High Performance Quadratic Classifier and the Application On PenDigits Recognition

ZhengYi John ZHAO, Jie SUN, Shuzhi Sam GE

Dept. of Electrical & Computer Engineering
National University of Singapore

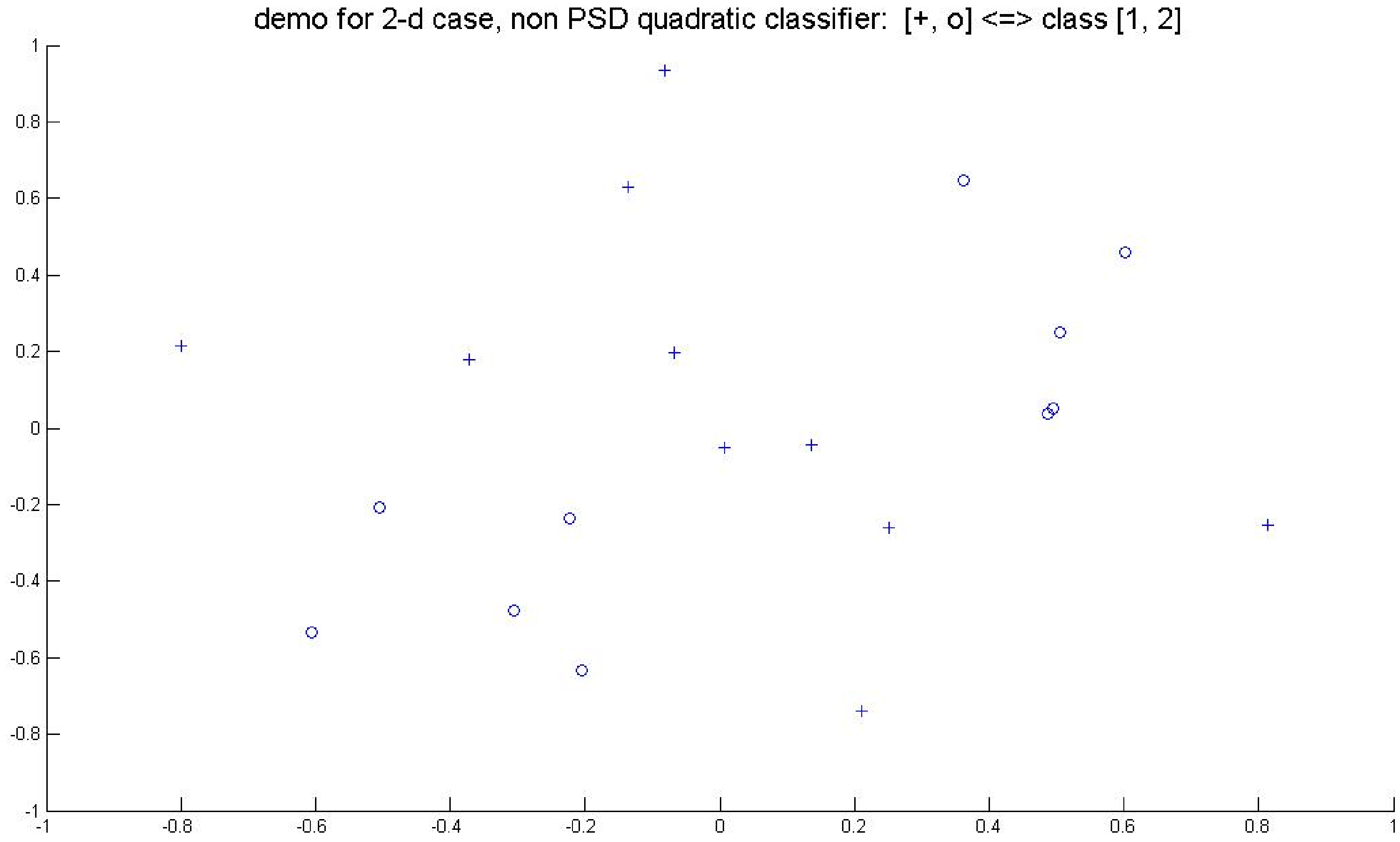
Contents

Introduction

- Pattern Classification
  - Preprocessing (Data Sampling, Noise reduction, Scaling, …)
  - Learning
  - Testing
- Current Models for Learning & Testing
  - Linear Classifier (Fisher)
  - K-NN
  - Gaussian-Bayesian

