Bandits and Preference Learning

By
ANIRUDDHA BHARGAVA

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY
(ELECTRICAL AND COMPUTER ENGINEERING)

at the
UNIVERSITY *of* WISCONSIN-MADISON
2017

Date of final oral examination: November 27, 2017.
The dissertation is approved by the following members of the Final Oral Committee:
Robert Nowak, Professor, Electrical and Computer Engineering,
William Sethares, Professor, Electrical and Computer Engineering,
Rebecca Willett, Professor, Electrical and Computer Engineering,
Stephen Wright, Professor, Computer Science,
Xiaojin (Jerry) Zhu, Professor, Computer Science.

The great Lord is boundless; the universe which comes from Her is also boundless.
Still the essence of creation continues to be boundless.

- Upanishads

Abstract

The internet revolution has brought a large population access to a vast array of information since the mid 1990s. More recently, with the advent of smartphones, it has become an essential part of our everyday life. This has lead to, among many other developments, the personalization of the online experience with great benefits to all involved. Companies have particular interest in showing products and advertisements that match what particular users are looking for, and users desire getting personalized recommendations from internet for entertainment and consumer goods that suit them as individuals. In machine learning, this is popularly achieved using the theory of the multi-armed bandits, methods which allow us to zero in on the consumer's personal preferences.

The last few decades have seen great advances in the theory and practice of multi-armed bandits exploiting either the context of the user, the context of the objects, or both. Great theoretical improvements have brought algorithms' performance close to their theoretical optimal. However, various challenges exist in the practical use of multi-armed bandits. In this thesis, we explore some of these challenges and endeavor to overcome them. First, we examine how multiple populations can be catered to simultaneously. We then address the issue of scaling multi-armed bandits to situations where there are many arms. We also look at how to incorporate generalized linear reward models while maintaining computational efficiency. Finally, we address how we can use feature feedback to focus the bandits exploration to a limited subset of features. This leads to algorithms that are still tractable for high-dimensional datasets where the preferences of the user are explained by a sparse subset of them.

Acknowledgements

My parents Vasumathi and Jamadagni are one of the biggest reasons I got into research from an early age. They instilled a spirit of curiosity and did all they could to put me in the best position for my education. My aunt Saraswati was pivotal in the upbringing too: shuttling me back and forth from schools. My wife Emily has been key to keeping me on track and helping me work hard. She has been a great companion through the last few years, having been there herself. To them, I owe a big debt of gratitude.

I would like to thank my advisor Rob. He gave me a chance to come to Wisconsin and allowed me to find my way in the world of research. He has been there to help me grow as a researcher and as a person. In my research journey I would like to thank other professors who have played a key role: Rebecca Willett, Jerry Zhu, Steve Wright, and many other.

Thanks to my co-authors Ravi Ganti and Kwang-Sung Jun for being great people and friends to work with on projects. I would like to thank Scott Beddia, Lalit Jain, Matt Malloy, Xin Hunt, Kevin Jamieson, John Ehrmanntraut, Urvashi Oswal, Blake Mason, Nikhil Rao, Gautam Dasarathy and many other. My friends have kept me grounded and provided me with constant support.

Thanks to the Kohler fellows for being great people to talk to and for being an outlet for creative ideas. I would like to thank the Madison community in general for allowing me to enjoy so many valuable moments. Finally, thanks to all the great forces that bring us together and to allow us to dig deep on research projects.

Contents

Chapter 1

Introduction

Multi-armed bandits first arose in the 1930s with the study of clinical trials (Thompson, 1933).), which required a small amount of data to be used as a guide to act on the effective collection of further data. Then, as it is now, this is a problem of sequential decision making, a problem that is often encountered in the modern internet.

In general, a multi-armed bandit problem consists of a set of possible actions, that we call the arms. At each time, one must propose an action. Then one gets to observe the reward for that action only. Based upon this information, one decides upon a next action to take, and the series continues.

The name multi-armed bandit itself comes from a term used for slot machines. Slot machines are, in fact, a great example for the concept of a bandit problem. Consider having $100 to spend on slot machines at a casino. In a room full of slot machines, there may be some that give a higher reward than others. Suppose you must pay 1 ¢to try a particular slot machine. The question addressed by bandit algorithms is this: how should we decide to spend the money? On one hand, we have to try different machines to estimate their rewards (exploration) in order to identify the best machines, and among those machines we have found empirically to be best, we want to spend the remaining money to get the best chance of getting a maximal payoff (exploitation). All bandit algorithms trade off these two factors, exploration and exploitation, in order to seek an optimum payout.

