle \leq \langle x, \lambda \rangle - \max_{d \in D} \langle x, d \rangle$$

$$\leq -(1 - \rho) \max_{d \in D} \langle x, d \rangle . \tag{4.16}$$

If we further assume that $x \notin C_M$, this leads to

$$\langle x, \ E \left[ \Delta(t) \mid Q(t) = x \right] \rangle \leq -(1 - \rho) \max_{d \in D} \langle x, d \rangle \leq -(1 - \rho) M. \tag{4.17}$$

We now set $M$ to be sufficiently large to achieve a uniformly negative drift. Fix $\epsilon > 0$ and set

$$M = \frac{c_1^2 n + \epsilon}{2(1 - \rho)}. \tag{4.18}$$
Combining (4.13) and (4.17), we conclude that if \( x \notin C_M \), then
\[
(PV - V)(x) \leq c_1^2n - 2(1 - \rho)M = -\epsilon.
\]
We are almost ready to evoke the Foster-Lyapunov theorem. What remains to verify is that if \( x \in C_M \), then \((PV - V)(x)\) is bounded from above by some constant \( b \). This follows easily by noticing that the changes in queue length from \( t \) to \( t - 1 \) are bounded with probability 1. We can therefore evoke Theorem 4.3.1, and conclude that \( Q \) is positive recurrent. Since this holds true for any \( \rho \in (0, 1) \) and \( \lambda \in \rho \Pi \), we have established the maximum stability of the Max-Weight policy.

Remark 4.3.3. We have cheated here slightly to simplify the proof. In general, the arrivals and potential departures might not have a finite support. In that case, the standard assumption is that the second moments of \( A(t) \) and \( D(t) \) be bounded. Then, the same proof steps should work out just fine, with a bit more tedious algebra to keep track of the error terms.

4.4 Multi-Hop Networks and the Back-pressure Algorithm

The original Max-Weight policy was actually proposed in [Tassiulas and Ephremides, 1990] to provide stability in a multi-hop queueing network, where it is also known as the back-pressure algorithm. We have actually already seen an example of a multi-hop network earlier in Figure 3.5, where a job needs to complete serve at a sequence of servers before leaving the system. This type of model comes up in communication systems, where a job can be thought of as a data packet that needs to travel from one node to another in a network. To do so, it must be transmitted along a route, which consists of a sequence of links. Each link is bandwidth-limited, and can therefore be modeled by a capacity-constrained server.

The stability analysis we have thus performed for the single-hop network can be adapted to the multi-hop setting. The key modification here is to express the network topology by ensuring that when a job is served from an upstream queue, the same job needs to be added to the down-stream queue. And we can again use the quadratic Lyapunov function. Carrying out the stability analysis rigorously will be left as an exercise.

4.5 Notes

The Max-Weight and Back-Pressure policies, as well as the notion of maximum stability were first proposed in the celebrated work [Tassiulas and Ephremides, 1990], which employed a quadratic Lyapunov function to establish stability. The version of the Foster-Lyapunov criterion used here is based on [Hajek, 2006]. In the operations research literature, a well-known version of the Max-Weight type policy is known as the \( c\mu \) rule [Mandelbaum and Stolyar, 2004].
Chapter 5

More on the Foster-Lyapunov Criterion

5.1 Proof of the Foster-Lyapunov Criterion

We now prove the Foster-Lyapunov criterion in Theorem 4.3.1, reproduced below for convenience.

**Theorem 5.1.1.** Let \( X \) be an irreducible discrete-time Markov process defined on a countable state-space \( S \), with transition probabilities \( P \). Fix \( V : S \to \mathbb{R}_+ \), and a finite set \( C \) of \( S \). Then,

1. If \( \{ i \in S : V(i) \leq K \} \) is finite for all \( K \), and
   \[ (PV - V)(i) \leq 0 \] for all \( i \in S \setminus C \), then \( X \) is recurrent.

2. Suppose there exist constants \( \epsilon \) and \( b > 0 \), such that
   \[ (PV - V)(i) \leq -\epsilon + b\mathbb{1}(i \in C), \quad \forall i \in S. \] Then, \( X \) is positive recurrent.

**Proof.** We first prove part (a), which relies on the martingale convergence theorem (Theorem 2.4.1). Suppose \( X \) starts in the initial state \( i_0 \in S \setminus C \). Define the stopping time \( \tau := \min\{ t \geq 0 : X(t) \in C \} \).

Define the derived process
\[ Y(t) = V(X(t \wedge \tau)). \]

That is, \( Y(t) \) tracks the value of \( V(X(t)) \) until \( X \) first visits \( C \), and thereafter its value stays at that of \( V(X(\tau)) \). Since the value of \( Y \) would stop changing after \( t \geq \tau \), we have
\[ Y(t+1) - Y(t) = (V(X(t+1))) - V(X(t)) \mathbb{1}(t < \tau). \]

Let \( \mathcal{F}_t \) be the filtration generated by \( \{X(s)\}_{s=0,...,t} \). By definition, the event \( \{t < \tau\} \) is measurable with respect to \( \mathcal{F}_t \). Therefore, we have
\[
\mathbb{E} \left[ Y(t+1) - Y(t) \mid \mathcal{F}_t \right] = \mathbb{E} \left[ (V(X(t+1)) - V(X(t))) \mathbb{1}(t < \tau) \mid \mathcal{F}_t \right] \\
= \mathbb{E} \left[ V(X(t+1)) - V(X(t)) \mid \mathcal{F}_t \right] \mathbb{1}(t < \tau) \\
\leq 0,
\]
where step (a) follows from the \( \mathcal{F}_t \)-measurability of the random variable \( I(t < \tau) \), and the last step from the assumption that \( (PV - V)(i) \leq 0 \) whenever \( i \notin C \).

We have thus shown that \( Y \) is a non-negative supermartingale. Applying the martingale convergence theorem (Theorem 2.4.1) to \( Y \), we conclude that \( Y(t) \) converges almost surely to a well-defined limit as \( t \to \infty \). This further implies that, almost surely, the process \( \sup_t V(X(t) \wedge \tau) \) is finite. Since we have assumed that the set \{ \( i : V(i) < K \) \} is finite for any finite \( K \), we conclude that the process \( X(t \wedge \tau) \) will visit a finite number of states in \( S \) with probability one. Because \( X \) is irreducible, this further shows that we must have

\[
\mathbb{P}(\tau < \infty \mid X(0) = i_0) = 1, \quad \forall i_0 \in S \setminus C.
\] (5.7)

Therefore, if we start the process from a state in \( C \), \( X \) will return to \( C \) infinitely often with probability one, which, again by irreducibility, shows that all states in \( C \) are recurrent for the original process \( X \). This completes the proof of (a).

The proof of part (b) of the theorem will be a bit more delicate, because not only do we need to demonstrate that starting from \( C \) the process \( X \) will eventually returns, we want to show that such return is sufficiently often. To start, the following lemma establishes a sufficient condition for positive recurrence, where instead of requiring the mean return time to each state be finite, it suffices to require that for the return time to a set of states.

**Lemma 5.1.2.** Let \( X \) be an irreducible and time-homogeneous Markov process. Suppose there exists a finite set \( C \subset S \), such that the mean hitting time to \( C \) starting from any states in \( C \) is finite, then \( X \) is positive recurrent.

**Proof.** (Sketch) Let \( X(0) = i \) for some \( i \in C \) and let \( \psi \) be the first time \( X(t) = i \) for \( t \geq 1 \). If we restrict the process to \( C \) only by ignoring all trajectories when \( X(t) \notin C \), we see that the restricted process is a finite-state, irreducible Markov process, and hence is positive recurrent. Let’s call this mean hitting time of the restricted chain \( m_1 \). This means that portion of hitting time \( \psi \) spent during this restricted process has finite mean. Now, at each step of this restricted process, the actual process could take an excursion outside of \( C \). However, the expected duration of these excursions are uniformly bounded by some constant \( m_2 \) across the states in \( C \). Stitching these two facts together with a version of Wald’s equality yields the fact that \( \mathbb{E} \left[ \tau \mid X(0) = i \right] \leq m_1 m_2 < \infty \).

In light of this lemma, we will focus on showing that the process \( X \) linger too long outside of \( C \). This is somewhat expected from the Lyapunov condition: for as long as \( X(t) \) is not in \( C \), the value of \( V(X(t)) \) will experience negative drift. If we think of the initial value of \( V(X(t)) \) at some \( t \) as a “potential budget”, then this budget should be depleted \( V(X(t))/\epsilon \) time slots later, should the process remain outside of \( C \) this whole time. Making this intuition rigorous would require some careful manipulation of expectations, as we do next:

**Lemma 5.1.3.** Let \( f, g \) be some non-negative functions defined over \( S \). Suppose

\[
P V - V \leq -f + g
\] (5.8)

over \( S \). Then, for any initial state \( i_0 \) and stopping time \( \tau \),

\[
\mathbb{E} \left[ \sum_{t=0}^{\tau-1} f(X(t)) \right] \leq V(i_0) + \mathbb{E} \left[ \sum_{t=0}^{\tau-1} g(X(t)) \right]
\] (5.9)

**Proof.** Let \( \mathcal{F}_t \) be the filtration generated by \( \{X(s)\}_{s=0,\ldots,t} \). Since by assumption \( (PV - V)(i) \leq -(f + g)(i) \) for all \( i \in S \), replacing \( i \) with \( X(t) \) and taking expectation on both sides of the inequality...
with conditioning on $\mathcal{F}_t$ yields

$$
E \left[ V(X(t + 1)) \mid \mathcal{F}_t \right] + f(X(t)) \leq V(X(t)) + g(X(t)).
$$

(5.10)

Define the stopping time

$$
\tau^n = \min\{\tau, n, \inf\{t \geq 0 : V(X(t)) \geq n\}\}.
$$

(5.11)

(The purpose of $\tau^n$ is to induce a sequence of well-defined monotone random variables so that we can evoke monotone convergence theorem down the road.) We now truncate (5.10) by incorporating the indicator variable $I(t < \tau^n)$ in the terms and take an (unconditional) expectation:

$$
E \left[ E \left[ V(X(t + 1)) \mid \mathcal{F}_t \right] I(t < \tau^n) \right] + E[f(X(t))I(t < \tau^n)] \\
\leq E[V(X(t))I(t < \tau^n)] + E[g(X(t))I(t < \tau^n)].
$$

(5.12)

Note that $I(t < \tau^n)$ is $\mathcal{F}_t$-measurable, and hence we have

$$
E \left[ E \left[ V(X(t + 1)) \mid \mathcal{F}_t \right] I(t < \tau^n) \right] = E \left[ E \left[ V(X(t + 1))I(t < \tau^n) \mid \mathcal{F}_t \right] \right] \\
= E[V(X(t + 1))I(t < \tau^n)] \\
\geq E[V(X(t + 1))I(t < \tau^n - 1)].
$$

(5.13)

Substituting this into (5.12), we get

$$
E [V(X(t + 1))I(t < \tau^n - 1)] + E[f(X(t))I(t < \tau^n)] \\
\leq E[V(X(t))I(t < \tau^n)] + E[g(X(t))I(t < \tau^n)].
$$

(5.14)

Note that by definition $\tau^n \leq n$ with probability one, and hence all terms in the above inequality become 0 for $t \geq n$. Furthermore, also by the definition of $\tau^n$, we have that $E[V(X(t + 1))I(t < \tau^n - 1)] \leq n < \infty$ for all $t$. We may therefore sum both sizes of (5.14) over $t \geq 0$, and cancel out identical terms involving $V(X(t))$ on both sides, to obtain:

$$
E \left[ \sum_{t=0}^{\tau^n-1} f(X(t)) \right] \leq V(i_0) + E \left[ \sum_{t=0}^{\tau^n-1} g(X(t)) \right].
$$

(5.15)

Finally, because $f$ and $g$ are non-negative, both $E \left[ \sum_{t=0}^{\tau^n-1} f(X(t)) \right]$ and $E \left[ \sum_{t=0}^{\tau^n-1} g(X(t)) \right]$ are monotone non-decreasing in $n$. Taking the limit as $n \to \infty$, and using the fact that $\tau^n \nearrow \tau$ as $n \to \infty$, we conclude the proof by evoking the monotone convergence theorem.

We are now ready to complete the proof of part (b). Let $f := \epsilon, g = bI(\cdot \in \mathcal{C})$ and $\tau = \min\{t \geq 1 : X(t) \in \mathcal{C}\}$. Using Lemma 5.1.3, we have that

$$
e E[\tau] \leq V(i_0) + b
$$

(5.16)

for any $i_0 \in \mathcal{C}$. In particular, the term $b$ on the right stems from the fact that $g(X(0)) = b$ and $g(X(t)) = 0$ for all $t = 1, \ldots, \tau - 1$ because $X(t)$ is outside of $\mathcal{C}$ during this period. This shows that the mean hitting time to set $\mathcal{C}$ starting from any state in $\mathcal{C}$ is bounded. Evoking Lemma 5.1.2 proves the positive recurrence of $X$. 

5.2 Steady-State Drift via Foster-Lyapunov

We have thus far used the negative drift of a Lyapunov function to establish positive recurrent (stability) of a queueing system. However, we have not quite used the full power of the drift, especially in the scenario where the magnitude of the negative drift outside of the finite set \( C \) is state dependence. Indeed, a closer inspection of the derivation in the previous chapter reveals that the magnitude of the negative drift increases as the queue length vector \( x \) grows. This is a straightforward consequence due to the curvature of a quadratic Lyapunov function: for the same differential change in queue lengths, the further away you are from the origin, the steeper a one-step reduction in \( V \) becomes.

Before illustrating this in the specific example of max-weight algorithm, let us introduce a more general result, which gives an upper bound on the steady-state drift. The result is a variant of Lemma 5.1.3, where the function \( f \) can be thought of as the magnitude of the (negative) drift in the Lyapunov function when \( X \) is outside \( C \).

Lemma 5.2.1 (Steady-State Drift Bound). Let \( f, g \) be some non-negative functions defined over \( S \). Suppose

\[
PV - V \leq -f + g \tag{5.17}
\]

over \( S \). Suppose, in addition, that the Markov process \( X \) is positive recurrent, with steady-state distribution \( X(\infty) \). Then

\[
E[f(X(\infty))] \leq E[g(X(\infty))] \tag{5.18}
\]

Proof. Sketch. Let us first look at a proof sketch for the main idea behind the result. Since \( X \) admits a valid steady-state distribution \( X(\infty) \), we can take expectation on both sides of the inequality with respect to \( X(\infty) \):

\[
E\left[ E[V(X(t+1)) \mid X(t) = X(\infty)] \right] - E[V(X(\infty))] \leq -E[f(X(\infty))] + E[g(X(\infty))] \tag{5.19}
\]

Because \( X \) is stationary with respect to the steady-state distribution \( X(\infty) \), we have

\[
E\left[ E[V(X(t+1)) \mid X(t) = X(\infty)] \right] = E[V(X(\infty))] \tag{5.20}
\]

Therefore the left-hand side of (5.19) is equal to zero, and the claim follows. Unfortunately, this argument requires that \( E[V(X(t+1)) \mid X(t) = X(\infty)] \) and \( E[V(X(\infty))] \) be finite, and hence isn’t rigorous.

Formal Proof. The issue of integrability was mitigated in the proof of Lemma 5.1.3 via the use of the stopping time \( \tau^n \) and the monotone convergence theorem, and we shall leverage this result in the proof. Fix \( i_0 \in C \) and suppose \( X \) is initialized in state \( i_0 \). Define \( \tilde{\tau}^m \) as the time \( X \) visits state \( i_0 \). Because the times between successive visits to \( i_0 \) are i.i.d., we have that \( \tilde{\tau}^m \) can be written as the sum of \( m \) i.i.d. random variables each with the distribution of \( \tilde{\tau}^1 \). If follows that the expected value of the time-average value of \( f(X(t)) \) during \( [\tilde{\tau}^1, \tilde{\tau}^{i+1}] \) is equal to \( E[f(X(\infty))] \). We thus have that

\[
E\left[ \sum_{t=0}^{\tilde{\tau}^m-1} f(X(t)) \right] = mE[\tilde{\tau}^1] E[f(X(\infty))],
\]

\[
E\left[ \sum_{t=0}^{\tilde{\tau}^m} -g(X(t)) \right] = mE[\tilde{\tau}^1] E[g(X(\infty))].
\]

We now apply Lemma 5.1.3 with the stopping time \( \tilde{\tau}^m \), and obtain

\[
mE[\tilde{\tau}^1] E[f(X(\infty))] \leq V(i_0) + mE[\tilde{\tau}^1] E[g(X(\infty))]. \tag{5.21}
\]

Dividing both sides of the above inequality by \( mE[\tilde{\tau}^1] \) and taking the limit as \( m \to \infty \) leads to the desired result. \( \square \)
5.3  Bounding Steady-State Queue Lengths with Drift

Let’s see how to use Lemma 5.2.1 to derive a bound on the steady-state queue lengths under the Max-Weight policy.

**Example 5.3.1 (Steady-State Queue Lengths under Max-Weight Policy).** Recall from the previous chapter ((4.13) and (4.16)), we have

\[
(PV - V)(x) \leq c_1^2 n - 2(1 - \rho) \max_{d \in D} \langle x, d \rangle \leq c_1^2 n - 2(1 - \rho) \langle x, \lambda \rangle. \tag{5.22}
\]

Applying Lemma 5.2.1 with \(f(x) = 2(1 - \rho) \langle x, \lambda \rangle\) and \(g(x) = c_1^2 n\) immediately yields the following result:

**Proposition 5.3.1.** Fix \(\rho \in (0, 1)\) and \(\lambda \in \rho \Pi\). Under the Max-Weight scheduling policy, the queue length process is positive recurrent, and its steady-state distribution satisfies

\[
\langle \mathbb{E}[Q(\infty)], \lambda \rangle \leq \frac{c_1^2 n}{2(1 - \rho)}. \tag{5.23}
\]

Proposition 5.3.1 has several interesting features. First, the characterization is given in the form of a weighted average queue lengths, with weight vector \(\lambda\). If we normalized both sides of the inequality by the \(L_1\) norm of \(\lambda\), this weighted average queue length can be interpreted as the average queue length seen from the perspective of a “typical” arrival joining the system, since such an arrival would join queue \(i\) with probability \(\propto \lambda_i\). Second, as the traffic intensity \(\rho \to 1\), the upper bound shows that

\[
\langle \mathbb{E}[Q(\infty)], \lambda \rangle = \mathcal{O}\left(\frac{1}{1 - \rho}\right), \tag{5.24}
\]

which coincides with the celebrated \(\mathcal{O}\left(\frac{1}{1 - \rho}\right)\) heavy-traffic scaling seen in many queueing systems. Finally, note that the upper bound depends on the number of queues \(n\) in a linear manner, which degenerates as \(n\) grows. We will see in later chapters that this dependence can be removed to a certain degree when we combine the Lyapunov drift method with so-called state-space collapse.

Lemma 5.2.1 is sometimes referred to as the moment bound, and its application indeed extends beyond bounding the expected queue length. In the example we just saw, the analysis of the Max-Weight policy uses a quadratic Lyapunov function, which leads to a drift term \((f)\) that is a linear function in the queue lengths. In general, if we can establish a drift condition where \(f\) is say, a higher order polynomial of \(x\) or even exponential, then we should be able to obtain bounds on the higher moments of the steady-state distribution, large-deviation coefficients, etc.

5.4 Notes

The proofs for the Foster-Lyapunov criterion as well as the steady-state drift bound are based on [Hajek, 2006].
Chapter 6

Drift in Complex Systems via Fluid Limits

In previous chapters, we have developed the following program for establishing stability (i.e., positive recurrence) of a particular policy. That is: (1) show that the system state evolves as a time-homogeneous Markov process \( X \) under the said policy; (2) identify a suitable Lyapunov function \( V \) and (3) show that the one-step expected drift of \( V(X(t)) \) is strictly negative whenever the state escapes a finite set. In practice, step (1) is often immediate, and Lyapunov function in (2) is also not too hard to find (one would typically start with a quadratic function). Challenge often come in step (3) when one tries to establish the negative drift.

Example 6.0.1. Here is a simple example of when step (3) can fail: We often encounter in practice a stochastic system where the drift depends on some modulating random state, such as weather. We may observe that when managing a single queue, the length drift down when the weather is good, but up when weather is bad. Suppose we know that the weather state is a simple two-state Markov process, oscillating between good and bad, and good weather shows up 99.9\% of the time in equilibrium. Any reasonable guess would suggest that the system should be stable as long as the average drift, weighted by the equilibrium probability of good and bad weather, and in this case it’s almost always good. Surprisingly enough, it’s actually quite hard to apply the Lyapunov technique to establish stability in this trivial example, and in particular, step (3) is especially hard. This is because that we cannot simply abstract away the weather state and substitute it with its “average”. On the one hand, if we only look at the queue length process and ignore the weather state, the system is not a Markov process, and hence the Foster-Lyapunov theorem fails. If, on the other hand, we consider the queue length and weather processes jointly as a single Markov process, then, the drift in the Lyapunov function is not negative whenever the weather state is bad, regardless how long the queue is. This makes it pretty hard to find a finite set of states outside of which the negative is always negative.

Indeed, the requirement that the Lyapunov function \( V(X(t)) \) over one step is pretty stringent and somewhat arbitrary. Why not showing a negative drift over 2, 3 or 10 steps, or even across a random number of steps? In a similar way, what if the system’s evolution depends on some modulating process as in the previous example, so that the drift might not be negative in every state of the world, but tends to be negative “on average”?

One could in principle derive a separate version of the Foster-Lyapunov theorem for each one of these variations, but this is clearly not ideal and gets messy pretty quickly. In this chapter, we will look at a unifying approach, which elegantly generalizes the Foster-Lyapunov criterion. The framework roughly consists of the following steps:
1. **Convergence to fluid limits.** Show the original Markov process $X(\cdot)$, under some appropriate limit and scaling, converges to a limiting “fluid” trajectory, $x(\cdot)$. The evolution of $x$ corresponds to the solution to a set of ordinary differential equations (ODE).

2. **Uniform attraction of fluid limit.** Show that for all bounded initial conditions, the fluid limit $x(t)$ converges to 0 as $t \to \infty$ uniformly fast. This step is often accomplished by analyzing the drift of $x$.

3. **Attraction to positive recurrence.** Argue that the attraction of the fluid limit readily implies the positive recurrence of the original, pre-limit process $X$.

As a result, we will be able to tackle the dynamics of more complex systems with a more streamlined set of arguments. As a bonus, by replacing expected drift in a finite system with the derivative of a limiting trajectory, the fluid limit perspective also tends to be algebraically simpler, and thus better illustrates the conceptual underpinnings of various adaptive algorithms.

In the remainder of this chapter, we will describe each of the above three steps in more detail. We will conclude the chapter by showing how this would manifest in the stochastic network model we have been studying thus far. I should preface this chapter by saying that the mathematical machinery involved here, especially those involving the convergence of sample paths, will be a bit more advanced and delicate. As such, our main goal would be to provide a road map, and refer the reader to various sources for the exact results.

### 6.1 From Attractiveness to Stability

Let $Q$ be an irreducible, aperiodic discrete-time Markov chain with a countable state space. We will further assume that the jumps of $Q$ from one step to another are uniformly bounded in $L_1$. We will consider a sequence of stochastic systems, indexed by $k$, which are essentially identical, except for that in system $k$, we initialize

$$Q^k(0) = kq^0,$$

for some $q^0 \in \mathbb{Z}^n_+$. Consider the following scaled process

$$\hat{Q}^k(t) := \frac{1}{k}Q(\lfloor kt \rfloor), \quad t \geq 0.$$  

This type of scaling essentially amounts to a “zoomed out” view of the system dynamics: we scale down “space” ($Q(t)$) by an order of $1/k$, while simultaneously shrinking down the length of each time step by the same factor. Imagine you were to simulate the trajectory of $Q(\cdot)$ with a large initial queue length, put it on a wall, and walk backwards from it, $\hat{Q}^k(\cdot)$ is what you will see. As $k$ becomes large, each unit of arrivals and departures becomes infinitesimally small, and the evolution of the queue lengths tends to become more “fluid like,” and hence the nomenclature. Panel (a) in Figure 6.1 shows what this limit might look like.

More precisely, let us assume that for any $q^0$, there exists a unique deterministic trajectory $q : \mathbb{R}_+ \to \mathbb{R}^n_+$, such that for any $T > 0$, almost surely,

$$\sup_{t \in [0, T]} \|\hat{Q}^k(t) - q(t)\| = 0,$$

with $q(0) = q^0$. We will refer to $q$ as the fluid limit, and denote by $Q_\infty$, the the set of all such limiting trajectories. Convergence of this type is also known as the **functional strong law of large numbers**.

