Curve fitting that minimizes the mean square of perpendicular distances from sample points

Shotaro Akaho

Electrotechnical Laboratory, Mathematical Informatics Section
1–1–4 Umezono, Tsukuba-shi, Ibaraki 305 Japan

ABSTRACT

This paper presents a new method of curve-fitting to a set of noisy samples. In the case of fitting a curved line (or a curved surface) to given sample points, it seems natural the curve is decided so as to minimize the mean square of perpendicular distances from the sample points. However it is difficult to get the optimal curve in the sense of this criterion. In this paper, the perpendicular distance is approximated by local linear approximation of function, and the algorithm for getting the near-optimal curve is proposed. Some simulation results are also shown.

1. INTRODUCTION

Curve fitting to noisy sample points is one of the most important and basic processes for image processing. Let us consider fitting a curve $f(\boldsymbol{x};\boldsymbol{a}) = 0$ to sample points $\boldsymbol{x}^1,\ldots,\boldsymbol{x}^m$, where $\boldsymbol{x}$ is a 2- or 3- (or higher-) dimensional vector and $\boldsymbol{a}$ is a parameter vector. It is natural that we choose parameter $\boldsymbol{a}$ such that the mean square value of perpendicular distance from sample points to the curve, i.e., suppose $\hat{\boldsymbol{x}}^p$ denote the foot of the perpendicular from $\boldsymbol{x}^p$ to $f(\boldsymbol{x};\boldsymbol{a})$,

$$\mathop{\mathrm{E}}_{p}[|\hat{\boldsymbol{x}}^p - \boldsymbol{x}^p|^2] \tag{1}$$

is minimized (fig.1).

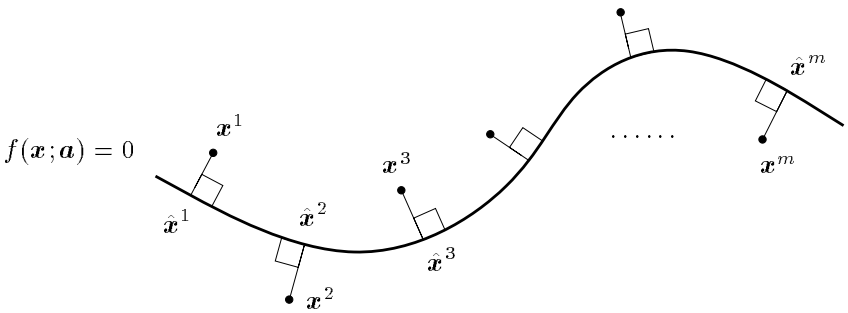


Figure 1: Perpendiculars from points to a curve

Generally this problem is difficult to solve directly because the perpendicular distance can not be solved explicitly and moreover the perpendicular to a curve is not always unique. Thus normally a rough approximate solution is derived under the criterion of minimizing the mean square value of $f(\boldsymbol{x};\boldsymbol{a})$. However, the curve $f(\boldsymbol{x};\boldsymbol{a}) = 0$ derived in such a way is often different from the curve expected intuitionally.

In this paper, the uniqueness of a perpendicular distance is ensured by locally linear approximation and we give a simple algorithm to minimize the mean square of perpendicular distances.

In sections 2 and 3, the approximate perpendicular in a simple case is presented, and in section 4 we give a primitive algorithm, and it is extended to the class of generalized linear functions in section 5. Although this algorithm converges under some assumption, it is sometimes necessary to be stabilized. Thus we discuss the

convergence property of the algorithm in section 6 and add some stabilization techniques to the algorithm such as cross-validation and regularization in section 7. In section 8, we show some simulation results.