This thesis will focus on the particular problem of contextual bandits in a stochastic setting. Contextual refers to the assumption that for each action or arm, we also get to observe the context or features associated with it. A stochastic setting refers to the assumption that we observe a noisy reward, where the outcome contains noise which is random and independent of the choice of the arm.

In stochastic multi-armed bandits, two objectives are often important: finding the best action, and minimizing the regret. In the first case the goal is clear: to find the best possible action with as few queries as possible. In the second case, minimizing regret, regret refers to the difference in rewards between the action we actually took and best possible action we could have taken. This thesis focuses on the problem of minimizing regret.

Work in this particular area of multi-armed bandits can be traced back to (Auer and Long, 2002) who proposed the linear-bandit model. Further seminal work in this area was done by (Dani et al., 2008), (Rusmevichientong and Tsitsiklis, 2010) and (Abbasi-Yadkori et al., 2011) leading to algorithms whose regrets scale with the square-root of time.

While this area of research started with the study of clinical trials, it has a multitude of applications today in many areas. For example, the spread of the internet has led to personalization of the information that we are exposed to. Netflix recommends movies based on our previous viewing profiles, Google personalizes search results based on our browsing history and physical location, Amazon recommends products based on our shopping history. The list goes on.

One key consideration presented in this thesis is how to make the most of human feedback in the process of bandit design. We propose the use of a mathematical structure to aid in aggregating information from multiple populations. We also propose modifications for existing algorithms in order to obtain better results. An example of such solicitation of human feedback can be seen in Figure 1.1.

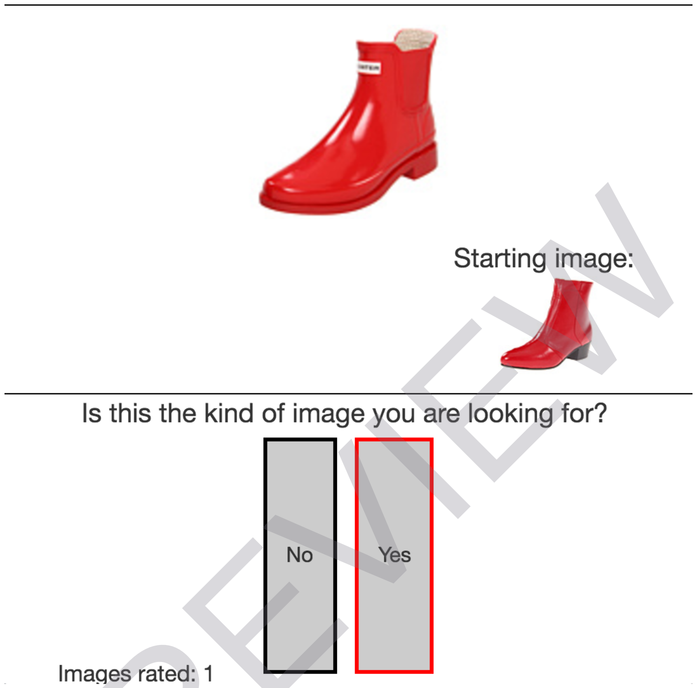

Figure 1.1: Example of reward in bandit setting for image search. We would like to propose as many images as possible that leads to the users liking them.

While bandits have comprised an important component in the enabling of personalized interfaces, many challenges exist in their use on a large scale. Critically, it is computationally expensive to run these algorithms. This thesis will address some of the issues that hinder a wider use of multi-armed bandits by exploiting the structure of multiple populations, enabling the use of hashing, and by making use of user feedback.

Below are some of the big themes addressed in this thesis.

1. **Estimating the preference of multiple-populations blindly:** (Chapter 2). This chapter addresses a more general problem: completing low-rank positive semi-definite matrices by asking queries actively. Then giving a max-norm bound on

the estimating a matrix when we sample elements of the matrix through a random process. Then it shows that estimating the preferences of multiple populations can be seen as a sub-problem of the matrix completion.