We next define the notion of uniform attraction for the fluid limit. Essentially, we would like that the fluid limit approach 0 quickly, wherever it starts.
Figure 6.1: This is a sample path of \( Q(t) \) in a system with two queues and one server, simulated over 30,000 time steps. In panel (a), the warmer colors indicate smaller \( t \). In each time step, the server can choose one of the two queues and serve one job there. \( Q(0) \) is initialized to \( Q_1(0) = 200, Q_1(0) = 600 \). \( \lambda_1 = \lambda_2 = 0.4935 \) and \( \rho = 97\% \).

**Definition 6.1.1.** The set of all fluid limits \( Q_{\infty} \) is said to be uniformly attractive, if there exists constant \( c > 0 \), such that for all \( q \in Q_{\infty} \):

\[
q(t) = 0, \quad \text{for all } t > c\|q(0)\|. \tag{6.4}
\]

In other words, a family of fluid limits is uniformly attractive if all fluid limits “drain” to 0 at a constant speed \( 1/c \).

The following theorem is the main driver of the stability theory via fluid limits.

**Theorem 6.1.1.** If \( Q_{\infty} \) is uniformly attractive, then the Markov process \( Q \) is positive recurrent.

**Proof sketch.** There are variants of the proof of this result, but most follow the same main idea. They key is to realize that, under the fluid convergence, along with the fact that \( Q \) has \( L_1 \) increments, \( \mathbb{E} \left[ \frac{1}{k} \|Q^k(kt)\| \right] \to \|q(t)\| \) as \( k \to \infty \). Therefore, fixing \( q_0 \) and \( t \), if we know that \( q(t) = 0 \), then we should expect \( \mathbb{E} \left[ \frac{1}{k} \|Q^k(kt)\| \right] \to 0 \), or equivalently

\[
\mathbb{E} \left[ \|Q^k(kt)\| \right| Q^k(0) = kq_0 = o(k). \tag{6.5}
\]

for all large \( k \). Let’s parse the limit involved in the above equation a bit: in the original system, we start with a queue that has \( O(k) \) jobs, and \( O(k) \) steps later, the queues have essentially been depleted...
and are now on the order of $o(k)$. This further implies that the $||Q^k(\cdot)||$ has had an average constant negative drift during the first $kt$ time steps.

Contrasting this with the Foster-Lyapunov criteria we saw earlier, we see that the key difference here is that instead of a constant negative drift in every step, we have weakened the requirement to a constant negative drift averaged across $k$ time steps. It is precisely this averaging property that underlies the power and generality of the fluid limit approach, one that would allow us to “smooth out” annoying, but non-consequential, aspects of the original stochastic process, such as those in the example at the beginning of this chapter.

The above sketch is of course not rigorous. References to formal proofs of this result, which essentially hinges upon showing that (6.5) implies positive recurrence, will be given in the Notes of this chapter.

### 6.2 Deploying the Fluid Limit Approach

The development of the previous section leaves two holes in the argument:

1. How to show that the pre-limit trajectory converges to a fluid limit?
2. How to show the set of fluid limits are universally attractive?

Question #1 is typically easier to resolve and follows standard arguments. The intuition is as follows: the evolution of the Markov process $Q(t)$ is derived from a telescopic sum of a sequence of increments $\Delta(t) = Q(t) - Q(t-1)$. The type of fluid limit scaling we have used is essentially that of law of large number type averaging, evaluated at all points of a compact interval. Therefore, one would expect that if the increments $\Delta(t)$ are independent, or almost independent, and with bounded moments, then this type of averaging would give rise to well-behaved fluid limits. Standard procedures for showing these limits indeed follow such intuition. One would typically break down further the updates of $Q(t)$ into separate primitives. In our example, for instance, one would write down

$$Q(t) = (Q(t-1) + A(t) - D(t))^+. \quad (6.6)$$

We can often use the problem structure to show that the scaled processes for the primitives $A$ and $D$ both converge to unique fluid limits, which would further imply the convergence of $Q$ itself. An example of this type of convergence is carried out in Section 8 of [Massoulié and Xu, 2018], and there are many of other examples.

Question # 2, on the other hand, is typically problem-specific, ranging from trivial to pretty difficult. Regardless, the typical approach would be to show that the fluid limit $q$ is the solution of a certain ordinary differential equation, and use the expression of the ODE to prove attraction. To take a simple example, suppose our system is a single queue with arrival rate $\lambda$ and service rate $\mu$. It is not difficult to guess that the fluid limit $q$ should satisfy:

$$\dot{q}(t) = \lambda - \mu, \quad (6.7)$$

with initial condition $q(0) = q_0$. The solution of this ODE is given by

$$q(t) = q_0 + (\lambda - \mu)t, \quad (6.8)$$

which is clearly universally attractive. with $c = (\mu - \lambda)^{-1}$. 

6.3. WHY FLUID LIMIT

6.2.1 Insights for Max-Weight Policy

Since the one-queue example is a bit boring, let us look at what happens if we were to find a fluid limit for the Max-Weight policy. It turns out that the fluid limit is given by

\[ \dot{q}(t) = \lambda - \arg\max_{d \in D} \langle q(t), d \rangle, \tag{6.9} \]

a most natural guess based on the definition of the policy. For now, let us assume that the fluid limit trajectory is unique (which it may not be depending on the structure of \( D \)). How do we show that this trajectory is universally attractive?

In this case, the original Lyapunov argument for the deterministic dynamical system is useful. Consider the following Lyapunov function, which is the \( L_2 \) of the queue lengths:

\[ V(x) = \sqrt{\sum_i x_i^2}. \tag{6.10} \]

Suppose the arrival rate \( \lambda \) is in the interior of the stability region: \( \lambda \in \rho \Pi \) for some \( \rho < 1 \). We have by the chain rule:

\[
\frac{d}{dt} V(q(t)) = \nabla V(q(t)) \cdot \dot{q}(t) \\
= \frac{1}{V(t)} \langle q(t), \dot{q}(t) \rangle \\
= \frac{1}{V(t)} \langle q(t), \lambda - \arg\max_{d \in D} \langle q(t), d \rangle \rangle \\
= -\frac{1}{V(t)} \max_{d \in D} \langle q(t), \lambda - d \rangle \\
\leq -(1 - \rho) d_{\min} \frac{1}{V(t)} \max_i q_i(t) \\
\leq -(1 - \rho) d_{\min} / n.
\]

where the second to last inequality follows from the fact that \( \lambda \in \rho \Pi \), and \( d_{\min} := \min_i \max_{d \in D} d_i \), and the last inequality from the definition of \( V \). Note moreover that \( V(x) \leq n \|x\| \). We thus conclude that it will take

\[ t = \frac{n^2}{(1 - \rho) d_{\min}} \|q_0\| \tag{6.11} \]

for \( q(t) \) to hit zero and stay thereafter. This proves the uniform attraction of the fluid limits, and thus the positive recurrence of the queueing length process under Max-Weight.

6.3 Why Fluid Limit

Compare this to the original proof we had using Foster-Lypunov theorem, the fluid limit analysis is quite a bit cleaner. Moreover, the fluid interpretation also clearly illustrates what the Max-Weight policy is trying to achieve, that is, to maximize the downward drift on the quadratic queue length.

The comparison is not entirely fair, of course, because we have transferred substantial complexity into proving convergence to the fluid limit, as well as showing Theorem 6.1.1 itself. The benefit of the fluid limit approach is much more evident if we are interested in analyzing a host of somewhat similar models, since presumably the heavy lifting has to be done only once, and we can simply repeat the analysis of the fluid limits themselves.
We have not touched upon some other more sophisticated applications of the fluid limit models, such as in a general stochastic network with complex routing topologies, general service times, non-Markovian arrivals, etc (cf. [Dai and Harrison, 2020]). In all of these cases, it is in principle possible do without the fluid limit, but it can often be quite a pain and obscure the simple intuition that underlies the stochastic processes.

In another example, such as the one in [Massoulié and Xu, 2018], the fluid model can be very handy when we want to analyze a system consisting of interconnected sub-components. If the system as a whole is a Markov process, but the individual sub-components are not, then again, carrying out a direct Foster-Lyapunov analysis can be difficult. Under the fluid limit framework, one simply has to establish the fluid limit for each individual component, which will then give rise to the fluid limit of the overall system. In other words, stitching together the (deterministic) fluid limits from sub-systems are often trivial, whereas stitching the pre-limit stochastic processes across these sub-systems are often impossible.

A final, and important, reason for using a fluid limit, is that showing Markov process converges to a fluid limit is often the first step towards establishing other, more refine, stochastic approximation results, such as the diffusion approximation and state-space collapse. For instance, in the next chapter we will encounter the use of state-space collapse to study the delay optimality of the Max-Weight policy, which relies on first showing the convergence of the system dynamics to a fluid limit.

### 6.4 Other Types of Fluid Limits

The idea of fluid limit shows up in many areas, and can bear names such as hydraulic limits, or mean-field limits. The type of scaling we see here is but one special case. For instance, we are not covering here the type of “spatial fluid limits” that have been deployed to study large-system limit of stochastic systems. Hopefully we will circle back to this in a future chapter, if time allows.

### 6.5 Notes

[Rybko and Stolyar, 1992] is one of the first papers to make the connection between the fluid limit attraction and the stability of the original Markov process, where a first-come-first-serve scheduling policy was analyzed. The analysis was subsequently generalized to other scheduling disciplines and network models in [Dai, 1995, Stolyar, 1995], where [Stolyar, 1995] also provides a sufficient condition that is a bit weaker than the uniform attraction used here. The exposition in this section generally follow the steps in these two sources. For a proof of Theorem 6.1.1, one can find excellent references in the original papers above as well as a more recent text: Section 6 [Dai and Harrison, 2020], which is also a great reference for the use of fluid models in stochastic networks. In the context of reinforcement learning, [Xu and Yun, 2020] offers an example where the fluid limit is used to study a system where the dynamics is modulated by an underlying Markov process, similar to the example given at the beginning of the chapter.
Chapter 7
State Space Collapse and Optimality of Drift Methods

As a quick recap, we have seen thus far that a drift-based scheduling policy, Max-Weight, is able to achieve maximum stability in a stochastic network. Furthermore, if we perform a more careful analysis, we may even obtain bounds on the steady-state moments of the queue lengths (Section 5.2). However, none of this seems to suggest that the Max-Weight policy is optimal. Do there exist other, possibly more sophisticated algorithms, that perform much better than Max-Weight?

We will see in this chapter a remarkable result due to [Stolyar et al., 2004], showing that the seemingly naive and greedy Max-Weight policy is, in some appropriate sense, optimal! More precisely, the result suggests that the Max-Weight policy minimize the weighted total queue length \( \sum Q_i(t) \), in a system that is critically loaded, i.e., where all service resources are nearly saturated. This regime of resource saturation is also known as the heavy-traffic limit. The main tool that enables this result is the state-space collapse (SSC). We will also revisit the the fluid limit that we encountered in the previous chapter, now in a different context and application.

7.1 Baby-Step SSC: An Example with Two Queues

To establish the heavy-traffic optimality of Max-Weight requires some pretty heavy machinery. Luckily, we can get a pretty accurate (though limited) glimpse of what is to come in a very simple example of scheduling with two parallel queues, illustrated in Figure 7.1. Here, we have one server capable of processing one job per time step, and two queues with arrival rates \( \lambda_1 \) and \( \lambda_2 \), respectively. For now, let us assume that

\[
\lambda_1 = \lambda_2 = \lambda/2,
\]

for some \( \lambda \in (0, 1) \). Since the total arrival rate is equal to \( \lambda \), the heavy-traffic regime in our example amounts to letting

\[
\lambda \to 1.
\]

We will take a close look at what happens with \( Q(t) \) in this system under the Max-Weight policy. More specifically, we will try to understand:

1. What is state space collapse in this simple example? Why does it happen under Max-Weight?
2. Why is state space collapse connected to optimal queue length in the heavy-traffic?
7.1.1 What does SSC look like here?

Let us take a look at the sample path in Panels (a) and (b) of Figure 7.2. Notice that after very short period at the beginning, $Q(t)$ stays very closely to the one-dimensional invariant manifold:

$$\{ (q_1, q_2) \geq 0 : q_1 = q_2 \}$$

In other worlds, the queue length process $Q$ with a native dimension of two “collapses” into an effective dimension of one.

7.1.2 Why does Max-Weight induce SSC?

In this example, Max-Weight amounts to the server simply picking at every time slot the longer of the two queues. In Figure 7.2, we started with $Q_2(0) > Q_1(0)$, and when such discrepancy exists between the two queues, services are exclusively dedicated to the longer queue until parity is reached: assuming the system is stable, then during this period, the queue length of the longer queue would gradually reduce, while the shorter queue would increase. It is easy to see that $Q_1(t) = Q_2(t)$ is a stable equilibrium: if the two queue lengths were to come apart, the above-mentioned dynamics will push them back to parity. In summary, we have heuristically argued that (1) Max-Weight policy will drive initially unequal queue lengths to parity and (2) once there, the policy will keep the two queues from diverging. Remarkably, SSC also happens in much more general settings beyond this simple two-queue exampl

Notice that the SSC is a direct consequence of the Max-Weight policy, not a universal feature for stable scheduling policies in general. For instance, the randomized allocation policy that chooses with probability 50% either queue to service in each iteration is also stable. However, in this case the two queues clearly evolves independently from one another, and the queue lengths processes “span” the native two-dimensional space, and no state-space collapse occurs. Panels (c) and (d) in Figure 7.2 illustrates this example.

7.1.3 What does SSC have to do with optimal heavy-traffic queue lengths?

At a high level, SSC is desirable because when $Q(t)$ stays on the invariant manifold, it’s quite hard for the system to “waste”resource unnecessarily. In the two-queue example, we see that while on the invariant manifold, all unprocessed jobs in the system are equally divided between the two queues, and when this occurs, whatever schedule we end up picking, the service will never be wasted unless
7.1. **BABY-STEP SSC: AN EXAMPLE WITH TWO QUEUES**

(a) Max-Weight. Warmer color = earlier.  
(b) Max-Weight  
(c) Random  
(d) Random

Figure 7.2: This is a sample path of $Q(t)$ in a system with two queues and one server, reproduced from Figure 6.1. The queues receive Poisson arrivals with rate $\lambda_1 = \lambda_2 = 0.49$, and operate at $\rho = 97\%$. However, instead of running it for 30,000 steps, we will run it up to 100,000 steps, so that the steady-state behavior of the system is more evident. It is visually clear that after some time, the two-dimensional queue-length process $Q(t)$ “collapses” to the one-dimensional manifold that is the diagonal ray $\{ q : q_1 = q_2 \}$. Moreover, the system spends vast majority of time in the collapsed state. The fact that the queue lengths “stay” on the manifold is due to the use of a Max-Weight policy ((a) and (b)). In contrast, there is no state-space collapse if we instead use a random scheduling policy, where the server chooses either queue with probability $1/2$, independent of their lengths.

The system is entirely empty ($Q(t) = 0$). That is, as long as there are jobs to process, the Max-Weight policy does not waste resources. In contrast, a less desirable state would be one where most of the jobs are in while queue, while the other queue is nearly empty. If services are not carefully chosen in this setting, it could lead to idling of precious processing resources. When the system approaches the heavy-traffic of $\rho \to 1$, a good algorithm would have to harness every bit of available resource, and any such unnecessary idling spells trouble.

Of course, one might argue that, in this two-queue example, the fact that Max-Weight is work-
conserving is obvious from the definition, and SSC seems to be a rather round-about way to arrive at this fact. However, as the system gets more complicated, it becomes far from obvious how to find a “work-conserving” policy, or even whether this notion of work-conservation is well defined. In these more general systems, SSC offers a much more powerful language to articulate the behavior of good scheduling policies.

7.2 Two ways lead to SSC

In this chapter and the next, we will look at two paths towards establishing and leveraging the SSC for performance analysis. Very roughly, the present chapter follows the program developed in [Stolyar et al., 2004], showing that an appropriately scaled queue length process under Max-Weight exhibits SSC over a finite time interval. In the next chapter, we will use the analysis of [Eryilmaz and Srikant, 2012] to directly analyze the steady-state distribution of the queue lengths and establish SSC of these distributions using a drift-based stochastic approximation. The two approaches are complimentary and focus on different aspects of the dynamics.

7.3 Preliminaries

The following notation will be useful. We will largely follow the convention set out in the original paper of [Stolyar et al., 2004], with some modifications to maintain consistency with the rest of the notes.

The model [Stolyar et al., 2004] addresses a more general version of Max-Weight, where the scheduling policy chooses the schedule \( d \in D \) in order to maximize the weight:

\[
\sum_{i=1}^{n} d_i \cdot \gamma_i Q_i^\beta(t).
\] (7.3)

To simplify the exposition, in this chapter we will only focus on the original Max-Weight policy, by setting \( \gamma_i = 1 \) and \( \beta = 1 \).

**Geometric aspects of the maximum stability set**

Let \( \Pi \) be the maximum stability set defined in Chapter 4, (4.4):

\[
\Pi = \{ \lambda : s \in \text{Conv}(D), s \geq \lambda \},
\] (7.4)

That is, all arrival rates that fall within the interior of \( \Pi \) admits a stabilizing state-independent randomized scheduling policy. Define \( \Pi^* \) to be the “north-east” boundary of \( \Pi \).

\[
\Pi^* = \{ x \in \Pi : \text{there exists } s \in \Pi, s > x \}.
\] (7.5)

Since the set \( \Pi \) is a bounded polyhedron with finitely many extreme points, we know that \( \Pi^* \) consists of a finite number of faces with dimension \( n - 1 \). For a point \( x \in \Pi^* \) that lies in the relative interior of some face, we will denote by \( L(x) \) the unique face containing \( x \).

Fix a vector \( x \in \Pi^* \), and denote by \( \xi \) an outer-normal vector to \( \Pi^* \) at \( x \), with

\[
\|\xi\|_2 = 1.
\] (7.6)

The following is an important assumption that will be crucial to our analysis.
7.3. PRELIMINARIES

Definition 7.3.1 (Resource Pooling Condition). We say that the vector $x$ satisfies the resource pooling (RP) condition, if

1. $x$ belongs to $\Pi^*$.
2. The outer normal unit vector $\xi$ at $x$ is uniquely defined.

Or, equivalently, if $x$ belongs to the relative interior of one of the faces of $\Pi^*$.

If, in addition, all components of $\xi$ are strictly positive, then we say that $x$ satisfies the complete resource pooling (CRP) condition.

The heavy-traffic regime

Fix $\lambda \in \Pi^*$ such that $\lambda$ satisfies the RP condition. We will consider a sequence of systems indexed by $k$. In the $k$th system, the arrival rate vector is denoted by $\lambda^k$ and belongs to the interior of $\Pi$. All quantities in the $k$th system will be indicated by the superscript $k$.

We will assume that for some $a \in \mathbb{R}$:

$$\lim_{k \to \infty} k \langle \xi, \lambda^k - \lambda \rangle = a. \quad \text{(7.7)}$$

That is, projected onto the direction of $\xi$, the sequence of arrival rate vectors converge to the limit $\lambda$ at rate $O(1/k)$.

Finally, as in Chapter 4, we will assume that the arrivals have bounded support, where

$$\text{Var}(A^k(0)) \to \sigma_i^2, \quad k \to \infty. \quad \text{(7.8)}$$

The assumption of bounded support can be easily relaxed and replaced by a condition on the tail distribution [Stolyar et al., 2004].

Scalings

We define the workload process, $X$:

$$X^r(t) := \langle \xi, Q^k(t) \rangle \quad \text{(7.9)}$$

Fix $k \in \mathbb{N}$, define the fluid scaled queue length process:

$$\hat{Q}^k(t) := \frac{1}{k}Q(kt), \quad \text{(7.10)}$$

and the diffusion scaled process:

$$\tilde{Q}^k(t) := \frac{1}{k}Q(k^2t). \quad \text{(7.11)}$$

The corresponding scalings of $X$: $\hat{X}^k$ and $\tilde{X}^k$ are defined analogously.

The invariant manifold $\mathcal{M}$ is defined to be the points in $\mathbb{R}_+^n$ that are proportional to $\xi$:

$$\mathcal{M} = \{x \in \mathbb{R}_+^n : x = c\xi, c \in \mathbb{R}_+\}. \quad \text{(7.12)}$$

In order for there to be a non-trivial limit, we will assume that the initial queue lengths scale accordingly. We will assume that $\hat{Q}^k(0)$ converges to a point on the invariant manifold.

$$\hat{Q}^k(0) \to \hat{Q}^\circ(0) = \hat{X}^\circ(0)\xi, \quad \text{(7.13)}$$
for some constant $\tilde{X}^\alpha(0) \geq 0$. The above convergence also implies the convergence of $\tilde{X}^k(0)$:

$$\tilde{X}^k(0) \to \tilde{X}^\alpha(0). \quad (7.14)$$

Define the Brownian motion:

$$\tilde{W} = \tilde{X}^\alpha(0) + at + \sigma B(t), \quad t \geq 0, \quad (7.15)$$

where $a$ is defined in (7.7), and

$$\sigma = \sqrt{\sum_i \xi_i^2 \sigma_i^2}. \quad (7.16)$$

where $\sigma_i^2$ is the limiting arrival variance. $B$ is the standard Brownian motion.

Finally, let $\tilde{X}^\alpha(\cdot)$ be the reflected Brownian motion (RBM) associated with $\tilde{W}$, so that

$$\tilde{X}^\alpha(t) = \tilde{W}^\alpha(t) + \tilde{Y}^\alpha(t), \quad (7.17)$$

where

$$\tilde{Y}^\alpha(t) := -\left(0 \wedge \inf_{0 \leq u \leq t} \tilde{W}^\alpha(u)\right). \quad (7.18)$$

One can think of $\tilde{Y}^\alpha$ here as keeping track of the cumulative amount of “idling” services.

### 7.4 Statement of Main Result

We now arrive at the main theorem.

**Theorem 7.4.1** (Heavy-traffic optimality of Max-Weight). Under the Max-Weight scheduling policy, we have that

1. $\tilde{X}^k \xrightarrow{(w)} \tilde{X}^\alpha$, $k \to \infty$, where $\xrightarrow{(w)}$ represents the weak convergence of random processes over a compact interval.

2. The following SSC holds:

$$\tilde{Q}^k \xrightarrow{(w)} \tilde{Q}^\alpha := \tilde{X}^\alpha \cdot \xi. \quad (7.20)$$

That is, $\tilde{Q}^k$ converges to a one-dimensional process evolving on the invariant manifold $M$.

3. Max-Weight policy minimizes workload in the following sense: Let $\tilde{X}_G^k$ be the workload process associated with an arbitrary scheduling policy $G$, then for all $t, u \geq 0$,

$$\liminf_{k \to \infty} P\left(\tilde{X}_G^k(t) > u\right) \geq P\left(\tilde{X}^\alpha(t) > u\right). \quad (7.21)$$

**Remark 7.4.2.** The initial condition is required to be on the invariant manifold. This condition can be relaxed to having $\tilde{Q}^k(0)$ converge to a fixed point not on the invariant manifold (Section 9.3, [Stolyar et al., 2004]). The weak convergence of the scaled processes can also be converted into a sample-path version, as almost-sure uniform convergence over compact sets (u.o.c), using the Skorohod representation theorem. In particular, there exists a coupling construction of the various system primitives, such that almost surely:

$$\liminf_{k \to \infty} \tilde{X}_G^k(t) \geq \tilde{X}^\alpha(t), \quad \forall t. \quad (7.22)$$
7.5  Main Steps of the Proof

The proof of Theorem 7.4.1 [Stolyar et al., 2004] is quite non-trivial and involved. Presenting the entire proof rigorously will unfortunately require too many tools that are beyond the scope of these notes. Instead, we will aim to explain the main steps of the proof and isolate technical challenges, so that interested readers can seek out the material in the references for their specific applications.

7.5.1 Step 1: The one-dimensional Skorohod Problem

It might sound counter-intuitive, but it’s perhaps easiest to go backwards. That is, instead of directly analyzing the dynamics induced by Max-Weight, we take a step back and look at the workload process induced by an arbitrary scheduling policy \( X_G \). We may ask: what features of \( X_G \) can tell us whether it is optimal?

The answer to this question is surprisingly elegant. At the high-level, we would expect that under a “good” policy, the system resources shouldn’t be sitting idle while \( X_G(t) \) is large. Indeed, this intuition can be formalized to characterize the \textit{optimal} workload process. This characterization is known as the one-dimensional Skorohod Problem, as follows. Let \( D([0, \infty, \mathbb{R}) \) be the family of right-continuous functions with left-limits (RCLL) from \([0, \infty)\) to \( \mathbb{R} \).

