

Careful Use of Machine Learning Methods  
is needed for Mobile Application  
A case study on Transportation-mode  
Detection

By Yu et al (2013)

Presented by  
Muhammad Awais Shafique

# Contents

1. Introduction
2. Transportation-mode detection
3. Practical use of SVM
4. Pitfall of CV accuracy
5. Model size reduction
6. Fast training by optimization
7. Multi-class SVM method
8. Non-machine learning issues
9. Conclusion

# Introduction

- Machine learning methods are often applied as a black box.
- Example is transportation-mode detection.
- Collect data, use algorithms and compare results.
- Default settings may not be the best one.
- Evaluation criterion (e.g. cross-validation) may not be appropriate.
- Some methods may not be applied due to resource constraints of mobile phones.

This paper focuses on using SVM and how the performance can be optimized.

# Transportation-mode Detection

- The detector can use only up to 16 KB of memory.
- Data consists of log files containing signals from gyroscope, accelerometer and magnetometer.
- Classification was done among
  - Still, Walk, Run, Bike, Others
- Five features were extracted by calculating mean or standard deviation of the signals.
- Decision trees, AdaBoost and SVM were employed.

# Transportation-mode Detection

- Results

Classifiers	CV accuracy (%)	Model size (KB)
Decision Tree	89.41	76.02
AdaBoost	91.11	1500.54
SVM	84.72	1379.97

# Practical use of SVM

- Worse SVM performance may be because of lacking
  - Data scaling
  - Parameter selection
- Given label-instance pairs  $(y \downarrow 1, \mathbf{x} \downarrow 1) \dots (y \downarrow l, \mathbf{x} \downarrow l)$  with  $y \downarrow i = \pm 1, \mathbf{x} \downarrow i \in \mathbb{R}^n, \forall i$  as the training set. (Primal problem)

$$\min_{w, b} \frac{1}{2} w^T w + C \sum_{i=1}^l \max(1 - y_i(w^T \phi(x_i) + b), 0).$$

# Practical use of SVM (Cont.)

- Because  $w$  becomes a huge vector so dual optimization problem is solved. (Dual Problem)

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \alpha^T Q \alpha - e^T \alpha \\ \text{subject to} \quad & y^T \alpha = 0, \\ & 0 \leq \alpha_i \leq C, i = 1, \dots, l, \end{aligned}$$

- Where  $Q_{ij} = y_i y_j \phi(x_i)^T \phi(x_j) = y_i y_j K(x_i, x_j)$ ,  $e = [1, \dots, 1]^T$

# Practical use of SVM (Cont.)

- $K(\mathbf{x}_i, \mathbf{x}_j)$  is the kernel function.
- Default kernel function in LIBSVM is RBF (Gaussian) kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2}$$