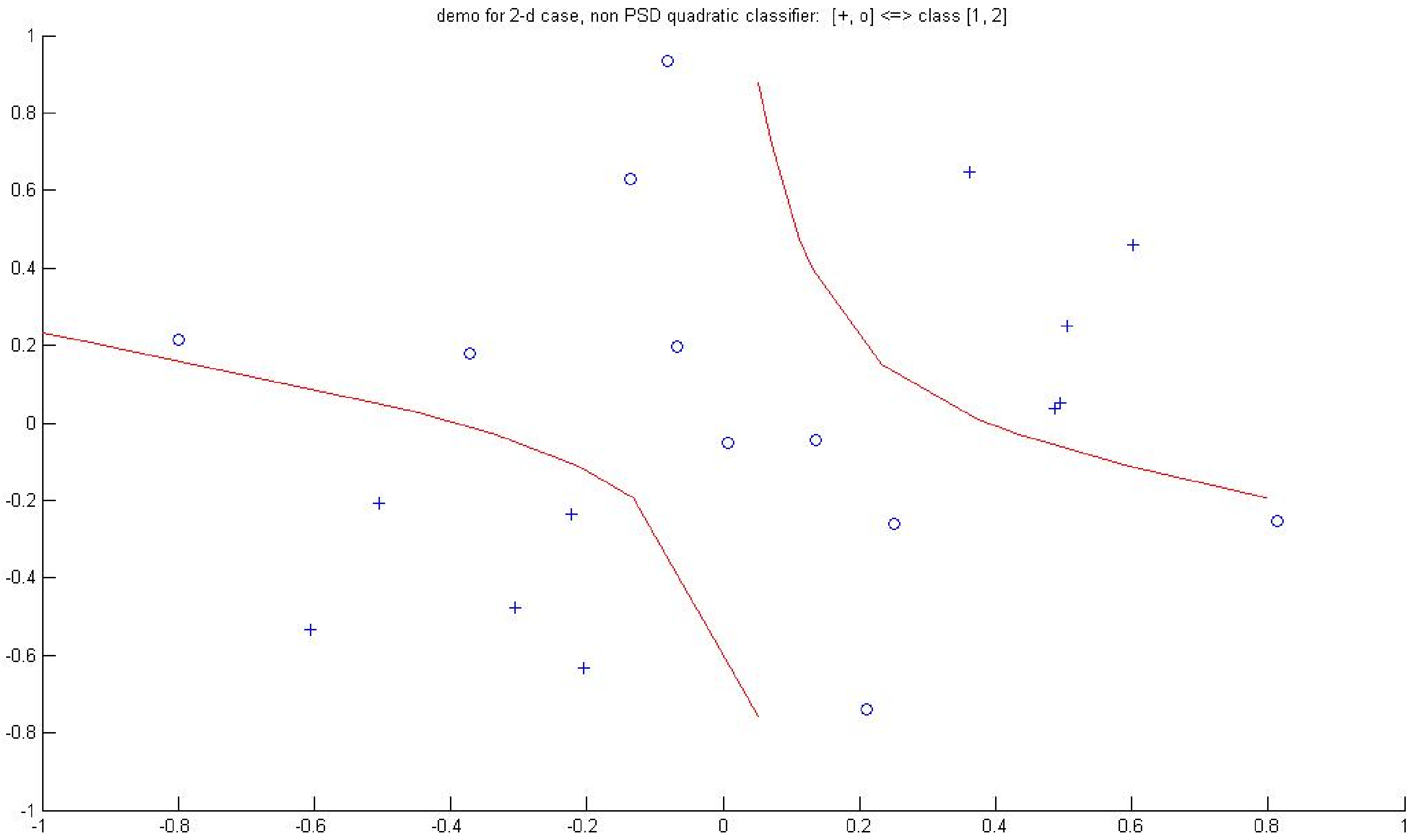
Why choose non PSD quadratic model

- A sample case hard to classify by any of current models



Why choose non PSD quadratic model

- It can be classified by general quadratic model



How to formulate
- intuition from Gauss Bayes

- Mean and variance of samples in class-k

$$\mu_k = \frac{\sum_{i:c_i=\omega_k} \mathbf{s}_i}{n(\omega_k)} \quad (1)$$

$$\Sigma_k = \frac{1}{n(\omega_k)-1} \sum_{i:c_i=\omega_k} (\mathbf{s}_i - \mu_k)(\mathbf{s}_i - \mu_k)^T \quad (2)$$

- PDF (Probability Density Function) assumption of Gaussian distribution

$$p(\mathbf{x}|\omega_k) = \frac{e^{\left(-\frac{1}{2}(\mathbf{x}-\mu_k)^T \Sigma_k{}^{-1}(\mathbf{x}-\mu_k)\right)}}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \quad (3)$$

How to formulate
- intuition from Gauss Bayes

- Bayesian Decision Rule

$$P(\omega_k|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_k)P(\omega_k)}{p(\mathbf{x})} \quad (4)$$

$$k^* = \operatorname{argmax}\{P(\omega_k|\mathbf{x}) : k = 1, ..., K\} \quad (5)$$

B1: The prior probability $P(\omega_k)$ is computed by $\frac{n(\omega_k)}{N}$.

B2: $p(\mathbf{x})$ is common in all posterior functions, $\{P(\omega_k|\mathbf{x}) : k = 1, ..., K\}$.

B3: The relative values of the posteriori are more important for decision making, for the final decision prefers the relatively largest one in (5).

How to formulate
- Further build Likelihood Function

$$L_k(\mathbf{x}) = P(\omega_k|\mathbf{x})p(\mathbf{x}) = p(\mathbf{x}|\omega_k)P(\omega_k)$$
$$= p(\mathbf{x}|\omega_k)\frac{n(\omega_k)}{N}$$

K functions defined for each class {k: 1,2, ..., K}

$$L_k^G(\mathbf{x}) = \frac{e^{\left(-\frac{1}{2}(\mathbf{x}-\mu_k)^T{\Sigma_k}^{-1}(\mathbf{x}-\mu_k)\right)}}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \cdot \frac{n(\omega_k)}{N} \qquad (6)$$

Substitute Gaussian PDF

$$\ln(L_k^G(\mathbf{x})) = -\frac{1}{2}(\mathbf{x}-\mu_k)^T{\Sigma_k}^{-1}(\mathbf{x}-\mu_k) + \ln(T_k)$$
$$= \underbrace{-\frac{1}{2}\mathbf{x}^T\left({\Sigma_k}^{-1}\right)\mathbf{x}}_{\text{2nd order, PSD}} + \underbrace{\left(\mu_k^T{\Sigma_k}^{-1}\right)\mathbf{x}}_{\text{1st order}} \underbrace{-\frac{1}{2}\mu_k^T\left({\Sigma_k}^{-1}\right)\mu_k + \ln(T_k)}_{\text{Constant}}$$

$$T_k = \frac{n(\omega_k)}{N(2\pi)^{d/2}|\Sigma_k|^{1/2}}.$$

2nd order, PSD

Constant

1st order

$$\ln(L_{k1}^G(\mathbf{x})) > \ln(L_{k2}^G(\mathbf{x})) \equiv L_{k1}^G(\mathbf{x}) > L_{k2}^G(\mathbf{x})$$