**Proposition 7.5.1** (Skorohod Problem). Let \( w \) be a continuous function in \( D([0, \infty, \mathbb{R}) \). We say that the pair of functions \((x, y)\) solve the Skorohod problem if

(a) \( x(t) = w(t) + y(t) \geq 0 \) for \( t \geq 0 \),

(b) \( y \) is nondecreasing and nonnegative,

(c) \( y(0) = 0 \)

(d) for any \( t \geq 0 \), if \( x(t) > 0 \), then \( t \) is not a point of increase of \( y \).

Then, the unique solution \((x^o, y^o)\) of the Skorohod problem is given by

\[
y^o(t) = -\left(0 \land \inf_{0 \leq u \leq t} w(u)\right), \quad x^o(t) = w(t) + y^o(t), \quad t \geq 0.
\] (7.23)

Moreover, for any functions \((x, y)\) in \( D([0, \infty, \mathbb{R}) \) satisfying conditions (a) and (b) above, we have

\[
y(t) \geq y^o(t), \quad x(t) \geq x^o(t), \quad t \geq 0.
\] (7.24)

If we view \( w \) as the cumulative potential departure process of a stochastic network, \( x \) the queue length process, and \( y \) the idling process, then Condition (d) in the Skorohod problem amounts to saying that no idling should occur when there are still jobs in the system to be processed. This condition is crucial to the notion of optimality as in (7.24).

Note that Theorem 7.4.1 claims that the workload process in the diffusion limit under Max-Weight solves the Skorohod problem with respect to \( \tilde{W}^o \). By Proposition 7.5.1, this would in term imply that the limiting workload is optimal.
7.5.2 Step 2: Workload decomposition and the idling process

The next step, then is to show that the workload process does converge to  \( \hat{X}^o \) which solves the Skorohod problem. To this end, let us make the role of \( X \) and \( Y \) more explicit in the pre-limit stochastic system.

In what follows, we will project all relevant processes onto the invariant manifold \( M \) (along \( \xi \)), the rationale being that, under SSC, essentially all the “actions” will take place on this manifold. Denote by \( F^k(t) \) the cumulative arrival, projected onto \( \xi \):

\[
F^k(t) = \sum_{s=0}^{t} \langle \xi, A^k(s) \rangle.
\]

(7.25)

Note that when \( t \) is large, by the (functional) law of large number,

\[
F^k(t) \approx \langle \xi, \lambda^k \rangle t.
\]

(7.26)

Next, define \( \bar{\mu} \) to be the maximum service rate the system is capable of inducing among the direction of \( \xi \):

\[
\bar{\mu} = \max_{d \in D} \langle \xi, d \rangle = \max_{d \in \Pi} \langle \xi, d \rangle = \langle \xi, \lambda \rangle.
\]

(7.27)

The last equality follows from the fact that \( \xi \) is normal to the face containing \( \lambda \), and therefore, any point \( v \) on that face would maximize \( \langle \xi, v \rangle \), including \( \lambda \). Intuitively, \( \bar{\mu} \) represents the maximum speed at which the system is capable of draining workload along \( \xi \).

We will define \( W^k(t) \) as the cumulative discrepancies:

\[
W^k(t) = X^k(0) + F^k(t) - \bar{\mu}t.
\]

(7.28)

That is, \( W^k(t) \) represents the workload at time \( t \) if we consistently use a schedule that induces a service of \( \bar{\mu} \) along \( \xi \) in each time step, and if the queues are allowed to go negative after hitting zero. Notice that the definition of \( W^k \) only depends on the problem primitives, and is independent of the scheduling policy.

The reality is, of course, that the queues do not go negative after hitting zero, and when such event occur, the potential departures simply get “wasted”. To capture this waste of resources, we will denote by \( Y(t) \) the cumulative amount of workload resulted from potential departures that did not materialize into actual departures, due to the queues being empty. Denote by \( S^k_i(t) \) the number of actual departures at queue \( i \) at time \( t \), then

\[
Y(t) = \sum_{s=0}^{t} \langle \xi, D^k(s) - S^k(s) \rangle.
\]

(7.29)

Unlike \( W^k \), the definition of \( Y^k \) does depend on the scheduling policy. Finally, we can write the actual workload process \( X \) as:

\[
X^k(t) = W^k(t) - Y^k(t).
\]

(7.30)

Now, let us check if we can tell whether the pre-limit idling process \( Y \) solves the Skorohod problem under Max-Weight. If it did, it would automatically imply that \( X \) is optimal, and we wouldn’t even have to do anything beyond this point! The first three conditions (a)-(c) are easily satisfied by essentially any scheduling policy. The problem comes in condition (d). Note that if the system happens to be in some “bad state”, where some queues are empty while others are very long, then \( Y \) can still increase due to inadvertently serving some empty queues, while \( X \) is arbitrarily large.
7.5. MAIN STEPS OF THE PROOF

7.5.3 Step 3: SSC implies convergence of idling process to Skorohod solution

The crux of the analysis is therefore going to be focused on showing that under Max-Weight and the heavy-traffic limit, condition \((d)\) will be satisfied in the asymptotic sense, and the limiting process \(\tilde{Y}\) will indeed solve the Skorohod problem.

This is where the state-space collapse property comes in. Specifically, we would like to argue that when \(\tilde{X}_k(t) > 0\) (for large \(k\)), then \(\tilde{Y}_k(t)\) shouldn’t be increasing in the small time interval \((t, t + \delta)\). Suppose for now that the queue length process \(Q_k(t)\) lies exactly on the invariant manifold \(M\) during \([t, t + \delta]\), then we know that if there is positive workload, then the actual departures projected onto \(\xi\) must be equal to \(\bar{\mu}\delta\). That is, no services are wasted during this time, and \(Y\) should stay non-increasing.

The bulk of the analysis is therefore to make this notion precise in a stochastic system, where \(Q_k(t)\) does not lie exactly on \(M\), but “close to” \(M\). This is accomplished by showing that the fluid-scaled process \(\hat{Q}_k(t)\) collapses to \(M\) as \(k \to \infty\). Detailed analysis on how to translate this closeness into condition \((d)\) are carried out in Lemmas 7 and 10 of [Stolyar et al., 2004].

7.5.4 Step 4: Show Max-Weight leads to SSC using fluid scaling

Finally, the last step is to prove SSC of the fluid-scaled process. This collapse is pretty obvious in our two-queue example above, with \(\xi = (1/\sqrt{2}, 1/\sqrt{2})\), but to establish it generally will require a more sophisticated machinery. We will invoke the following Lyapunov functions.

Let

\[
V(x) = \|x\|_2^2 = \sum_i x_i^2, \tag{7.31}
\]

and

\[
G(x) = \frac{V(x)}{V(\langle x, \xi \rangle x)} - 1, \quad x \in \mathbb{R}_+^n. \tag{7.32}
\]

In particular, let \(\alpha(x)\) be the angle between \(x\) and \(\xi\), then

\[
G(x) = \frac{1}{\cos(\alpha(x))^2} - 1. \tag{7.33}
\]

In this way, \(G(x)\) measures how close \(x\) is to the invariant manifold, with \(G(x) = 0\) if any only if \(x \in M\). To add some geometric intuition, let us further decompose any \(x \in \mathbb{R}_+^n\) into two components. Define the linear operators \(\|\) and \(\perp: \mathbb{R}^n \mapsto \mathbb{R}^n:\)

\[
x = x\| + x\perp. \tag{7.34}
\]

We can thus write:

\[
x = x\| + x\perp. \tag{7.35}
\]

By Pythagoras theorem, we have

\[
V(x) = V(x\|) + V(x\perp), \tag{7.36}
\]

and

\[
G(x) = \frac{V(x\|) + V(x\perp)}{V(x\|)} - 1 = \frac{V(x\perp)}{V(x\|)}. \tag{7.37}
\]

Let us look at, \(\hat{Q}_k\), the fluid-scaled limit of \(Q\). Suppose for now that its limit \(q\) as \(k \to \infty\) is well defined as unique. Then following the analysis of the last chapter (see (6.11)) we have

\[
\dot{q}(t) = \lambda - \arg\max_{d \in D} \langle q(t), d \rangle = \lambda - \arg\max_{d \in \Pi} \langle q(t), d \rangle. \tag{7.38}
\]
CHAPTER 7. STATE SPACE COLLAPSE AND OPTIMALITY OF DRIFT METHODS

**Proposition 7.5.2.** Fix a \( \lambda \) that satisfies the RP condition.

1. There exists \( \epsilon > 0 \), which only depends on \( \lambda \), such that \( \frac{d}{dt} V(t) \leq -\epsilon \|q^\perp(t)\|_2 \).

2. \( \frac{d}{dt} V(q^\parallel(t)) \geq 0 \).

As a direct consequence of this proposition, we arrive at a main result, establishing the SSC. We state here a weaker version of the uniform convergence of \( G(q(t)) \) than is necessary: we will assume that all coordinates of \( \xi \) are positive (that is, the CRP condition is satisfied); the original proof does not require this assumption (assuming only RP), and is a bit more involved (Theorem 2, [Stolyar et al., 2004]). The key to this result lies in showing that the perpendicular component \( q^\perp(t) \) decays to 0 sufficiently fast. Indeed, this is exactly what we would expect from SSC.

**Theorem 7.5.3.** Suppose all coordinates of \( \xi \) are strictly positive. Then \( G \) exhibits uniform attraction. That is, \( G(q(t)) \to 0 \) uniformly fast as \( t \to \infty \) over all initial conditions with \( \|q(0)\|_2 = 1 \).

**Proof** Because \( V(q^\parallel(t)) \) does not decrease by Proposition 7.5.2, we have that

\[
G(q(t)) = \frac{V(q^\perp(t))}{V(q^\parallel(t))} \leq \frac{V(q^\perp(t))}{V(q^\parallel(0))}, \quad t \geq 0. \tag{7.39}
\]

Proposition 7.5.2 also implies that \( V(q^\perp(t)) \) is nonincreasing, while \( V(q^\parallel(t)) \) nondecreasing. This implies that \( V(q^\perp(t)) = V(q(t)) - V(q^\parallel(t)) \) must be nonincreasing. With a initial condition \( \|q(0)\|_2 = 1 \), we have

\[
V(q(t)) = V(q(0)) + \int_s^t \frac{d}{dt} V(q(t))|_{t=s}ds \leq 1 - \epsilon t \sqrt{V(q^\perp(t))}, \tag{7.40}
\]

where in the last step we used the fact that \( V(q^\perp(t)) \) is nonincreasing. Because the Lyapunov function \( V \) is non-negative, this shows that

\[
V(q^\perp(t)) \leq \left( \frac{1}{\epsilon t} \right)^2. \tag{7.41}
\]

Substituting the above into (7.39), we obtain

\[
G(q(t)) \leq \frac{1}{V(q^\parallel(0))} \left( \frac{1}{\epsilon t} \right)^2 \tag{7.42}
\]

Finally, note that because we have assumed that \( \xi \) is coordinate-wise positive, \( V(q^\parallel(0)) \) is bounded from below over all \( q(0) \) with unit norm. We have thus established the uniform convergence of \( G(q(t)) \).

**Proof of Proposition 7.5.2.** For the first claim, recall that \( L(\lambda) \) is the face of \( \Pi \) containing \( \lambda \), and let \( U(\epsilon) \) be defined by:

\[
U(\epsilon) = \{ \lambda + \gamma : \|\gamma\| = \epsilon, (\xi, \epsilon) = 0 \}. \tag{7.43}
\]

That is, \( U(\epsilon) \) is a small ball around \( \lambda \) that lies in the same plane as \( L(\lambda) \). Since \( \lambda \) belongs to the relative interior of \( L(\lambda) \), we know that for some sufficiently small but positive \( \epsilon \), we have

\[
U(\epsilon) \subset L(\lambda). \tag{7.44}
\]

From (7.38) we have
\[
\frac{d}{dt} V(q(t)) = \langle q(t), \dot{q}(t) \rangle
\]
\[
= \min_{d \in \Pi} \langle q(t), \lambda - d \rangle
\]
\[
\leq \min_{d \in U(\epsilon)} \langle q(t), \lambda - d \rangle
\]
\[
\leq \min_{d \in U(\epsilon)} \langle q^\perp(t), \lambda - d \rangle
\]
\[
= -\epsilon \|q^\perp(t)\|_2^2,
\]
(7.45)

where the last equality follows from the fact that \(q^\perp\) belongs to the plane of \(L(\lambda)\) and that the set \(\lambda - U(\epsilon)\) contains vector of \(\epsilon\) length in all directions within \(L(\lambda)\). This proves the first claim.

For the second claim, we have
\[
\frac{d}{dt} V(q^\parallel(t)) = \left\langle \nabla V(q^\parallel(t)), q^\parallel(t) \right\rangle
\]
\[
= 2 \left\langle q^\parallel(t), (\lambda - \operatorname{argmax}_{d \in \Pi} \langle q(t), d \rangle)^\parallel \right\rangle
\]
\[
= 2 \left\| q^\parallel(t) \right\|_2 \left\langle \xi, \lambda - (\operatorname{argmax}_{d \in \Pi} \langle q(t), d \rangle) \right\rangle
\]
\[
\overset{(a)}{=} 2 \left\| q^\parallel(t) \right\|_2 \left\| \xi \right\|_2 \min_{d \in L(\lambda)} \langle \xi, \lambda - d \rangle
\]
\[
\overset{(b)}{\geq} 2 \left\| q^\parallel(t) \right\|_2 \left\| \xi \right\|_2 \min_{d \in L(\lambda)} \langle \xi, \lambda - d \rangle
\]
\[
= 0
\]
(7.46)

where step (a) follows from the fact that \(\xi\) is of unique length, (b) from the fact that \(\xi\) is normal to \(L(\lambda)\), and thus the most efficient way to induce a negative along \(\xi\) is to choose \(d\) on \(L(\lambda)\). The final equality follows again from the normality of \(\xi\) with respect to \(L(\lambda)\), which implies that \(\langle \psi, \lambda - x \rangle = 0\) for all \(x \in L(\lambda)\).

\begin{itemize}
\item 7.6 Notes
\end{itemize}

The heavy-traffic workload optimality of the Max-Weight algorithm was established in [Stolyar et al., 2004]; the analysis of this chapter is based on that paper. The use of state-space collapse in analyzing heavy-traffic limits in queueing systems had appeared in earlier work [Bramson, 1998, Williams, 1998]. While [Stolyar et al., 2004] considers a single-hop stochastic network, the results have since been extended to multi-hop stochastic networks by [Dai and Lin, 2008].
Chapter 8

Double Drift: Stationary Lyapunov Analysis with SSC

We have seen in Chapter 7 how to use the diffusion limit to establish that the Max-Weight scheduling policy induces an optimal workload process in finite time, in the heavy-traffic regime. In this chapter, we will explore a related, but different, analysis proposed by [Eryilmaz and Srikant, 2012], which will establish a form of heavy-traffic workload optimality of the Max-Weight policy in steady state.

There are several advantages to the results introduced in this chapter. First, as the name suggests, it allows us to ascertain the algorithm’s performance in steady state. While a result can be naturally conjectured from the transient analysis we conducted in Chapter 7, establishing it rigorously would require additional work even with the transient diffusion limit in hand. Second, as it turns out that we will be able to partially bypass the analysis of the transient sample-paths, and investigate properties of the steady-state distributions directly. This simplifies the analysis, and also reveals different insights. Finally, note that the characterization of the workload process we obtained in Chapter 7 only holds in the heavy-traffic limit. In contrast, the bounds we obtain in this chapter will also be valid (though not tight) for the sub-critical regime where the traffic intensity is bounded away from one.

To be fair, there are disadvantages of the steady-state analysis as well. For one, because we will bypass the sample-path characterizations, the results will say little of how the queues actually evolve over a finite time interval. Also, since we will not be evoking formal functional limits, we do not know exactly what the steady-state distribution is, but will be able to give bounds on its moments. In constraint, the diffusion limits taken in Chapter 7 pins down the distributions of the workload process more explicitly.

Differences aside, the main driving force behind both the transient and the steady-state analyses of the heavy-traffic optimality remain the state-space collapse (SSC) properties of the system dynamics. As such, we will see concepts and ideas from Chapter 7, such as the invariant manifold and specific Lyapunov functions, resurface here.

8.1 A Parallel Load Balancing Model

The steady-steady SSC analysis of Max-Weight-type scheduling policies was first proposed in [Eryilmaz and Srikant, 2012]. The original paper considers both the Max-Weight policy in single-hop network, the model we studied back in Chapter 7, as well as a simpler routing model, which can be thought of a mirror version of the scheduling problem, where instead of choosing which jobs to serve, the decision maker chooses which queues newly arriving jobs will join. The analysis follows the same
steps in both models, but we will focus on the routing model, since it is simpler to analyze and helps to elucidate the main insights.

The parallel routing model is depicted in Figure 8.1, which is essentially a mirrored version of the parallel queue scheduling problem we have encountered in Figure 7.1. In the stochastic network models we have seen so far, the decision maker always picks the schedule, while the arrivals to each queue are exogenously generated. In the routing model, we will simply flip the roles of the departures and arrivals:

1. The system runs in discrete time and consists of $n$ queues.

2. **Arrivals and dispatching**: In time step $t$, a total of $A_\Sigma(t)$ jobs arrive to the system. A dispatcher is to decide how many jobs will join each of the $n$ queues. Denote by $A_i(t)$ the number of jobs to join queue $i$. Then the dispatcher’s task is to generate an arrival vector $A(t) = (A_1(t), \ldots, A_n(t))$ such that

   $$\sum_i A_i(t) = A_\Sigma(t).$$

   We will assume that the total arrivals $A_\Sigma(t)$ are i.i.d., with

   $$E[A_\Sigma(0)] = \lambda_\Sigma, \quad \text{Var}(A_\Sigma(0)) = \sigma_\Sigma^2.$$

3. **Departures**: We will assume that each of the $n$ queues are equipped with a server. The servers do not coordinate with each other, and simply process jobs in their respective queue at maximum speed. As a result, the departures in this system are modeled in the same way as in our original single-hop stochastic network model. However, instead of having the decision maker choosing a schedule from among a schedule set, we will assume that the potential departure vector $D(t)$ is generated exogenously from some fixed distribution. We will assume $D(t)$ are i.i.d., with

   $$E[D(0)] = \mu, \quad \mu_\Sigma := \sum_i \mu_i,$$

   $$\text{Var}(D(0)) = \nu^2, \quad \nu^2_\Sigma := \sum_i \nu_i^2.$$

   Note that here both $\mu$ and $\nu$ are vectors in $\mathbb{R}^n_+$. As usual, we will assume that all arrivals and potential departures are non-negative integers and have bounded support. In particular, we will assume that for some $d_{max} < \infty$,

   $$d_{max} \geq \max_i D_i(t), \quad (8.1)$$

   with probability one.

   We will look at a routing algorithm that is essentially the counter-part of Max-Weight in this model:

   **Definition 8.1.1 (Join Shortest Queue).** Under the Joint-Shortest-Queue policy (JSQ), all arrivals are routed to the queue with the least number of jobs. I.e.,

   $$A(t) = \arg\min_{A: \|A\| = A_\Sigma(t)} \langle A, Q(t) \rangle,$$

   with ties broken uniformly at random.
8.2 Main Result: Performance of the JSQ Policy

Fix the service rate vector $\mu$. Consider a sequence of arrival processes $A_\epsilon(\cdot)$ parameterized by $\epsilon > 0$, with $A_\epsilon$ has variance $(\sigma_\epsilon^2)^2$ and $\lambda_\epsilon = \mu - \epsilon$.

\[ \lambda_\epsilon = \mu - \epsilon. \tag{8.3} \]

**Theorem 8.2.1.** Fix $\mu > 0$ and $\epsilon < \mu$. Denote by $Q^f(\infty)$ the steady-state queue length distribution, if it exists and is unique. Define

\[ c^\epsilon = (\sigma_\epsilon^2)^2 + \nu_\epsilon^2 + \epsilon^2. \]

The following is true.

1. **Upper bound.** Under the JSQ policy, we have

\[ \mathbb{E}\left[ \sum_i Q^f_i(\infty) \right] \leq \frac{c^\epsilon}{2\epsilon} - b^\epsilon, \tag{8.4} \]

where $b^\epsilon = o(1/\epsilon)$, i.e.,

\[ \lim_{\epsilon \downarrow 0} \epsilon b^\epsilon = 0. \tag{8.5} \]

2. **Lower bound.** Fix any stationary routing policy that induces a well-defined steady-state queue length distribution. Then,

\[ \mathbb{E}\left[ \sum_i Q^f_i(\infty) \right] \geq \frac{c^\epsilon}{2\epsilon} - \frac{d_{\text{max}}}{2} n. \tag{8.6} \]

The following corollary is immediate.

**Corollary 8.2.2** (Heavy-traffic optimality of JSQ). Fix $\mu > 0$. Under the heavy-traffic limit of $\epsilon \rightarrow 0$, the total queue length in steady-state under an optimal routing policy satisfies\(^1\)

\[ \mathbb{E}\left[ \sum_i Q^f_i(\infty) \right] \sim \frac{(\sigma_\epsilon^2)^2 + \nu_\epsilon^2}{2\epsilon}, \tag{8.7} \]

as $\epsilon \rightarrow 0$. Furthermore, this scaling is achieved using JSQ.

\(^1\)Here, by $f(x) \sim g(x)$, we mean $\lim_{x \rightarrow 0} \frac{f(x)}{g(x)} = 1$. 
8.3 New Technique: Drift Conservation

We next examine the main ideas that lead to the proof of Theorem 8.2.1. A simple and yet powerful idea that we will encounter repeatedly is that of drift conservation, a concept that we have already encountered partially back in Section 5.3. Suppose we are dealing with a Markov process $X(\cdot)$, and that $X$ is positive recurrent and admits a unique steady-state distribution. Then, suppose that we initialize $X(0)$ in this steady-state distribution, by stationarity, we must have that $X(t)$ has the same distribution for all $t$. This must further imply that, for any Lyapunov function $V$, 

$$E[V(X(1))] = E[V(X(0))]$$

provided $V(X(0))$ has a finite expectation. Noting that $V(X(1))$ can be further expressed as $V(X(0))$ plus a drift:

$$V(X(1)) = V(X(0)) + (V(X(1)) - V(X(0)))$$

We see that

$$E[V(X(1)) - V(X(0))] = 0.$$ 

That is, the average drift in one step in a stationary Markov process must be equal to zero.

Why is this useful for us? I’m using the term “drift conservation” to signify a parallel with the flow conservation principle of the stationary measure in discrete-state Markov chains. Just like how we use flow conservation to set up a system of equations to calculate the steady-state distribution of a Markov chain, we can use the drift conservation such as (8.10) to pin down properties of the stationary distribution! Moreover, we have the extra degree of freedom here to choose the Lyapunov function $V$, over which the drift is defined. The method is powerful, because the function $V$ serves as a “microscope”, that will allow us to “zoom” in onto, and accentuate, different aspects of the stationary distribution that may be of interest to us.

Going forward, some notation will be helpful. Recall the operator $P$ from (4.6), such that $Pf(i) = E[f(X(t + 1)) \mid X(t) = i]$. Define

$$\Delta V(x) = (PV - V)(x) = E[V(X(t + 1)) - V(x) \mid X(t) = x].$$

that is, $\Delta V(x)$ is the one-step expected drift in $V$ starting in state $x$. In this notation, (8.10) is equivalent to, by the towering property of expectation,

$$E[\Delta V(X(\infty))] = 0,$$

where $X(\infty)$ is the unique steady state distribution of the Markov process $X$.

8.4 Drift Conservation for the Lower Bound

To start, let us see how the drift conservation principle is used in proving the queue length lower bound in Theorem 8.2.1. We will begin with a reduction and coupling by considering a fictitious system that produces a total queue length than any routing policy in the actual system. This part is fairly straightforward. Imagine that instead of having $n$ queues with $n$ servers, we can simply pool together all the arrivals and potential departures into a single queue. That is, this single queue system, labeled by the superscript $r$, has in each time step

1. Arrivals $A^r(t) = A^r[1](t)$.
2. Potential departures $D^r(t) = D^r[1](t)$.
3. Queue length evolution: \( Q^r(t + 1) = (Q^r(t) + A^r(t) - D^r(t))^+ \).

It is easy to show that this system performs no worse than any policy in the original system, in the sense that, almost surely, for all \( t \)
\[
Q^r(t) \leq \sum_i Q_i(t), \tag{8.13}
\]
assuming that \( Q^r(0) = \sum_i Q_i(0) \).

