Experiment on Pseudo Relevance Feedback Method Using Taylor Formula at NTCIR-3 Patent Retrieval Task

Kazuaki KISHIDA
Surugadai University* / National Institute of Informatics
*698 Azu, Hanno, Saitama 357-8555 Japan
kishida@surugadai.ac.jp

Abstract

Pseudo relevance feedback is empirically known as a useful method for enhancing retrieval performance. For example, we can apply the Rocchio method, which is well-known relevance feedback method, to the results of an initial search by assuming that the top-ranked documents are relevant a priori. In this paper, for searching NTCIR-3 patent test collection through pseudo feedback, we try to employ two relevance feedback mechanism; (1) the Rocchio method, and (2) a new method that is based on Taylor formula of linear search functions. The test collection consists of near 700,000 records including full text of Japanese patent materials. Unfortunately, effectiveness of our pseudo feedback methods was not empirically observed at all in the experiment. One of the reasons may be that all words from the documents to be assumed as relevant were used without any selection process.

Keywords: *Pseudo relevance feedback, Patent retrieval, Rocchio method*

1 Introduction

Relevance feedback is widely recognized as an effective method for improving retrieval effectiveness in the context of interactive IR. As often pointed out, it is difficult for users to represent her/his own information needs into a well-defined set of search terms or statements. The resulting short or poor queries would bring them only unsatisfactory results. However, if a few relevant documents happen to be found by the search, we could automatically or manually extract some useful terms from the documents, and add them to the initial search expression. It is obviously expected that search effectiveness of the second search using the extended query will be improved significantly. This is a basic idea of relevance feedback.

Inevitably, for executing automatic relevance feedback, the system has to obtain relevance information, i.e., relevant or irrelevant documents, from the users interactively. However, some researchers have tried to employ relevance feedback techniques with no relevance information. The main purpose is to enhance search performance of retrieval models such as vector space without interaction between system and users for relevance information. The technique is usually called *pseudo relevance feedback* (or *automatic relevance feedback*), in which a standard feedback method (e.g., Rocchio method) is applied by assuming that top-ranked documents searched by the initial search are relevant *a priori*.

The purpose of the paper is to examine effectiveness of pseudo relevance feedback empirically by using the patent test collection of NTCIR-3. In particular, we attempts to compare search performance of the traditional Rocchio method with that of an alternative method proposed by Kishida[1]. Kishida[1] has reported that the alternative method outperforms slightly the Rocchio method in an experiment of normal relevance feedback mechanism (not pseudo feedback) employing NTCIR-1 test collection, which consists of about 330,000 bibliographic records of proceedings at conferences held in Japan[2]. We try to ascertain if the method has a positive effect in the case of pseudo relevance feedback.

The rest of this paper is organized as follows. In Section 2, the Rocchio method and the alternative method proposed by Kishida[1] will be introduced. In Section 3, outline of retrieval experiment using NTICIR-3 patent test collection will be shown. Finally, we will discuss the experiment results and implications in Section 4.

2 Relevance Feedback Methods

2.1 Rocchio method

The most typical approach to relevance feedback would be so-called the Rocchio method [3], which is based on the vector space model. The basic idea of this method is to add an average weight of each term within the set of relevant documents to the original (or initial) query vector, and to subtract an average weight within the set of irrelevant ones from the vector.

We denote a document vector by $\mathbf{d}_i = (w_{i1},..., w_{iM})^T$ where w_{ij} is a weight of a term t_j within the document d_i, and the original query vector $\mathbf{q} = (w_{q1},..., w_{qM})^T$ where w_{qj} is a weight of a term t_j within the query (M is total number of distinct terms in the collection and T indicates transposition). A modified query vector $\tilde{\mathbf{q}}$ is obtained by the formula

$$\tilde{\mathbf{q}} = \alpha\mathbf{q} + \frac{\beta}{|D_1|}\sum_{i:d_i \in D_1}\mathbf{d}_i - \frac{\gamma}{|D_0|}\sum_{i:d_i \in D_0}\mathbf{d}_i \qquad (1)$$

where D_1 is the set of relevant documents, D_0 is the set of irrelevant documents, and α , β , and γ are constants.

It has been empirically shown that the performance of the Rocchio method is fairly good [4], and in recent, many researchers have examined the method directly or indirectly [5-8]. Also, due to its effectiveness and simplicity, the Rocchio method has also been applied in other research areas, for example, image retrieval [9] or text categorization [10].

