Similarity and Recommender Systems

Hiroshi Shimodaira*

January-March 2020

In this chapter we shall look at how to measure the *similarity* between items. To be precise we'll look at a measure of the *dissimilarity* or *distance* between feature vectors, as well as a direct measurement of similarity. We shall then see how such measures can be used to suggest items in collaborative filtering and recommender systems.

2.1 Distances

We often want to compare two feature vectors, to measure how different (or how similar) they are. We hope that similar patterns will behave in a similar way. For example if we are performing handwriting recognition, a low distance between digit feature vectors (derived from images) might indicate that they should be given the same label. If we are building a recommender system for an online shop, similar user feature vectors (derived from their purchasing or browsing histories) might indicate users with similar tastes. The distance between two items depends on both the representation used by the feature vectors and on the distance measure used.

If the feature vectors are binary (i.e., all elements are 0 or 1) then the *Hamming distance* is a possible distance measure. For real valued vectors, the Euclidean distance is often used: this is familiar from 2- or 3-dimensional geometry, and may also be generalised to higher dimensions.

2.1.1 Hamming distance

The Hamming distance between two binary sequences of equal length is the number of positions for which the corresponding symbols are different. For example the Hamming Distance between 10101010 and 11101001 is 3.

2.1.2 Euclidean distance

The Euclidean distance is already familiar to you from 2- and 3-dimensional geometry. The Euclidean distance $r_2(\mathbf{u}, \mathbf{v})$ between two 2-dimensional vectors $\mathbf{u} = (u_1, u_2)^T$ and $\mathbf{v} = (v_1, v_2)^T$ is given by:

$$r_2(\mathbf{u}, \mathbf{v}) = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2} = \sqrt{\sum_{k=1}^{2} (u_k - v_k)^2}\,. \tag{2.1}$$

(Superscript T symbolises the transpose operation so we can write a column vector easily within a line, e.g., $\binom{x}{y}$ as $(x, y)^T$.)

Generalising to higher dimensions, the *Euclidean distance* between two D-dimensional vectors $\mathbf{u} = (u_1, u_2, u_3, \ldots, u_D)^T$ and $\mathbf{v} = (v_1, v_2, v_3, \ldots, v_D)^T$ is given by:

$$r_2(\mathbf{u}, \mathbf{v}) = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2 + \cdots + (u_D - v_D)^2} = \sqrt{\sum_{d=1}^{D} (u_d - v_d)^2}\,. \tag{2.2}$$

It is often the case that we are not interested in the precise distances, just in a comparison between distances. For example, we may be interested in finding the closest point (nearest neighbour) to a point in a data set. In this case it is not necessary to take the square root.

Other distance measures are possible for example the Manhattan (or city-block) metric:

$$r_1(\mathbf{u}, \mathbf{v}) = |u_1 - v_1| + |u_2 - v_2| + \cdots + |u_D - v_D| = \sum_{d=1}^{D} |u_d - v_d|\,. \tag{2.3}$$

The notation $|a|$ indicates the absolute value of a. More generally it is possible to use other powers, giving rise to a more general form (known as the p-norm or L^p-norm):

$$r_p(\mathbf{u}, \mathbf{v}) = \left(\sum_{d=1}^{D} |u_d - v_d|^p \right)^{1/p}\,. \tag{2.4}$$

We will be mostly concerned with the familiar Euclidean distance in this course.

2.2 A simple recommender system

Table 2.1 shows the ratings that a few well-known US film critics gave to a small group of movies. We shall use this data to develop a simple recommender system. Unlike a realistic system, in this case every person has rated every film.

We can represent this data as a matrix, whose rows correspond to a particular critic and whose columns correspond to a film, and where the value of each element (c, m) is the score given by critic c to film (movie) m:

$$\begin{pmatrix} 3 & 7 & 4 & 9 & 9 & 7 \\ 7 & 5 & 5 & 3 & 8 & 8 \\ 7 & 5 & 5 & 0 & 8 & 4 \\ 5 & 6 & 8 & 5 & 9 & 8 \\ 5 & 8 & 8 & 8 & 10 & 9 \\ 7 & 7 & 8 & 4 & 7 & 8 \end{pmatrix}.$$

If we want a feature vector per critic then we can just take the rows:

$$\begin{aligned} \mathbf{x}_1 &= (3, 7, 4, 9, 9, 7)^T \\ \mathbf{x}_2 &= (7, 5, 5, 3, 8, 8)^T \\ &\vdots \\ \mathbf{x}_6 &= (7, 7, 8, 4, 7, 8)^T\,, \end{aligned}$$

where $\mathbf{x}_1$ corresponds to David Denby, $\mathbf{x}_2$ corresponds to Todd McCarthy, and so on.