What remains to be shown is a lower bound on \( \mathbb{E}[Q^r(\infty)] \) in this fictitious system. So far, we have only seen the Foster-Lyapunov drift analysis being used to establish upper bounds on queue lengths. Here, we will instead use drift methods to establish lower bounds; the new ingredients here will be the drift conservation principle.

We will omit the superscript \( r \) in the remainder of this section, knowing that we will be exclusively looking at the single-queue fictitious system. It can be easily shown that \( Q \) is positive recurrent with all moments finite in steady state. Denote by \( Y(t) \) the amount of idling in time step \( t \), i.e., the amount of potential departures that is wasted:
\[
Y(t) = (Q(t) + A(t) - D(t))^-. \tag{8.14}
\]
We have that \( Q(1) = Q(0) + A(0) - D(0) + Y(0) \).

Consider the quadratic Lyapunov function \( V(x) = x^2 \), and the quantity \( \Delta V \) defined in (8.11).
We have
\[
\begin{align*}
\Delta V(x) &= \mathbb{E}[Q(1)^2 - x^2 | Q(0) = x] \\
&= \mathbb{E}[(x + A(0) - D(0) + Y(0))^2 - x^2 | Q(0) = x] \\
&= \mathbb{E}[(x + A(0) - D(0))^2 + 2(x + A(0) - D(0))Y(0) + Y(0)^2 - x^2 | Q(0) = x] \\
&= \mathbb{E}[(A(0) - D(0))^2] + 2x\mathbb{E}[A(0) - D(0)] - \mathbb{E}[Y(0)^2 | Q(0) = x]. \tag{8.15}
\end{align*}
\]
In the last step, we have used the fact that
\[
Y(0)(Q(0) + A(0) - D(0)) = -Y(0)^2. \tag{8.16}
\]
To see why this is true, we note
\[
Y(0)(Q(0) + A(0) - D(0)) = Y(0)(Q(1) - Y(0)).
\]
If \( Q(1) > 0 \), we know that there must have been no idling in time step 0, and \( Y(0) = 0 \). This shows that
\[
Y(0)Q(1) = 0. \tag{8.16}
\]
And (8.15) follows.

We now evoke the drift conservation principle. Let \( \lambda_{\Sigma} = \lambda_{\Sigma}^c \). Let \( X \) be a random variable distribution according to the law of \( Q(\infty) \). By the towering property of expectation, we have
\[
0 = \mathbb{E}[\Delta V(X)] \\
= \mathbb{E}[(A(0) - D(0))^2] + 2\mathbb{E}[X]\mathbb{E}[(A(0) - D(0))] - \mathbb{E}[\mathbb{E}[Y(0)^2 | Q(0) = X]] \\
= \mathbb{E}[(A(0) - D(0))^2] - 2\mathbb{E}[X]\epsilon - \mathbb{E}[\mathbb{E}[Y(0)^2 | Q(0) = X]] \\
\geq c^\epsilon - 2\mathbb{E}[X]\epsilon - \mathbb{E}[\mathbb{E}[Y(0)^2 | Q(0) = X]] \\
\geq c^\epsilon - 2\mathbb{E}[X]\epsilon - (nd_{max})\epsilon. \tag{8.17}
\]
For the first inequality, observe
\[
E[(A(0) - D(0))^2] = E[(A(0) - \lambda\Sigma - (D(0) - \mu\Sigma) + (\lambda\Sigma - \mu\Sigma))^2] \\
= \sigma^2_\Sigma + \nu^2_\Sigma + \epsilon^2 + E[2(A(0) - \lambda\Sigma)(D(0) - \mu\Sigma)] \\
= \sigma^2_\Sigma + \nu^2_\Sigma + \epsilon^2, \tag{8.18}
\]
where \(E[2(A(0) - \lambda\Sigma)(D(0) - \mu\Sigma)]\) due to the independence between \(A\) and \(D\).

The last inequality in (8.17) is based on the observation that \(Y(0) \leq D(0) \leq nd_{\text{max}}\), and the fact that
\[
E[Y(0)] = \epsilon \tag{8.19}
\]
since we know that on average potential departures will be wasted at a rate \(\epsilon\).

Rearranging the above inequalities and dividing both sides by \(\epsilon\), we get
\[
E[Q^r(\infty)] = E[X] \geq \frac{c\epsilon - (nd_{\text{max}})^2}{2\epsilon}, \tag{8.20}
\]
as claimed.

Use of stationarity. It is interesting to note that the stationarity of the Markov process \(Q^r\) was evoked in two separate places in the proof:

1. First, it is used in evoking drift conservation. This we have already discussed.
2. A second, more subtle, usage is in observing that \(E[Y(0)] = \epsilon\) in stationarity. Both steps are crucial for the optimality result that follows. For instance, if we had skipped the second step and used the cruder bound
\[
E[Y(0)] \leq (nd_{\text{max}})^2, \tag{8.21}
\]
then the lower bound would have become
\[
E[Q^r(\infty)] \geq \frac{c\epsilon - (nd_{\text{max}})^2}{2\epsilon}, \tag{8.22}
\]
which is much too weak for any good.

8.5 Upper Bound: Drift Conservation with State Space Collapse

We now turn to the proof of the upper bound on the steady state queue lengths under JSQ. We will again use the drift conservation condition, i.e., by setting the steady-state expected drift of a Lyapunov function to zero, and see what structural property manifests.

However, naively doing so over the original queue length process \(Q\) under a quadratic Lyapunov function wouldn’t work. In fact, we have done exactly that back in Section 5.3 in bounding the steady-state queue length under the Max-Weight policy. There, we saw that the naive application of the drift conservation principle gives us some upper bound, but the bound scales as \(O(n)\). In contrast, the lower bound we have established just now does not depend on the dimension \(n\) (number of queues) at all!

Recall that the lower bound proof in the previous section relies on comparing to a fictitious system with one queue, where all the service resources and arrivals are centrally “pooled”. To have a matching upper bound, we probably would like that the queues under JSQ behave just like the one queue in the fictitious system? This may sound similar to you - because this is exactly the intuition behind the state space collapse phenomenon we had seen in the last chapter!
We are now ready to explain what “extra twist” we need to add to drift conservation to make it work. Instead of setting drift to zero for the steady state distribution of the original $n$-dimensional queue length process, $Q(\infty)$, we will

1. first establish an SSC result, showing $\epsilon \to 0$, the $Q^\epsilon(\infty)$ collapses to a one-dimensional invariant manifold $M$, along the direction of some unit vector $\xi$, and

2. then apply a drift conservation argument to the one dimensional, projected queue length process $\langle Q^\epsilon(\infty), \xi \rangle$, in a manner similar to the one we have seen in the lower bound.

The “double drift” in the title comes from this step-step procedure: in both steps, we will use a drift argument to establish the claim. The first “drift” shows SSC to $M$, and the second bounds the queue length along $M$.

### 8.5.1 Steady State SSC

When it comes to the SSC, many of the ideas, intuition and notation coincide with what we have already seen in the transient version, in Chapter 7. In the remainder of the section, we may suppress the superscript $\epsilon$ when the context is clear.

Consider the normal vector

$$\xi = (1/\sqrt{n}, \ldots, 1/\sqrt{n}).$$

(8.23)

And define the projections of $Q^\epsilon$ using the operators we had encountered in (7.34):

$$Q^\epsilon_\parallel(t) = \langle Q(t), \xi \rangle \xi, \quad Q^\epsilon_\perp(t) = Q(t) - Q^\epsilon_\parallel(t).$$

(8.24)

Importantly, note that $Q^\epsilon_\parallel(\infty)$ is essentially the average queue length:

$$Q^\epsilon_\parallel(\infty) = \frac{1}{\sqrt{n}} \sum_i Q^\epsilon_i(\infty).$$

(8.25)

And to obtain our upper bound, it suffices to develop a good bound on $Q^\epsilon_\perp(\infty)$.

The main result establishing SSC under JSQ is stated below. The proof of the result follow similar intuition as the drift arguments carried out in the fluid limits in Section 7.5.4; refer to Proposition 1 of [Eryilmaz and Srikant, 2012] a detailed proof.

**Lemma 8.5.1.** Let $\mu_{\min} := \min_i \mu_i$. Then, there a sequence with finite elements $\{N_k\}_{k \in \mathbb{N}}$, such that for any $\delta \in (0, \mu_{\min})$, we have for $\epsilon \in (0, n(\mu_{\min} - \delta))$,

$$E\left[\|Q^\epsilon_\perp(\infty)\|_2^k\right] \leq N_k,$$

(8.26)

for all $k \in \mathbb{N}$.

The key feature of the above result is that the bounds on the moments of $\|Q^\epsilon_\perp(\infty)\|_2^k$ do not depend on $\epsilon$. That is, the component of $Q$ perpendicular to $\xi$ remains bounded in its various moments even in the heavy-traffic limit of $\epsilon \to 0$. This is a key step towards showing that, in the heavy-traffic limit, essentially “all of $Q$” lies along the direction of $\xi$. 

8.5.2 From SSC to the Upper Bound

We now use the SSC in Lemma 8.5.1 to establish the upper bound in Theorem 8.2.1. Define the following notation: for a vector $x \in \mathbb{R}_+$, define the scalar

$$x_{|\xi|} := \langle x, \xi \rangle. \tag{8.27}$$

In other words, $x_{|\xi|}$ is the signed length of $x_{||}$ along $\xi$. We also have, by definition,

$$(x_{|\xi|})^2 = \|x_{||}\|^2. \tag{8.28}$$

Let us define the idling process, similar to what we did for the fictitious system:

$$Y(t) = (Q(t) + A(t) - D(t))^- . \tag{8.29}$$

Let $\Gamma$ be the gap between the potential departure $D$ and arrivals $A$:

$$\Gamma(t) := D(t) - A(t). \tag{8.30}$$

Note that $Y(t)$ and $A(t)$ can both depend on $Q(t)$.

Our first lemma is a direct consequence of the drift conservation property in steady state.

**Lemma 8.5.2.** Fix $\epsilon$. Suppose that all finite moments of $\|Q(\infty)\|_2$ are finite, then following holds in steady state (omitting ($\infty$) in all quantities)

$$E \left[ Q_{|\xi|} \cdot \Gamma_{|\xi|} \right] = \frac{1}{2} E \left[ \|\Gamma_{||}\|^2 \right] + \frac{1}{2} E \left[ \|Y_{||}\|^2 \right] + E \left[ (Q - \Gamma)_{|\xi|} \cdot Y_{|\xi|} \right]. \tag{8.31}$$

**Proof.** The proof will make use of the same quadratic Lyapunov function $V$ from the lower bound. Define

$$V_{||}(x) = \|x_{||}\|^2. \tag{8.32}$$

Let $\Delta V_{||}$ be defined analogously:

$$\Delta V_{||}(x) = (PV_{||} - V_{||})(x). \tag{8.33}$$

In the following equation, let us use the short-hand notation $E_x \left[ \cdot \right] = E \left[ \cdot \mid Q(0) = x \right]$, and assume all index random variables have $t = 0$ and suppress that from our notation. After expanding lots of quadratic equations, we get:

$$E_x \left[ \Delta V_{||}(x) \right] = E_x \left[ \|x_{||} + A_{|\xi|} + Y_{|\xi|}\|^2 - \|x_{||}\|^2 \right]$$

$$= E_x \left[ \|x_{||} + A_{|\xi|} + Y_{|\xi|}\|^2 - \|x_{||}\|^2 + 2 \|x_{||} + A_{|\xi|} + Y_{|\xi|}\|_{|\xi|} \cdot Y_{|\xi|} + \|Y_{||}\|^2 - \|x_{||}\|^2 \right]$$

$$= E_x \left[ 2x_{|\xi|} \cdot (A_{|\xi|} + Y_{|\xi|}) + \|A_{|\xi|} + Y_{|\xi|}\|^2 + 2 \|x_{||} + A_{|\xi|} + Y_{|\xi|}\|_{|\xi|} \cdot Y_{|\xi|} + \|Y_{||}\|^2 - \|x_{||}\|^2 \right]$$

$$= -2E_x \left[ x_{|\xi|} \cdot \Gamma_{|\xi|} \right] + E_x \left[ \|\Gamma_{||}\|^2 + \|Y_{||}\|^2 + 2 \|x_{||} + A_{|\xi|} + Y_{|\xi|}\|_{|\xi|} \cdot Y_{|\xi|} \right] \tag{8.34}$$

Let $X$ be distributed according to $Q(\infty)$ and independent from $Q$. We note that it is not difficult to show that $E \left[ \|Q(\infty)\|_2^2 \right]$ is bounded and hence $E \left[ \Delta V(X) \right] < \infty$. This allows us to conclude that, by drift conservation, $E \left[ \Delta V(X) \right] = 0$. Substituting this fact into the above derivation leads to the desirable result. \qed
Remark 8.5.3 (Comparison to vanilla Lyapunov drift). Note that (8.33) amounts to saying

\[ E_x \Delta V(x) = -2E_x \left[ x|_\xi \cdot \Gamma|_\xi \right] + E_x \left[ \| \Gamma \|_2^2 + \| Y \|_2^2 \right] + 2E_x \left[ (x - \Gamma)|_\xi \cdot Y|_\xi \right] \]

Compare this to the very first Lyapunov-Foster drift condition we encountered back in equation (4.13), we see that the form of the two expressions are very similar. The first term corresponds to the "negative drift", and the second term to positive drifts due to fluctuations in the system. The new element comes in the third term: Whereas the drift back in (4.13) was measured in the native space of \( Q(t) \), here we are measuring drift along the projected process \( x|_\xi \).

Roughly speaking, the third term \( E_x \left[ (x - \Gamma)|_\xi \cdot Y|_\xi \right] \) measures the alignment between \( x \) and the invariant manifold along \( \xi \). On the one extreme, if \( x \) is perfectly aligned with \( \xi \), then all queues should be of equal length and there should be no idling when \( x \) is large. As a result, the product between the two should be small. On the other hand, if \( x \) is poorly aligned with \( \xi \), e.g., when half of the coordinates of \( x \) are zero while the other half are large, then \( Y|_\xi \) can be large due to the resulting idling in those empty queues. In this case, the product can be large.

The above intuition suggests a natural bound on this last term, using state space collapse. In the following analysis, we will show that the SSC property outlined in Lemma 8.5.1 is sufficient for ensuring that the third term remains small.

Let us now explain why Lemma 8.5.2 is useful for upper bounding the total queue length. Since \( \xi \) has identical entries, every time we project a vector onto \( \xi \), we can think of the resulting projection as a form of total sum or average of the vector’s entries. Using this logic, we see that \( \Gamma(t)|_\xi \) is nothing but the difference between total service and arrivals:

\[ \Gamma(t)|_\xi = \frac{1}{\sqrt{n}} \left( \sum_i D_i(t) - \sum_i A_i(t) \right) = \frac{1}{\sqrt{n}} (D_\Sigma(t) - A_\Sigma(t)) \quad \text{(8.34)} \]

Therefore, while the vector \( \Gamma(t) \) can depend on \( Q(t) \) through \( A(t) \), the scalar \( \Gamma(t)|_\xi \) is independent of \( Q(t) \)! The same is true in steady state if we take \( t \to \infty \). We thus have

\[ E \left[ Q|_\xi \cdot \Gamma|_\xi \right] = E \left[ Q|_\xi \right] E \left[ \Gamma|_\xi \right] = E \left[ Q|_\xi \right] \frac{1}{\sqrt{n}} (\mu_\Sigma - \lambda_\Sigma) = E \left[ Q|_\xi \right] \frac{\epsilon}{\sqrt{n}} = \frac{\epsilon}{n} E \left[ \sum_i Q_i \right]\quad \text{(8.35)} \]

This means that to get a handle on the total queue length, it suffices to control the right-hand side of (8.34):

\[ \frac{1}{2} E \left[ \| \Gamma \|_2^2 \right] + \frac{1}{2} E \left[ \| Y \|_2^2 \right] + E \left[ (Q - \Gamma)|_\xi \cdot Y|_\xi \right] \quad \text{(8.36)} \]

Let us examine each of these three terms:

1. \( E \left[ \| \Gamma \|_2^2 \right] \) should fairly simple to handle, since the distribution of \( \Gamma \) does not even depend on the policy. By (8.18) we get

\[ E \left[ \| \Gamma \|_2^2 \right] = \frac{1}{n} E \left[ (D_\Sigma(t) - A_\Sigma(t))^2 \right] = \frac{\sigma^2 + \nu^2 + c^2}{n} = \frac{c}{n} \quad \text{(8.37)} \]

2. For the second term, we will use the stationarity of \( Q \) similar to that in (8.19). Note that if we initialize \( Q(0) \) in its steady state, then

\[ E \left[ Q(0)|_\xi \right] = E \left[ Q(1)|_\xi \right] = E \left[ Q(0)|_\xi + A(0)|_\xi - D(0)|_\xi + Y(0)|_\xi \right] \\
= E \left[ Q(0)|_\xi \right] - E \left[ \Gamma(0)|_\xi \right] + E \left[ Y(0)|_\xi \right] \quad \text{(8.38)} \]
On the one extreme, if \( x \) is aligned closely along \( \xi \), then there should be little idling in the system when \( x|_{\xi} \) is large. Canceling out \( \mathbb{E} \left[ Q_{|\xi}(0) \right] \) on both sides and we obtain

\[
\mathbb{E} \left[ Y(0)|_{\xi} \right] = \mathbb{E} \left[ \Gamma(0)|_{\xi} \right] = \mathbb{E} \left[ \frac{1}{\sqrt{n}} (D_{\Sigma}(0) - A_{\Sigma}(0)) \right] = \frac{\epsilon}{\sqrt{n}}. \tag{8.39}
\]

Using this fact, we get

\[
\mathbb{E} \left[ \|Y\|^2 \right] \leq \mathbb{E} \left[ Y_{|\xi} \right] \cdot \langle \xi, d_{\max} 1 \rangle = \frac{\epsilon d_{\max}}{n}. \tag{8.40}
\]

3. Finally, the term \( \mathbb{E} \left[ (Q - \Gamma)|_{\xi} \cdot Y_{|\xi} \right] \) is the most difficult. Essentially, we would like to show that when \( Q_{|\xi} - \Gamma_{|\xi} \) is large, then \( Y_{|\xi} \) must be small, thus making the expectation small overall. As described in Remark 8.5.3, our plan of attack will be to use the SSC property in bounding this term.\(^2\)

We can rewrite the term as

\[
\mathbb{E} \left[ (Q(t)|_{\xi} - \Gamma(t)|_{\xi}) Y(t)|_{\xi} \right] = \mathbb{E} \left[ (Q(t+1)|_{\xi} - Y(t)|_{\xi}) Y(t)|_{\xi} \right] \\
= \mathbb{E} \left[ Q(t+1)|_{\xi} Y(t)|_{\xi} \right] - \mathbb{E} \left[ (Y(t)|_{\xi})^2 \right] \\
\leq \mathbb{E} \left[ Q(t+1)|_{\xi} Y(t)|_{\xi} \right]. \tag{8.41}
\]

That is, we want to show that \( Q(t+1)|_{\xi} \) and \( Y(t)|_{\xi} \) are nearly orthogonal, so that the inner product \( \mathbb{E} \left[ Q(t+1)|_{\xi} Y(t)|_{\xi} \right] \) is small. Note that we have already seen the same orthogonality property back in the lower bound proof, in (8.16). The additional challenge in this case is to do so for the projected queue length vector that has an original dimension of \( n \), in which idling is much more likely to occur, as opposed to a system that has only one queue to begin with.

Before developing a rigorous bound on \( \mathbb{E} \left[ Q(t+1)|_{\xi} Y(t)|_{\xi} \right] \), let us develop first some key geometric intuition. A key idea is to realize the following orthogonality, a direct generalization of (8.16) in high dimensions:

\[
\langle Q(t+1), Y(t) \rangle = 0. \tag{8.42}
\]

To see why this is true, it is clear that if there’s any unused service in queue \( i \) at the end of time \( t \) (\( Y_{|i}(t) > 0 \)), then \( Q_{|i}(t+1) \) must be equal to zero. With this in mind, the relationships between \( Q(t+1), Y(t) \) and \( \xi \) can be visualized as in Figure 8.2. Even though \( \langle Q(t+1), Y(t) \rangle = 0 \), it does not mean that \( Q(t+1)|_{\xi} Y(t)|_{\xi} = 0 \); the latter can still be large if both \( Q(t+1) \) and \( Y(t) \) have some moderate angles away from \( \xi \), such as the case illustrated by the left panel. However, if \( Q(t+1) \) is aligned with \( \xi \) (under SSC), then the fact that \( \langle Q(t+1), Y(t) \rangle = 0 \) means that \( Y_{|\xi} \) must be almost zero (right panel). In this case, it is not difficult to see that \( Q(t+1)|_{\xi} Y(t)|_{\xi} \) should be small.

Let us now make the above argument rigorous. We begin by observing the following property:

**Lemma 8.5.4.** For any \( t \)

\[
Q(t+1)|_{\xi} \cdot Y(t)|_{\xi} = \langle -Q_{\perp}(t+1), Y(t) \rangle \tag{8.43}
\]

\(^2\)This is actually a version of the condition (d) of the Skorohod problem (no increase in idling when there are jobs present), and precisely the intuition we had discussed in the last chapter when it comes to the benefits of SSC.
8.5. UPPER BOUND: DRIFT CONSERVATION WITH STATE SPACE COLLAPSE

Recall the definition, such that $\langle x_\perp, y_\parallel \rangle = 0$ for any $x, y \in \mathbb{R}^n$. Step (b) follows from the orthogonality between $Q(t + 1)$ and $Y(t)$ above.

Returning to our task in (8.41), and using the lemma above, we obtain that, assuming $Q(t)$ is distributed according to $Q(\infty)$,

$$
E \left[ (Q(t)\xi - \Gamma(t)\xi)Y(t)\xi \right] \leq E \left[ Q(t + 1)\xi Y(t)\xi \right] \\
= E \left[ (-Q_\perp(t + 1), Y(t)) \right] \\
\leq \sqrt{ E \left[ ||Q_\perp(t + 1)||_2^2 \right] E \left[ ||Y(t)||_2^2 \right]} \\
= \sqrt{ E \left[ ||Q_\perp(t)||_2^2 \right] E \left[ ||Y(t)||_2^2 \right]} \quad (8.45)
$$

where the inequality follows from Cauchy-Schwartz, and the last equality from the stationarity of $Q$. 

**Proof.** Using this fact, we have

$$
Q(t + 1)\xi \cdot Y(t)\xi = \langle Q_\parallel(t + 1), Y_\parallel(t) \rangle \\
= \langle Q(t + 1) - Q_\perp(t + 1), Y_\parallel(t) \rangle \\
\overset{(a)}{=} \langle Q(t + 1), Y_\parallel(t) \rangle \\
= \langle Q(t + 1), Y(t) - Y_\perp(t) \rangle \\
\overset{(b)}{=} \langle Q(t + 1) - Y_\perp(t) \rangle \\
= \langle Q_\parallel(t + 1) + Q_\perp(t + 1), -Y_\perp(t) \rangle \\
\overset{(c)}{=} \langle Q_\perp(t + 1), -Y_\perp(t) \rangle \\
\overset{(d)}{=} \langle -Q_\perp(t + 1), Y(t) + Y_\perp(t) \rangle \\
= \langle -Q_\perp(t + 1), Y(t) \rangle. 
\quad (8.44)
$$

Here, step (a), (c) and (d) all follow from the definition, such that $\langle x_\perp, y_\parallel \rangle = 0$ for any $x, y \in \mathbb{R}^n$. Step (b) follows from the orthogonality between $Q(t + 1)$ and $Y(t)$ above. \( \square \)
Let us now evoke the SSC result from Lemma 8.5.1, and get
\[
E \left[ (Q(t)_{\xi} - \Gamma(t)_{\xi}) Y(t)_{\xi} \right] \leq \sqrt{E \left[ \|Q_{\perp}(t)\|_2^2 \right] \cdot E \left[ \|Y(t)\|_2^2 \right]}
\leq \sqrt{E \left[ \|Q_{\perp}(t)\|_2^2 \right] d_{\max} \cdot \sqrt{E \left[ Y(t)_{\xi} \right]}}
\leq \sqrt{\epsilon N_2 d_{\max}}.
\tag{8.46}
\]

We are now ready to put everything back together. Substituting the bounds for the three terms back in (8.36), we get
\[
E \left[ \sum_i Q_i \right] = \frac{n}{\epsilon} E \left[ Q_{\xi} \cdot \Gamma_{\xi} \right]
\leq \frac{n}{\epsilon} \left( \frac{c^2}{2n} + \frac{\epsilon d_{\max}}{2n} + \sqrt{\epsilon N_2 d_{\max}} \right)
= \frac{c^2}{2\epsilon} + \frac{d_{\max}}{2} + n \sqrt{\frac{N_2 d_{\max}}{\epsilon}}.
\tag{8.47}
\]
This establishes the upper bound.