2.2 Feedback method using Taylor formula of retrieval function

Kishida[1] has proposed an alternative relevance feedback method, which is suitable for the situation that the degree of relevance is given as a numerical value, not dichotomous value (i.e., relevance or not), from actual users. Also, as another feature, Kishida[1] has suggested that the method is able to be applied to the Okapi formula [11] rationally as well as vector space model.

According to Kishida[1], details of the method will be explained in the rest of this section.

2.2.1 Retrieval model based on linear matching function

In the vector space model, typical formulas for determining the term weights w_{ij} and w_{qj} are respectively,

$$w_{ij} = \log x_{ij} + 1.0, \text{ and} \qquad (2)$$

$$w_{qj} = (\log x_{qj} + 1.0)\log\left(N/n_j\right), \qquad (3)$$

where x_{ij} is frequency of term t_j within the document d_i , x_{qj} is frequency of term t_j within the query, n_j is the number of documents including term t_j , and N is the total number of documents in the database [12]. For calculating the degree of similarity between a document vector $\mathbf{d}_i$ and the query vector $\mathbf{q}$, a cosine formula is normally used such that

$$s_i = \sum_{j=1}^{M} w_{ij} w_{qj} \Big/ \sqrt{\sum_{j=1}^{M} w_{ij}^2 \sum_{j=1}^{M} w_{qj}^2}, \qquad (4)$$

where s_i is a numerical score indicating the similarity or relevance probability of the document d_i given a query. The cosine formula (4) is a matching function of the vector model.

On the other hand, in the case of the Okapi formula,

$$s_i = \sum_{j=1}^{M}\left(\frac{3.0x_{ij}}{(0.5 + 1.5 l_i / \bar{l}) + x_{ij}} \times x_{qj} \log\frac{N - n_j + 0.5}{n_j + 0.5}\right), \qquad (5)$$

where

$$l_i = \sum_{j=1}^{M} x_{ij}, \text{ and } \bar{l} = N^{-1}\sum_{i=1}^{N} l_i ,$$

i.e., l_i is document length, and $\bar{l}$ is the average within the database. The formula (5) is one version of Okapi formula, and it should be noted that there are some different formulas according to its way of setting its parameters.

Kishida[1] has shown that we can represent uniformly these retrieval models as a linear function of vector,

$$\mathbf{s} = f(\mathbf{b}) = \mathbf{Ab} \qquad (6)$$

where $\mathbf{s} = (s_1,\ldots, s_N)^T$, f is a linear function of vector ($f : R^{M\times1} \to R^{N\times1}$) and $\mathbf{A}$ is a $N \times M$ matrix of which element a_{ij} is

$$a_{ij} = (\log x_{ij} + 1.0) \Big/ \sqrt{\sum_{j=1}^{M} (\log x_{ij} + 1.0)^2} \qquad (7)$$

in the case of vector space model (see (2) and (4)), or

$$a_{ij} = \frac{3.0x_{ij}}{(0.5 + 1.5 l_i / \bar{l}) + x_{ij}} \qquad (8)$$

in the case of Okapi formula (see (5)).

Also, $\mathbf{b}$ is a M dimensional vector of which element b_j （ $j = 1,\ldots, M$ ） is defined as

$$b_j = w_{qj} \Big/ \sqrt{\sum_{j=1}^{M} w_{qj}^2} \qquad (9)$$

where $w_{qj} = (\log x_{qj} + 1.0)\log\left(N/n_j\right)$ in the case of vector space model (see (3)), or

$$b_j = x_{qj} \log \frac{N - n_j + 0.5}{n_j + 0.5} \tag{10}$$

in the case of Okapi formula.

The most important thing is that the two very well known formulas for calculating document scores in ranked output can be represented into a fairly simple unified form (6).

2.2.2 Use of Taylor formula

In ranked output, documents are sorted in decreasing order of their scores s_i ($i = 1, \ldots, N$). This means that each s_i is assumed to indicate the degree of relevance. In other words, s_i is to be expected as an estimate of 'true' value of relevance degree r_i. Let $\mathbf{r} = (r_1, \ldots, r_N)^T$ be a vector representing the true relevance degree. By using these notations, we can describe operationally the purpose of retrieval system as "an estimation of a vector $\mathbf{s}$ that is the closest to the vector $\mathbf{r}$ given a search request."