2. **Incorporating hashing and more general linear models into contextual bandits:** (Chapters 3 and, Chapter 4) When we can come up with good query words, we can use many search algorithms. When we have a hard time formulating what we are looking for, the search problem becomes much harder. We propose a bandit framework, where we get answers to the question: (how much) do you like the following image? Our response could be a score (which translates to standard linear bandits) or Yes/No answers.

   Besides, all current bandit algorithms scale linearly (Thompson sampling (Agrawal and Goyal, 2013)) or super-linearly (OFUL (Abbasi-Yadkori et al., 2011)) in the number of arms. In the area of uncertainty sampling, people have previously used hashing to scale sub-linearly in the number of query items. We propose an algorithm whose regret is slightly worse that state-of-the-art but whose computational complexity scales sub-linearly. Besides, we also show that practically, Thompson sampling can be used with hashing with little penalty on the regret on the Zappos50K dataset.

   There has been theory of bandits that can deal with generalized linear models but to the best of our knowledge, it applied to either logistic models only or, needed the storage of all previous queries. We therefore propose a new algorithm that can deal with much more general models while not requiring all previous data.

3. **Using what the user says:** When we are running bandits on arms whose features are high dimensional, it may be impractical to run them in real-world situations. This is because, at every iteration, for every person, we need to update a $d \times d$ matrix. We look at how we may restrict ourselves to a much small dimension k and how we could use feedback about the features themselves, in addition to the

linear rewards we get, in order to identify the k relevant features. We propose a model for getting feature feedback for documents and show on both a synthetic dataset and a dataset generated from real documents that we can improve the regret by using this feature feedback.

1.1 Chapters at a Glance

Chapter 2. Bandits and Low Rank Matrix Completion

Completion of low-rank matrices using active queries. Bound on the max-norm between the difference of the completed matrix and the original matrix. Connection to the problem of estimating the preferences of more than one population without the knowledge of which population the user belongs to.

Chapter 3. Image Search

Using linear bandits to perform personalized search. Biasing linear bandit algorithms towards the starting point to potentially get better regret scaling. Changing the objective function of OFUL ((Abbasi-Yadkori et al., 2011)) in order to allow the use of hashing. Showing that hashing, under certain assumptions, does not degrade the regret guarantees. The application of our algorithm and, other state-of-the-art algorithms to the Zappos50k dataset. Real life data from the NEXT system using these algorithms.

Chapter 4. Generalized Linear Bandits

Our premise is that it is hard for the users to search large datasets with keywords. Therefore, it our belief that it is more reasonable to expect Yes/No answers to questions rather than getting a real number in $[-1, 1]$. We propose a new algorithm that does not require the storage of all previously pulled arms and their rewards, and which applies to a broad class of generalized linear models. We apply our algorithm to synthetic data to show the advantages of the new method.

Chapter 5. Bandits with Feature Feedback

How do we use feature feedback to recognize relevant dimensions to run bandit algorithms? How do we apply this to searching for documents from a large corpus? We apply the proposed algorithms to a synthetic dataset and to the Newsgroup 20 dataset.

Appendix: Chapter 6. Socioscope

Can we estimate real-world signals from Twitter? More specifically, if we model event intensities as happening in space and time and people as sensors of those events, can we estimate the intensity of event at different locations across different times? We apply our algorithm to estimating animal extant estimates using tweets about roadkill. We then show that these are close to the maps produced by domain scientists. We also show that we can see the diurnal-nocturnal behavior of animals.

Appendix: Chapter 7

This chapter summarizes the contribution to NEXT. It documents the modifications that have been introduced to enable the running of contextual linear bandits.

Appendix: Chapter 8

This chapter talks about the contributions made to the software called IDTaxa. It is a classification algorithm for getting taxonomy of biological samples. It talks about some of the problems of existing software and what we addressed in order to improve classification leading to state-of-the-art results.

Chapter 2

Bandits and Low-Rank Matrix Completion [1]

[1] Paper presented at AISTATS 2017

2.1 Introduction

The problem of matrix completion is a fundamental problem in machine learning and data mining where one needs to estimate an unknown matrix using only a few entries from the matrix. This problem has seen an explosion in interest in recent years perhaps fueled by the famous Netflix prize challenge Bell and Koren (2007) which required predicting the missing entries of a large movie-user rating matrix. Candès and Recht (2009) showed that by solving an appropriate semidefinite programming problem it is possible to recover a low-rank matrix given a few entries at random. Many improvements have since been made both on the theoretical side (Keshavan et al., 2009; Foygel and Srebro, 2011) as well as on the algorithmic side (Tan et al., 2014; Vandereycken, 2013; Wen et al., 2012).