Ln() is monotonic

How to formulate
- a generalized quadratic model

$$L_k^Q(\mathbf{x}) = \mathbf{x}^T \mathcal{M}_k \mathbf{x} + \mathbf{p}_k^T \mathbf{x} + q_k \qquad (7)$$

- K functions defined for each class {k: 1,2, …, K}

$\mathcal{M}_k \in \mathbf{R}^{d\times d}$ is symmetric matrix

- May not be PSD (Positive Semi-Definite)

Q1 : $L_k^Q(\mathbf{x}) \geq 1$, if $\mathbf{x}$ belongs to class k.
Q2 : $L_k^Q(\mathbf{x}) \leq -1$, if $\mathbf{x}$ doesnot belong to class k.
Q3 : $-1 < L_k^Q(\mathbf{x}) < 1$, if not clear whether $\mathbf{x}$ belongs to class k or not.

Q4 : $L_{k1}^Q(\mathbf{x}) > L_{k2}^Q(\mathbf{x})$, if it is more likely that $\mathbf{x}$ belongs to class $k1$ than that $\mathbf{x}$ belongs to class $k2$.

How to solve
- Second Order Cone Programming

$$\text{minimize}_{\mathcal{M}_k, \mathbf{p}_k, q_k, e_i, \epsilon} \qquad \epsilon + C \sum_{i=1}^{N} e_i$$

subject to

$$\mathcal{M}_k(m,n) = \mathcal{M}_k(n,m), \forall m < n, \text{and } m,n \in \{1,2,...,d\}$$

$$\epsilon \geq \sqrt{\sum_{1 \leq i \leq j \leq d} \mathcal{M}_k(i,j)^2 + \sum_{1 \leq i \leq d} \mathbf{p}_k(i)^2}$$

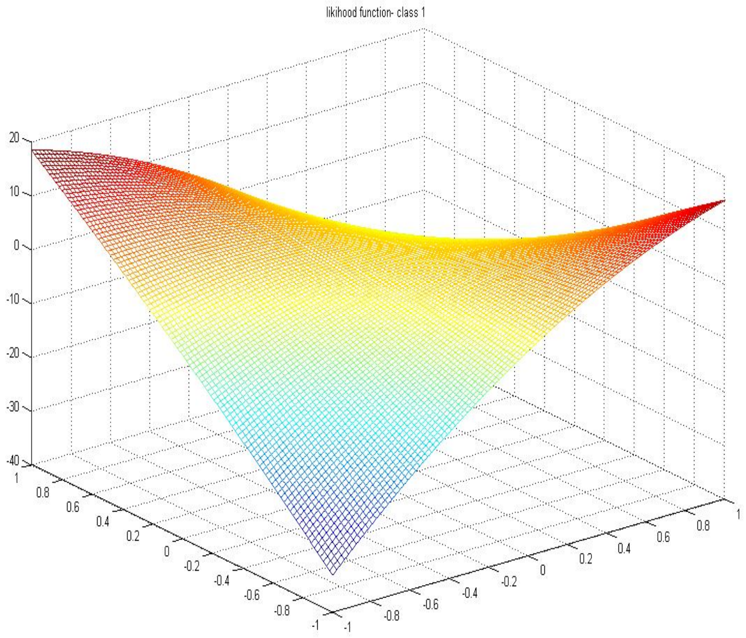
$$\mathbf{s}_i^T \mathcal{M}_k \mathbf{s}_i + \mathbf{p}_k^T \mathbf{s}_i + q_k \geq 1 - e_i, \text{ if } c_i = \omega_k, \forall i \in \{1,2,...,N\}$$

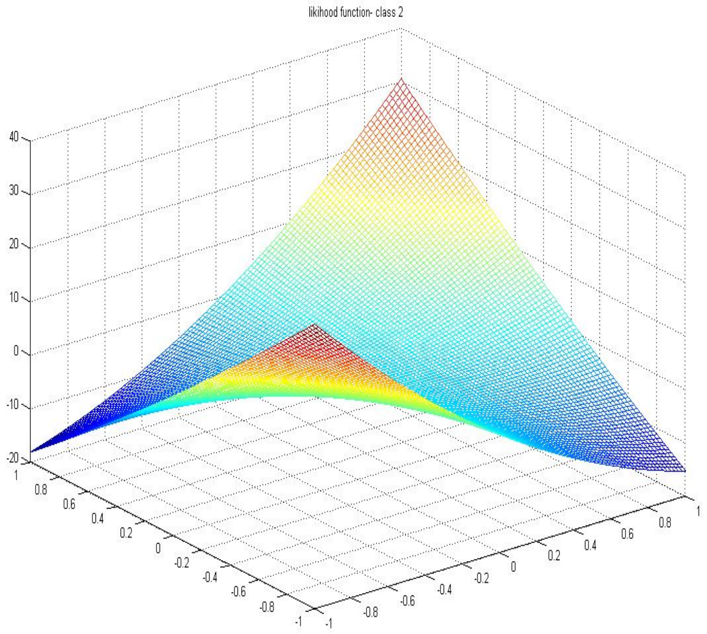
$$\mathbf{s}_j^T \mathcal{M}_k \mathbf{s}_j + \mathbf{p}_k^T \mathbf{s}_j + q_k \leq -1 + e_j, \text{ if } c_j \neq \omega_k, \forall j \in \{1,2,...,N\}$$