8.6 Discussion

Because we made extensive uses of quadratic Lyapunov functions in our analysis, we often had to show that cross product terms were small or zero. To do so, we made repeated use of the orthogonality of \(Q(t + 1)\) and \(Y(t)\) in various forms. Is this an ad-hoc trick that happens to work? Or is there something more fundamental? The latter seems more likely, because we can interpret this type of orthogonality as another way of saying that the system is fully utilizing the resources, by ensuring that there be no idling in step \(t\) as long as the queue lengths in step \(t + 1\) remain non-zero. In a way, this is equivalent to the condition in the Skorohod problem in one dimension, which a necessary and sufficient functional condition on optimal workload processes.

8.7 Notes
Chapter 9

Potential Functions in Online Learning

9.1 A Marching Call: From Stochastic Networks to Online Learning

We will now make the transition from stochastic networks to machine learning. In the next two chapters, we will focus on fundamental models of online learning and prediction: prediction with expert advice and multi-arm bandit. The term “online” is to emphasize two key aspects of learning problem:

1. The learning process unfolds in time in a sequential manner.
2. Future decisions or predictions can depend on past observation and knowledge in an adaptive manner.

There is huge variety of online learning problems, but they are largely driven by a central tension between the past and future: on the one hand, the decision maker would like to make the best use of past data to inform future decisions, while on the other hand, one would want to build in some safeguards and not “overfit” too much to past observations, in case they do not offer a complete picture and turn out to be unrepresentative of future events. Significant efforts have thus been devoted to understanding how to balance these opposing forces in various problem formulations.

In this chapter, we will begin by examining a corner stone of online learning, the problem of prediction with expert advice (PEA). Unlike the multi-arm bandit problem, the information that the decision maker gathers in PEA does not depend on their action, which takes off one additional layer of complexity. As we will see shortly, a central concept in online learning is that of regret. Interestingly, there is a strong parallel between the evolution of the regret process in a learning problem and that of the queue lengths in a stochastic network. Whereas in a queueing network one strives to allocate resources in such a way that drive queues towards zero, the ultimate goal of online learning is to make decisions so as to drive the regret vector towards zero. Viewed from this vantage point, one may expect that drift methods and Lyapunov functions should play a similarly important role in online learning, helping us understand just how the regret process evolves under various algorithms.

With this in mind, in this chapter, we will indeed see a striking parallel between PEA and Max Weight algorithm, a deep connection that may be best explored and unified through the theory of approachability pioneered by David Blackwell. But, the story is not as simple as saying that the two fields are after all one and the same. Along the way, we will see that potential functions and drift analysis being employed in very different ways than what we had experienced in the world of stochastic networks. Just why they diverge and what commonalities lie beneath such differences? These are ever enduring and intriguing research questions for us to explore...
9.2 Prediction with Expert Advice

We now define model of prediction with expert advice. The system consists of a decision maker operating in discrete time \( t \in \mathbb{N} \), whom we refer to as the forecaster. The forecaster’s goal is to make accurate predictions for an unknown sequence, \( y_1, y_2, \ldots \), whose values lie in an outcome space \( \mathcal{Y} \). In time step \( t \), the forecaster produces a forecast \( \hat{p}_t \), taking values in a decision space, \( \mathcal{D} \). We will assume that \( \mathcal{D} \) and \( \mathcal{Y} \) are convex subsets of vector spaces. Finally, the forecaster incurs a loss \( \ell(\hat{p}_t, y_t) \), where \( \ell(\cdot, \cdot) : \mathcal{D} \times \mathcal{Y} \to \mathbb{R} \) is the loss function that measures the discrepancy between the true value \( y_t \) and the prediction \( \hat{p}_t \). We will assume that \( \ell \) is uniformly bounded over \( \mathcal{D} \times \mathcal{L} \).

We now introduce the notion of experts, which serves both as a key source of information for the forecaster, as well as a relative performance benchmark against which the forecaster will be measured. Suppose we have a finite group of \( K \) experts, indexed by the set \( \mathcal{E} = \{1, 2, \ldots, K\} \), such that in time step \( t \), the prediction produced by expert \( E \in \mathcal{E} \) is \( f_{E,t} \). We say that the forecaster predicts with expert advice if the prediction \( \hat{p}_t \) is a (possibly random) function of

1. Past and current expert predictions: \( \{f_{s,E} : s = 1, \ldots, t, E \in \mathcal{E}\} \).
2. Past sequence values: \( \{y_s : s = 1, \ldots, t-1\} \).

To simplify notation, we will use \( f_t \) to denote the vector \( f_t = (f_{1,t}, f_{2,t}, \ldots, f_{K,t}) \).

The same convention will be used in analogous manner for vectors with double subscripts, such as \( R_{E,t} \).

**Definition 9.2.1.** The cumulative regret (or simply regret) of the forecaster at time \( t \), \( R_t \in \mathbb{R}^K \), is defined as the difference between their cumulative loss and those of the individual experts:

\[
R_{E,t} = \sum_{s=1}^{t} (\ell(\hat{p}_s, y_s) - \ell(f_{E,s}, y_s)), \quad E \in \mathcal{E}.
\]

That is, \( R_{E,t} \) captures the additional loss incurred by the forecaster as compared to following the advice of the same expert, \( E \), the entire time. Define the instantaneous regret \( R_t \):

\[
r_{E,t} = \ell(\hat{p}_t, y_t) - \ell(f_{E,t}, y_t).
\]

Then we can write

\[
R_t = \sum_{s=1}^{n} r_s.
\]

The forecaster’s goal is to achieve cumulative losses that are comparable to the single best expert in hindsight. We say that the forecaster achieves sub-linear regret if

\[
\limsup_{t \to \infty} \max_{E \in \mathcal{E}} \frac{1}{t} R_{E,t} = 0. \tag{9.4}
\]

This sub-linear regret criterion is also known as Hannan consistency [Hannan, 1957], and a forecaster that satisfies (9.4) is called Hannan consistent.
9.3 Weighted Average Forecasters

Let us take a step back and look at how a good forecast strategy might look like. Because the forecaster has at its disposal all past expert predictions, it is not difficult to find the expert, $E^*$, who has been performing the best so far. If past performance is indeed indicative of the future, it would make sense for the forecast to bestow more trust into $E^*$ and weigh its prediction more heavily. On the other hand, because we are not imposing any stochastic assumption on the value sequence $y$ or the experts’ predictions, there is always the possibility that $E^*$ suddenly starts to perform poorly. In the most extreme case, one can create pathological examples where the best expert up to time $t-1$ always has the worst performance at time $t$; in that case, it’s clear that a naive strategy of following the best expert will lead to disastrous (linear) regret compared to the best expert in hindsight.

With this tradeoff in mind, it is reasonable for us to conjecture that a good prediction strategy would consist of giving more weights to the better experts so far, while building in enough randomization so as not to put all the eggs in one basket. This is accomplished by a family of weighted average forecasters (WAF), where the forecaster produces the prediction by weighing the expert advice, where the weights are functions of the experts’ past regrets.

**Definition 9.3.1.** Fix $K \in \mathbb{N}$ and

1. $\phi: \mathbb{R} \mapsto \mathbb{R}$, a nonnegative, nondecreasing and twice differentiable function, and

2. $\psi: \mathbb{R} \mapsto \mathbb{R}$, a nonnegative, concave, strictly increasing, and twice differentiable function.

We define the $(\phi, \psi)$-weighted Lyapunov function $V: \mathbb{R}^K \to \mathbb{R}^+$ to be:

$$V(x) = \psi\left(\sum_{i=1}^{K} \phi(x_i)\right), \quad x \in \mathbb{R}^K. \quad (9.5)$$

**Definition 9.3.2 (Weighted Average Forecaster (WAF)).** Fix $\psi$ and $\phi$ as in Definition 9.3.1, and let $V$ be the corresponding $(\phi, \psi)$-weighted Lyapunov function. Define $q_t$ to be the probability vector:

$$q_{i,t} := \frac{\nabla V(R_{t-1})_i}{\sum_{j \in E} \nabla V(R_{t-1})_j} = \frac{\phi'(R_{i,t-1})}{\sum_{j=1}^{K} \phi'(R_{j,t-1})}. \quad (9.6)$$

If $\nabla V(R_{t-1}) = 0$, all entries of $q_t$ are set to $1/d$. Then, the weighted average forecaster induced by $V$ is defined by

$$\hat{p}_t = \langle q_t, f_t \rangle. \quad (9.7)$$

The WAF prediction $\hat{p}_t$ can be interpreted as the expected value of some random variable $\hat{P}$, where

$$\mathbb{P}(\hat{P} = f_{i,t}) = q_{i,t}. \quad (9.8)$$

Despite this interpretation involving a random variable, the prediction $\hat{p}_t$ is a deterministic function of $f_t$ and $q_t$.

The WAF is based on the premise that if the cumulative regret against an expert $i$ is fairly large, it suggests that expert $i$ is performing particularly well relative to the forecaster, and hence probably deserves more imitation. Note that the value of $q_t$, and therefore the forecaster, does not depend on the function $\psi$.

The main result of this section is to show that by using different functions $\phi$ and $\psi$, we can create various WAFs that achieve sub-linear regrets.
Theorem 9.3.1 (Potential-based Forecasters). Fix $K \in \mathbb{N}$ and a loss function $\ell$ that is bounded within $[0, 1]$ and convex in its first argument. The following holds:

1. Polynomally weighted forecaster: Fix $p \geq 2$ and let
   \[ \phi(x) = (x^+)^p \]
   \[ \psi(x) = (x)^{2/p} \]

   so that
   \[ V(x) = \|x^+\|_p^2, \quad x \in \mathbb{R}^K, \]
   where $\| \cdot \|_p$ is the $L_p$ norm. Then, for any $T \in \mathbb{N}$
   \[ \max_i R_i,T \leq \sqrt{(p - 1)K^2/pT} \]

2. Exponentially weighted forecaster: Fix $\eta > 0$, and let
   \[ \phi(x) = \exp(\eta x) \]
   \[ \psi(x) = \frac{1}{\eta} \ln x \]

   so that
   \[ V(x) = \frac{1}{\eta} \ln \left( \sum_{i=1}^K \exp(\eta x_i) \right), \quad x \in \mathbb{R}^K. \]

   Then, for any $T \in \mathbb{N}$
   \[ \max_i R_i,T \leq \frac{\ln K}{\eta} + \frac{\eta T}{2}. \]

   In particular, setting $\eta = \sqrt{2 \ln T / K}$ in the above inequality, we have
   \[ \max_i R_i,T \leq \sqrt{2(\ln K)T} \]

9.4 Drift Analysis of Weighted Average Forecasters

In a style reminiscent of the stability analysis of stochastic networks, we will prove Theorem 9.3.1 by analyzing the dynamics of the Lyapunov function $V$. We begin with a simple but crucial property, a direct consequence of the definition of the Lyapunov function and the WAF.

Lemma 9.4.1 (Blackwell Condition). Suppose $\ell$ is convex in its first argument. Fix any sequence of expert predictions $\{f_{i,t}\}_{t \in \mathbb{N}}$. We have that, for any $t$,
   \[ \sup_{y_t \in \mathcal{Y}} \langle \nabla V(R_{t-1}), r_t \rangle \leq 0. \]

Proof. Fix any $y \in \mathcal{Y}$. Define $\tilde{\ell}(f_t, y)$ to be the vector where
   \[ \tilde{\ell}_i(f_t, y) = \ell(f_{i,t}, y). \]

Since $\ell$ is convex in its first argument, by Jensen’s inequality, we have
   \[ \ell(\hat{p}_t, y) = \ell(\langle q_t, f_t \rangle, y) \leq \langle q_t, \tilde{\ell}(f_t, y) \rangle. \]
9.4. Drift Analysis of Weighted Average Forecasters

Recall the definition of the instantaneous regret: \( r_{i,t} = \ell(\hat{p}_{i,t}, y_t) - \ell(f_{i,t}, y_t) \). Rearranging the terms in the inequality above, we obtain

\[
0 \geq \ell(\hat{p}_t, y_t) - \langle q_t, \ell(f_t, y_t) \rangle \\
= \langle q_t, r_t \rangle \\
= \langle \nabla V(R_{t-1}), r_t \rangle / \| \nabla V(R_{t-1}) \|. 
\] (9.19)

Note that \( R_t = R_{t-1} + r_t \), and when \( r_t \) is small, we would expect that

\[
V(R_t) - V(R_{t-1}) \approx \langle \nabla V(R_{t-1}), r_t \rangle . 
\] (9.20)

Lemma 9.4.1 thus suggests that, under the weighted average forecaster, the drift in the Lyapunov function \( V(R_t) \) is always approximately nonpositive! If this is true, then we would hope that the value of \( V(R_t) \) stays “small” even as \( t \) grows. In particular, if \( V(x) \) were linear in \( x \), we would have deduced, from Lemma 9.4.1, that

\[
V(R_T) = V(0) + \sum_{t=1}^{T} \langle \nabla V(0), r_t \rangle \leq V(0). 
\] (9.21)

Unfortunately, as we illustrate in the following example, a linear Lyapunov function in this case is not a good idea, because the nonnegative criterion on \( \phi \) and \( \psi \) cannot be enforced.

**Example 9.4.1** (Perils of Linear Lyapunov Functions). Consider a simple example where \( \phi(x) = \psi(x) = x \), and consequently \( V(x) = \sum_{i=1}^{K} x_i \) is a linear function. By (9.21), we see that \( V(R_t) \leq V(0) = 0 \) for all \( t \). This is great, right? While we may rejoice at the fact that the Lyapunov function does not grow in \( t \), something does not seem right: under this WAF, \( q_t = (1/K, 1/K, \ldots, 1/K) \), and the forecaster is simply outputting the average prediction across all experts. It is easy to see why this is a bad idea. If there is one good expert that consistently gives accurate predictions, while the rest \( K-1 \) experts simply give random predictions, then by taking the average there is no way that the forecaster can asymptotically match the performance of the good expert. So, what went wrong?

The key is to realize that by using linear \( \phi \) and \( \psi \), we have missed an important feature that motivated the use of a Lyapunov function in the first place. In order for \( V(x) \) to meaningfully describe the property of \( x \), we need the property that \( V(x) \) being small also implies that \( \| x \| \) is small. A linear function does not possess this property. Take for instance \( V(x) = x_1 + x_2 \), the set \( \{ x : V(x) = 0 \} \) is a line that contains points of arbitrarily large norm. In the context of the WAF, this crucial property is ensured by the combination of \( \phi \) and \( \psi \) being both increasing and nonnegative. Any linear choices for \( \phi \) and \( \psi \) may be increasing, but will surely fail to satisfy the nonnegativity requirement.

The example above demonstrates any nondegenerate \( V \) must be nonlinear. In this case, the nonpositive drift property of (9.20) cannot hold exactly, and we will need to further incorporate error terms that are results of the curvature of \( V \). This leads to the following lemma, which bounds \( V(R_t) \) using the higher order derivatives of \( \phi \) and \( \psi \), following a Taylor series analysis.

**Lemma 9.4.2.** Fix \( T \in \mathbb{N} \). We have

\[
V(R_T) \leq V(0) + \frac{1}{2} \sum_{t=1}^{T} C(r_t),
\] (9.22)

where

\[
C(r_t) := \sup_{x \in \mathbb{R}_+^K} \psi'' \left( \sum_{i=1}^{K} \phi(x_i) \right) \left( \sum_{i=1}^{K} \phi''(x_i)r_{i,t}^2 \right). 
\] (9.23)
Let us first provide some intuition as to why Lemma 9.4.2 is useful. Suppose under some appropriate choice of $\phi$ and $\psi$, such that $C(r_t)$ is uniformly bounded over all $t \geq 1$. The lemma would imply that

$$V(R_T) \leq \beta + \alpha T$$  \hspace{1cm} (9.24)

for some constants $\alpha, \beta > 0$. That is, the Lyapunov function $V(R_t)$ has an at most constant positive per-step drift. If we can further demonstrate that the Lyapunov function has sufficient curvature so that it grows super-linearly in the norm of its argument, e.g.,

$$V(x) \geq \|x\|^p$$  \hspace{1cm} (9.25)

for some $p > 1$, then we could “invert” (9.24) and obtain

$$\|R_T\| \leq (\beta + \alpha T)^{1/p},$$  \hspace{1cm} (9.26)

implying a sub-linear growth in regret. More precisely, with an upper bound on $V(R_T)$, we may obtain an upper bound on the regret by using the monotonicity of $\psi$ and $\phi$:

$$\psi \left( \phi \left( \max_{i=1,\ldots,K} R_{i,T} \right) \right) = \psi \left( \max_{i=1,\ldots,K} \phi(R_{i,T}) \right) \leq \psi \left( \sum_{i=1,\ldots,K} \phi(R_{i,T}) \right) = V(R_T).$$  \hspace{1cm} (9.27)

If we assume further that $\phi$ is strictly increasing and thus invertible, then the above inequality implies

$$\max_{i=1,\ldots,K} R_{i,T} \leq \phi^{-1} \left( \psi^{-1} \left( V(R_T) \right) \right).$$  \hspace{1cm} (9.28)

**Proof of Lemma 9.4.2.** Using Taylor expansion, we have that for all $t \leq T$, there exists $\beta \in \mathbb{R}^K$, such that

$$V(R_t) = V(R_{t-1} + r_t)$$
$$= V(R_{t-1}) + \langle \nabla V(R_{t-1}), r_t \rangle + \frac{1}{2} \sum_{i=1,\ldots,K} \sum_{j=1,\ldots,K} \left( \frac{\partial^2 V}{\partial x_i x_j} \bigg|_{x=\beta} \right) r_{i,t} r_{j,t}$$
$$\leq V(R_{t-1}) + \frac{1}{2} \sum_{i=1,\ldots,K} \sum_{j=1,\ldots,K} \left( \frac{\partial^2 V}{\partial x_i x_j} \bigg|_{x=\beta} \right) r_{i,t} r_{j,t},$$  \hspace{1cm} (9.29)

where the last inequality follows from Lemma 9.4.1. It suffices to bound the term involving second-order derivatives. We have that

$$\frac{\partial^2 V}{\partial x_i x_j}(x) = \psi'' \left( \sum_{i=1,\ldots,K} \phi(x_i) \right) \phi'(x_i) \phi'(x_j) + \mathbb{I}\{i = j\} \psi' \left( \sum_{i=1,\ldots,K} \phi(x_i) \right) (\psi'(x_i))^2.$$
9.4. DRIFT ANALYSIS OF WEIGHTED AVERAGE FORECASTERS

Summing over all \( i \) and \( j \), we get

\[
\sum_{i=1}^{K} \sum_{j=1}^{K} \left( \frac{\partial^2 V}{\partial x_i \partial x_j} \right)_{x=\beta} r_{i,t} r_{j,t} = \psi'' \left( \sum_{i=1}^{K} \phi(\beta_i) \right) \sum_{i=1}^{K} \phi'(\beta_i) \phi'(\beta_j) r_{i,t} r_{j,t} + \psi' \left( \sum_{i=1}^{K} \phi(\beta_i) \right) \sum_{i=1}^{K} \phi''(\beta_i) r_{i,t}^2 \\
+ \psi' \left( \sum_{i=1}^{K} \phi(\beta_i) \right) \sum_{i=1}^{K} \phi''(\beta_i) r_{i,t}^2 \\
= \psi'' \left( \sum_{i=1}^{K} \phi(\beta_i) \right) \left( \sum_{i=1}^{K} \phi'(\beta_i) r_{i,t} \right)^2 + \psi' \left( \sum_{i=1}^{K} \phi(\beta_i) \right) \sum_{i=1}^{K} \phi''(\beta_i) r_{i,t}^2 \\
\leq \sup_{x \in \mathbb{R}^K} \psi' \left( \sum_{i=1}^{K} \phi(x_i) \right) \sum_{i=1}^{K} \phi''(x_i) r_{i,t}^2 \\
= C(r_t), \quad (9.30)
\]

where in the inequality \((a)\) follows from \( \psi \) being concave. Substituting this inequality into (9.29) and summing over \( t = 1, \ldots, T \) proves the result.

**Proof of Theorem 9.3.1.** With Lemma 9.4.2 in hand, we are now in a place to prove Theorem 9.3.1. What remains is simply to bound the value of \( C(r_t) \) under different choices of \( \phi \) and \( \psi \).

1. **Polynomially weighted forecaster.** Fix any \( x \in \mathbb{R}^K \). In this case, the derivatives are given by:

\[
\psi'(x) = \frac{2}{px^p(x^p-2)/p}, \\
\phi''(x) = p(p-1)(x^p)^{p-2} \quad (9.31)
\]

We have

\[
\psi' \left( \sum_{i=1}^{K} \phi(x_i) \right) = \frac{2}{p} \left( \sum_{i=1}^{K} (x_i^p)^{-(p-2)/p} \right). \quad (9.32)
\]

Recall Hölder’s inequality, which states that for \( a, b > 1 \) with \( 1/a + 1/b = 1 \), we have

\[
\sum_{i=1}^{K} |x_i|^a y_i^b \leq \left( \sum_{i=1}^{K} |x_i|^a \right)^{1/a} \left( \sum_{i=1}^{K} |y_i|^b \right)^{1/b}, \quad x, y \in \mathbb{R}^K. \quad (9.33)
\]
We have that
\[
\sum_{i=1}^{K} \phi''(x_i) r_{i,t}^2 = p(p-1) \sum_{i=1}^{K} (x_i^+)^{p-2} r_{i,t}^2
\]
\[
\leq p(p-1) \left( \sum_{i=1}^{K} \left( (x_i^+)^{p-2} \right)^{p/(p-2)} \right) \left( \sum_{i=1}^{K} |r_{i,t}|^{p/2} \right)^{2/p}
\]
\[
= p(p-1) \left( \sum_{i=1}^{K} (x_i^+) \right)^{(p-2)/p} \left( \sum_{i=1}^{K} |r_{i,t}| \right)^{2/p}
\]
(9.34)

where we have evoked Hölder’s inequality with \( a = \frac{p}{p-2} \) and \( b = p/2 \). Putting everything together, we get
\[
C(r_t) \leq \sup_{x \in \mathbb{R}^K} \psi'(\sum_{i=1}^{K} \phi(x_i)) \sum_{i=1}^{K} \phi''(x_i) r_{i,t}^2
\]
\[
\leq \sup_{x \in \mathbb{R}^K} 2(p-1) \left( \sum_{i=1}^{K} (x_i^+) \right)^{(p-2)/p} \left( \sum_{i=1}^{K} (x_i^+) \right)^{(p-2)/p} \left( \sum_{i=1}^{K} |r_{i,t}| \right)^{2/p}
\]
\[
= \left( \sum_{i=1}^{K} |r_{i,t}| \right)^{2/p}
\]
\[
= 2(p-1) \|r_t\|_p^2.
\]
(9.35)

Evoking Lemma 9.4.2 and noting that \( V(0) = 0 \) in this case, we have
\[
\left( \sum_{i=1}^{K} (R_{i,T}^+) \right)^{2/p} = V(R_T) \leq (p-1) \sum_{i=1}^{T} \|r_t\|_p^2 \leq (p-1)TK^{2/p},
\]
(9.36)

where the last inequality follows from the fact that \( \ell \) and \( |r_t| \) by consequence, is bounded from above by 1. Finally, note that for any \( x \in \mathbb{R}^K \):
\[
\max_{i} x_i \leq \left( \sum_{i=1}^{K} (x_i^+) \right)^{1/p}.
\]
(9.37)

Using this fact, we have that
\[
\max_{i=1,...,K} R_{i,T} \leq \left( \sum_{i=1}^{K} (R_{i,T}^+) \right)^{1/p} \leq \sqrt{(p-1)K^{2/p}T}.
\]
(9.38)

This proves our claim.