- The optimal solution satisfies

$$\mathbf{w} = \sum_{i=1}^l \alpha_i y_i \phi(\mathbf{x}_i).$$

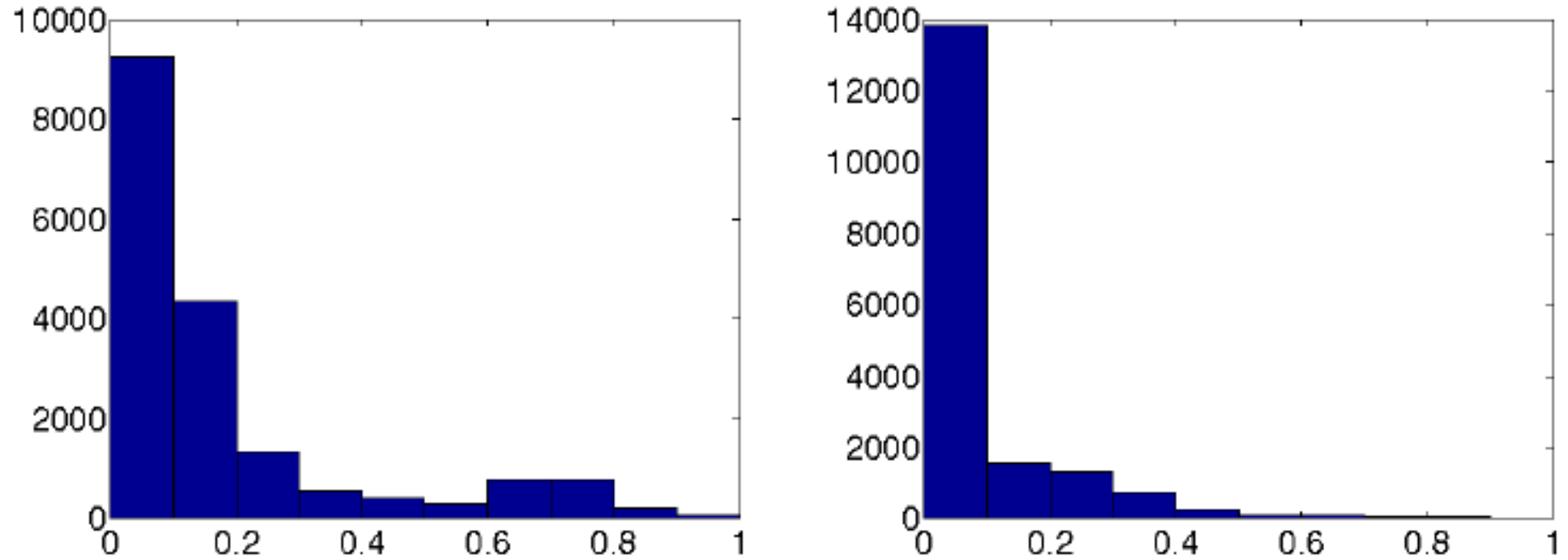


# Practical use of SVM (Cont.)

- Linear scaling of features is done

$$\frac{(\mathbf{x}_i)_s - \min(\mathbf{x}_t)_s}{\max(\mathbf{x}_t)_s - \min(\mathbf{x}_t)_s}, \forall s = 1, \dots, n.$$