	Australia	*Body of Lies*	*Burn After Reading*	*Hancock*	*Milk*	*Revolutionary Road*
David Denby (New Yorker)	3	7	4	9	9	7
Todd McCarthy (Variety)	7	5	5	3	8	8
Joe Morgenstern (Wall St Journal)	7	5	5	0	8	4
Claudia Puig (USA Today)	5	6	8	5	9	8
Peter Travers (Rolling Stone)	5	8	8	8	10	9
Kenneth Turan (LA Times)	7	7	8	4	7	8

Table 2.1: Ratings given to six movies by six film critics (from `http://www.metacritic.com`).

If the critics have each reviewed a set of M films (movies), then we can imagine each critic defining a point in an M-dimensional space, given by that critic's review scores. The points are hard to visualise in more than three dimensions (three films). Figure 2.1 shows a two dimensional version in which the six critics are placed in a space defined by their reviews of two films. As more users are made available they can be plotted on the chart, according to their ratings for those films. We make the assumption that the closer two people are in this review space, then the more similar are their tastes.

Consider a new user who rates *Hancock* as 2, and *Revolutionary Road* as 7. This user is also shown in the review space in Figure 2.1. Based on these two films, to which critic is the user most similar? It is easy to see from the graph that the closest critic to the user is McCarthy. In this 2-dimensional case Euclidean distance is:

$$r_2(\text{User}, \text{McCarthy}) = \sqrt{(2-3)^2 + (7-8)^2} = \sqrt{2} \approx 1.4$$

We can go ahead and use Equation (2.1) to compute the Euclidean distances between six critics in the 6-dimensional review space:

	Denby	McCarthy	Morgenstern	Puig	Travers	Turan
Denby		7.7	10.6	6.2	5.2	7.9
McCarthy	7.7		5.0	4.4	7.2	3.9
Morgenstern	10.6	5.0		7.5	10.7	6.8
Puig	6.2	4.4	7.5		3.9	3.2
Travers	5.2	7.2	10.7	3.9		5.6
Turan	7.9	3.9	6.8	3.2	5.6	

Thus the two closest critics, based on these films, are Claudia Puig and Kenneth Turan.

Consider a new user who has not seen Hancock, Australia or Milk, but has supplied ratings to the other three films:

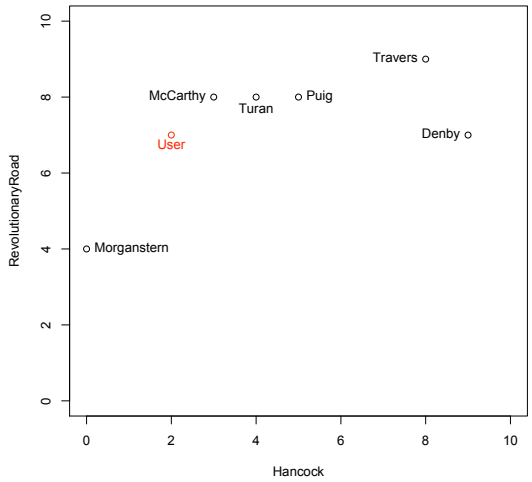

Figure 2.1: Plot of critics in a 2-dimensional review space for *Hancock* and *Revolutionary Road*. For the pair of films under consideration the Euclidean distance between two points provides a measure of how different two reviewers are.

	Body of Lies	*Burn After Reading*	*Revolutionary Road*
User2	6	9	6

Using a 3-dimensional space defined by the films that User2 has rated, we can compute the distances to each critic:

Critic	$r_2(\text{critic}, \text{user2})$
Denby	$\sqrt{27} \approx 5.2$
McCarthy	$\sqrt{21} \approx 4.6$
Morgenstern	$\sqrt{21} \approx 4.6$
Puig	$\sqrt{5} \approx 2.2$
Travers	$\sqrt{14} \approx 3.7$
Turan	$\sqrt{6} \approx 2.4$

Thus, based on these three films, User2 is most similar to Claudia Puig. Can we use this information to build a simple recommender system? Or, more specifically, can we use this information to decide which film out of *Milk*, *Hancock* and *Australia* the system should recommend to User2 based on their expressed preferences?