2. Exponentially weighted forecaster. The analysis in this case follows the same steps as the polynomially weighted forecaster, though the expressions are significantly simpler. We have
\[
\psi'(x) = \frac{1}{\eta x}, \quad \text{and} \quad \phi''(x) = \eta^2 \exp(\eta x) = \eta^2 \phi(x).
\]
(9.39)
Moreover
\[ \phi^{-1}(\psi^{-1}(x)) = x. \] (9.40)

We have that for \( x \in \mathbb{R}^K \),
\[
\phi' \left( \sum_{i=1}^{K} \phi(x_i) \right) \sum_{i=1}^{K} \phi''(x_i) r_{i,t}^2 = \frac{1}{\eta \left( \sum_{i=1}^{K} \phi(x_i) \right)} \left( \frac{\eta^2 \sum_{i=1}^{K} \phi(x_i)}{2} \right) \]
\[ \leq \frac{1}{\eta \left( \sum_{i=1}^{K} \phi(x_i) \right)} \left( \frac{\eta^2 \sum_{i=1}^{K} \phi(x_i)}{2} \right) \max_{i=1, \ldots, K} r_{i,t}^2 \]
\[ = \eta \max_{i=1, \ldots, K} r_{i,t}^2 \leq \eta. \] (9.41)

Note that \( V(0) = \frac{1}{\eta} \ln K \). Using Lemma 9.4.2, we get
\[ V(R_T) \leq \frac{1}{\eta} \ln K + \frac{\eta T}{2}, \] (9.42)
and consequently
\[ \max_{i=1, \ldots, K} R_{i,T} \leq \phi^{-1}(\psi^{-1}(V(R_T))) = V(R_T) \leq \frac{1}{\eta} \ln K + \frac{\eta T}{2}. \] (9.43)

\[ \square \]

### 9.5 Blackwell Approachability

The concept of approachability turns out to be a crucial tool in unifying our understanding of various potential-based drift methods. The setup concerns a sequential game between two players. We will refer to them as the row and column player, respectively, and their sets of actions are finite and denoted by \( I \) and \( J \), respectively. In each time step \( t \), the row player chooses action \( i_t \in I \) and the column player \( j_t \in J \). We will assume that the sequence of actions are represented by the sequence of probability distributions \( \{p_t\}_t \) and \( \{q_t\}_t \), where \( p_t \) and \( q_t \) are probability distributions over \( I \) and \( J \), respectively. When the two players choose actions \( i \in I \) and \( j \in J \), a vector payoff \( L(i, j) \in \mathbb{R}^d \) is realized. With a slight abuse of notation, we similarly define the expected payoff as
\[ L(p,q) = p^T Lq = \sum_{i \in I, j \in J} L(i, j)p(i)q(j). \] (9.44)

\( L(p, j) = \langle p, L(\cdot, j) \rangle \) and \( L(i, q) = \langle q, L(i, \cdot) \rangle \) are defined analogously. Define the time-average payoff
\[ \overline{L}_t = \frac{1}{t} \sum_{s=1}^{t} L(I_s, J_s). \] (9.45)

We assume that the row player’s actions \( p_t \) are chosen according to a strategy; the strategy may depend on the payoff function \( L(\cdot, \cdot) \), and take as input the entire past history of previous actions \( \{(I_s, J_s)\}_{s=1, \ldots, t-1} \) when generating \( p_t \). Define the distance measure between a point and a set:
\[ d(x, \mathcal{S}) = \inf_{y \in \mathcal{S}} \|x - y\|_2, \quad x \in \mathbb{R}. \] (9.46)
Definition 9.5.1 (Approachability). Fix a set $\mathcal{A} \subset \mathbb{R}^d$, we say that the sequence of time-average payoffs $\{\bar{L}_t\}$ approaches $\mathcal{A}$ if as $t \to \infty$,

$$\lim_{t \to \infty} d(\bar{L}_t, \mathcal{A}) = 0. \quad (9.47)$$

We say that a set $\mathcal{A}$ is approachable, if for any column player $(q)$ strategy, there exists a row player $(p)$ strategy that, $L_t$ approaches $\mathcal{A}$ with probability one.

A half space $\mathcal{H}(b, c)$ is a set defined by

$$\mathcal{H}(b, c) = \{x \in \mathbb{R}^d : \langle b, x \rangle \leq c\}, \quad b \in \mathbb{R}^d, c \in \mathbb{R}. \quad (9.48)$$

We call $\{x \in \mathbb{R}^d : \langle b, x \rangle = c\}$ the hyperplane associated with $\mathcal{H}(b, c)$.

The following remarkable theorem is due to David Blackwell [Blackwell, 1956].

Theorem 9.5.1. Let $\mathcal{A}$ be a closed convex subset of $\mathbb{R}^d$. Suppose that all payoffs and $\mathcal{A}$ belong to the unit ball. The following statements are equivalent.

1. $\mathcal{A}$ is approachable.
2. For every $q$, there exists $p$ such that $L(p, q) \in \mathcal{A}$.
3. Every half space containing $\mathcal{A}$ is approachable.

Proof. We will show that $2. \iff 3.$ and $1. \iff 3.$

(a) $2. \Rightarrow 3.$ Fix a half space $\mathcal{H}(b, c)$ such that $\mathcal{A} \subset \mathcal{H}(b, c)$. Define

$$f(p, q) := \langle L(p, q), b \rangle. \quad (9.49)$$

Notice that $f(p, q)$ is blinear (linear in each of the two arguments, respectively). Fix $q \in J$. By assumption, there exists $p$ such that $L(p, q) \in \mathcal{A}$. In other words, we have shown that

$$\max_q \min_p f(p, q) \leq c. \quad (9.50)$$

By the minimax theorem and the bilinearity of $f$,

$$\min_q \max_p f(p, q) = \max_p \min_q f(p, q) \leq c. \quad (9.51)$$

That is, there exists $p'$ such that

$$\langle L(p', q), b \rangle \leq c, \quad \text{for all } q. \quad (9.52)$$

Suppose the row player adopts the strategy where $p_t = p'$ for all $t$. The above inequality shows that for all $t$

$$\sum_{s=1}^{t} \mathbb{E}[L(I_t, J_t)] \in \mathcal{H}(b, c). \quad (9.53)$$

We may then use a martingale concentration result to convert the above to the almost sure convergence of $\bar{L}_t$, by observing that $X_t := \langle \sum_{s=1}^{t} L(I_s, J_s), b \rangle - c$ is a supermartingale with respect to the filtration induced by $\{I_t, J_t\}_{t \in \mathbb{N}}$.
(b) \(3 \Rightarrow 2\). Suppose \(\mathcal{H}(b, c)\) contains \(\mathcal{A}\). We know that \(\mathcal{H}(b, c)\) is approachable by assumption. Then, suppose \(q_t = q\) for some \(q\). The approachability implies there exists a row player strategy under which
\[
\|\langle \bar{L}_t, b \rangle - c \| \to 0
\]
almost surely. Since payoffs are bounded, we conclude that
\[
\left\langle \mathbb{E} \left[ \sum_{s=1}^{t} L(p_s, q) \right], b \right\rangle = c + o(t).
\] (9.55)
Define \(p' := \lim_{t \to \infty} \mathbb{E} \left[ \frac{1}{t} \sum_{s=1}^{t} p_s \right]\). We have thus established that for all \(q\) and any \(\mathcal{H}(b, c)\) containing \(\mathcal{A}\), there exists \(p'\) such that
\[
L(p', q) \in \mathcal{H}(b, c).
\] (9.56)
It remains to argue that (9.56) implies our claim, that is for all \(q\) there exists \(p\) such that \(L(p, q) \in \mathcal{A}\). To show this, suppose, for the sake of contradiction that there exists \(q'\) such that \(L(p, q') \notin \mathcal{A}\) for all \(p\). Denote by \(\mathcal{S}\) the set of all expected payoffs under \(q'\):
\[
\mathcal{S} = \{ L(p, q') : p \in \Delta^{d-1} \}.
\] (9.57)
It follows immediately from the bilinearity of \(L(\cdot, \cdot)\) that \(\mathcal{S}\) is convex. Our assumption implies that
\[
\mathcal{S} \cap \mathcal{A} = \emptyset,
\] (9.58)
which, combined with the convexity of \(\mathcal{S}\) implies that that there exists a half space \(\mathcal{H}\) such that (illustrated in Figure 9.1)
\[
\mathcal{A} \subset \mathcal{H}, \quad \text{and} \quad \mathcal{S} \cap \mathcal{H} = \emptyset.
\] (9.59)
The latter claim leads to a contraction with (9.56). We have thus proven our claim.

(c) \(1 \Rightarrow 3\). Note that for two sets \(\mathcal{S}_1 \subset \mathcal{S}_2\), we have that
\[
d(x, \mathcal{S}_2) \leq d(x, \mathcal{S}_2), \quad \forall x \in \mathbb{R}^d.
\] (9.60)
The claim immediately follows by applying this reasoning to \(\mathcal{A}\) and any half space containing \(\mathcal{A}\).

(d) \(3 \Rightarrow 1\). While step (c) is fairly straightforward, its converse is far from obvious. We will pursue a constructive proof, by creating a row player strategy that approaches \(\mathcal{A}\) under any column player strategy.
To start, a similar reasoning to (9.56) shows that a half space $H(b, c)$ is approachable if any only if there exists a probability vector $p$ such that
\[ \max_{j \in J} \langle L(p, j), b \rangle \leq c. \] (9.61)

In other words, there should exist $p$ such that no matter what the column player does, $L(p, j)$ always lies inside the half space.

Let $\pi_A(x)$ be the $L_2$ projection of $x$ onto $A$
\[ \pi_A(x) = \arg\min_{y \in A} \|x - y\|_2, \] (9.62)
which is unique when $A$ is convex. Define
\[ b_t = \frac{\bar{L}_t - \pi_A(\bar{L}_t)}{\|\bar{L}_t - \pi_A(\bar{L}_t)\|_2} \text{ and } c_t = \langle b_t, \pi_A(\bar{L}_t) \rangle \] (9.63)

where $\bar{L}_0 := 0$. In particular, if $\bar{L}_t \notin A$, $b_t$ and $c_t$ are defined such that $H(b_t, c_t)$ corresponds to the half space created by the hyper plane passing through $\pi_A(\bar{L}_t)$ and tangent to $A$ (Figure 9.2). Note that by definition
\[ A \subset H(b_t, c_t) \] (9.64)
whenever $\bar{L}_t \notin A$.

The row player strategy is defined as follows: at the start of time step $t$, if $\bar{L}_{t-1} \in A$ then the row player can choose an arbitrary action. Otherwise, the row player would choose $I_t$ sampled from $p_t$, where $p_t$ is a probability vector satisfying
\[ p_t \in \arg\min_p \max_{j \in J} \langle L(p, j), b_{t-1} \rangle. \] (9.65)

Importantly, by (9.61), we know that
\[ \max_{j \in J} \langle L(p_t, j), b_{t-1} \rangle \leq c_{t-1}. \] (9.66)

Intuitively, the objective of this choice of $p_t$ is to ensure that the one-step expected payoff $L(p_t, J_t)$ lies inside $H(b_{t-1}, c_{t-1})$, thus “nudging” $\bar{L}_t$ towards $A$ (see Figure 9.2).
Note that by definition $\bar{L}_t = \frac{t-1}{t} \bar{L}_{t-1} + \frac{1}{t} L(I_t, J_t)$. We have that

$$d(\bar{L}_t, \mathcal{A})^2 = \| \bar{L}_t - \pi_\mathcal{A}(\bar{L}_t) \|^2_2$$

$$\leq \| \bar{L}_t - \pi_\mathcal{A}(\bar{L}_{t-1}) \|^2_2$$

$$= \left\| \frac{t-1}{t} \bar{L}_{t-1} + \frac{L(I_t, J_t)}{t} - \pi_\mathcal{A}(\bar{L}_{t-1}) \right\|^2_2$$

$$= \left\| \frac{t-1}{t} (\bar{L}_{t-1} - \pi_\mathcal{A}(\bar{L}_{t-1})) + \frac{1}{t} (L(I_t, J_t) - \pi_\mathcal{A}(\bar{L}_{t-1})) \right\|^2_2$$

$$= \left( \frac{t-1}{t} \right)^2 d(\bar{L}_{t-1}, \mathcal{A})^2 + \frac{1}{t^2} \| L(I_t, J_t) - \pi_\mathcal{A}(\bar{L}_{t-1}) \|^2_2$$

$$+ 2 \frac{t-1}{t^2} \langle \bar{L}_{t-1} - \pi_\mathcal{A}(\bar{L}_{t-1}), L(I_t, J_t) - \pi_\mathcal{A}(\bar{L}_{t-1}) \rangle$$

$$= \left( \frac{t-1}{t} \right)^2 d(\bar{L}_{t-1}, \mathcal{A})^2 + \frac{1}{t^2} \| L(I_t, J_t) - \pi_\mathcal{A}(\bar{L}_{t-1}) \|^2_2$$

$$+ 2 \frac{t-1}{t^2} d(\bar{L}_{t-1}, \mathcal{A}) \langle L(I_t, J_t) - \pi_\mathcal{A}(\bar{L}_{t-1}), b_{t-1} \rangle.$$  \hspace{1cm} (9.67)

**Some intuition...** Ideally, we would like the above equation to show that the distance between $\bar{L}_t$ and $\mathcal{A}$ shrinks as $t \to \infty$. Before getting into more algebra, let us take a break and look at each of the three terms and see how they stack up against this objective:

1. $\left( \frac{t-1}{t} \right)^2 d(\bar{L}_{t-1}, \mathcal{A})^2$: this term shows a multiplicative shrinkage of the previous distance. We will show that this speed of shrinkage is sufficient for us.

2. $\frac{1}{t^2} \| L(I_t, J_t) - \pi_\mathcal{A}(\bar{L}_{t-1}) \|^2_2$: this term decays at speed $O(1/t^2)$ due to the boundedness of the payoffs, so it shouldn’t be a problem.

3. $2 \frac{t-1}{t^2} d(\bar{L}_{t-1}, \mathcal{A}) \langle L(I_t, J_t) - \pi_\mathcal{A}(\bar{L}_{t-1}), b_{t-1} \rangle$: note that by the design of our policy (see Figure 9.2)

$$\langle L(p_t, J_t) - \pi_\mathcal{A}(\bar{L}_{t-1}), b_{t-1} \rangle \leq 0.$$  \hspace{1cm} (9.68)

This is because $L(p_t, J_t)$ and $\bar{L}_{t-1}$ lie on opposite sides of the hyperplane defining $\mathcal{H}(b_{t-1}, c_{t-1})$. Granted that in this inequality we have changed $I_t$ to its distribution $p_t$, the above analysis suggests that this last term should be negative in expectation (with respect to $p_t$). This suggests that we may be able to use a martingale-based concentration inequality down the road to bound the impact of this term.

Back to the proof. We now put everything together. Since all payoffs and $\mathcal{A}$ belong to the unit ball, we may assume $\| L(I_t, J_t) - \pi_\mathcal{A}(\bar{L}_{t-1}) \|^2_2 \leq 4$. From (9.67) we obtain

$$t^2 d(\bar{L}_t, \mathcal{A})^2 - (t - 1)^2 d(\bar{L}_{t-1}, \mathcal{A})^2$$

$$\leq 4 + 2(t - 1) d(\bar{L}_{t-1}, \mathcal{A}) \langle L(I_t, J_t) - \pi_\mathcal{A}(\bar{L}_{t-1}), b_{t-1} \rangle.$$  \hspace{1cm} (9.69)

Summing both sides of the inequality above over $t = 1, \ldots, T$, and defining

$$K_t := \frac{t}{T} d(\bar{L}_t, \mathcal{A}),$$  \hspace{1cm} (9.70)
we get
\[
d(L_T, A)^2 \leq \frac{4}{T} + \frac{2}{T} \sum_{t=1}^{T} K_t \left\langle L(I_t, J_t) - L(p_t, J_t), b_t - b_{t-1} \right\rangle
\]
(9.71)
The last step follows from the definition of \(p_t\). In particular, note that \(\pi_A(\tilde{L}_{t-1})\) lies on the hyperplane \(\{x : \langle x, b_{t-1} \rangle = c_{t-1}\}\) by definition, and hence \(\langle \pi_A(\tilde{L}_{t-1}), b_{t-1} \rangle = c_{t-1}\). On the other hand, by (9.66) we have that \(\max_{j \in J} \langle L(p_t, j), b_{t-1} \rangle \leq c_{t-1}\). Together they imply that with probability one
\[
\langle \pi_A(\tilde{L}_{t-1}), b_{t-1} \rangle \geq \langle L(p_t, J_t), b_{t-1} \rangle.
\]
(9.72)
Finally, note that \(K_t\) is at most 2 for all \(t = 1, \ldots, T\). Therefore, the term second term on the right-hand side of (9.71) corresponds to the average of a bounded zero-mean martingale difference sequence. Using Azuma-Hoeffding inequality and the Borel-Cantelli lemma, we can then show that
\[
d(L_T, A)^2 \to 0 \text{ almost surely as } T \to \infty.
\]
This shows the approachability of \(A\), thus proving our claim.

9.6 A Unified View through Approachability

The concept of approachability provides a unified interpretation of some of important problems we have encountered in stochastic networks and online learning. In the next two sub-sections, we will show that in some sense both Max-Weight and the WAF algorithms can be viewed as special cases of the Blackwell approachability theorem.

To make these connections more evident, we begin by considering a slight generalization of Blackwell approachability to a broader family of distance functions. Fix a convex set \(A \subset \mathbb{R}^d\). Let \(V : \mathbb{R}^d \to \mathbb{R}\) be a nonnegative, convex and twice differentiable Lyapunov function, such that \(V(x) = 0\) for all \(x \in A\). Define the Bergman divergence:
\[
D_V(y, x) := V(y) - V(x) - \langle \nabla V(x), y - x \rangle.
\]
(9.73)
We will define the distance between a point \(x\) and a set \(S\) with respect to the Bergman divergence \(D_V(.)\):
\[
d(x, S) = \inf_{y \in S} \sqrt{D_V(y, x)}
\]
(9.74)
In particular, we have replaced the Euclidean distance in the original definition of approachability with \(D_V\). This is a generalization, in the sense that if we let
\[
V(x) = \|x\|^2_2 \mathbb{1}\{x \notin A\},
\]
(9.75)
we recover the same distance measure as before.
Under the new distance measure $D_V$, we have that

\[
\pi^V_A(x) = \arg\min_{y \in A} D_V(x, y) \\
= \arg\min_{y \in A} (V(y) - V(x) - \langle \nabla V(x), y - x \rangle) \\
= \arg\min_{y \in A} (V(y) - \langle \nabla V(x), y - x \rangle) \\
= \arg\max_{y \in A} \langle \nabla V(x), y - x \rangle,
\]

where the last equality is due to the fact that $V(y) = 0$ for all $y \in A$. Projection of this type is also known as the Bergman projection.

Figure 9.3: Illustration for the design of a row player strategy under the distance measure derived from Bergman divergence.

You may ask why bother making the problem more complex? After all, changing the distance measure from $L_2$ to $D_V$ does not fundamentally change the approachability of a certain set, at least for most “nice” Lyapunov functions, since if two points are very close according to one distance measure then they are probably also quite close under another. Indeed, but what this generalization does afford us is more flexibility on the algorithmic side: by using different Lyapunov functions $V$, the same “meta-strategy” of approaching a target set as described in the original proof of the approachability theorem could now give rise to a family of different row strategies, each with potentially different convergence properties.

Let us see this in action. Recall from the proof of Theorem 9.5.1, we used the following row player strategy by setting:

\[
p_t \in \arg\min_p \max_j \langle L(p, j), b_t, j \rangle,
\]
where
\[ b_t = \frac{L_t - \pi_A(L_t)}{\|L_t - \pi_A(L_t)\|_2} \text{ and } c_t = \langle b_t, \pi_A(L_t) \rangle. \] (9.78)

In the generalized version, all we will do is to suitably generalize the definition of \( b_t \) and \( c_t \), by incorporating the geometry induced by \( D_V \), as follows:
\[ b_t = \frac{\nabla V(L_t)}{\|\nabla V(L_t)\|_2} \text{ and } c_t := \sup_{y \in A} \langle b_t, y \rangle = \langle b_t, \pi_A(L_t) \rangle. \] (9.79)

In particular, the norm vector defining the half space \( H(b_t, c_t) \) points along the direction of \( \nabla V(L_t) \), and \( c_t \) is chosen such that the hyperplane passes through \( \pi_A(L_t) \). It is easy to show that such a hyperplane is tangent to \( A \). Note that in the special case of \( V \) being the squared \( L_2 \) norm,
\[ \nabla V(L_t) \propto L_t - \pi_A(L_t) \] (9.80)
so we recover the original definition.

With the new definition of \( b_t \), we have now arrived at a crucial row player strategy:
\[ p_t \in \arg\min_p \max_j \langle L(p, j), \nabla V(L_{t-1}) \rangle, \] (9.81)

Viewing \( \nabla V(L_{t-1}) \) as the direction of the steepest ascent in \( V \), we see that the row player is essentially trying to minimize the worst-case positive drift in the Lyapunov function.

The following theorem states that the strategy in (9.81) is able to approach any approachable set, and furthermore, provides a quantitative bound on the rate of convergence. The theorem is stated for a slightly different, “expected” version of \( \bar{L}_t \). See Section 7.8 of [Cesa-Bianchi and Lugosi, 2006] for a proof.

**Theorem 9.6.1.** Let \( A \) be a closed, convex and approachable subset of the unit ball in \( \mathbb{R}^d \). Let \( V : \mathbb{R}^d \to \mathbb{R} \) be a nonnegative, convex and twice differentiable Lyapunov function, such that \( V(x) = 0 \) for all \( x \in A \) and positive outside of \( A \). Denote by \( H_V(x) \) the Hessian of \( V \) at \( x \). Let \( B := \sup_{x: \|x\|_2 \leq 1} \|H(x)\|_2 \) be the maximum operator norm of the Hessian in the unit ball, and suppose that \( B < \infty \). Let
\[ \bar{L}_t = \frac{1}{t} \sum_{s=1}^{t} L(p_s, J_s). \] (9.82)

Then, under the row player strategy where
\[ p_t \in \arg\min_p \max_j \langle L(p, j), \nabla V(L_{t-1}) \rangle, \] (9.83)
we have that, under any column player strategy,
\[ V(\bar{L}_t) \leq \frac{2B(ln t + 1)}{t}, \quad \forall t \in \mathbb{N}. \] (9.84)

Furthermore, the average payoff approaches \( A \):
\[ \lim_{t \to \infty} \inf_{y \in A} \left\| \frac{1}{t} \sum_{s=1}^{t} L(I_s, J_s) - y \right\|_2 = 0, \quad \text{almost surely.} \] (9.85)
9.6.1 Maximum stability and Max-Weight algorithm

Let us now look at how the notion of maximum stability in a stochastic network can be interpreted as a special case of Blackwell approachability. To do so, let us look at a slightly modified scheduling problem to make this link more obvious. We make the following substitutions:

1. Dimension $d$: number of queues.
2. Row player action set $I$: the set of schedules or potential departures, $D$.
3. Column action set $J$: set of allowable expected arrival rate vector in each time slot. For now, let us suppose $J$ is finite.
4. Vector payoff function:

$$L(i, j) = j - i, \quad i \in I, j \in J. \quad (9.86)$$

That is $L(i, j)$ is the expected net change in queue lengths when the arrival rate is $j$ and potential departure vector is $i$.

For simplicity, let us ignore for now the boundary effect where queues cannot go negative once reaching zero, so that the queue lengths is simply the cumulative payoffs. In this formulation, $L_t$ is the time-average net discrepancy between arrival and potential departures. If for some coordinate $k$, $\lim_{t \to \infty} (\bar{a}_t)_k > 0$, then we know that the length of queue $k$ must be diverging to infinity at rate $\mathcal{O}(t)$. That is, the system cannot possibly be stable if any coordinate of $\bar{a}_t$ remains positive as $t \to \infty$. This naturally leads to the following definition of the target set $A$, the negative orthant:

$$A = \{x \in \mathbb{R}^d : x \leq 0\}. \quad (9.87)$$

That is, if $\bar{L}_t$ approaches $A$, we know that across all queues, the long-term departure rate should be no less than the long-term arrival rate. If we can show that $A$ is approachable, then we have effectively shown that a maximumly stable scheduling policy (under this slightly altered / weakened stability criterion) exists.

**Existence of maximum stable policy** We now evoke Theorem 9.5.1 to immediately conclude that a maximum stable scheduling policy exists. Suppose that the set of arrival rate vectors $J$ can be served by a combination of schedules, that is, for any $\lambda \in J$, there exists $\mu \in \text{Conv}(J)$ such that

$$\lambda \leq \mu. \quad (9.88)$$

This immediately implies that for any distribution $q$ over $J$, there exists a distribution $p$ over $I$ such that

$$L(p, q) \leq 0, \text{ or, equivalently, } L(p, q) \in A. \quad (9.89)$$

The above statement is condition #2 of the Blackwell Approachability Theorem, and we thus conclude that the set $A$ is approachable. This shows that there exists a maximumly stable scheduling policy.