Of course, $\mathbf{r}$ is unknown in real situations, but it is possible to get information on a part of $\mathbf{r}$ through relevance feedback process. For example, if the user replies a set of scores indicating each degree of relevance for top-ranked n documents after an initial search, the scores allow us to estimate a part of $\mathbf{r}$ corresponding to the n documents.

We denote the set of top-ranked n documents by X and a part of $\mathbf{r}$ corresponding to the set X by $\mathbf{r}_X$. In similar with (6), we can define that

$$\mathbf{s}_X = f_X(\mathbf{b}) = \mathbf{A}_X \mathbf{b}, \tag{11}$$

where $\mathbf{A}_X$ is a $n \times M$ matrix and $\mathbf{s}_X$ is an n dimensional vector, of which elements are a_{ij} and s_i respectively, where $d_i \in X$ ($f_X : R^{M\times 1} \rightarrow R^{n\times 1}$). If we establish a distance measure ϕ between $\mathbf{r}_X$ and $\mathbf{s}_X$, the purpose of relevance feedback can be formally described as follows: the relevance feedback aims at calculating $\tilde{\mathbf{b}}$ such that

$$\begin{aligned} \tilde{\mathbf{b}} &= \arg\min_{\mathbf{b}} \phi(\mathbf{r}_X, \mathbf{s}_X) \\ &= \arg\min_{\mathbf{b}} \phi(\mathbf{r}_X, f_X(\mathbf{b})). \end{aligned} \tag{12}$$

Then $\tilde{\mathbf{b}}$ is to be used for the second search.

The approach to calculating $\tilde{\mathbf{b}}$ suggested in Kishida[1] is to focus on the difference between the initial document scores $f_X(\mathbf{b})$ and the secondary scores $f_X(\tilde{\mathbf{b}})$, and to apply Taylor formula for obtaining a vector function $f_X(\tilde{\mathbf{b}})$, i.e.,

$$f_X(\tilde{\mathbf{b}}) = f_X(\mathbf{b}) + \frac{\partial f_X(\mathbf{b})}{\partial \mathbf{b}^T}(\tilde{\mathbf{b}} - \mathbf{b}) + K \tag{13}$$

where K is a residual term (see Harville[13]). If we employ (11) and assume that $\mathbf{r}_X$ is equal to $f_X(\tilde{\mathbf{b}})$ according to our target condition (12), we obtain

$$\tilde{\mathbf{b}} = \mathbf{b} + \mathbf{A}_X^{-1}(\mathbf{r}_X - \mathbf{s}_X). \tag{14}$$

(see Appendix for details).

The equation (14) contains an abnormal inverse matrix $\mathbf{A}_X^{-1}$, which is a $M \times n$ matrix and $\mathbf{A}_X^{-1}\mathbf{A}_X = \mathbf{I}_M$ where $\mathbf{I}_M$ is a $M \times M$ matrix of which all diagonal elements are 1 and others are 0. Using singular value decomposition (SVD), the transpose matrix of $\mathbf{A}_X$ can be represented as $\mathbf{A}_X^T = \mathbf{U}\Lambda\mathbf{V}^T$ where $\mathbf{U}$ is a $M \times n$ orthogonal matrix, Λ is a $n \times n$ diagonal matrix and $\mathbf{V}$ is a $n \times n$ orthogonal matrix. By employing the decomposition, we can finally represent (14) as

$$\tilde{\mathbf{b}} = \mathbf{b} + \mathbf{U}\Lambda^{-1}\mathbf{V}^T(\mathbf{r}_X - \mathbf{s}_X). \tag{15}$$

This is a final formula of our relevance feedback algorithm. For convenience, we call the algorithm the *Taylor formula based method*.

3 Outline of Retrieval Experiment

3.1 Test Collection

We use the patent test collection prepared for NTCIR-3 project [14], which consists of over 690,000 records containing full text of Japanese patent materials. All 31 queries in Japanese were used for the experiment in this paper.

3.2 Procedure and type of runs

The procedure of the experiment is as follows:

(a) Initial search: two initial search runs are carried out, i.e., the first is based on vector space model from (2) to (4) and the second is Okapi formula (5). We denote the initial search runs by VECTOR and OKAPI, respectively.

(b) Query modification by pseudo relevance feedback; initial queries are modified by assuming that top-ranked n documents of each initial run are relevant. In this paper we set $n = 10$.

  (i) In the case of vector space model, the Rocchio method (1) is applied. The run is denoted as ROCCHIO.