$$e_i \geq 0, \forall i \in \{1,2,...,N\}$$

Solution to Sample Problem

- Likelihood functions



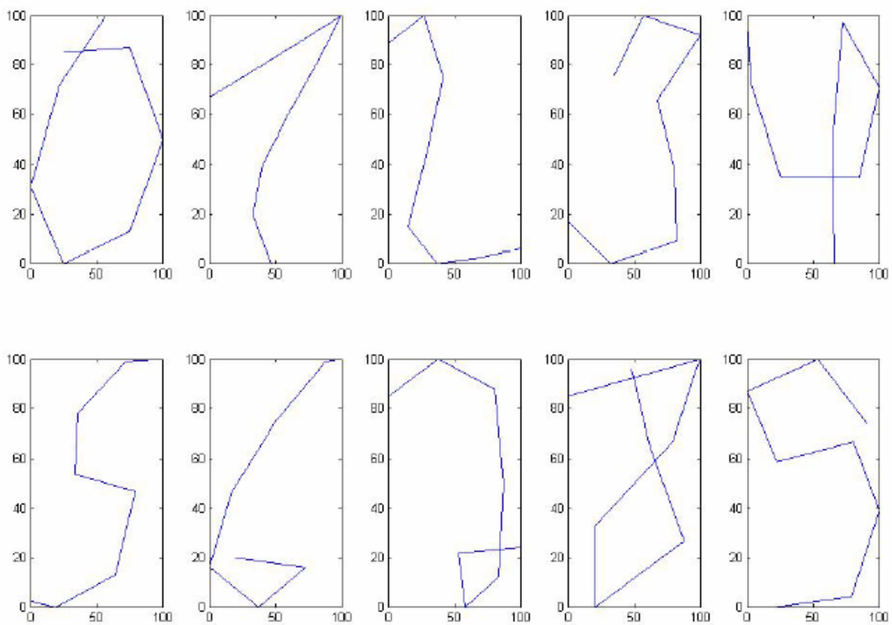


Class 1 {+}

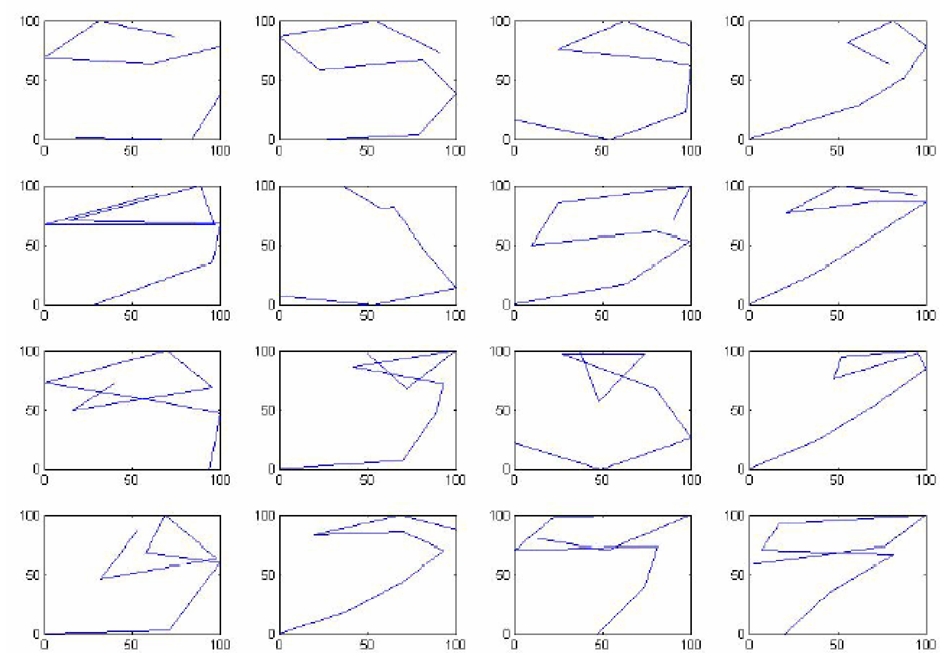
Class 2 {o}

Benchmark on pen-digits recognition

- Database from ftp://ftp.ics.uci.edu/pub/machine-learning-databases/pendigits/
- 7494 samples: each sample is 16 dimension array $\{(x_1,y_1), (x_2,y_2),\ldots, (x_8,y_8)\}$, totally 10 classes {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}