**Using approachability to obtain Max-Weight** We can go one step further. Beyond just saying that the negative orthant is approachable, we know what kind of row player strategy does the job! Let $V$ be a suitable Lyapunov function, and let us consider the row player strategy as per Theorem 9.6.1

$$p_t \in \arg\min_p \max_{j \in J} \langle L(p, j), \nabla V(\bar{L}_{t-1}) \rangle, \quad (9.90)$$
CHAPTER 9. POTENTIAL FUNCTIONS IN ONLINE LEARNING

Recall that in this model
\[ Q(t) = t\bar{L}_t, \]
\[ L(i,j) = j - i. \]
and define the average departure under the randomized action \( p \):
\[ \mu_p = \sum_{i \in I} i p_i = \mathbb{E}_p[I]. \]  
(9.91)

We have that
\[ p_t \in \arg\min_{p} \max_{j \in J} \langle L(p,j), \nabla V(\bar{L}_{t-1}) \rangle \]
\[ = \arg\min_{p} \max_{j \in J} \left( j - \mu_p, \nabla V(\bar{L}_{t-1}) \right) \]
\[ = \arg\min_{p} \left( -\langle \mu_p, \nabla V(\bar{L}_{t-1}) \rangle - \max_{j \in J} \langle j, \nabla V(\bar{L}_{t-1}) \rangle \right) \]
\[ = \arg\max_p \langle \mu_p, \nabla V(\bar{L}_{t-1}) \rangle \]
\[ = \arg\max_p \left( \mu_p, \nabla V(\frac{1}{t-1}Q(t-1)) \right) \].  
(9.92)

Most remarkably, because \( L(i,j) \) is linear, the maximization step by the column player \( \max_{j \in J} \) has no influence on the consideration of the column player when it comes to choosing \( p_t \). In fact, the optimal \( p_t \) does not even depend on the structure of the column player action set \( J \).

Under this choice of \( p \), we see that the resulting expected departure \( \mu_p \) satisfies:
\[ \mu_{p_t} \in \arg\max_{\mu \in \Pi} \left( \mu, \nabla V\left(\frac{1}{t-1}Q(t-1)\right)\right), \]  
(9.93)
where \( \Pi \) is the maximum stability set \( \Pi = \{ \lambda : \exists s \in \text{Conv}(D), s \geq \lambda \} \). If you think the above equation looks familiar, you are not alone. Suppose \( V \) is the quadratic Lyapunov function and \( Q(t-1) > 0 \), then we obtain
\[ \mu_{p_t} \in \arg\max_{\mu \in \Pi} \left( \mu, \nabla V\left(\frac{1}{t-1}Q(t-1)\right)\right) = \arg\max_{\mu \in \Pi} \langle \mu, Q(t-1) \rangle, \]  
(9.94)
which is precisely the Max-Weight algorithm! More generally, if \( V \) is scale-invariant in the sense that
\[ V(ax) = f(a)V(x), \quad a \in \mathbb{R}_+, x \in \mathbb{R}^d, \]  
(9.95)
for some function \( f(a) \). Then, we obtain a generalized family of Max-Weight algorithms, where
\[ \mu_{p_t} \in \arg\max_{\mu \in \Pi} \langle \mu, \nabla V(Q(t-1)) \rangle. \]  
(9.96)

9.6.2 Weighted average forecasting in online prediction

We now apply Blackwell approachability to the problem designing weighted average forecasters in online prediction.
Randomized Online Prediction

We will consider a slightly different version of the online prediction problem, which can be viewed as a special case of the problem of prediction with expert advice. In this version, the forecaster aims to predict an unknown sequence of vectors. Instead of choosing a prediction by combining those from a set of experts, the forecaster will instead choose one $I_t$ from a finite set $I$. The accuracy of a prediction $i$ is measured by the loss function $\ell$ as $\ell(i, y)$. Furthermore, instead of outputting a prediction that is a deterministic function of history (such as the WAFs defined earlier), we will allow the forecaster to randomize, so that in each time slot, the forecaster chooses a probability vector $p_t$ over $I$, and the prediction $I_t$ is sampled according to the distribution associated with $p_t$. Finally, the regret is measured by comparing the forecaster’s cumulative expected loss against the best single vector in hindsight, where the cumulative regret vector is now defined as:

$$R_{i,t} = \sum_{s=1}^{t} (\ell(I_s, y_s) - \ell(i, s)), \quad (9.97)$$

and we may also define a related notion of expected regret:

$$R_{i,t} = \sum_{s=1}^{t} (\ell^E(p_s, y_s) - \ell(i, s)), \quad (9.98)$$

where $\ell^E$ indicates the expected loss where the expectation is taken with respect to $p_t$:

$$\ell^E(p_t, y) := \sum_{i \in I} p_{i,t} \ell(i, y). \quad (9.99)$$

**Remark 9.6.2 (Online prediction as a special PEA).** To see how the above model is a special case of the problem of prediction with expert advice (PEA), we will show how the online prediction problem can be transformed into an instance of PEA. The decision space $D$ is equal to the probability simplex over $I$. For any $p \in D$ and $y \in Y$, with the loss function $\ell^E$. Furthermore, we will fix the prediction of the $i$th expert $f_{i,t}$ to be the point probability measure concentrated on $i \in I$, for all $t$:

$$f_{i,t} = \delta_i \in D. \quad (9.100)$$

The instantaneous regret in this PEA is defined by:

$$r_{i,t}^E = \ell^E(\hat{p}_t, y_t) - \ell^E(\delta_i, y_t) = \ell^E(\hat{p}_t, y_t) - \ell(i, y). \quad (9.101)$$

This means that the loss under the PEA problem is equal to the expected loss in the online prediction problem. Moreover, $\ell^E$ is linear in its first argument and hence always convex, so all of our guarantees on the PEA regret can be applied here. This reduction shows that any online prediction problem can be solved for an effective PEA algorithm, with the same guarantees applied to the expected regret.

**Remark 9.6.3 (Some interesting subtitles).** We note here a few more interesting subtitles with the online prediction formulation:

1. By the reduction above, the strength of this formulation is that we no longer require the original loss function $\ell$ to be convex: the expectation with respect to $p_t$ takes care of this problem, since the resulting expected loss $\ell^E(p, y)$ is always convex in $p$. 


2. Because the forecaster’s actions are now randomized, it opens the possibility that the nature could choose \( y_t \) based on the forecaster’s past prediction, which means that \( y_t \) could also be random themselves. Surprisingly, it turns out that whether or the sequence is allowed to depend on past forecasts has little effect on the worst-case performance of the forecaster. In the example with PEA above, we had assumed that \( y_t \) is deterministic, but in the subsequent development we will generalize so as to allow it to be random.

3. We have shown that the online prediction problem is a special case of the PEA. It would seem that the converse is not true: the forecaster in an online prediction problem is compared against a fixed prediction, where as in the PEA the forecaster’s benchmark is an expert who may produce time-varying prediction. So one would expect that we can do better in the online prediction problem. Surprisingly, this is actually not true. This is because the loss function in the online prediction problem is arbitrary. In particular, let \( \ell \) be a loss function in a PEA problem, then we may reduce it to an online prediction problem with the loss function \( \ell' \)

\[
\ell'(i, y'_t) = \ell'(f_{i,t}, y_t).
\]

(9.102)

where \( y'_t = (f_t, y_t) \). That is, we can simply encode the expert predictions inside the value of \( y_t \) in an online prediction problem and thus make the problem just as hard.

**Existence of forecaster with sub-linear regret.** Let us see how to cast online prediction as a question of approachability, as follows.

1. Dimension \( d \): number of feasible predictions that the forecaster is allowed to choose from.
2. Row player action set \( \mathcal{I} \): the set of feasible predictions. The row player is forecaster.
3. Column action set \( \mathcal{J} \): possible values of the target sequence. The column player is nature who chooses the sequence value.
4. Vector payoff function: one-step regret, where

\[
L(i, j)_k = \ell(i, j) - \ell(k, j), \quad k \in \mathcal{I}.
\]

(9.103)

where \( \ell : \mathcal{I} \times \mathcal{J} \to \mathbb{R} \) is the loss function. That is, \( L(i, j)_k \) is the regret incurred by choosing prediction \( i \) compared to an alternative prediction \( k \).

In this setting, our goal is for the maximum regret across all feasible predictions to grow sub-linearly. That is, we want to achieve

\[
\lim_{t \to \infty} \max_i \hat{L}_{i,t} = 0.
\]

(9.104)

This objective leads to the same target set as we had seen in the stochastic network model just now:

\[
\mathcal{A} = \{ x \in \mathbb{R}^d : x \leq 0 \}.
\]

(9.105)

To see whether we can find a prediction strategy with a sub-linear regret, we can use Theorem 9.5.1 to verify that \( \mathcal{A} \) is approachable. Fix probability vectors \( p \) and \( q \) over \( \mathcal{I} \) and \( \mathcal{J} \), respectively.

\[
L(p, q)_k = \sum_{i,j} p_i q_j (\ell(i, j) - \ell(k, j)) = \ell(p, q) - \ell(k, q),
\]

(9.106)
where \( \ell(p, j) := \sum_i p_i \ell(i, j) \). For any \( q \), let \( k^* = \arg \min_k \ell(k, q) \), and \( p^* \) be the probability vector that puts unit mass on the vector \( k^* \) and zero everywhere else. We then have
\[
L(p^*, q) = (\min_i \ell(i, q)) - \ell(k, q) \leq 0, \quad \forall k \in \mathcal{I},
\]
and hence \( L(p^*, q) \in \mathcal{A} \). This verifies condition \#2 in Theorem 9.5.1, showing that \( \mathcal{A} \) is approachable.

**Interpretation of the Lyapunov-based WAF.** Viewing the online prediction problem as an approachability problem also helps to clarify the nature of the weighted average forecasters that we had seen earlier. There is a direct analogue of the WAF in the online prediction setting, where the only difference is that instead of averaging over the expert predictions, we use the same weights as a probability vector and sample a random prediction according to that. In particular, let \( \phi, \psi \) and \( V \) be defined as in Definition 9.3.1. In each round, we would set \( I_t = i \) with probability
\[
p_{i,t} = \frac{\nabla V(R_{t-1})_i}{\|\nabla V(R_{t-1})\|} = \frac{\phi'(R_{i,t-1})}{\sum_{j=1}^d \phi'(R_{j,t-1})}.
\]

It is easy to draw the analogy with the approachability setting: if we simply replace \( R_t \) in the with \( L_t = \frac{1}{L} R_t \), then we will have arrived at the strategy for the row player:
\[
p_{i,t} = \frac{\nabla V(L_{t-1})_i}{\|\nabla V(L_{t-1})\|}.
\]

Let us compare WAF-inspired row player strategy in (9.109) to the row player strategy in (9.81):
\[
p_t \in \arg\min_p \max_{j \in \mathcal{J}} \langle L(p, j), \nabla V(L_{t-1}) \rangle.
\]

A crucial difference between the two is that solving the above optimization problem requires the full structure of the payoff function \( L \), while the WAF strategy only depends on \( L \) through the cumulative average payoff up to time \( t - 1 \). Therefore, in general the WAF strategy does not achieve the optimal value of (9.110).

Remarkably though, even though WAF is not directly solving (9.110), it turns out that it is sufficiently powerful to ensure that the first-order drift in the Lyapunov function is non-positive. In particular, fix \( x \in \mathbb{R}^d \) and suppose that \( p_t = \frac{\nabla V(x)}{\|\nabla V(x)\|} \), then for any \( j \)
\[
\langle L(p_t, j), \nabla V(L_{t-1}) \rangle = \sum_{k=1}^d \left( \sum_{i=1}^d p_{i,t} \ell(i, j) - \ell(k, j) \right) \nabla V(x)_k
= \left( \|\nabla V(x)\| \sum_{i=1}^d \frac{\nabla V(x)_i}{\|\nabla V(x)\|} \ell(i, j) \right) - \sum_{k=1}^d \nabla V(x)_k \ell(k, j)
= \sum_{i=1}^d \nabla V(x)_i \ell(i, j) - \sum_{k=1}^d \nabla V(x)_k \ell(k, j)
= 0.
\]

That is, we know that
\[
\max_{j \in \mathcal{J}} \langle L(p_t, j), \nabla V(x) \rangle = 0.
\]

Note that this is exactly the Blackwell condition that we encountered back in Lemma 9.4.1! Interestingly, here the online prediction formulation no longer requires that the loss function be convex in the first argument, and more over, the above property holds for any \( V \) with positive gradients.
9.7 Notes

The materials on online prediction models and most of the exposition on approachability theorems are based on the wonderful text [Cesa-Bianchi and Lugosi, 2006], though claim #2 of the Blackwell theorem is not treated there. The connection between Max-Weight and Blackwell approachability is based on the exposition by Neil Walton [Walton, 2020], and the approachability theorem was stated and proved in the seminal paper of David Blackwell [Blackwell, 1956], where the adaptive row player strategy as outlined here was first introduced. The Lyapunov-based generalization of approachability in Theorem 9.6.1 is due to [Hart and Mas-Colell, 2000].
Chapter 10

Drift Analysis in Bayesian Bandits with Information Ratio

In this chapter, we will turn our attention to another well known online learning problem, known as the multi-arm bandit problem. Recall that in the online prediction with expert advice model, the decision maker has access to the predictions made by all experts after each time slot. This type of model where the regret vector is fully observed by the decision maker is often referred to as full information. In the multi-arm bandit model, this assumption of full information will be relaxed, and the decision maker will only be able to observe the effect of the action that was chosen. The objective here again is to minimize the regret, defined as the performance gap with respect to the best single action in hindsight. Within the world of multi-arm bandit problems, we will further focus our attention on a family of models called the Bayesian stochastic bandits, to be defined shortly.

This chapter stands apart from our journey so far, in terms of the way in which drift method is evoked. Previously in the PEA problem, we have seen how drift analysis and Lyapunov functions were used to both design forecasting algorithms, as well as in the analysis of their performance. The key idea there was to apply the Lyapunov function on the regret vector, and subsequently use bounds on the drift of the Lyapunov function to arrive at an upper bound on the growth rate of regret. In some sense, this line of reasoning is somewhat “classical” and one sees a strong parallel to those in the original control theory framework of Lyapunov, as well as the stochastic stability analysis of queueing systems. All of these analyses are centered around showing that the one-step drift in the Lyapunov function is small, thus culminating in a small Lyapunov function as time tends to infinity.

The drift method used in this chapter will behave very differently from the above line of attack. The idea of small or negative one-step drift will not play a role at all in the success in our algorithm. In fact, the Lyapunov function will not be used on the regret process at all, but instead on a certain function of the optimal action. This is perhaps enough spoiler to get you interested, and we shall contrast these different uses of the drift method once we have seen how it is used in the Bayesian stochastic bandit problem.

10.1 Bayesian Stochastic Multi-arm Bandit

In a multi-arm bandit problem consists of the following elements:

1. A player that operates in discrete time \( t \in \mathbb{N} \).
2. A finite set of \( K \) arms, indexed by the set \( A = \{1, \ldots, K\} \).
3. Each arm \( a \in A \) is associated with it a reward distribution, \( \gamma_a \), which is a probability measure over \( \mathbb{R}^+ \). For simplicity, we will assume that the support of \( \gamma_a \) lies in the unit interval \([0, 1]\). We will denote by \( \mu_a \) the expectation of the random variable associated with \( \gamma_a \).

4. Let \( \{ R_t \}_{t \in \mathbb{N}} \) be a sequence of random vectors, where \( R_t = (R_{a,t})_{a \in A} \), and
\[
\mathbb{P}(R_{a,t} \in \cdot) = \gamma_a(\cdot),
\]
(10.1)
In particular, \( R_{a,t} \) represents the reward the player would have obtained if they choose arm \( a \) at time \( t \). For any fixed \( a \) and \( t \), the realization of \( R_t \) is assumed to be independent from all other parts of the system. That is, in this case, the randomness in \( R_{a,t} \) is solely due to a single random draw from the measure \( \gamma_a \).

5. We assume that the player is causal in the sense that \( A_t \) is only a function of the past observations and actions. Denote by \( H_t \) the history of the system up to time \( t - 1 \),
\[
H_t = ((A_1, R_{A_1, 1}), \ldots, (A_{t-1}, R_{A_{t-1}, t-1})),
\]
(10.2)
with \( H_1 := \emptyset \), and denote by \( \mathcal{F}_t \) the \( \sigma \)-algebra generated by \( H_t \). That is, \( H_t \) captures the player’s information at the start of the \( t \)th time step. We assume that \( A_t \) is a random variable that is measurable with respect to \( H_t \) for all \( t \in \mathbb{N} \). More formally, the player’s behavior can be modeled by an algorithm, \( \pi \), which is a deterministic mapping that generates \( A_t \) as a function of the history and any internal randomness:
\[
A_t = \pi(H_t, Z_t),
\]
(10.3)
where \( \{ Z_t \} \) is a sequence of i.i.d. uniform random variables that captures any internal randomization that \( \pi \) employs.

6. Finally, we assume that the set of reward distributions \( \{ \gamma_a \}_{a \in A} \) is parameterized by \( \theta \in \Theta \), and that before the interaction begins, \( \theta \) is drawn randomly from a prior distribution \( \eta \) over \( \Theta \). After \( \theta \) is drawn, it will be fixed throughout, and the reward distributions will be given by \( \{ \gamma^\theta_a \}_{a \in A} \).

For any \( \theta \), denote by \( A^*(\theta) \) the arm with the maximum expected reward under \( \theta \), with ties broken in an arbitrary and deterministic manner:
\[
A^*(\theta) = \arg\max_{a \in A} \mathbb{E}_\theta [R_{a, 1}] .
\]
(10.4)
When the context is clear, we will omit the dependence on \( \theta \) and write simply \( A^* \), with the understanding that \( A^* \) is always a deterministic function of \( \theta \).

We define the Bayesian regret (or simply regret for short) of the player as the expected difference between the player’s cumulative rewards versus that of the best arm in hindsight.
\[
\mathcal{R}(T; \pi) = \mathbb{E} \left[ \sum_{t=1}^T (R_{A^*(\theta), t} - R_{A_t, t}) \right] ,
\]
(10.5)
where the expectation is taken with respect to the randomness in the parameter \( \theta \), \( \{ \gamma_a \} \), and the internal randomness employed by the algorithm, \( \{ Z_t \} \).

Just like in the online prediction problem, the player’s goal in a multi-arm bandit problem is to identify an algorithm \( \pi \) that achieves sub-linear Bayesian regret (in \( T \)), which also scales favorably on the number of arms \( K \).
10.2 Thompson Sampling

There are many algorithms that will be able to achieve sub-linear regret for the Bayesian bandit problem described earlier. Our interest, however, will be focused on a particularly elegant heuristic, known as the Thompson sampling algorithm, or posterior-sampling, due to William R. Thompson [Thompson, 1933]. The original algorithm was developed for what we might refer to today as a two-arm Bayesian stochastic bandit problem.

For any \( t \in \mathbb{N} \), denote by \( \eta_t \) the posterior probability distribution of \( \theta \) conditional on all past observations:

\[
\eta_t(\cdot) = \mathbb{P}(\theta \in \cdot | \mathcal{H}_t),
\]

where \( \eta_1 \) is equal to the prior distribution \( \eta \).

The Thompson sampling algorithm is an exceedingly simple procedure: for each \( t \), perform the following

1. Sample \( \theta_t \) from the posterior distribution \( \eta_t \).
2. Choose \( A_t \) to be the optimal arm pretending that \( \theta_t \) is the ground truth for \( \theta \). That is, we let \( A_t = A^*(\theta_t) \).
3. Repeat.

There is a useful, alternative description of the above procedure. Instead of sampling \( \theta_t \), we might as well directly sample from the posterior distribution of the optimal arm \( A^*(\theta) \). Define \( \xi_t \) to be the posterior distribution of \( A^* \) at the start of time step \( t \):

\[
\xi_t(\cdot) = \mathbb{P}(A^*(\theta) \in \cdot | \mathcal{H}_t).
\]

Then, steps 1 and 2 of the above procedure can be condensed into one: “Sample \( A_t \) from the posterior distribution \( \xi_t \). Play \( A_t \).”

We will denote by \( \pi_{TS} \) the Thompson sampling algorithm.

10.3 Regret Upperbound under Thompson Sampling

The machinery to be developed shortly will allow us to prove the following result.

**Theorem 10.3.1.** Fix \( K \) and \( \eta \). Denote by \( H(A^*(\theta)) \) the entropy of \( A^*(\theta) \). Then

\[
\mathcal{R}(T, \pi_{TS}) \leq \sqrt{H(A^*(\theta))KT/2}, \quad T \in \mathbb{N}.
\]

Moreover, because \( H(A^*(\theta)) \leq \log K \) under any prior, we also have that

\[
\mathcal{R}(T, \pi_{TS}) \leq \sqrt{(K \log K)T/2}, \quad T \in \mathbb{N}.
\]

The first inequality in Theorem 10.3.1 was first established in [Russo and Van Roy, 2016]. Note that the bounds in (10.10) does not depend on the prior distribution \( \eta \). For this class of problems, it is known that for any \( K \) and \( T \), there exists a prior distribution under which any algorithm would suffer a regret of at least \( \frac{1}{20} \sqrt{KT} \) [Bubeck and Liu, 2013]. In this sense, the upper bound in (10.10) matches this lower bound up to a factor of \( \sqrt{\log K} \). This logarithmic factor can be removed by using a different Lyapunov function [Lattimore and Szepesvári, 2019].
types of analyses that could lead to the type of regret bounds in (10.10). Our main objective in this
chapter is more than just to show that Theorem 10.3.1 holds, but rather how to prove it using a novel
style of drift analysis, pioneered by [Russo and Van Roy, 2014].

In what follows, we will focus on the Thompson sampling algorithm only, and thus suppress \( \pi_{TS} \)
from our notation.

## 10.4 Coupling to an Information Piggy Bank

The crux of the idea is rooted in an understanding of the information dynamics induced by the
Thompson sampling algorithm. Since we would like to derive an upper bound on the regret, we would
like to control how often arms of low expected rewards are chosen by the player. Unfortunately, this
in general is fairly complex task to tackle head-on, especially given that we have not even endowed
the family of reward distributions with any structure! However, the following observation is going to
help us: whenever a very bad arm is chosen, say arm \( a' \), we will likely know that this has occurred by
simply observing that the resulting reward is rather small.

This seemingly trivial observation is in fact crucial. More formally, we are claiming that, because
the player is fully aware of the prior distribution, if we happen to incur relatively large regret in a
single time step, it must imply that in the same time step, we gained useful information about the
optimal arm \( A^* \), by noticing, for instance, that the arm we had just chosen is probably not \( A^* \).

This observation inspires a “coupling” approach, where we couple the evolution of the regret to
that of a certain information process. If every time we incur regret, we are always forced to deposit
some “information” about the location of the best arm into an “information piggy bank”, then, to
get a sense of how much regret has been accrued, it suffices for us to simply check the amount of
savings in the information piggy bank (which, hopefully, is easier to do!). To put this logic into
pseudo-equations, let us define \( F_t \) loosely as

\[
V_t = \text{“total amount of information learned about } A^* \text{ by time } t."
\]

Then, we can factor regret as

\[
\mathcal{R}(T) = \frac{\mathcal{R}(T)}{V_T} \cdot V_T.
\]

To bound \( \mathcal{R}(T) \), our strategy is to, roughly speaking, bound separately these two factors.

What does all of this have to do with Lyapunov and drifts? You have probably guessed from the
above notation: the information measure \( V_T \) will serve as our Lyapunov function, and incremental
changes in information from step to step will be our drift.

Remarkably, unlike in previous examples of Lyapunov functions that are unbounded over their
domain, in this case we will be dealing with Lyapunov functions that are uniformly bounded, and that
is precisely the key observation of this point of thinking. Indeed, as the player learns more and more
about \( A^* \), the total amount of information will eventually stop growing and saturate as \( t \to \infty \) (The
information piggy bank has a finite capacity by design!) This immediately tells us that the number of
times that the players chooses “bad” actions must be sub-linear in \( T \), for otherwise the piggy bank
would have exploded!

In summary, in what follows, we will establish that under Thompson sampling, and a Lyapunov
function that captures “information” about \( A^* \), the following holds

1. The Lyapunov function is uniformly bounded from above by a constant.

2. If the player experiences a large one-step regret at \( t \), then there must be a sufficiently large
   positive drift in the Lyapunov function.
10.5 Drift Accounting via Information Ratio

We now make the above discussion rigorous by defining a notion of information ratio. The Bergman divergence of a differentiable function $V$ is again defined as

$$D_V(y, x) = V(y) - V(x) - \langle \nabla V(x), y - x \rangle.$$ (10.13)

The Bergman divergence can be defined more generally in a way that does not assume the differentiability of the underlying function. The directional derivative of a function $f : \mathbb{R}^d \mapsto \mathbb{R}$ at $x$ along a vector $v \in \mathbb{R}^d$ is defined to be the limit

$$\nabla_v f(x) := \lim_{h \downarrow 0} \frac{f(x + hv) - f(x)}{h}.$$ (10.14)

Then, the Bergman divergence of a function $V$ is defined to be

$$D_V(y, x) = V(y) - V(x) - \nabla_y - x V(x).$$ (10.15)

Note that if $V$ is differentiable at $x$, then the two definitions coincide.