  (ii) In the case of the Okapi formula, the Taylor formula based method (15) is employed. We denote this run TAYLOR.

(c) Second search: each modified query is used for the second run.

(i) In the case of ROCCHIO, modified queries are matched with document vectors by cosine formula (4).

(ii) In the case of TAYLOR, the linear function (6) is used for matching operation.

Furthermore, we discern two kinds of run according to query (topic) fields used for run; (I) <ARTICLE> and <SUPPLEMENT> fields and (II) <DESCRIOTION> and <NARRATIVE> fields. As a result, in the experiment, six runs in total were executed as shown in Table 1. It should be noted that Okapi-none runs (i.e., normal Okapi formula without any feedback) were added as a baseline for evaluating pseudo feedback methods.

Table 1 Search runs in the experiment

initial run	feedback	Topic fields	
		<A><S>*	<D><N>**
OKAPI	TAYLOR	Run1	Run2
VECTOR	ROCCHIO	Run3	Run4
OKAPI	none	Run5	Run6

*<A>:<ARTICLE>, <S>:<SUPPLEMENT>
**<D>:<DESCRIPTION>,<N>:<NARRATIVE>

3.3 Implementation of pseudo relevance feedback method

As to the Rocchio method, we can directly use (1) for pseudo relevance feedback by assuming that the set of top-ranked n documents equals with D_1, and D_0 is empty. In the experiment, we suppose $\alpha = 8$, $\beta = 16$ and $\gamma = 0$.

In the case of Taylor formula based method, we need to determine the value of variable r_i in the equation (15), which means 'true' degree of relevance ($i = 1,...,n$). A simple way is to assume a simple linear function such that

$$r_i = As_i + B, \tag{16}$$

and to estimate each relevance degree r_i from the corresponding document score s_i calculated for the initial search.

let $r_{\max}$ and $s_{\max}$ be the maximum values of r_i and s_i in the set of top-ranked n documents, respectively. Similarly, the minimum values of r_i and s_i are denoted as $r_{\min}$ and $s_{\min}$, respectively. Thus, the constants A and B in (16) are determined as a solution of equations,

$$\begin{cases} r_{\max} = As_{\max} + B \\ r_{\min} = As_{\min} + B \end{cases}. \tag{17}$$

It is easy to show that

$$A = (r_{\max} - r_{\min})/(s_{\max} - s_{\min}), \tag{18}$$

$$B = (s_{\max} r_{\min} - r_{\max} s_{\min}) / (s_{\max} - s_{\min}). \tag{19}$$

To employ this method for estimating constants, we have to determine values of $r_{\max}$ and $r_{\min}$, but there is no reasonable way in the context of pseudo feedback. Thus, as a trial, we set that $r_{\max} = 2 \times s_{\max}$ and $r_{\min} = s_{\max}$ in the experiment.

In both of the Rocchio method and Taylor formula based method, no tern selection that extracts 'useful' words from the n documents according to its weight is executed.

3.4 Segmentation of Japanese text

The patent test collection for NTCIR-3 basically consists of documents written in Japanese language, and query statements are also in Japanese. As well known, Japanese text has no explicit boundary between terms unlike English. Thus we need to segment the text into a set of terms automatically for indexing documents and query. In this paper, each term is identified through an operation of matching strings in the text with entries included in a machine-readable dictionary. We used the dictionary of ChaSen [15] and selected as an index term the longest entry matched with a portion of text (longest-matching method). Also, two heuristic rules were applied additionally; (A) unmatched string is decomposed according to change of type of characters, and (B) for identifying compound words as content-bearing terms, a pair of adjacent two terms

Table 2 Average precision and R-precision of each run

Topic Fields	initial run	feedback	Average precision	R-precision
<ARTICLE> <SUPPLEMENT>	OKAPI	TAYLOR	0.1152	0.1421
	VECTOR	ROCCHIO	0.1281	0.1565
	OKAPI	none	0.1282	0.1565
<DESCRIPTION> <NARRATIVE>	OKAPI	TAYLOR	0.1370	0.1820
	VECTOR	ROCCHIO	0.1581	0.1896
	OKAPI	none	0.1583	0.1813

Note: The values in the table are calculated based on both of relevant and partially relevant documents.

pair of adjacent two terms identified by dictionary matching or rule (A) is automatically combined into a compound word.