2. APPROXIMATE PERPENDICULAR

By taking the first order Taylor expansion of $f(\hat{\boldsymbol{x}}^p;\boldsymbol{a})=0$, we get

$$f(\hat{\boldsymbol{x}}^p;\boldsymbol{a}) \simeq f(\boldsymbol{x}^p;\boldsymbol{a}) + \boldsymbol{d}^{p\prime}\frac{\partial f}{\partial \boldsymbol{x}} = 0, \qquad \boldsymbol{d}^p = \hat{\boldsymbol{x}}^p - \boldsymbol{x}^p. \tag{2}$$

The first (nearly-)equation of (2) defines the hyperplane that approximates hypersurface $z=f(\boldsymbol{x};\boldsymbol{a})$ in $\mathbf{R}^{n+1}$ (z is the $(n+1)$-th component of $\mathbf{R}^{n+1}$). The second equation represents the $n-1$ dimensional subspace which is the intersection of that hyperplane and $z=0$.

Thus, $\hat{\boldsymbol{x}}^p$ is approximated uniquely by the perpendicular from $\boldsymbol{x}^p$ to the linear subspace of $\mathbf{R}^{n-1}$ defined by the second equation of (2). Moreover, if we can find the parameter $\boldsymbol{a}$ such that $\min_{\boldsymbol{a}} \mathrm{E}_p[\|\boldsymbol{d}^p\|^2]$, the problem is solved in the sense of locally linear approximation.

3. SIMPLE CASE

In this section, we consider a quadratic function on two dimensional space as a simple case.

3.1. Two dimensional space

Let (x,y) denote the two dimensional coordinates, and let $\boldsymbol{d}^p$ be (d_x^p, d_y^p). From the equation (2) we get

$$d_x^p \frac{\partial f}{\partial x}(\boldsymbol{x}^p;\boldsymbol{a}) + d_y^p \frac{\partial f}{\partial y}(\boldsymbol{x}^p;\boldsymbol{a}) = -f(\boldsymbol{x}^p;\boldsymbol{a}). \tag{3}$$

This equation and the condition $(d_x^p)^2+(d_y^p)^2 \to \min$, it follows

$$\|\boldsymbol{d}^p\|^2 = \frac{\{f(\boldsymbol{x}^p;\boldsymbol{a})\}^2}{\{\partial f(\boldsymbol{x}^p;\boldsymbol{a})/\partial x\}^2 + \{\partial f(\boldsymbol{x}^p;\boldsymbol{a})/\partial y\}^2}. \tag{4}$$

Hence the problem is approximated by

$$\mathcal{E}(\boldsymbol{a}) = \frac{1}{m}\sum_{p=1}^{m} \frac{\{f(\boldsymbol{x}^p;\boldsymbol{a})\}^2}{\{\partial f(\boldsymbol{x}^p;\boldsymbol{a})/\partial x\}^2 + \{\partial f(\boldsymbol{x}^p;\boldsymbol{a})/\partial y\}^2} \longrightarrow \min_{\boldsymbol{a}} \tag{5}$$

3.2. Quadratic function

Let $f(x,y;\boldsymbol{a}) = a_1x^2 + a_2xy + a_3y^2 + a_4x + a_5y + a_6$,

$$f(x,y;\boldsymbol{a}) = \boldsymbol{a}'\tilde{\boldsymbol{x}}, \qquad \tilde{\boldsymbol{x}} = (x^2, xy, y^2, x, y, 1), \tag{6}$$

where $\boldsymbol{a}'$ denotes the transpose of $\boldsymbol{a}$. Similarly

$$\frac{\partial f}{\partial x}(x,y;\boldsymbol{a}) = \boldsymbol{a}'\tilde{\boldsymbol{u}}, \qquad \tilde{\boldsymbol{u}} = (2x, y, 0, 1, 0, 0), \tag{7}$$

$$\frac{\partial f}{\partial y}(x,y;\boldsymbol{a}) = \boldsymbol{a}'\tilde{\boldsymbol{v}}, \qquad \tilde{\boldsymbol{v}} = (0, x, 2y, 0, 1, 0). \tag{8}$$