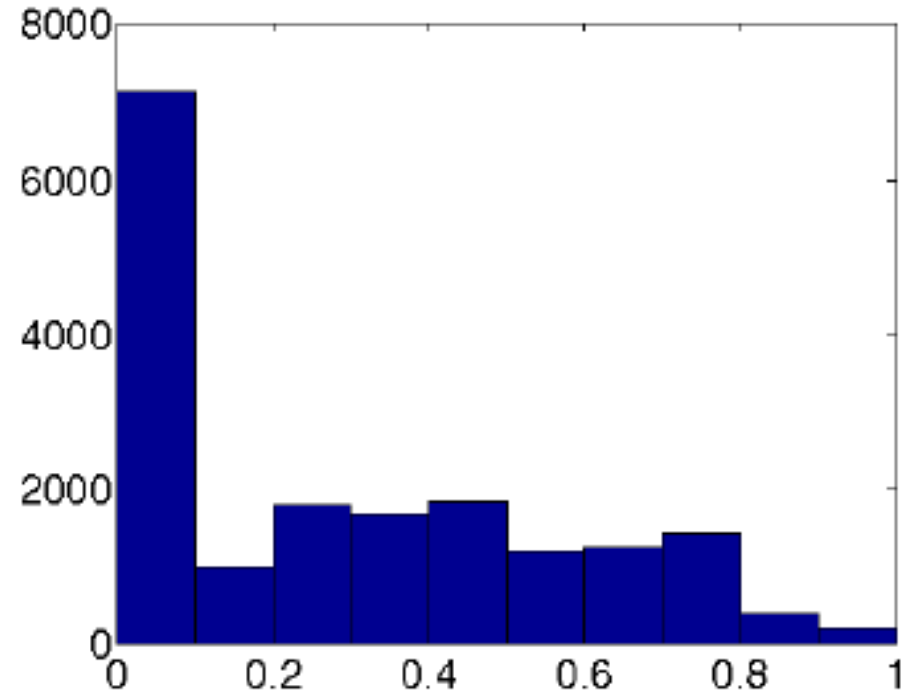
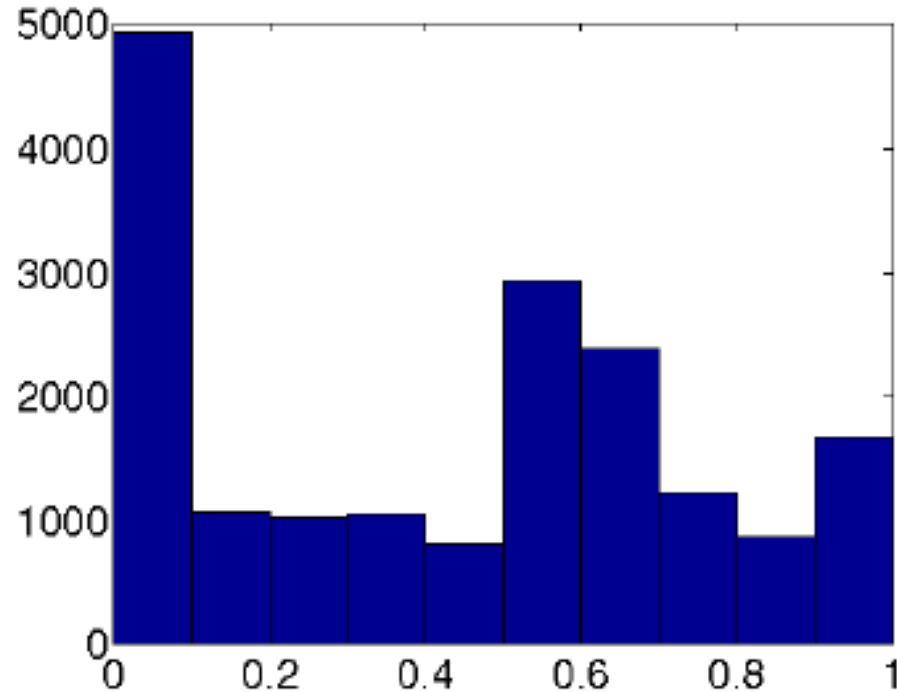
# Practical use of SVM (Cont.)



(a) Linearly scaled to [0,1].

# Practical use of SVM (Cont.)

linear scaling  $\rightarrow$   $\log(\text{feature value} + 0.01)$   $\rightarrow$  linear scaling.



(b) A log-scaling procedure by (7).

# Practical use of SVM (Cont.)

- Parameter selection
  - Regularization parameter (C)
  - Kernel parameter ( $\gamma$  in case of RBF kernel)

$$C \in \{2^{-1}, 2^0, \dots, 2^9\} \text{ and } \gamma \in \{2^0, 2^1, \dots, 2^8\}$$

- Select the one achieving the best five-fold CV accuracy

# Practical use of SVM (Cont.)

- Results

SVM procedures	CV accuracy (%)
Linear scaling + parameter selection	89.20
Log scaling + parameter selection	90.48