In the experiment, index terms were extracted from only title and claim fields in the text of each document. This means that full text included in the records was not used for search. However, it is anticipated that most of important terms are contained in the title and claim fields. In the term of subject search, limiting the fields for indexing does not necessarily cause low performance.

3.5 The System

In the experiment, all tasks were executed on a personal computer SONY VAIO PCV-LX53/BP (CPU:1.50GHz, MEMORY:256MB, HDD:80GB) using Microsoft Visual C++ on Windows XP.

For executing runs, an inverted file of indexing term and some other files were constructed using the technique of B-tree.

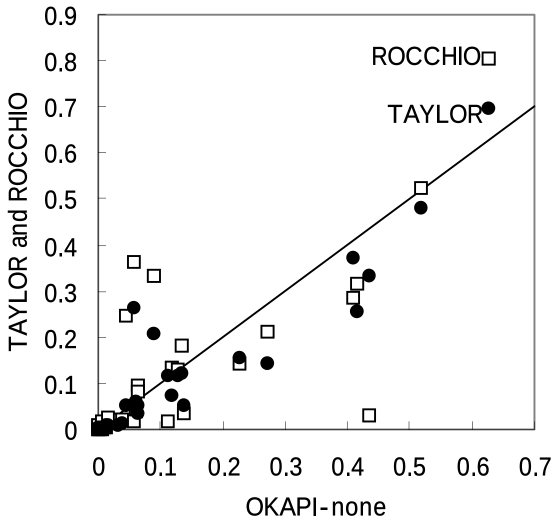

Figure 1 Topic-by-topic analysis (average precision)

4 Results and Discussion

4.1 Basic Statistics

In our indexing phase, 697,262 records were processed and average length of documents is 393.32.

4.2 Overall Performance

Table 2 shows search performance of each run. Unfortunately, pseudo relevance feedback using relevance feedback techniques has no effect on the performance. It seems that there are no statistically significant differences between any pairs of runs. However, performance of Taylor formula based method may be slightly low.

4.3 Topic-by-topic analysis

Figure 1 is a plot of values of average precision by topic. We can compare the Taylor formula based method (OKAPI-TAYLOR) and the Rocchio method (VECTOR-ROCCHIO) with normal Okapi formula (OKAPI-none), in level of each topic. It should be noted that, in Figure 1, square indicates ROCCHIO and circle TAYLOR.

Figure 1 shows that for most of topics, normal Okapi formula with no pseudo feedback outperforms the Rocchio method and Taylor formula based method although there are a few topics for which the Rocchio and Taylor dominate.

4.4 Discussion

Adding all words that appear in 10 top-ranked documents would cause the failure of our pseudo feedback technique. We should have attempted to select significant words by a method proposed in the literatures on pseudo relevance feedback. A simple method would be to add only some top-ranked words in the decreasing order of term weights (e.g., tf-idf). Another method is to assume ‘irrelevant’ documents as well as relevant ones and to take both sets of documents into account in the process of pseudo feedback.

Unfortunately, the failure prevents us from comparing empirically effectiveness of the Taylor formula based method with that of traditional Rocchio method. As shown in Table 2, it seems that there is no statistically significant difference between the two methods within the range of results obtained from our experiment.

5 Concluding remarks

An alternative feedback method based on Taylor formula proposed by Kishida [1] has unique characteristics as follows;

(a) the method is suitable when the degree of relevance is represented as a continuous value, not dichotomous value.

(b) the method is applicable to a large class of retrieval models including vector space model and the Okapi weighting.

As to (b), in the model, equation for calculating document score has to be a linear function of query vectors.

As discussed above, the new method did not work well as a method for pseudo relevance feedback in the experiment. Further study would be needed for applying such relatively complicated methods to pseudo relevance feedback.

Acknowledgement

The author would like to appreciate organizers of NTICR-3 patent retrieval task for preparing a valuable test collection.

References

[1] K. Kishida. Feedback method for document retrieval using numerical values on relevance given by users. In *IPSJ SIG Notes Fundamental Infology*, No.61, pages 189-196, 2001. (*In Japanese*)

[2] N. Kando, and T. Nozue eds. *Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition*. Tokyo, National Center for Scientific Information Systems. http://www.nii.ac.jp/ntcir/

[3] J. J. Rocchio, J.J. Jr. Relevance feedback in information retrieval. In G. Salton ed., *The SMART Retrieval System: Experiments in Automatic Document Processing*, Englewood Cliffs, New Jersey, Prentice-Hall, pages 313-323, 1971.