Equation (5) becomes

$$\mathcal{E}(\boldsymbol{a}) = \frac{1}{m}\sum_{p=1}^{m} \frac{\boldsymbol{a}'(\tilde{\boldsymbol{x}}^p\tilde{\boldsymbol{x}}^{p\prime})\boldsymbol{a}}{\boldsymbol{a}'(\tilde{\boldsymbol{u}}^p\tilde{\boldsymbol{u}}^{p\prime} + \tilde{\boldsymbol{v}}^p\tilde{\boldsymbol{v}}^{p\prime})\boldsymbol{a}} = \frac{1}{m}\sum_{p=1}^{m} \frac{\boldsymbol{a}'X^p\boldsymbol{a}}{\boldsymbol{a}'D^p\boldsymbol{a}} \longrightarrow \min_{\boldsymbol{a}}, \tag{9}$$

where

$$X^p = \tilde{\boldsymbol{x}}^p \tilde{\boldsymbol{x}}^{p\prime}, \quad D^p = \tilde{\boldsymbol{u}}^p \tilde{\boldsymbol{u}}^{p\prime} + \tilde{\boldsymbol{v}}^p \tilde{\boldsymbol{v}}^{p\prime}. \tag{10}$$

However, each term of (9) is a nonlinear form with respect to $\boldsymbol{a}$ and it is difficult to solve explicitly.

4. PRIMITIVE ALGORITHM

In this section, we give an algorithm to solve the minimization problem (9) approximately. First of all, we make an assumption as follows. The case that the assumption is not fulfilled is considered in section 7.

Assumption $f(\boldsymbol{x};\boldsymbol{a})$ is sufficiently smooth with respect to $\boldsymbol{a}$, so that the change of $\partial f/\partial x$ is sufficiently small when $\boldsymbol{a}$ changes slightly.

Suppose a rough approximation $\boldsymbol{a}_0$ is given, we can substitute $\boldsymbol{a}$ in the numerator of equation (9) by $\boldsymbol{a}_0$ under the assumption; hence the problem is reduced to

$$\mathcal{E}(\boldsymbol{a}) \simeq \hat{\mathcal{E}}(\boldsymbol{a}) = \frac{1}{m}\sum_{p=1}^{m} \frac{1}{\mu^p} \boldsymbol{a}' X^p \boldsymbol{a} \to \min_{\boldsymbol{a}} \tag{11}$$

$$\mu^p = \boldsymbol{a}_0{}' D^p \boldsymbol{a}_0. \tag{12}$$

The solution of this problem is derived by means of the method of Lagrangian undetermined coefficient, i.e., we take such $\boldsymbol{a}$ that minimizes $\hat{\mathcal{E}}(\boldsymbol{a})$ among the solution of the following equation,

$$\frac{\partial}{\partial \boldsymbol{a}}\{\hat{\mathcal{E}}(\boldsymbol{a}) - \lambda(\boldsymbol{a}'\boldsymbol{a} - 1)\} = 2\left(\frac{1}{m}\sum_{p=1}^{m}\frac{X^p}{\mu^p}\right)\boldsymbol{a} - 2\lambda\boldsymbol{a} = 0, \tag{13}$$

where λ is an undetermined coefficient. Consequently, the solution is given by the eigen vector that corresponds to the minimal eiven value of $(1/m)\sum_{p=1}^{m}(X^p/\mu^p)$.

$1/\mu^p$ can be regarded as the weight for each sample. The classical method[3] that minimizes $\mathrm{E}[f(\boldsymbol{x};\boldsymbol{a})^2]$ is the case of $\mu^p = 1$ for all p. In this sence, the present algorithm includes the classical method as a special case.

We complete this section by giving the algorithm explicitly. Starting with the initial solution derived by the classical method, we develop the solution iteratively by the method above.

Algorithm 1 (primitive version)

1. $\mu^p := 1 \qquad (p = 1, \cdots, m)$
2. Repeat (a), (b)
   (a) Solve the eigenvalue problem $(\frac{1}{m}\sum_{p=1}^{m}(X^p/\mu^p))\boldsymbol{a} = \lambda\boldsymbol{a}$ and let $\hat{\boldsymbol{a}}$ be the eigen vector that corresponds to the minimal eigen value. $(\|\hat{\boldsymbol{a}}\| = 1)$.
   (b) $\mu^p := \hat{\boldsymbol{a}}' D^p \hat{\boldsymbol{a}} \qquad (p = 1, \cdots, m)$

where X^p and D^p are defined in (10).