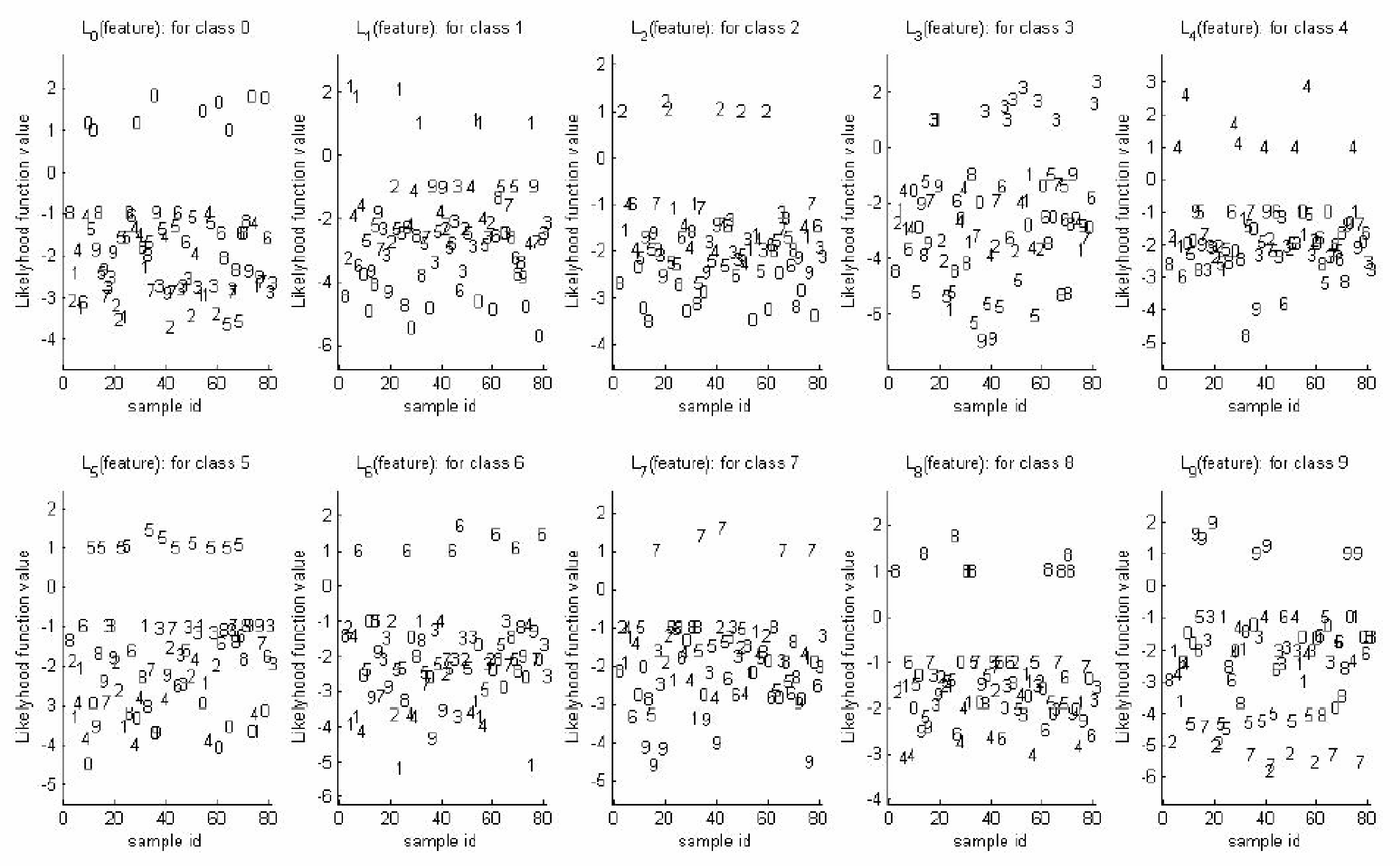
(a) From 0 to 9



(b) Randomly pick 16 samples of digit 9

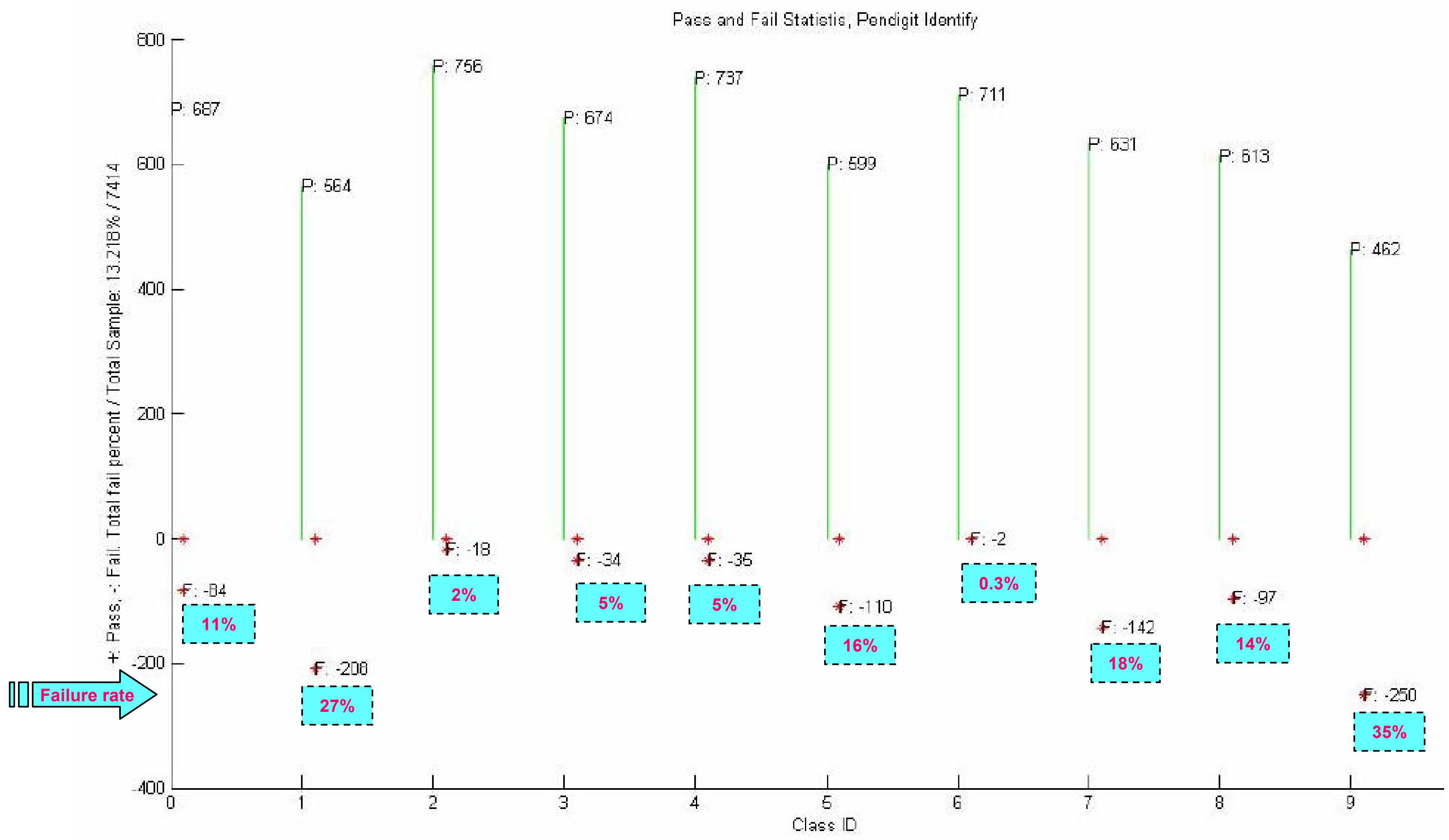
Solution Likelihood Function

- Learning Result for first 80 samples



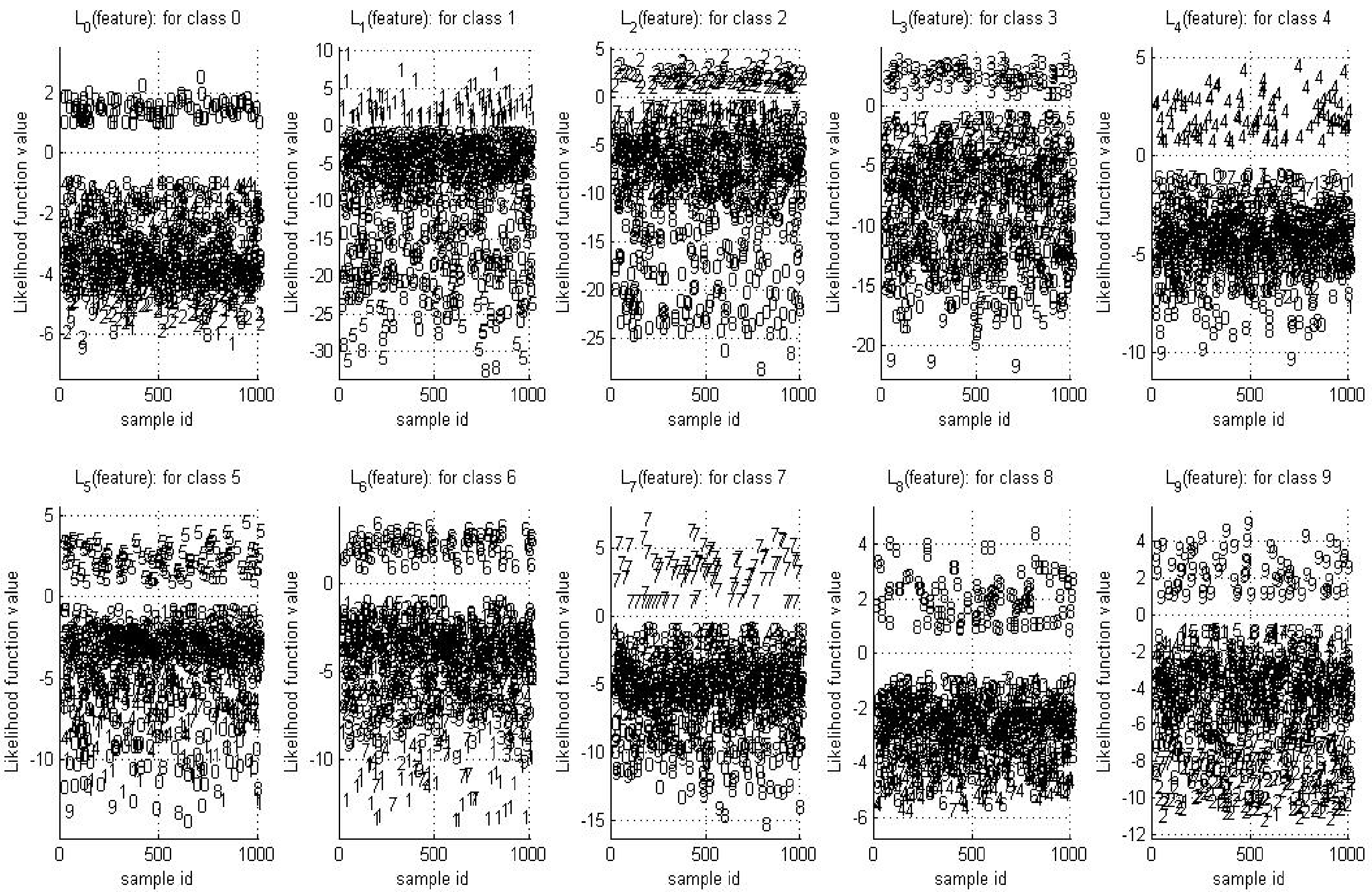
Solution Performance

