6ZC-03

Computational Complexity of College Math Eigenvalue Problems

Yuki MORI*, Shuichi YUKITA
Graduate School of Computer and Information Sciences, Hosei University, Japan

1 Introduction

We propose a framework to evaluate computational complexity of college math problems, aiming to apply our framework to automatic generation of college math problems with controlled computational complexity. Providing students with suitable complex exercise problems is crucial to keeping them motivated and leading them to deeper understanding. Making such problems manually takes mathematics teachers's precious time which can otherwise be used to mentor students.

Many software tools and services for automatic generation of math problems are found on the Web, but all of them provide only materials of a high school level or below. In addition, no standardized methods are provided to evaluate and control the computational complexity of the generated problems. Compared to those popular tools, our framework is new in dealing with mathematics at a college level and in introducing suitable methods for the evaluation of computational complexity from learners' view.

The user chooses parameters such as the algebraic number field used in the calculation, the number of the calculation steps, the dimension of the matrix, and so on, which determine the computational complexity of the generated problem. The user chooses the problem category and provides the parameters that control the computational complexity like these. To avoid an excess freedom of choice for busy users, predefined sets of recommended parameters are stored in the system, which eases the user's choice. In addition to the problems of required complexity, their model answers are also generated. Eigenvalue problem is topics usually taught in linear algebra courses at engineering departments, but most textbooks do not give enough exercise problems. Therefore teachers should make additional exercise problems to be used in class, assignments, and term exams. In this paper, we present an automatic generator of diagonalization problems for Hermitian matrices to show the relevance of our framework.

2 Complexity from Learners' View

Rigorous concepts of complexity in various forms can be found in standard textbooks. We treat a different flavor of complexity, subjective complexity, where complexity is measured by the difficulty that learners feel. Designing exercise problems enough but not excessively complex is crucial to keeping the leaner motivated. We propose a new framework for estimating computational complexity and demonstrate its relevance by developing a framework for automatic generation of complexity-controlled exercise problems. With the new framework we can

1. control the number of the calculation steps,
2. limit the height of rational numbers involved in the whole calculation, and
3. handle algebraic numbers.

Computational complexity of generated problem is roughly defined by the sum of the height of rational numbers (the maximum absolute number of the denominator and numerator) appearing in the model solution of the problem. In the hope of extending our work to other math problems, we incorporated algebraic number fields in our system. The user can select the calculation field from the rational number field and other algebraic fields extended by irrational numbers, especially, quadratic irrational numbers, and 4th irrational numbers.

3 Automatic Generation of Eigenvalue Problems

Eigenvalue problems are usually taught in linear algebra courses at engineering departments. Eigenvalue problems appear in two forms: diagonalization of Hermitian matrix and Jordan canonicalization of linear transformations. In this paper, we deal with diagonalization of Hermitian matrix. The process of generating eigenvalue problems with controlled complexity is to

1. predefine unitary matrices,
2. generate eigenvalues,
3. generate n Hermitian matrices, and
4. select ones suitable for exercises.

We explain each step in detail. First, we generate almost entire set of tractable unitary matrices and classify them by algebraic number field. The number of calculation steps does not vary once the dimension is determined.

General Hermitian matrices can be generated by a diagonal matrix D and a unitary matrix U as

$$H = UDU^{\dagger} \tag{1}$$

where $U^\dagger$ is the conjugate transposed matrix of U. Equation (1) is rewritten as $D = U^\dagger HU$. The most difficult part is to generate the unitary matrix with the specified properties. However, the number of matrices suitable for this purpose is relatively small because the entries of those matrices should be taken from a given algebraic number field, and the heights of the involved rationals should be restricted, and further, all the column vectors should form an orthonormal system. Therefore it is possible to predefine almost entire sets of the unitary matrices which can be used to generate Hermitian matrices which can be diagonalized with specified complexity.

4 Generation of Unitary Matrices

This section describes the generation process of 3×3 unitary matrices by way of example. The procedure of generating a matrix consists of four major steps:

1. generate a unit column vector,
2. verify that all the entries belong to the given number field,
3. get an orthonormal basis out of other two linearly independent column vectors via the Gram-Schmidt procedure, and
4. verify again that all the entries of those basis vectors belong to the given number field.

In step 1, we generate various unit vectors that have the form in Equation (2).

$$\boldsymbol{e}_1{}^t = (\frac{i}{j\sqrt{k}}, \frac{l}{m\sqrt{n}}, \sqrt{1-((\frac{i}{j\sqrt{k}})^2+(\frac{l}{m\sqrt{n}})^2)}) \tag{2}$$

where i, j, l, and m are rational integers and k, n are $2, 3, 5, 7$, or 1. In step 2, all the irrational numbers are extracted such as $\sqrt{2}$, $\sqrt{3}$, and $\sqrt{-1}$ from the vector and matrix. We select the vectors whose entries belong to the specified number field. In step 3, e_1 and other 2 vectors are orthogonalized. Additional two vectors are only required to form a linearly independent triple together with e_1. So, for ease of calculation, we can take them from sparse matrices, where only the positions of nonzero entries are important.

We generated 3 patterns of unitary matrices by changing the position of the non-zero element of each vector. If the user wants a complex number field, one has to add $\sqrt{-1}$ in some entry of an initial vector. In step 4, the number field of the components is verified again. This step is necessary since the Gram-Schmidt orthogonalization involves taking square roots which may cause further algebraic extension of fields. We select matrices all whose entries have rationals of low heights in their subexpressions. This forms the basic set of tractable unitary matrices. Out of generated 500,000 unitary matrices in a preliminary stage, the filter selects 681 matrices according to the criterion described in later sections.

Though the basic set is relatively small (681), we can generate other tractable matrices by multiplying among themselves and by taking direct sums as follows: given two matrices of the same dimension

$$U_1 \text{ and } U_2 \in U(n), \tag{3}$$

we get

$$U_1U_2 \in U(n). \tag{4}$$

and given two matrices of possibly the different dimensions

$$U_1 \in U(m) \text{ and } U_2 \in U(n), \tag{5}$$

we get

$$U_1 \bigoplus U_2 \in U(m+n). \tag{6}$$

By using such methods, we can get 58583 unitary matrices.

5 Conclusion

Our contributions are as follows.

1. A new framework to evaluate and control the computational complexity from learners' view
2. To prove the relevance of our concepts

We developed an automatic generation system of eigenvalue problems based on our framework. The automatic generation of eigenvalue problems with controlled computational complexity is one of the sample implementation of proof of our framework. Controlling the complexity of eigenvalue problems involves restricting the number of the calculation steps, the height of the involved rationals, and algebraic number field appearing in the model solution of the problem. The small-scale experiments showed the relevant of our framework. It is important to make problems with adequate computational complexity to keep learners' motivation. Our framework helps teachers to prepare teaching materials and thereby to save their time for the interaction with their students.

Our technique is expected applicable to other subjects in linear algebra and analysis. Our system makes a path to strict quantitative control on various complexity from the learners' view. Future work includes the application of our methods to other areas such as differential equations, number theory and so on. We also should prove that this system has a positive effect on the engineering education at college level. Therefore, demonstration experiments in classroom will be scheduled in the next step of our research.

References

[1] Paul J.McCarthy (1996), Algebraic Extensions of Fields, Dover Publications, Inc., New York.