5. GENERAL CASE

In the preceding sections, we have described the case of fitting a quadratic function in two dimensional space. It can be extended for generalized linear models in $\mathbf{R}^n$.

Consider the problem fitting $f(\boldsymbol{x};\boldsymbol{a}) = \boldsymbol{a}'\boldsymbol{\phi}(\boldsymbol{x}) = 0$ to m sample points in $\mathbf{R}^n$, where

$$\boldsymbol{\phi}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \cdots, \phi_l(\boldsymbol{x})), \quad \boldsymbol{a} = (a_1, \ldots, a_l), \quad \boldsymbol{x} = (x_1, \ldots, x_n), \tag{14}$$

and $\phi_i(\boldsymbol{x})$ are independent base functions. Let $\partial_j \boldsymbol{\phi}(\boldsymbol{x}) \equiv \partial \boldsymbol{\phi}(\boldsymbol{x})/\partial x_j$, similarly to (9) we get

$$\mathcal{E}(\boldsymbol{a}) = \frac{1}{m}\sum_{p=1}^{m} \frac{\boldsymbol{a}' X^p \boldsymbol{a}}{\boldsymbol{a}' D^p \boldsymbol{a}} \longrightarrow \min_{\boldsymbol{a}}, \tag{15}$$

where

$$X^p = \boldsymbol{\phi}^p \boldsymbol{\phi}^{p\prime}, \quad D^p = \sum_j \partial_j \boldsymbol{\phi}^p \partial_j \boldsymbol{\phi}^{p\prime}. \tag{16}$$

The algorithm for this problem is just the same as that of the previous section, but the definition of X^p, D^p should be substituted from (10) to (16).

6. CONVERGENCE

If the assumption is fulfilled, we can show the local convergence as follows. Let $\boldsymbol{a}^{(k)}$ denote the approximate solution given in k-th iteration and let

$$\mathcal{E}^{(k)} \equiv \mathcal{E}(\boldsymbol{a}^{(k)}), \tag{17}$$

we derive

$$\mathcal{E}^{(k+1)} \simeq \mathop{\mathrm{E}}_{p}[\frac{\boldsymbol{a}^{(k+1)\prime} X^p \boldsymbol{a}^{(k+1)}}{\boldsymbol{a}^{(k)\prime} D^p \boldsymbol{a}^{(k)}}] < \mathop{\mathrm{E}}_{p}[\frac{\boldsymbol{a}^{(k)\prime} X^p \boldsymbol{a}^{(k)}}{\boldsymbol{a}^{(k)\prime} D^p \boldsymbol{a}^{(k)}}] = \mathcal{E}^{(k)}, \tag{18}$$

which implies that $\mathcal{E}^{(k)}$ is a decreasing sequence.

7. STABILIZATION

The primitive algorithm does not converge when the initial solution does not approximate samples well or base functions are of complicated form.

A lot of methods to stabilize unstable algorithms have been proposed. In this paper, we consider two basic techniques: cross validation and regularization. We use cross validation in order to get a good initial solution and regularization in order to stabilize the algorithm. We begin with summarizing those techniques and next we give the stabilized algorithm.

7.1. Cross validation

Cross validation is a kind of resampling method[2) 1)] and it consists of two phases.

1. **Subsampling** : A certain number of samples are drawn randomly from the whole data set, and a curve is constructed so as to approximate this subset. This procedure is performed many times.
2. **Selection** : Each curve derived in the first phase is tested for the whole data, and the best curve is selected, which is adopted as an initial solution in our algorithm.