We would like to rely on the most similar critics, so we convert our distance measure into a similarity measure:

$$\text{sim}(\mathbf{u}, \mathbf{v}) = \frac{1}{1 + r_2(\mathbf{u}, \mathbf{v})}\,. \tag{2.5}$$

We have chosen an ad hoc measure of similarity based on Euclidean distance. However, it has some desirable properties: a distance of 0 corresponds to a similarity of 1 (the largest value it can take); a distance of ∞ corresponds to a similarity of 0 (the smallest it can take). We now use this measure to list the critics' similarity to User2:

Critic	sim(critic, user2)
Denby	0.16
McCarthy	0.18
Morgenstern	0.18
Puig	0.31
Travers	0.21
Turan	0.29

One way to make a recommendation for User2 would be to choose the most similar critic, and then to choose the movie that they ranked most highly. This approach has a couple of drawbacks: (1) it does nothing to smooth away any peculiarities of that critic's rankings; and (2) in the case when recommender systems are applied (e.g., online shops) the population of 'critics' may be very large indeed, and there may be quite a large number of similar user profiles. An alternative way to make a ranking for User2 would be to weight the rating of each critic by the similarity to User2. An overall score for each film can be obtained by summing these weighted rankings. If u is the user, and we have C critics, then the estimated score given to film m by u, $\mathrm{sc}_u(m)$, is obtained as follows:

$$\mathrm{sc}_u(m) = \frac{1}{\sum_{c=1}^{C} \mathrm{sim}(\tilde{\mathbf{x}}_u, \tilde{\mathbf{x}}_c)} \sum_{c=1}^{C} \mathrm{sim}(\tilde{\mathbf{x}}_u, \tilde{\mathbf{x}}_c) \cdot x_{cm}\,, \tag{2.6}$$

where $\tilde{\mathbf{x}}_u$ is a vector of ratings for the films seen by u, and $\tilde{\mathbf{x}}_c$ is the vector of corresponding film ratings from critic c. In this example, $\tilde{\mathbf{x}}_u$ and $\tilde{\mathbf{x}}_c$ are 3-dimensional vectors, whereas the original $\mathbf{x}_c$ is a 6-dimensional vector. The term $1/\sum_{c=1}^{C} \mathrm{sim}(\tilde{\mathbf{x}}_u, \tilde{\mathbf{x}}_c)$ is used to normalise the weighted sum of scores to estimate the user's score for the film.

We can compute an estimate of User2's score for each film using Equation (2.6). We'll make the computation explicit in a table:

		Australia		*Hancock*		*Milk*	
	Similarity	Score	Sim · Score	Score	Sim · Score	Score	Sim · Score
Denby	0.16	3	0.48	9	1.44	9	1.44
McCarthy	0.18	7	1.26	3	0.54	8	1.44
Morgenstern	0.18	7	1.26	0	0.00	8	1.44
Puig	0.31	5	1.55	5	1.55	9	2.79
Travers	0.21	5	1.05	8	1.68	10	2.10
Turan	0.29	7	2.03	4	1.16	7	2.03
Total	1.33		7.63		6.37		11.24
Est. Score			5.7		4.8		8.5

So the recommender system would propose *Milk* to User2. But more than just proposing a single film, it provides an estimate of the rating that User2 would provide to each film based on User2's ratings of other films, the estimated similarity of User2 to each critic, and the ratings of the critics to films unseen by User2.

Some questions:

- What do we do if not all the critics have seen the same set of movies? Are the distance/similarity methods between different pairs of people comparable if they are computed across different spaces (i.e., different sets of ratings)? Is there something we can do to make them more comparable?
- How do we deal with the fact that some critics may score more highly on average than others? Or that some critics have a wider spread of scores than others?