For a convex set $\mathcal{K}$, we define the diameter of $\mathcal{K}$ with respect to $V$ as

$$\text{diam}_V(\mathcal{K}) = \sup_{x, y \in \mathcal{K}} V(x) - V(y).$$ (10.16)

We will use the subscript $t$ in a probability measure to indicate the conditional probability given $\mathcal{H}_t$:

$$\mathbb{P}_t(\cdot) := \mathbb{P}(\cdot \mid \mathcal{H}_t).$$ (10.17)

The meaning of $\mathbb{E}_t[\cdot]$ is defined analogously.

**Definition 10.5.1 (Information Ratio).** Fix $d \in \mathbb{N}$, a convex set $\mathcal{D} \subseteq \mathbb{R}^d$. Let $\{M_t\}_{t \in \mathbb{N}}$ be an $\mathbb{R}^d$-valued martingale adapted to the filtration $\{\mathcal{F}_t\}_{t \in \mathbb{Z}^+}$ such that $M_t \in \mathcal{D}$ for all $t$ with probability one. Let $V : \mathbb{R}^d \mapsto \mathbb{R}$ be a convex function with

$$\text{diam}_V(\mathcal{D}) < \infty.$$ (10.18)

Fix a slack parameter, $\alpha > 0$. Then information ration at time $t$ is defined as

$$\Gamma_t = \frac{\mathbb{E}_t [R_{A,t} - R_{A,t}] - \alpha^2}{\mathbb{E}_t [D_V(M_{t+1}, M_t)]}.$$ (10.19)

**Interpretation of the information ratio** Informally, the martingale $M$ captures the type of information that we would like to keep track of at each time step. The quantity $\mathbb{E}_t [D_V(M_{t+1}, M_t)]$ serves as a measure of the incremental change in information as a result of the new observations made in time step $t$. As we show shortly, $\mathbb{E}_t [D_V(M_{t+1}, M_t)]$ is indeed a lower bound on the one-step drift in the Lyapunov function $V(M_t)$.

Therefore, if we treat the denominator of the information ratio as a form of incremental information gain, then the information ratio can be thought as measuring the relative magnitude between regret and the information gain in one step. Importantly, if we know that $\Gamma_t$ is small for all $t$, then it would mean that whenever a large one-step regret is incurred (in expectation), it must be accompanied by a sizable gain in information, as well! What remains to be shown is that for carefully selected Lyapunov functions, the information gain cannot continue forever. This is formalized in the following theorem, which will serve as primary workhorse. The theorem was first proven in a more restricted setting by [Russo and Van Roy, 2016] and subsequently generalized to the present form in [Lattimore and Szepesvári, 2019].
**Theorem 10.5.1 (Regret Bound via Information Ratio).** Fix $\alpha > 0$ and $T \in \mathbb{N}$. Suppose that there exists $\bar{\Gamma}$ such that

$$\max_{1 \leq t \leq T} \Gamma_t \leq \bar{\Gamma}, \quad \text{almost surely.} \quad (10.20)$$

Then,

$$\mathcal{R}(T) \leq \alpha T + \sqrt{T \bar{\Gamma} \mathbb{E} [V(M_{T+1}) - V(M_1)]}, \quad (10.21)$$

In particular, since $\mathbb{E} [V(M_{T+1}) - V(M_1)] \leq \text{diam}_V(D)$ under any prior, we have a prior-free version:

$$\mathcal{R}(T) \leq \alpha T + \sqrt{T \bar{\Gamma} \text{diam}_V(D)}. \quad (10.22)$$

**Proof.** We will cheat a little here by assuming that $V$ is differentiable everywhere:

$$\mathbb{E}_t [D_V(M_{t+1}, M_t)] = \mathbb{E}_t [V(M_{t+1}) - V(M_t) - \langle \nabla V(M_t), (M_{t+1} - M_t) \rangle]$$

$$= \mathbb{E}_t [V(M_{t+1})] - V(M_t) - \langle \nabla V(M_t), \mathbb{E}_t [(M_{t+1} - M_t)] \rangle$$

$$= \mathbb{E}_t [V(M_{t+1})] - V(M_t) \quad (10.23)$$

where the last step follows from $M$’s being a martingale, and hence $\mathbb{E}_t [(M_{t+1} - M_t)] = 0$.

Given the assumption on the maximum information ratio, we know that

$$\mathbb{E}_t [R_{A^*, t} - R_{A_t, t}] \leq \alpha + \sqrt{\bar{\Gamma} \mathbb{E}_t [D_V(M_{t+1}, M_t)]}. \quad (10.24)$$

We have

$$\mathcal{R}(T) = \mathbb{E} \left[ \sum_{t=1}^{T} (R_{A^*, t} - R_{A_t, t}) \right]$$

$$\leq \alpha T + \mathbb{E} \left[ \sum_{t=1}^{T} \sqrt{\bar{\Gamma} \mathbb{E}_t [D_V(M_{t+1}, M_t)]} \right]$$

$$\leq \alpha T + \sqrt{T \bar{\Gamma} \mathbb{E} \left[ \sum_{t=1}^{T} \mathbb{E}_t [D_V(M_{t+1}, M_t)] \right]} \quad (10.25)$$

where the last step follows from Cauchy-Schwartz inequality. We can further bound the summation in the second term using (10.28) and a telescopic sum:

$$\mathbb{E} \left[ \sum_{t=1}^{T} \mathbb{E}_t [D_V(M_{t+1}, M_t)] \right] \leq \mathbb{E} \left[ \sum_{t=1}^{T} (\mathbb{E}_t [V(M_{t+1})] - V(M_t)) \right]$$

$$= \mathbb{E} [V(M_{T+1}) - V(M_1)] \quad (10.26)$$

Substituting this inequality into the regret bound, we obtain the desired result:

$$\mathcal{R}(T) \leq \alpha T + \sqrt{T \bar{\Gamma} \mathbb{E} [V(M_{T+1}) - V(M_1)]}. \quad (10.27)$$
Remark 10.5.2. (The case of non-differentiable Lyapunov functions) We have cheated a bit in a key step in the proof of Theorem 10.5.1, in (10.23), by assuming that $V$ is differentiable everywhere. What happens if it is isn’t? This can occur even for common Lyapunov functions. For instance, when

$$\lim_{t \to 0} V(M_t) = 0,$$

with probability one. We thus have obtained a strict inequality in (10.28):

$$E_t [D_V(M_{t+1}, M_t)] \leq E_t [V(M_{t+1})] - V(M_t)$$  \hspace{1cm} (10.28)

and the rest of the proof would go through. The proof for this result follows similar intuition, but relies on the general definition of the Bergman divergence using directional derivatives. We have

$$E_t [D_V(M_{t+1}, M_t)] = E_t [V(M_{t+1}) - V(M_t) - \nabla_{M_{t+1} - M_t} V(M_t)]$$

(in the proof of Theorem 10.5.1, in (10.23), by assuming that $V$ is differentiable everywhere. Unfortunately, this is not possible, as illustrated by the following counterexample. Let $d = 1$, $t = 0$, $M_t = 0$ and $M_{t+1}$ is independent from all other parts of the system and follow the Radamacher distribution, taking the value of 1 or $-1$ with probability 1/2 each. In this case, we have that for all $h \in (0, 1)$,

$$(1 - h)M_t + hM_{t+1} = (1 - h)M_t + hM_t = M_t.$$  \hspace{1cm} (10.30)

Since we had an equality in (10.23) in the special case where $V$ is differentiable, it may be tempting to ask if we can improve the analysis to obtain an equality in general case. Unfortunately, this is not possible, as illustrated by the following counterexample. Let $d = 1$, $t = 0$, $M_t = 0$ and $M_{t+1}$ is independent from all other parts of the system and follow the Radamacher distribution, taking the value of 1 or $-1$ with probability 1/2 each. In this case, we have that for all $h \in (0, 1)$,

$$\frac{V((1 - h)M_t + hM_{t+1}) - V(M_t)}{h} = \frac{|hM_{t+1}|}{h} = |M_{t+1}| = 1,$$  \hspace{1cm} (10.31)

with probability one. We thus have obtained a strict inequality in (10.28):

$$E_t [D_V(M_{t+1}, M_t)] = E_t [V(M_{t+1})] - V(M_t) - 1 < E_t [V(M_{t+1})] - V(M_t).$$  \hspace{1cm} (10.32)
10.6 Entropy as Information Measure

We now apply Theorem 10.5.1, and the template therein, to prove Theorem 10.3.1. We will make the following assignments.

1. \( M_t \): posterior distribution of the optimal arm \( A^* \) at time \( t \), \( \xi_t \), taking values in \( \Delta^{K-1} \).

2. \( V \): the negentropy function

\[
V(p) = \sum_{i=1}^{K} p_i \log p_i, \quad p \in \Delta^{K-1}.
\]

(10.33)

Let us observe some useful properties. First, we have that

\[
E \left[ \xi_{t+1} \mid F_t \right] = E \left[ P(A^* = a \mid H_{t+1}) \mid F_t \right] = E \left[ P \left( A^* = a \mid H_t, (R_{A_{t+1}}, A_t) \right) \mid F_t \right] = P(A^* = a \mid H_t) = \xi_{a,t},
\]

(10.34)

where the second inequality follows from towering property of expectation as well as the fact that \((R_{A_{t+1}}, A_t)\) is \( F_t \)-measurable. That is, we have shown that \( \{ \xi_t \}_{t \in \mathbb{N}} \) is indeed a martingale.

Next, note that when \( V \) is the negentropy function,

\[
\nabla V(p)_i = 1 + \log p_i,
\]

(10.35)

and its Bergman divergence is precisely the relative entropy, also known as the Kullback–Leibler divergence, \( D_{KL} \). For \( p, q \in \Delta^{K-1} \),

\[
D_V(q, p) = V(q) - V(p) - \langle \nabla V(p), (q - p) \rangle
\]

\[
= \sum_i q_i \log q_i - \sum_i p_i \log p_i - \sum_i (q_i - p_i) \log p_i
\]

\[
= \sum_i q_i \log q_i - \sum_i p_i \log p_i - \sum_i (q_i - p_i) \log p_i
\]

\[
= \sum_i q_i \log q_i - \sum_i p_i \log p_i - \sum_i (q_i - p_i) \log p_i
\]

\[
= \sum_i q_i \log q_i - \sum_i p_i \log p_i - \sum_i (q_i - p_i) \log p_i
\]

\[
= D_{KL}(q \| p)
\]

(10.36)

**Lemma 10.6.1.** Suppose \( V \) is the negentropy function, then

1. \( E_t [D_V(\xi_{t+1}, \xi_t)] = E_t [D_{KL}(\xi_{t+1} \| \xi_t)] = I_t(A^*; (R_{A_{t+1}}, A_t)), \)

(10.37)

where \( I_t(X; Y) \) is defined to be the mutual information between random variables \( X \) and \( Y \) conditional on \( H_t \):

\[
I_t(X; Y) := E_t [D_{KL}(P_{X,Y} \| P_X P_Y)]
\]

(10.38)

2. \( \text{diam}_V(\Delta^{K-1}) = \log K. \)

(10.39)

Finally, we can prove the following upper bound on the information ratio when the player uses Thompson sampling.
Lemma 10.6.2 (Information Upper Bound under Negentropy). Let $V$ be the negentropy function, and $\alpha = 0$. Suppose the player uses the Thompson sampling strategy. Then, under any prior distribution $\eta$,}

\[
\Gamma_t = \frac{\mathbb{E}_t [R_{A^*, t} - R_{A, t}]}{I_t(A^*; (R_{A^*, t}, A_t))} \leq K/2. \tag{10.40}
\]

We are now ready to prove Theorem 10.3.1. Substituting the following into Theorem 10.5.1,

1. $\alpha = 0$
2. $M_t = \xi_t$
3. $\mathcal{D} = \Delta^{K-1}$
4. $\bar{\Gamma} = K/2$
5. $\text{diam}_V(\Delta^{K-1}) = \log K$

We have thus prove the prior independent bound (10.10):

\[
\mathcal{R}(T) \leq \alpha T + \sqrt{T \text{diam}_V(\mathcal{D})} = \sqrt{(K \log K)T/2}. \tag{10.41}
\]

For the more refined, prior-dependent bound in (10.52), we will use (10.21) and note that

\[
\mathbb{E} [V(\xi_t) - V(\xi_1) \leq -\mathbb{E} [V(\xi_1)] = H(A^*). \tag{10.42}
\]

10.6.1 Proof of Lemma 10.6.2

To avoid clutter, we will fix $t$ and suppress the subscript $t$ in $\mathbb{E}_t \cdot$, $\mathbb{P}_t(\cdot)$ and $I_t(\cdot)$ in this proof, while assuming that all probabilities are measured with the conditioning on $\mathcal{H}_t$. With this notation in mind, we will also use the short-hands $A := A_t$, $R := R_t$ and $R_A := R_{A^*, t}$. We begin with some useful properties of mutual information and one-step regret under Thompson sampling. The first part of the lemma roughly states that the amount of mutual information gain would be bigger if we expect the reward distribution of an optimal arm to be significantly different from an “average” arm, as measured by KL-divergence.

Lemma 10.6.3. The following holds true under Thompson sampling.

1.

\[
I(A^*; (R_A, A)) = \sum_{a \in A} \mathbb{P}(A = a) I(A^*; R_a)
= \sum_{a, a^* \in A} \mathbb{P}(A^* = a) \mathbb{P}(A^* = a^*) D \left( \mathbb{P}(R_a \in \cdot \mid A^* = a^*) \| \mathbb{P}(R_a \in \cdot) \right)
\tag{10.43}
\]

where $\mathbb{P}(X \in \cdot)$ denotes the probability measure associated with $X$.

2.

\[
\mathbb{E} [R_{A^*} - R_A] = \sum_{a \in A} \mathbb{P}(A^* = a) \left( \mathbb{E} [R_a \mid A^* = a] - \mathbb{E} [R_a] \right) \tag{10.44}
\]
CHAPTER 10. DRIFT ANALYSIS IN BAYESIAN BANDITS WITH INFORMATION RATIO

Figure 10.1: This is a Markov diagram that illustrates the dependency relationships among several key quantities. The optimal action $A^*$ is a deterministic function of $\theta$, and so are the distribution of the reward vector $R_t$. The action $A_t$ depends on $\theta$ only through the history $\mathcal{H}_t$. Therefore, conditional on $\mathcal{H}_t$, $A_t$ becomes independent from $\theta$, $A^*$ and $R_t$.

Proof. The proof relies on the following key insight. Because the policy only depends on the past history $\mathcal{H}_t$ and external random noise, conditional on the history $\mathcal{H}_t$, the action $A$ is independent of actual parameter, $\theta$. A useful mental picture is that $\mathcal{H}_t$ summarizes all the information we have about $\theta$ so far. Therefore, over a single step, one can imagine that $\theta$ is drawn “fresh” from the posterior distribution of $\mathbb{P}_t(\theta \in \cdot)$. This relationship is illustrated in the Markov diagram in Figure 10.1.

For the first part of the lemma, we have that

$$I(A^*; (R_A, A)) = I(A^*; A) + I(A^*; R_A | A)$$

$$= \sum_{a \in A} \mathbb{P}(A = a) I(A^*; R_a | A = a)$$

$$= \sum_{a \in A} \mathbb{P}(A = a) I(A^*; R_a)$$

$$= \sum_{a \in A} \mathbb{P}(A = a) \mathbb{E}_{A^*} [D(\mathbb{P}(R_a \in \cdot | A^*) \parallel \mathbb{P}(R_a \in \cdot))]$$

$$= \sum_{a, a^* \in A} \mathbb{P}(A^* = a) \mathbb{P}(A^* = a^*) D(\mathbb{P}(R_a \in \cdot | A^* = a^*) \parallel \mathbb{P}(R_a \in \cdot))$$

$$= \sum_{a, a^* \in A} \mathbb{P}(A^* = a) \mathbb{P}(A^* = a^*) D(\mathbb{P}(R_a \in \cdot | A^* = a^*) \parallel \mathbb{P}(R_a \in \cdot)).$$

(10.45)

The steps are based on:

1. (a): chain rule of mutual information: $I(X; (Y, Z)) = I(X; Y) + I(X; Z | Y)$.
2. (b): conditional independence between $A$ and $A^*$, as per Figure 10.1.
3. (c): conditional independence between $A$ and $(A^*, R)$, as per Figure 10.1.
4. (d): the KL-divergence form of mutual information:

$$I(X; Y) = \mathbb{E}_X \left[ D(\mathbb{P}(Y \in \cdot | X) \parallel \mathbb{P}(Y \in \cdot)) \right]$$

$$= \sum_x \mathbb{P}(X = x) D(\mathbb{P}(Y \in \cdot | X = x) \parallel \mathbb{P}(Y \in \cdot)).$$
5. \((e)\): the fact that we use Thomson sampling. That is, \(A\) is sampled from the posterior distribution and hence
\[
P(A \in \cdot) = P(A^* \in \cdot). \tag{10.46}
\]

For the second claim in the lemma, observe that
\[
\mathbb{E}[R_{A^*} - R_A] = \sum_{a \in A} P(A^* = a) \mathbb{E}[R_a | A^* = a] - \sum_{a \in A} P(A = a) \mathbb{E}[R_a | A = a]
\]
\[
\stackrel{(a)}{=} \sum_{a \in A} P(A^* = a) \mathbb{E}[R_a | A^* = a] - \sum_{a \in A} P(A^* = a) \mathbb{E}[R_a | A = a]
\]
\[
\stackrel{(b)}{=} \sum_{a \in A} P(A^* = a) \mathbb{E}[R_a | A^* = a] - \sum_{a \in A} P(A^* = a) \mathbb{E}[R_a]
\]
\[
= \sum_{a \in A} P(A^* = a) (\mathbb{E}[R_a | A^* = a] - \mathbb{E}[R_a]) \tag{10.47}
\]

where in (\(a\)) we again evoked the definition of Thompson sampling (10.46), and in (\(b\)) the fact that \(R\) is conditionally independent from the chosen action \(A\).

The next lemma is a consequence of Pinsker’s inequality.

**Lemma 10.6.4.** Fix probability distributions \(\mu\) and \(\gamma\) over \([0, 1]\) such that \(\mu\) is absolutely continuous with respect to \(\gamma\). Let \(X\) and \(Y\) be random variables distributed according to \(\mu\) and \(\gamma\), respectively. Then
\[
\mathbb{E}[X] - \mathbb{E}[Y] \leq \sqrt{\frac{1}{2} D(\mu \| \gamma)}. \tag{10.48}
\]

The significance of the above lemma is that it implies that whenever we have two random variables with diverging expectations, then such difference must also reflect in the KL-divergence between the underlying distributions as well.

We are now ready to prove the upper bound on information ratio in Lemma 10.6.2. We have that
\[
\mathbb{E}[R_{A^*} - R_A]^2 \stackrel{(a)}{=} \left( \sum_{a \in A} P(A^* = a) (\mathbb{E}[R_a | A^* = a] - \mathbb{E}[R_a]) \right)^2
\]
\[
\leq K \sum_{a, a' \in A} P(A^* = a) P(A^* = a') (\mathbb{E}[R_a | A^* = a] - \mathbb{E}[R_a])^2 \tag{10.49}
\]

The steps are based on
1. \((a)\): second claim in Lemma 10.6.3.
2. \((b)\): Cauchy-Schwartz inequality.
3. \((c)\): Lemma 10.6.4.
4. \((d)\): first claim in Lemma 10.6.3.

Dividing both sides of this inequality by \(I(A^*; (R_A, A))\) yields Lemma 10.6.2.

\section{10.7 Improved Distribution-Free Regret Bound}

In the previous section, we saw that the negentropy Lyapunov function combined with Theorem 10.5.1 leads to a distribution-free regret upper bound of \(\sqrt{TK \log K}\). In this section, we will experiment with a different Lyapunov function which will allow us to improve this bound to \(\sqrt{TK}\), which is optimal up to a constant factor.

We will follow a similar recipe as before, by plugging in an upper bound on the information ratio into Theorem 10.5.1. Consider the following Lyapunov function, we will refer to as the \(\ell_{1/2}\) function for its similarity to the \(\ell_p\) norms (though this function does not form a norm):

\[
V(x) = -2 \sum_{i=1,\ldots,d} \sqrt{x_i}, \quad x \in \mathbb{R}^d.
\] (10.50)

\textbf{Lemma 10.7.1.} Under the \(\ell_{1/2}\) Lyapunov function. We have that

1. \(\text{diam}_V(\Delta^{K-1}) \leq 2\sqrt{K}\).

2. Fix \(\alpha = 0\), we have that

\[
\Gamma_t \leq \sqrt{K}.
\] (10.51)

\textit{Proof.} For the first claim, notice that \(\max_{x \in \Delta^{K-1}} V(x) = -2\), which is attained when all mass of \(x\) concentrates on one element, and \(\min_{x \in \Delta^{K-1}} V(x) = -2\sqrt{K}\), attained when \(x = \frac{1}{K}1\). This immediately implies \(\text{diam}_V(\Delta^{K-1}) = 2\sqrt{K} - 2 \leq 2\sqrt{K}\).

The proof of the second claim requires more delicate calculation, and can be found in the proof of Lemma 7 of [Lattimore and Szepesvári, 2019]. \(\square\)

We now evoke Theorem 10.5.1 with the following substitutions:

1. \(\alpha = 0\)
2. \(M_t = \xi_t\)
3. \(D = \Delta^{K-1}\)
4. \(\hat{\Gamma} = \sqrt{K}\)
5. \(\text{diam}_V(\Delta^{K-1}) = \sqrt{K}\)

Note that the first three substitutions are the same as in the case of the negentropy Lyapunov function. We obtain the following result.

\textbf{Theorem 10.7.2.} Fix \(K\). Then under any prior distribution \(\eta\), we have

\[
\mathcal{R}(T, \pi_{TS}) \leq \sqrt{2KT}, \quad T \in \mathbb{N}.
\] (10.52)
10.8 Notes

The concept of information ratio was first proposed in [Russo and Van Roy, 2014] for designing algorithms in multi-arm bandit problems. It was subsequently employed in [Russo and Van Roy, 2016] for deriving prior-dependent regret bounds on Thompson sampling, where the authors established Theorem 10.3.1, and our proof for Lemma 10.6.2 follows this paper. In both papers, the corresponding Lyapunov function $V$ is the negentropy function of the $K-1$ dimensional simplex, and the slack parameter $\alpha$ is set to be zero. The framework in [Russo and Van Roy, 2016] is a bit more general, where the decision maker receives feedback in a certain set, and the rewards are derived as a function of the feedback; in our exposition, the feedback is simply equal to the reward. [Lattimore and Szepesvári, 2019] further builds on this idea and generalizes the analysis to partial monitoring, while also obtaining the mini-max optimal regret bound for Thompson sampling (Theorem 10.7.2). The regret bounds using general potential functions in 10.5 follow the development in [Lattimore and Szepesvári, 2019], though we have used a different presentation in the theorem to make the connection to earlier results on information ratio more evident. The results in [Lattimore and Szepesvári, 2019] are also more general and extend to Lyapunov functions that are not differentiable, though by assuming differentiability simplifies our exposition significantly. The Thompson sampling algorithm was proposed by William R. Thompson in [Thompson, 1933]. The counterexample in Remark 10.5.2 is suggested by Tor Lattimore in private email communications, gratefully acknowledged here.
Bibliography


Index

Bergman divergence, 72
  non-differentiable functions, 85
  projection, 73
Blackwell approachability
  approach, 68
  Approachability thoerem, 68
capacity region, 18
counting process, 8
drift conservation, 50
Foster-Lyapunov theorem
  proof, 23
  Foster-Lyapunov theorem: statement, 19
Hannan consistency, 60
job size, 12
Lyapunov function, 19
Markov process: generator, 8
  Max-Weight algorithm, 18
  maximum stability, 18
Poisson process, 8
potential departure, 12
prediction with expert advice
  expert, 60
  model, 60
pure-jump Markov process, 7
queues
  introduction, 11
recurrence
  null recurrence, 9
  positive recurrence, 9

state space collapse
  steady state, 47
  transient, 35
station, 13
stochastic network
  generalized switch, 17
  single hop, 17
time homogeneous, 7
traffic intensity, 13
transient
  Markov process, 9