7.2. Regularization

Regularization is a general technique to solve unstable problems[5)4)]. In our problem, the estimated norm of gradient μ^p is very unstable: especially when $|f(\boldsymbol{x}, \boldsymbol{a})|$ is large or $\mu^p \sim 0$, then $1/\mu^p$ has only little reliability. Thus for example, we can use $\hat{\mu}^p$ defined below instead of μ^p,

$$\hat{\mu}^p = \Omega_{\alpha,\beta}(\mu^p) = (\mu^p + \alpha)^{\frac{1}{1+\beta f(\boldsymbol{x}^p;\boldsymbol{a})^2}}, \tag{19}$$

where α and β are parameters. α avoids $\hat{\mu}^p$ to get close to 0, and β makes $\hat{\mu}^p$ be close to 1 if $|f|$ is large. If $\alpha = \beta = 0$, then $\hat{\mu}^p = \mu^p$ holds. The problem how large those parameters should be depends on each problem and it is a difficult problem in general.

7.3. Stabilized algorithm

Now we construct the algorithm using stabilization techniques. Let I_{CV} denote the number of subsets for cross validation, and let $m_{\mathrm{sub}}(< m)$ denote the number of elements of each subset.

Algorithm 2 (stabilized version)

1. Repeat for $i = 1, 2, \ldots, I_{\mathrm{CV}}$
   (a) $S_i :=$ Random subset of the whole samples. $(\#S_i = m_{\mathrm{sub}})$
   (b) Apply Algorithm 1 (primitive algorithm) for S_i. Let $\boldsymbol{a}_i$ be the solution.
2. $\boldsymbol{a}_{\mathrm{CV}} := \underset{\{\boldsymbol{a}_i\}}{\mathrm{Argmin}}\, \mathcal{E}(\boldsymbol{a}_i)$
3. Let an initial solution be $\boldsymbol{a}_{\mathrm{CV}}$ and apply Algorithm 1 for the whole samples.

Remark: $\hat{\mu}^p$ defined in (19) is used instead of μ^p in Algorithm 1 and in calculations of $\mathcal{E}(\boldsymbol{a})$.

8. SIMULATION RESULTS

Figure 2–4 are computer simulation results of our algorithms. 100 sample points are generated from the quadratic function $y = x^2 + x + 1$, where $x \sim U[-5, 5]$ ($U[a, b]$ is the uniform distribution between a and b) and independent noise according to $U[-0.5, 0.5]^2$ is added for each sample point. If we use the classical method for the quadratic function, an ellipse tends to be selected more than a parabola. Hence this example is the case that an initial solution does not fit samples well.

Fig.2 is a result of Algorithm 1. (a) is the initial solution derived by the classical algorithm. (b) and (c) are solutions in the first and the fifth step respectively. As a step goes on, solution is getting close to a parabola from a ellipse and a curve comes to fit samples well.

Fig.3 and fig.4 is a result of Algorithm 2. We have run the algorithm for twenty sample sets corresponding to different random seeds. Cross validation selection is performed 3 times for 30 random subsamples (i.e., $I_{\mathrm{CV}} = 3$ and $m_{\mathrm{sub}} = 30$ in Algorithm 2). In this simulation, no regularization is carried out (i.e., $\hat{\mu}^p = \hat{\mu}$), but more complicated base functions may need regularization. We performed two steps in each application of Algorithm 1. Fig.3 shows the ratio of improved MSE (mean square of perpendicular distance) to initial MSE. The average ratio is about 0.16. We could not improve a solution in one case (upper-left point in fig.3). Fig.4 shows the validity of approximation of perpendicular distances. This plot consists of values for all $\boldsymbol{a}$ appeared in the simulation. As the value becomes larger, the validity is getting lower, and in this case an approximate value is getting smaller than a real value.