- Testing with remaining 7414 (=7494-80) Samples



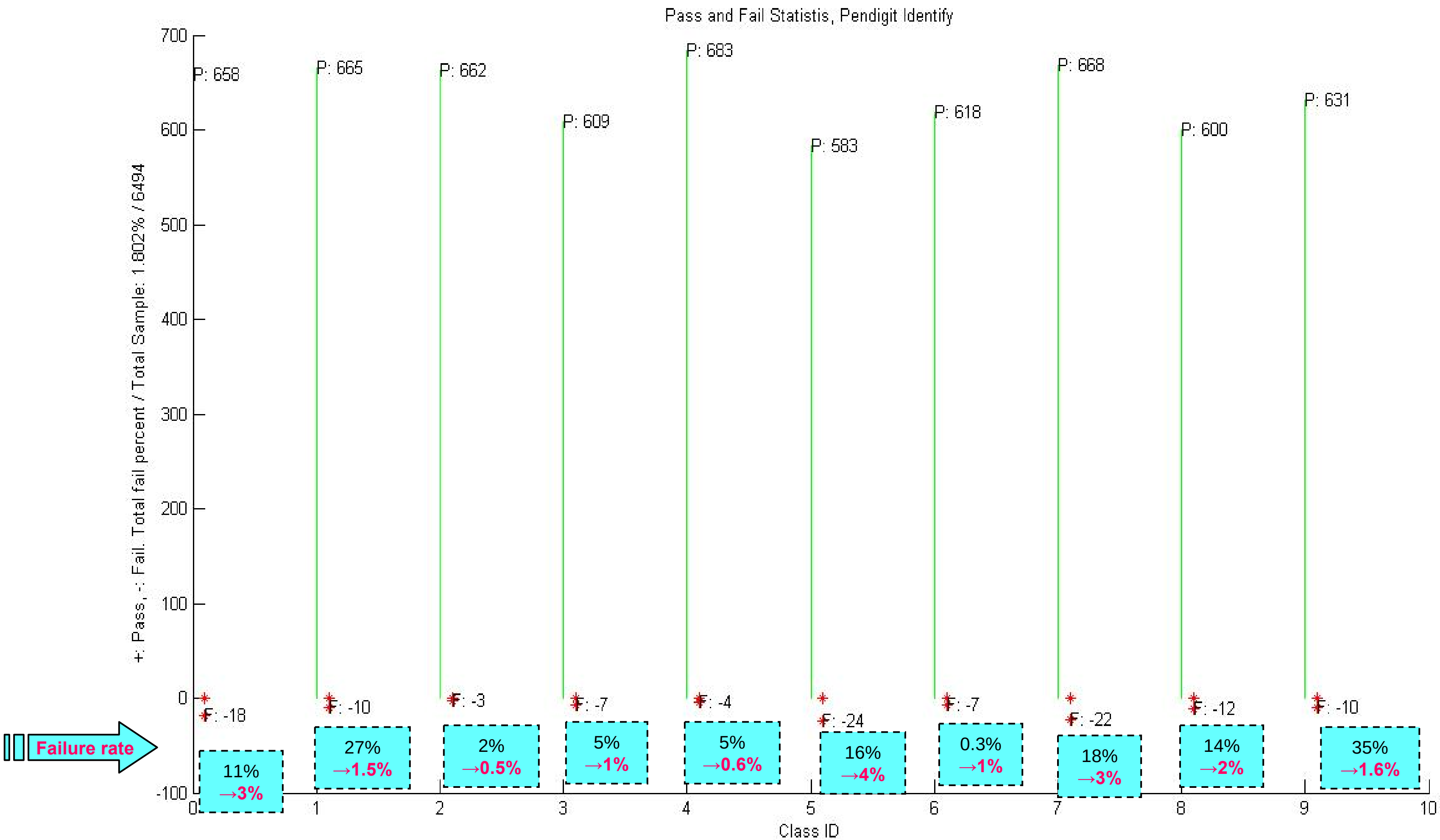
Solution Likelihood Function

- Learning Result for 1000 samples



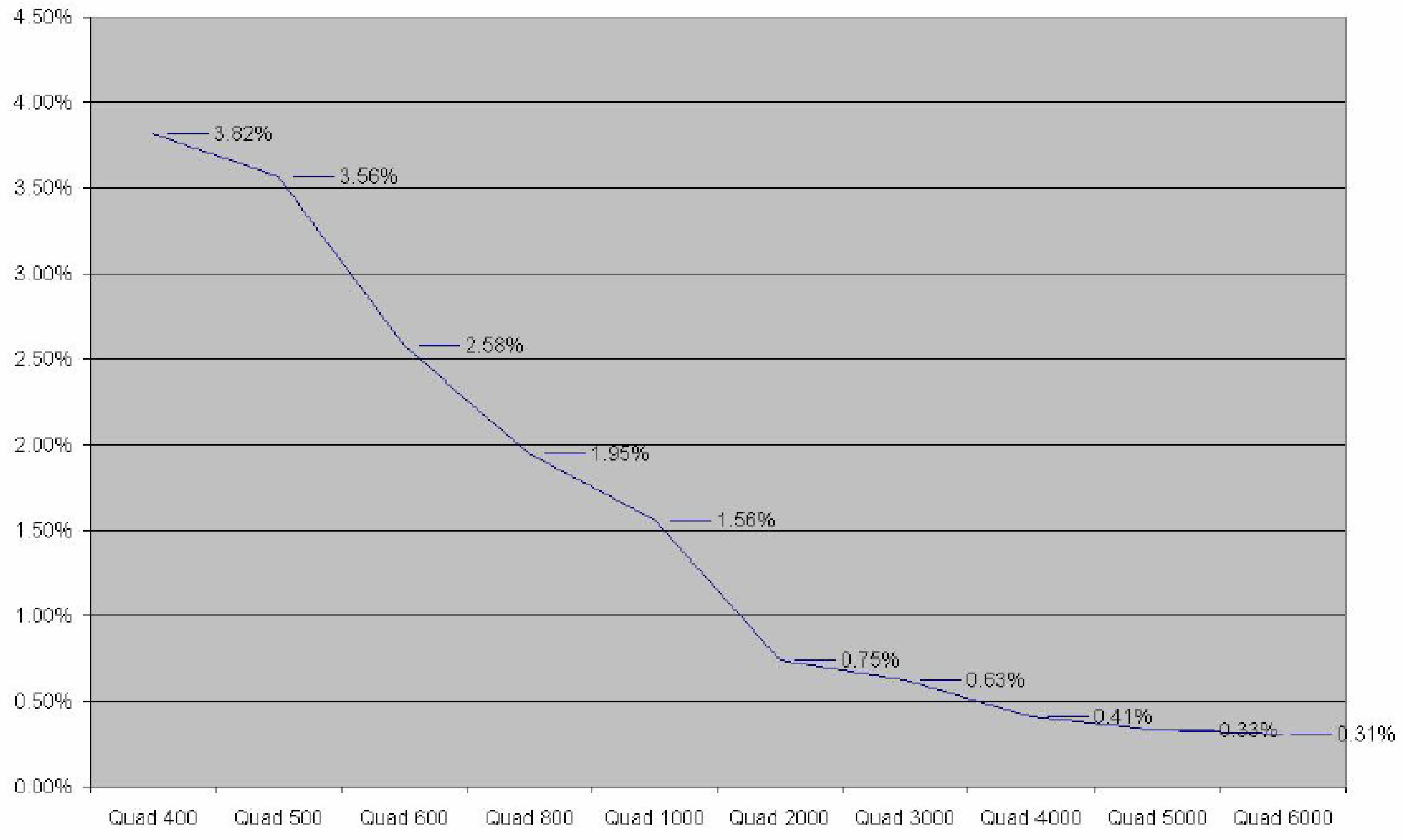
Solution Performance

- Testing with remaining 6414 (=7494-1000) Samples



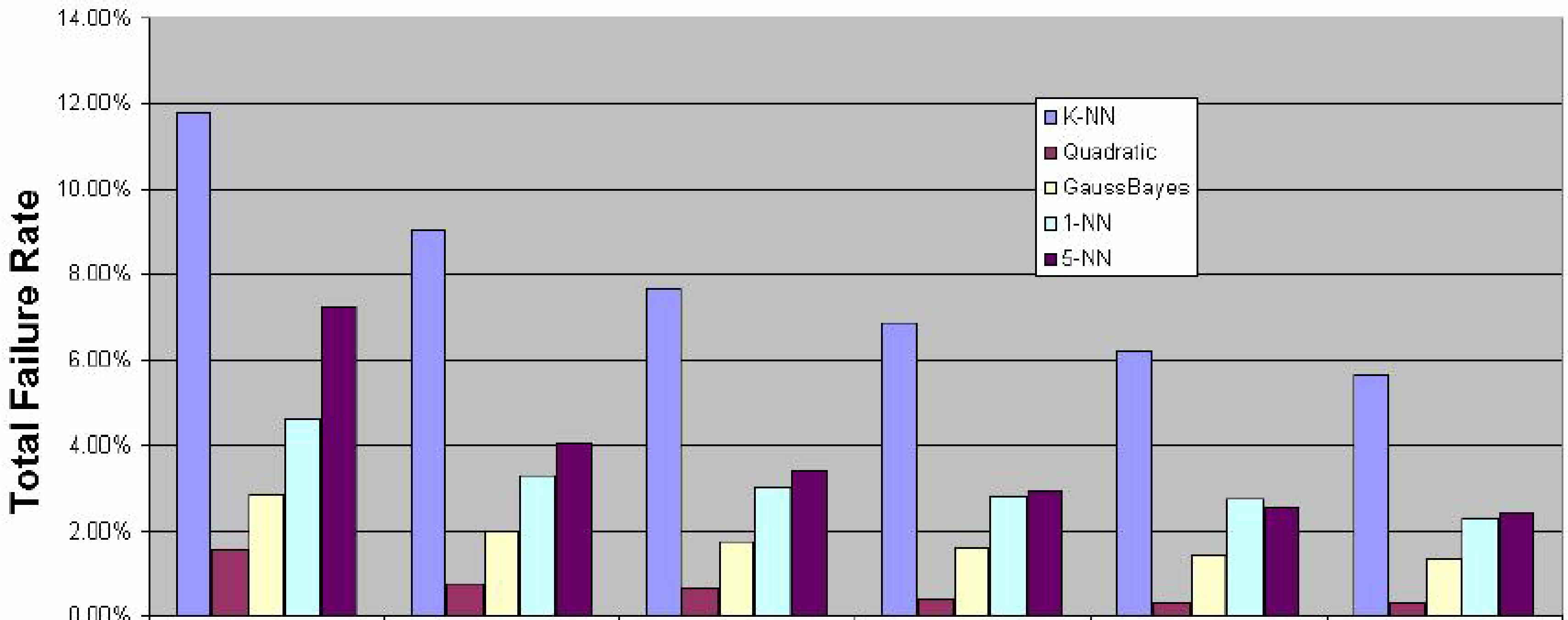
Improved Performance with Increasing Number of Learning Samples

- Total Failure Percentage, Quadratic Classifier



Compared with Other Classifiers

- Best performance with lowest failure rate



	1000	2000	3000	4000	5000	6000
K-NN	11.77%	9.03%	7.67%	6.85%	6.19%	5.62%
Quadratic	1.56%	0.75%	0.63%	0.41%	0.33%	0.31%
GaussBayes	2.84%	1.97%	1.76%	1.60%	1.43%	1.36%
1-NN	4.60%	3.29%	3.03%	2.80%	2.72%	2.32%
5-NN	7.23%	4.03%	3.43%	2.92%	2.54%	2.40%

Num. Training Samples

Tradeoff and Further Research

- The computation time is the longest
- Further research
  - Parallel computing
  - Robust classifier

Thanks!