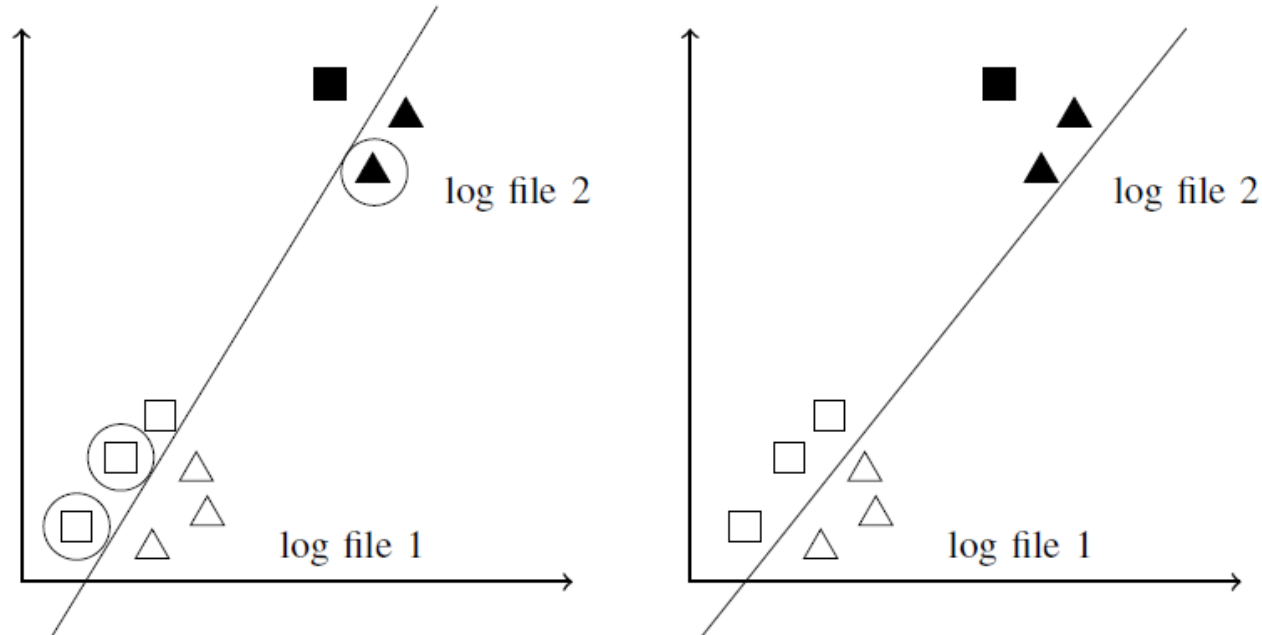
# Pitfall of CV accuracy

- Although CV accuracy is most widely used evaluation measure but it can over-estimate the real performance.
- Assume each user records 10 log files and each log file generates 100 feature vectors.

user 1	log file 1	$\mathbf{x}_1, \dots, \mathbf{x}_{100}$
	log file 2	$\mathbf{x}_{101}, \dots, \mathbf{x}_{200}$
	$\vdots$	
	log file 10	$\mathbf{x}_{901}, \dots, \mathbf{x}_{1000}$
user 2	log file 11	$\mathbf{x}_{1001}, \dots, \mathbf{x}_{1100}$
	$\vdots$	

# Pitfall of CV accuracy (Cont.)

- Feature vector in the same log file shares some information.
- In CV procedure if data from one log file appear in both training and validation sets, then the prediction becomes easy.



# Pitfall of CV accuracy (Cont.)

- Therefore the standard instance-wise split of data may easily overestimate the real performance.
- To eliminate the sharing of meta-information, data split should be made at higher level such as logs or users.

CV strategy	SVM CV accuracy (%)
Instance-wise CV	90.48
Log-wise CV	83.37



# Pitfall of CV accuracy (Cont.)

- Although log-wise CV is more reasonable but its better to have an independent test set collected by a completely different group of users.

Classifiers	CV accuracy (%)	Test accuracy (%)	Model size (KB)
Decision Tree	89.41	77.77	76.02
AdaBoost	91.11	78.84	1500.54
SVM	90.48	85.14	1379.97

- The result confirms that instance-wise CV may severely over-estimate.

# Pitfall of CV accuracy (Cont.)

- Similarly in “Towards physical activity diary: motion recognition using simple acceleration features with mobile phones” by J. Yang (2009)

Reported CV accuracy	80 – 90 %
Reported Test accuracy	< 70 %

	2 folds	5 folds	8 folds
CV accuracy	85.05	83.37	82.21
Test accuracy	85.33	85.14	84.66

# Model size reduction

- Although good accuracy achieved but the model size is much larger than 16 KB.
- Large size due to storage of optimal solution  $\alpha$  and support vectors.
- Because it is a multi-class problem and LIBSVM uses one-against-one method so for  $k$ -class problem the model size is

$$\binom{k}{2} \times \# \text{ support vectors} \times (k + n) \times 4\text{bytes}$$

- Where  $n$  is the number of features.