9. CONCLUSION

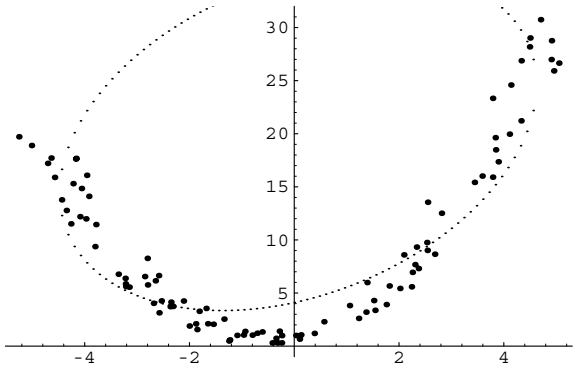
We have presented a new method of curve fitting that minimizes (approximate) mean perpendicular distance. This method includes the classical method as a special case. As shown in simulation results, the classical method does not work well when a target function is not a certain form (e.g. an ellipse in quadratic function). Our method can improve the solution in such a case.

We have not adopted an algorithm such as Newton method, because there are a lot of local minima. However, there still is a possibility to use such a gradient method.

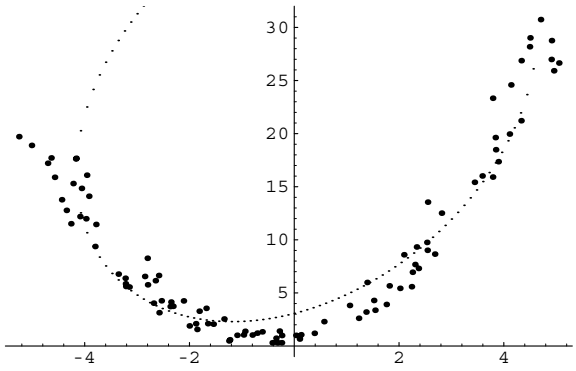
In practical applications, there are a lot of objects to fit curves in one image, so it should be divided by clustering in order to apply our algorithm. This is a different but an interesting problem.

10. ACKNOWLEDGEMENTS

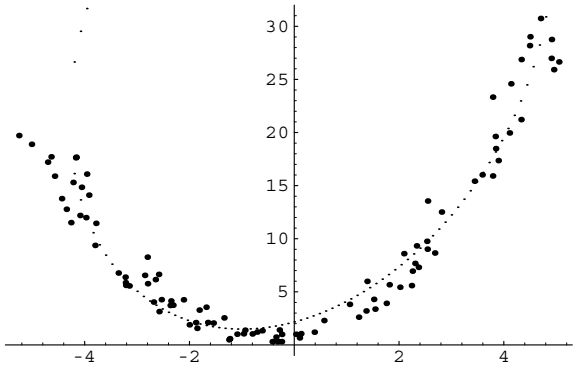
The author would like to thank M. Suwa, Director of Information Science Division of ETL, for affording an



(a) initial solution



(b) after the first step



(c) after the fifth step

Figure 2: Simulation result of Algorithm 1 for 100 random samples from $y = x^2 + x + 1$ ($x \sim U[-5, 5]$, additive noise to (x, y): $U[-0.5, 0.5]^2$). large dots: sample points; small dots: solution curve for each step.