Very often in applications the matrix of interest has more structure than just low rank. One such structure is positive semi-definiteness which appears when dealing with covariance matrices in applications like PCA, and kernel matrices when dealing with kernel learning. *In this chapter we study the problem of matrix completion of low-rank, symmetric positive semidefinite (PSD) matrices and provide simple, and computationally efficient algorithms that actively query a few elements of the matrix and output an estimate of the matrix that is provably close to the true PSD matrix.* More precisely, we are interested in algorithms that ouptput a matrix that is provably (ϵ, δ) close to the true underlying matrix in the max norm [2]. This means that if $\boldsymbol{L}$ is the true, underlying PSD matrix then we want our algorithms to output a matrix $\hat{\boldsymbol{L}}$ such that $\|\hat{\boldsymbol{L}} - \boldsymbol{L}\|_{\max} \leq \epsilon$, with probability at least $1 - \delta$. Our goal is strongly motivated by applications to certain multi-armed bandit problem where there are a large number of arms. In certain cases the losses of these arms can be arranged as a PSD matrix and finding the (ϵ, δ) best arm can be reduced to the above defined (ϵ, δ) PSD matrix completion (PSD-MC) problem. Our contributions are as follows

[2]The max norm of a matrix is the maximum of the absolute value of all the elements in a matrix

1. Let $\boldsymbol{L}$ be a $K \times K$ rank r PSD matrix, which is apriori unknown. We propose two models for the PSD-MC problem. In both the models the algorithm has access to an oracle $\mathcal{O}$ which when queried with a pair-of-indices (i, j) obtains a response $y_{i,j}$. The main difference between these two oracle models is the power of the oracle. In the first model, which we call as a deterministic oracle model, the oracle is a powerful, deterministic, but expensive oracle where $y_{i,j} = L_{i,j}$. In the second model, called as the stochastic oracle model, we shall assume that all the elements of the matrix $\boldsymbol{L}$ are in $[0, 1]$, and we have access to a less powerful, but cheaper oracle, whose output $y_{i,j}$ is sampled from a Bernoulli distribution with parameter $\boldsymbol{L}_{i,j}$. These models are sketched in Figure (1).

2. We propose algorithms for PSD-MC problem, under the above two models. Our algorithms, called MCANS [3], in the deterministic oracle model, and S-MCANS [4] in the stochastic oracle model are both based on the following key insight: In the case of PSD matrices it is possible to find linearly independent columns by using few, adaptively chosen queries. In the case of S-MCANS we use the above insight along with techniques from multi-armed bandits literature in order to tackle the randomness of the stochastic oracle.

3. We prove that the MCANS algorithm outputs a $(\epsilon = 0, \delta = 0)$ estimate of the matrix $\boldsymbol{L}$ (exact recovery) after making at most $K(r + 1)$ queries, and the S-MCANS algorithm outputs $\hat{\boldsymbol{L}}$ that is (ϵ, δ) close to $\boldsymbol{L}$ using queries that is linear in K and a low-order polynomial in the rank r of matrix $\boldsymbol{L}$. Establishing such sample complexity bounds leads to interesting problems in matrix approximation in the max-norm. The contributions we make here could be of independent interest in the low-rank matrix approximation literature.

4. We introduce a multi-armed bandit (MAB) problem motivated by applications

[3] MCANS stands for Matrix Completion via Adaptive Nystrom Sampling
[4] S in S-MCANS stands for stochastic

to advertising in Figure (5) and show how this MAB problem can be reduced to a PSD-MC problem. This reduction allows us to use MCANS and S-MCANS algorithms to solve the problem of finding an (ϵ, δ) optimal arm using far fewer queries than what standard multi-armed bandit based algorithms such as successive elimination or median elimination would need. Our experiments show that exploiting the spectral structure in the MAB problem allows us to design algorithms that are much more query efficient than state-of-art bandit algorithms that do not exploit the spectral structure in the bandit problem.