# Model size reduction (Cont.)

- To reduce size use polynomial kernel.

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i^T \mathbf{x}_j + 1)^d$$

- Where  $\gamma$  is the kernel parameter and  $d$  is the degree.
- The kernel is the inner product of two vectors  $\phi(\mathbf{x}_i)$  and  $\phi(\mathbf{x}_j)$
- If  $d = 3$

$$\phi(\mathbf{x}) = [1, \sqrt{3\gamma}x_1, \dots, \sqrt{3\gamma}x_n, \sqrt{3\gamma}x_1^2, \dots, \sqrt{3\gamma}x_n^2,$$

$$\sqrt{6\gamma}x_1x_2, \dots, \sqrt{6\gamma}x_{n-1}x_n, \gamma^{3/2}x_1^3, \dots, \gamma^{3/2}x_n^3,$$

$$\sqrt{3\gamma}^{3/2}x_1^2x_2, \dots, \sqrt{3\gamma}^{3/2}x_n^2x_{n-1}, \sqrt{6\gamma}^{3/2}x_1x_2x_3, \dots, \sqrt{6\gamma}^{3/2}x_{n-2}x_{n-1}x_n]^T.$$

# Model size reduction (Cont.)

- Only  $w$  and  $b$  need to be stored.

$$\begin{aligned} & \binom{k}{2} \times (\text{length of } w + 1) \times 4\text{bytes} \\ &= \binom{k}{2} \times \left( \binom{n+d}{d} + 1 \right) \times 4\text{bytes}. \end{aligned}$$

- For  $d = 3$ , the model size turns out to be 2.28 KB

# Model size reduction (Cont.)

- Comparison among kernels

SVM method	Test accuracy (%)	Model size (KB)
RBF kernel	85.33	1287.15
Polynomial kernel	84.79	2.28
Linear kernel	78.51	0.24

# Fast training by optimization

1. The training of kernel SVM is known to be slow.
  2. Because of using  $K(\mathbf{x}_i, \mathbf{x}_j)$  rather than  $\phi(\mathbf{x}_i)$  or  $\phi(\mathbf{x}_j)$ , the setting is very restricted.
- For linear SVM the optimization problem becomes

$$\min_{\mathbf{w}} \quad \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^l \xi(\mathbf{w}; \mathbf{x}_i, y_i)$$

- Where  $\xi(\mathbf{w}; \mathbf{x}_i, y_i)$  is the loss function

# Fast training by optimization (Cont.)

- Commonly used loss functions

$$e^{-y_i \mathbf{w}^T \mathbf{x}_i}$$

logistic regression

$$\max(1 - y_i \mathbf{w}^T \mathbf{x}_i, 0)$$

hinge-loss (l1-loss) SVM

$$\max(1 - y_i \mathbf{w}^T \mathbf{x}_i, 0)^2$$

squared hinge-loss (l2-loss) SVM

- The three loss functions are related so they give similar test result.



# Fast training by optimization (Cont.)

- Comparison scenarios
  - I. LIBSVM: polynomial kernel with hinge loss.
  - II. LIBLINEAR (primal): Linear SVM with squared hinge loss.
  - III. LIBLINEAR (dual): Linear SVM with squared hinge loss.

# Fast training by optimization (Cont.)

	LIBSVM	LIBLINEAR	
		Primal	Dual
Test accuracy	84.79	84.52	84.31
Training time	30519.10	1368.25	4039.20

- LIBSVM and LIBLINEAR (primal) give similar accuracy.
- Training time of LIBSVM is significantly high.
- In theory, both primal and dual solvers give exactly same accuracy.