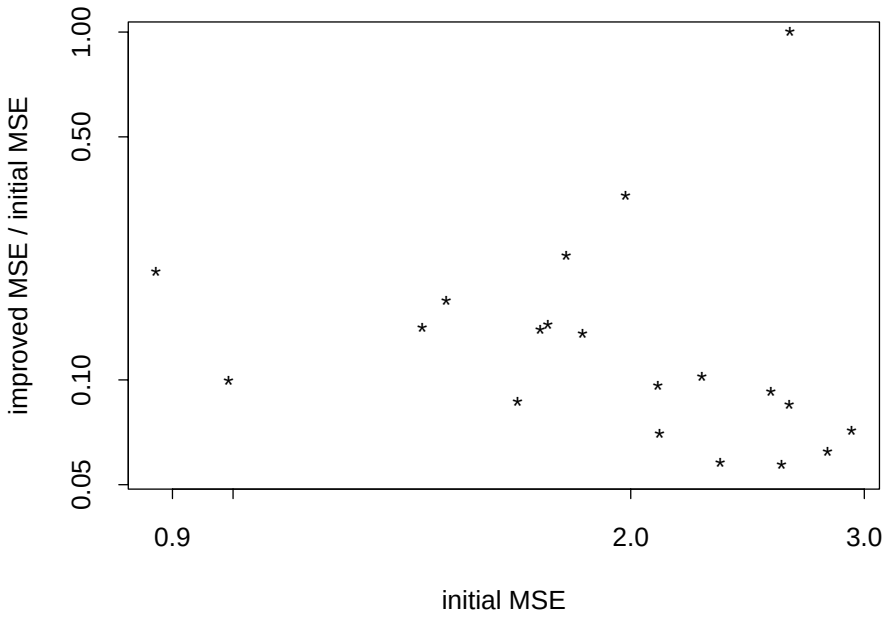


Figure 3: Simulation results of Algorithm 2. Initial MSE (mean square of real perpendicular distance) versus the ratio of improved MSE to initial MSE. Sample points are generated as similarly as in fig.2 for twenty different random seeds.

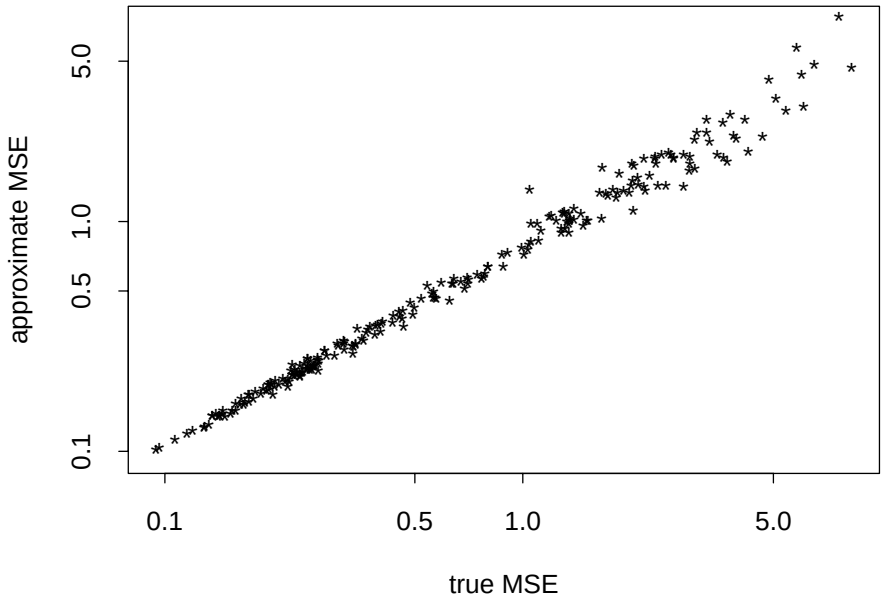


Figure 4: Approximation validity of perpendicular distances. Mean squares of real perpendicular distance versus mean squares of approximate perpendicular distance $\mathcal{E}$.

opportunity of this study. He is also deeply indebted to N. Otsu, Director of Machine Understanding Division, and all members of Mathematical Informatics Section for their helpful discussions.

REFERENCES

[1] B. Efron: *The Jackknife, the Bootstrap and Other Resampling Plans.* SIAM, Philadelphia, 1982.

[2] S. Geisser: The predictive sample reuse method with applications. *J. Amer. Statist. Assoc.*, Vol. 70, pp. 320–328, 1975.

[3] N. Otsu: Karhunen-loeve line fitting and a linearly measure. In *IEEE Proceedings of the seventh international conference on pattern recognition*, pp. 486–489, 1984.

[4] T. Poggio and F. Girosi: Networks for approximation and learning. *Proceedings of the IEEE*, Vol. 78, No. 9, pp. 1481–1497, 1990.

[5] A.N. Tikhonov and V.Ya. Arsenin: *Solutions of Ill-posed Problems.* Winston, Washington, 1977.