We can also solve the transposed problem: instead of measuring the similarity between people, we can measure the similarity between films. In this case we will have a space whose dimension is the number of critics, and each point in the space corresponds to a film. Transposing the previous data matrix:

$$\begin{pmatrix} 3 & 7 & 4 & 9 & 9 & 7 \\ 7 & 5 & 5 & 3 & 8 & 8 \\ 7 & 5 & 5 & 0 & 8 & 4 \\ 5 & 6 & 8 & 5 & 9 & 8 \\ 5 & 8 & 8 & 8 & 10 & 9 \\ 7 & 7 & 8 & 4 & 7 & 8 \end{pmatrix}^T = \begin{pmatrix} 3 & 7 & 7 & 5 & 5 & 7 \\ 7 & 5 & 5 & 6 & 8 & 7 \\ 4 & 5 & 5 & 8 & 8 & 8 \\ 9 & 3 & 0 & 5 & 8 & 4 \\ 9 & 8 & 8 & 9 & 10 & 7 \\ 7 & 8 & 4 & 8 & 9 & 8 \end{pmatrix},$$

each row corresponds to a feature vector for a film:

$$\mathbf{w}_1 = (3, 7, 7, 5, 5, 7)^T$$
$$\vdots$$
$$\mathbf{w}_6 = (7, 8, 4, 8, 9, 8)^T\,,$$

where $\mathbf{w}_1$ corresponds to *Australia*, $\mathbf{w}_2$ corresponds to *Body of Lies*, and so on.

We can then go ahead and compute the distances between films in the space of critics' ratings, in a similar way to before:

	Australia	*Body of Lies*	*Burn After Reading*	*Hancock*	*Milk*	*Revolutionary Road*
Australia		5.8	5.3	10.9	8.9	7.2
Body of Lies	5.8		3.7	6.6	5.9	4.0
Burn After Reading	5.3	3.7		8.9	7.0	4.5
Hancock	10.9	6.6	8.9		10.9	8.4
Milk	8.9	5.9	7.0	10.9		4.8
Revolutionary Road	7.2	4.0	4.5	8.4	4.8	

Therefore if a user, for whom we have no history of ratings chooses *Body of Lies*, then based on our stored critics' ratings we would recommend *Burn After Reading* and *Revolutionary Road* as the two most similar films.

To summarise:

1. We represented the rating data from C critics about M films (movies) as a $C \times M$ matrix.
2. Each row of this data matrix corresponds to a critic's feature vector in 'review space'.
3. We can compute the distance between feature vectors, to give measure of dissimilarity between critics.

4. We can use this information to construct a simple recommender system.
5. If we take the transpose of the data matrix, then each row corresponds to a film's feature vector; distance measures between these vectors correspond to dissimilarity between films.

2.3 Similarity using correlation

2.3.1 Normalisation

So far our estimate of similarity has been based on the Euclidean distance between feature vectors in a review space. But this distance is not well normalised. For example, two critics may rank a set of films in the same order, but if one critic gives consistently higher scores to all movies than the other, then the Euclidean distance will be large and the estimated similarity will be small. In the data we have been working with (Table 2.1) some critics do give higher scores on average: the mean review ratings per critic range from 4.8 to 8.0.

One way to *normalise* the scores given by each critic is to transform each score into a *standard score*[1]. The standard scores are defined such that the set of scores given by each critic have the same sample mean and sample standard deviation. We first compute the sample mean and sample standard deviation for each critic. Consider an M-dimensional feature vector corresponding to critic c, $\mathbf{x}_c = (x_{c1}, x_{c2}, \ldots, x_{cM})$, where x_{cm} is critic c's rating for movie m. We can compute the sample mean $\bar{x}_c$ and sample standard deviation s_c for critic c as follows[2]:

$$\bar{x}_c = \frac{1}{M}\sum_{m=1}^{M} x_{cm} \tag{2.7}$$

$$s_c = \sqrt{\frac{1}{M-1}\sum_{m=1}^{M}(x_{cm} - \bar{x}_c)^2}\,. \tag{2.8}$$

We then use these statistics to normalise x_{cm} (the critic c's score for movie m) to a standard score,

$$z_{cm} = \frac{x_{cm} - \bar{x}_c}{s_c}\,. \tag{2.9}$$

The z scores for a critic c are normalised with a mean of 0 (obtained by subtracting the mean score from the x_c scores) and a sample standard deviation of 1 (obtained by dividing by the sample standard deviation of the x_c scores). Thus using these normalised scores for each critic removes the offset effect of differing means and the spread effect of differing variances.