[4] G. Salton, and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4): 288-297, 1990.

[5] P. Sarinivasan. Query expansion and MEDLINE. *Information Processing and Management*, 32(4): 431-443, 1996.

[6] J. H. Lee. Combining the evidence of different relevance feedback methods for information retrieval. *Information Processing and Management*, 34(6): 681-691, 1998.

[7] R. Mandala, T. Tokunaga and H. Tanaka. Query expansion using heterogeneous thesauri. *Information Processing and Management*, 36, 361-378, 2000.

[8] Makoto Iwayama. Relevance feedback with a small number of relevance judgments: incremental relevance feedback vs. Document clustering. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 10-16, 2000.

[9] G. Ciocca and R. Schettini, R. A relevance feedback mechanism for content-based image retrieval. *Information Processing and Management*, 35: 605-632, 1999.

[10] Marie-Francine Moens and Jos Dumortier. Text categorization: the assignment of subject descriptors to magazine articles. *Information Processing and Management*, 36(6):841-861, 2000.

[11] S. Robertson, et al. Okapi at TERC-3. In D.K. Harman ed. *Overview of the Third Text Retrieval Conference (TREC-3)*, pages 109-126, 1995.

[12] C. Buckley, J. Allan and G. Salton. Automatic routing and ad-hoc retrieval using SMART: TREC2. In D.K. Harman ed. *The Second Text Retrieval Conference (TREC2)* , pages 45-55, 1994.

[13] D. A. Harville. *Matrix Algebra from a Statistician's Perspective*. New York, Springer, 1994.

[14]http://research.nii.ac.jp/ntcir/workshop/patent/

[15] Y. Matsumoto, A. Kitauchi, T. Yamashita, Y. Hirano, H. Matsuda and M. Asahara. Japanese morphological analysis system ChaSen version 2.0 manual 2nd ed. Technical Report NAIST-IS-TR99009, NAIST, 1999. http://cl.aist-nara.ac.jp/lab/nlt/chasen/

Appendix

If we assume a linear function (11), then

$$\frac{\partial f_X(\mathbf{b})}{\partial \mathbf{b}^T} = \frac{\partial(\mathbf{A}_X \mathbf{b})}{\partial \mathbf{b}^T} = \mathbf{A}_X ,$$

which is a well known result in the field of linear algebra [13]. Therefore (11) becomes that

$$f_X(\tilde{\mathbf{b}}) = f_X(\mathbf{b}) + \mathbf{A}_X(\tilde{\mathbf{b}} - \mathbf{b})$$

(it should be noted that $K = 0$).

By following our assumption that $\mathbf{r}_X$ is equal to $f_X(\tilde{\mathbf{b}})$ and noting that $f_X(\mathbf{b}) = \mathbf{s}_X$,

$$\mathbf{r}_X = \mathbf{s}_X + \mathbf{A}_X(\tilde{\mathbf{b}} - \mathbf{b})$$

(this assumption means that $\phi = 0$), we can obtain

$$\mathbf{A}_X(\tilde{\mathbf{b}} - \mathbf{b}) = \mathbf{r}_X - \mathbf{s}_X . \quad \text{(A.1)}$$

The equation (14) is easily derived from (A.1).

By using singular value decomposition we can obtain that $\mathbf{A}_X^T = \mathbf{U}\Lambda\mathbf{V}^T$. The transposition is

$$\mathbf{A}_X = \left(\mathbf{A}_X^T\right)^T = \left(\mathbf{U}\Lambda\mathbf{V}^T\right)^T = \mathbf{V}\Lambda\mathbf{U}^T \quad \text{(A.2)}$$

because $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrixes and Λ is a diagonal matrix. Substituting (A.2) into (A.1), we finally obtain that

$$\mathbf{V}\Lambda\mathbf{U}^T(\tilde{\mathbf{b}} - \mathbf{b}) = \mathbf{r}_X - \mathbf{s}_X$$

$$\tilde{\mathbf{b}} - \mathbf{b} = \mathbf{U}\Lambda^{-1}\mathbf{V}^T(\mathbf{r}_X - \mathbf{s}_X)$$

$$\therefore \tilde{\mathbf{b}} = \mathbf{b} + \mathbf{U}\Lambda^{-1}\mathbf{V}^T(\mathbf{r}_X - \mathbf{s}_X) .$$

The last equation is equivalent to (15).