5. We also show that the MCANS algorithm can be effectively used to complete kernel matrices, under budget constraints. The completed kernel matrix when used in a kernel dimensionality reduction task leads to better results than a kernel matrix which has been completed using standard low-rank matrix completion.

Notation. Δ_r represents the r dimensional probability simplex. Matrices and vectors are represented in bold font. For a matrix $\boldsymbol{L}$, unless otherwise stated, the notation $\boldsymbol{L}_{i,j}$ represents (i, j) element of $\boldsymbol{L}$, and $\boldsymbol{L}_{i:j,k:l}$ is the submatrix consisting of rows $i, i+1, \ldots, j$ and columns $k, k+1, \ldots, l$. The matrix $\|\cdot\|_1$ and $\|\cdot\|_2$ norms are always operator norms. The matrix $\|\cdot\|_{\max}$ is the element wise infinity norm. Finally, let $\mathbb{1}$ be the all 1 column vector.

2.2 Related Work

The problem of PSD-MC has been considered by many other authors (Bishop and Byron, 2014; Laurent and Varvitsiotis, 2014a,b). However, all of these papers consider the passive case, i.e. the entries of the matrix that have been revealed are not under their control. In contrast, we have an active setup, where we can decide which entries in the matrix to reveal. The Nyström algorithm for approximation of low rank PSD matrices has been well studied both empirically and theoretically. Nyström methods typically choose random columns to approximate the original low-rank matrix (Gittens and Mahoney, 2013; Drineas and Mahoney, 2005). Adaptive schemes where the columns used for Nystrom approximation are chosen adaptively have also been considered in the literature. To the best of our knowledge these algorithms either need the knowledge of the full matrix (Deshpande et al., 2006) or have no provable theoretical guarantees (Kumar et al., 2012). Moreover, to the best of our knowledge all analysis of Nystrom approximation that has appeared in the literature assume that one can get error free values for entries in the matrix. Adaptive matrix completion algorithms have also been proposed and such algorithms have been shown to be less sensitive to the incoherence in the matrix (Krishnamurthy and Singh, 2013). The bandit problem that we study in the latter half of the chapter is related to the problem of pure exploration in multi-armed bandits. In such pure exploration problems one is interested in designing algorithms with low, simple regret or designing algorithms with low (ϵ, δ) query complexity. Algorithms with small simple regret have been designed in the past (Audibert and Bubeck, 2010; Gabillon et al., 2011; Bubeck et al., 2013). Even-Dar et al. (2006) suggested the Successive Elimination (SE) and Median Elimination (ME) to find near optimal arms with provable sample complexity guarantees. These sample complexity guarantees typically scale linearly with the number of arms. In principal, one could naively reduce our problem to a pure exploration problem where we need to find an

(ϵ, δ) good arm. However, such naive reductions ignore any dependency information among the arms. The S-MCANS algorithm that we design builds on the SE algorithm but crucially exploits the matrix structure in the problem to give much better algorithms than a naive reduction.

2.3 Algorithms in the deterministic oracle model

Our deterministic oracle model is shown in Figure (1) and assumes the existence of a powerful, deterministic oracle that returns queried entries of the unknown matrix accurately. Our algorithm in this model, called MCANS, is shown in Figure (2). It

---

Model 1 Description of deterministic and stochastic oracle models

---

Figure (1)

while TRUE **do**

  Algorithm chooses a pair-of-indices (i_t, j_t).

  Algorithm receives the response y_t defined as follows

$$y_{t,\text{det}} = \boldsymbol{L}_{i_t,j_t} \text{ // if model is deterministic} \tag{2.1}$$

$$y_{t,\text{stoc}} = \text{Bern}(\boldsymbol{L}_{i_t,j_t}) \text{ // if model is stochastic} \tag{2.2}$$

  Algorithm stops if it has found a good approximation to the unknown matrix $\boldsymbol{L}$.