2.3.2 Pearson Correlation Coefficient

There are many other ways that we could measure similarity. One measure that has been used a lot in data mining, and collaborative filtering in particular, is a measure based on the *correlation* between users' ratings. Rather than considering the distance between feature vectors as a way to estimate similarity, we can consider the correlation between the critics scores. Figures 2.2 and 2.3 each plot films in terms of the ratings of two specified critics, along with a best fit straight line. If the ratings of the two critics are closely related (similar) then the best-fit line will (almost) touch every item; if the films are generally far from the best fit line then the review scores are not well associated (dissimilar). We can see that the scores of Travers and Denby (Figure 2.2) are much better correlated than the scores of McCarthy and Denby (Figure 2.3); although Denby has lower ratings on average than that of Travers, this does not affect the correlation.

[1] Standard scores are also called *z-values*, *z-scores*, and *normal scores*.

[2] The 'sample standard deviation' is the square-root of the 'sample variance'. Different books give different definitions of the 'sample variance', which is an estimate of the 'true' variance of a population from a limited number of samples. Equation (2.8) uses an 'unbiased' estimate of the true variance. Another version of variance replaces '$M-1$' with 'M', which is normally called a 'population variance', meaning the samples you have got are assumed to be the whole population rather than its subset. This way of calculating variance gives a 'biased' estimate if it is used as an estimate of the true variance of a population. For large sample sizes, as commonly-found in machine learning, the difference usually doesn't matter. Matlab/Octave use the '$M-1$' estimator by default, and Python's `scipy` package uses the 'M' estimator by default, although both have options to use the other.

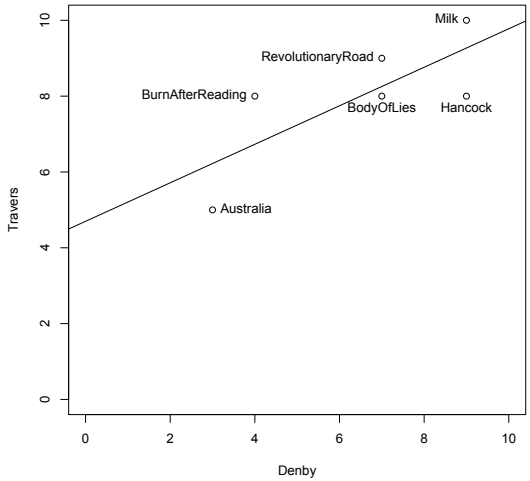

Figure 2.2: Plot of films in a 2-dimensional space defined by the ratings of David Denby and Peter Travers. The best fit straight line for these points is also plotted.

To estimate the correlation between two sets of scores we use the *Pearson Correlation Coefficient*. To estimate this we first normalise the scores for each critic to stand scores, using Equations (2.7), (2.8) and (2.9). We can then compute the Pearson correlation coefficient between critics c and d, r_{cd} as [3] :

$$r_{cd} = \frac{1}{M-1}\sum_{m=1}^{M} z_{cm}\, z_{dm} \tag{2.10}$$

$$= \frac{1}{M-1}\sum_{m=1}^{M}\left(\frac{x_{cm} - \bar{x}_c}{s_c}\right)\left(\frac{x_{dm} - \bar{x}_d}{s_d}\right). \tag{2.11}$$

If z_{cm} tends to be large when z_{dm} is large and z_{cm} tends to be small when z_{dm} is small, then the correlation coefficient will tend towards 1. If z_{cm} tends to be large when z_{dm} is small and z_{cm} tends to be small when z_{dm} is large, then the correlation coefficient will tend towards -1. If there is no relation between critics c and d, then their correlation coefficient will tend towards 0.

In the above examples, the correlation between Denby and Travers is 0.76 and the correlation between

[3] If we define *sample covariance* as $s_{cd} = \frac{1}{M-1}\sum_{m=1}^{M}(x_{cm} - \bar{x}_c)(x_{dm} - \bar{x}_d)$, Equation (2.11) can be rewritten as $r_{cd} = \frac{s_{cd}}{s_c s_d}$.