# Multi-class SVM

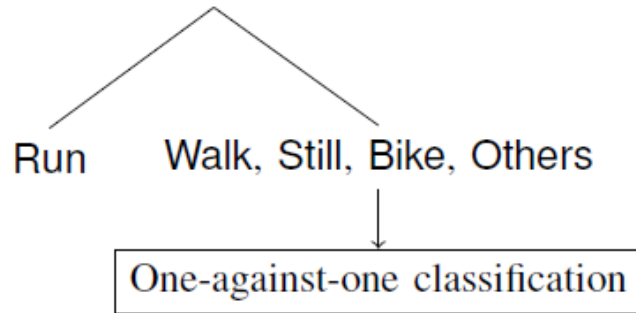
- SVM is designed for two-class classification.
- For multi-class two methods are used
  - One-against-one (Stores  $k(k-1)/2$  weight vectors)
  - One-against-rest (Stores  $k$  weight vectors)
- For 5 transport modes, we need 10 and 5 vectors respectively.

# Multi-class SVM (Cont.)

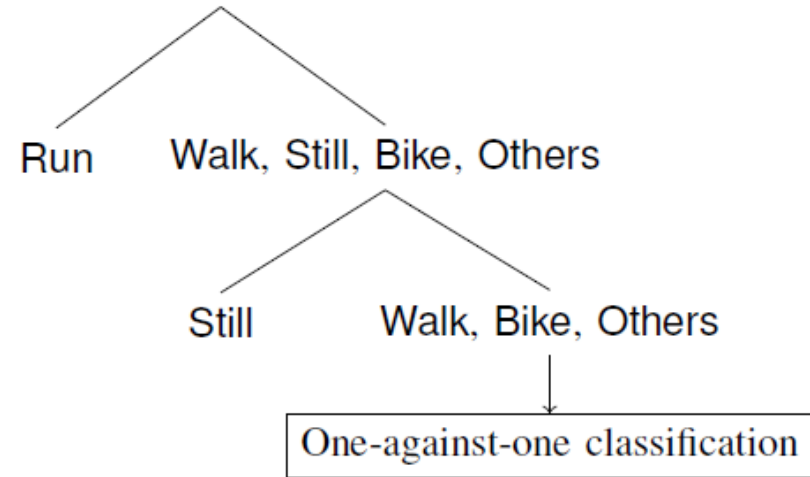
- Results

SVM method	Test accuracy (%)	Model size (KB)
One-against-one	84.52	2.24
One-against-rest	83.95	1.12

# Multi-class SVM (Cont.)



(a) A hierarchical setting to identify the mode Run first.



(b) A hierarchical setting to identify the modes Run and Still first.

a.  $1 + 4(4 - 1)/2 = 7$  weight vectors

b.  $1 + 1 + 3(3 - 1)/2 = 5$  weight vectors

# Multi-class SVM (Cont.)

- Results

SVM method	Test accuracy (%)	Model size (KB)
One-against-one	84.52	2.24
One-against-rest	83.95	1.12
Hierarchy 1	84.46	1.57
Hierarchy 2	84.53	1.12

# Non-machine learning issues

- Feature engineering
- Extracting important features is one of the most crucial steps.
- Added two frequency-domain features.
  - a. Peak magnitude: index of the highest FFT value.
  - b. Ratio: ratio between largest and second largest FFT values.

# Non-machine learning issues (Cont.)

- Results

CV strategy	5 features	Adding 2 FFT features
Instance-wise CV	89.90	92.98
Log-wise CV	85.05	89.26
Test accuracy	85.33	91.53



# Non-machine learning issues (Cont.)

- Use of Domain knowledge
- Using information from past predictions.
- Power saving by not enabling the classifier in some situations.

# Conclusion

- Direct use of a machine learning method may not give satisfactory results.
- Careful evaluation criterion must be chosen as this study showed that standard CV accuracy can slightly over-estimate.
- Practitioner should take care while employing classifiers and should have deeper understanding of the methodology.