end while

---

is an iterative algorithm that determines which columns of the matrix are independent. MCANS maintains a set of indices (denoted as $\mathcal{C}$ in the pseudo-code) corresponding to independent columns of matrix $\boldsymbol{L}$. Initially $\mathcal{C} = \{1\}$. MCANS then makes a single pass over the columns in $\boldsymbol{L}$ and checks if the current column is independent of the columns in $\boldsymbol{C}$. This check is done in line 5 of Figure (2) and most importantly requires *only the principal sub-matrix*, of $\boldsymbol{L}$, indexed by the set $\mathcal{C} \cup \{c\}$. If the column passes this test then all the elements in this column i whose values have not been queried in the past are queried and the matrix $\hat{\boldsymbol{L}}$ is updated with these values. The test in line 5 is the column selection step of the MCANS algorithm and is justified by Proposition (2.3.1). Finally, once r independent columns have been chosen, we impute the matrix by using Nystrom extension. Nystrom based methods have been proposed in the past to handle large scale kernel matrices in the kernel based learning literature Drineas and Mahoney (2005); Kumar et al. (2012). The major difference between this work and ours is that the column selection procedure in our algorithms is deterministic, whereas in Nystrom methods columns are chosen at random. The following proposition simply follows from the fact that any principal submatrix of an PSD matrix is also PSD and hence admits an eigen-decomposition.

Algorithm 2 Matrix Completion via Adaptive Nystrom Sampling (MCANS)

Input: A deterministic oracle that takes a pair of indices (i, j) and outputs $\boldsymbol{L}_{i,j}$.
Output: $\hat{\boldsymbol{L}}$

Choose the pairs $(j, 1)$ for $j = 1, 2, \ldots, K$ and set $\hat{\boldsymbol{L}}_{j,1} = \boldsymbol{L}_{j,1}$. Also set $\hat{\boldsymbol{L}}_{1,j} = \boldsymbol{L}_{j,1}$
$\mathcal{C} = \{1\}$ {Set of independent columns discovered till now}
for $(c = 2;\ c \leftarrow c + 1;\ c \leq K)$ **do**
  Query the oracle for (c, c) and set $\hat{\boldsymbol{L}}_{c,c} \leftarrow \boldsymbol{L}_{c,c}$
  if $\sigma_{\min}\left(\hat{\boldsymbol{L}}_{\mathcal{C}\cup\{c\},\mathcal{C}\cup\{c\}}\right) > 0$ **then**
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{c\}$
    Query $\mathcal{O}$ for the pairs $(\cdot, c)$ and set $\hat{\boldsymbol{L}}(\cdot, c) \leftarrow \boldsymbol{L}(\cdot, c)$ and by symmetry $\hat{\boldsymbol{L}}(c, \cdot) \leftarrow \boldsymbol{L}(\cdot, c)$.
  end if
  if $(|\mathcal{C}| = r)$ **then**
    break
  end if
end for
Let $\boldsymbol{C}$ denote the tall matrix comprised of the columns of $\boldsymbol{L}$ indexed by $\mathcal{C}$ and let $\boldsymbol{W}$ be the principle submatrix of $\boldsymbol{L}$ corresponding to the indices in $\mathcal{C}$. Then, construct the Nystrom extension $\mathbf{L} = \boldsymbol{C}\boldsymbol{W}^{-1}\boldsymbol{C}^\top$.

Proposition 2.3.1. *Let $\mathbf{L}$ be any PSD matrix of size K. Given a subset $\mathcal{C} \subset \{1, 2, \ldots, K\}$, the columns of the matrix $\mathbf{L}$ indexed by the set $\mathcal{C}$ are independent iff the principal submatrix $\boldsymbol{L}_{\mathcal{C},\mathcal{C}}$ is non-degenerate, equivalently iff, $\lambda_{\min}(\boldsymbol{L}_{\mathcal{C},\mathcal{C}}) > 0$.*

It is not hard to verify the following theorem. The proof has been relegated to the appendix.

Theorem 2.3.2. *If $\boldsymbol{L} \in \mathbf{R}^{K\times K}$ is an PSD matrix of rank r, then the matrix $\hat{\boldsymbol{L}}$ output by the MCANS algorithm* (2) *satisfies $\hat{\boldsymbol{L}} = \boldsymbol{L}$. Moreover, the number of oracle calls made by MCANS is at most $K(r+1)$. The sampling algorithm* (2) *requires: $K + (K - 1) + (K - 2) + \ldots + (K - (r - 1)) + (K - r) \leq (r + 1)K$ samples from the matrix $\boldsymbol{L}$.*