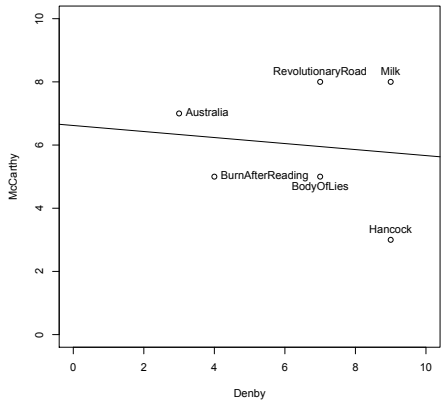

Figure 2.3: Plot of films in a 2-dimensional space defined by the ratings of David Denby and Todd McCarthy. The best fit straight line for these points is also plotted

Denby and McCarthy is −0.12. For comparison the similarities based on Euclidean distance are 0.16 and 0.11.

Exercise: Using the Python function you will write in Tutorial 3, show that the similarities between critics obtained using the correlation coefficient are as below. Compare these similarities with those obtained using the Euclidean distance.

Similarities based on Euclidean distances between critics:

	Denby	McCarthy	Morgenstern	Puig	Travers	Turan
Denby	1	0.11	0.09	0.14	0.16	0.11
McCarthy	0.11	1	0.17	0.19	0.12	0.20
Morgenstern	0.09	0.17	1	0.12	0.09	0.13
Puig	0.14	0.19	0.12	1	0.20	0.24
Travers	0.16	0.12	0.09	0.20	1	0.15
Turan	0.11	0.20	0.13	0.24	0.15	1

Pearson correlation coefficient similarities between critics

	Denby	McCarthy	Morgenstern	Puig	Travers	Turan
Denby	1	-0.12	-0.36	0.14	0.76	-0.55
McCarthy	-0.12	1	0.75	0.53	0.18	0.61
Morgenstern	-0.36	0.75	1	0.44	-0.04	0.64
Puig	0.14	0.53	0.44	1	0.73	0.65
Travers	0.76	0.18	-0.04	0.73	1	0.08
Turan	-0.55	0.61	0.64	0.65	0.08	1

2.4 Reading

Further reading on recommender systems in chapter 2 of Segaran.

Pearson's correlation is widely used to find relationship between two data, but it should be noted that the measure does not tell you causation (cause and effect). Many people, even some scientists, confuse correlation and causation, and derive wrong conclusions. See those references for details:

`https://en.wikipedia.org/wiki/Correlation_does_not_imply_causation`

`https://www.americanscientist.org/article/what-everyone-should-know-about-statistical-correlatio`

Exercises

1. Consider two column vectors such that $\mathbf{a} = (1, 2, 3)^T$ and $\mathbf{b} = (-3, 3, -1)^T$.

   (a) Find $r_p(\mathbf{a}, \mathbf{b})$ for $p = 1$ and 2.

2. We have studied that the Pearson correlation coefficient for two data sets represented as vectors, $\mathbf{x} = \{x_n\}_1^N$ and $\mathbf{y} = \{y_n\}_1^N$, is given by

$$r_{\mathbf{xy}} = \frac{1}{N-1} \sum_{n=1}^{N} \left(\frac{x_n - \bar{x}}{s_x} \right) \left(\frac{y_n - \bar{y}}{s_y} \right).$$

   Now, show that it can be rewritten as

$$r_{\mathbf{xy}} = \frac{\sum_{n=1}^{N} (x_n - \bar{x})(y_n - \bar{y})}{\sqrt{\sum_{n=1}^{N} (x_n - \bar{x})^2} \sqrt{\sum_{n=1}^{N} (y_n - \bar{y})^2}}.$$

3. Consider five vectors such that $\mathbf{x}_1 = (1, 2, 3)^T, \mathbf{x}_2 = (2, 3, 1)^T, \mathbf{x}_3 = (1, 2, 1)^T, \mathbf{x}_4 = (2, 4, 6)^T, \mathbf{x}_5 = (-1, -2, -3)^T$. Find the Pearson correlation coefficient for each of the following pairs.

   (a) $\mathbf{x}_1$ and $\mathbf{x}_2$.

   (b) $\mathbf{x}_1$ and $\mathbf{x}_3$.

   (c) $\mathbf{x}_1$ and $\mathbf{x}_4$.

   (d) $\mathbf{x}_1$ and $\mathbf{x}_5$.