Using Support Vector Machine for Emotion Classification in presence of Noise Label

Po-sheng Wang

Advisor: **Dr. Ehsan Tarkesh Esfahani**

Committee Member: **Dr Rahul Rai**

Committee Member: **Dr. Amin Karami**

Human In Loop Laboratory

M.S. in Mechanical Engineering. University at Buffalo

August 13, 2015

Problem Definition

Facial Expression Recognition

- Ambiguity of data classes : even human may be confused

Challenge

- Classification of data with label noise

Approach

- Human subject study to estimate the noise in the label of data
- Compared Standard SVM with Noise Label Robust SVM

Noise Label issue

Where is the noise coming from … ambiguity of data classes

True label : Disgust

True label : Anger

True label : Sad

Happy, Disgust, Anger Sad, Neutral

Data Set and Classes
Japanese Female Expression (JAFFE)

Class 1: Happy (30 images)

Class 2: Disgust (29 images)

Class 3: Anger (30 images)

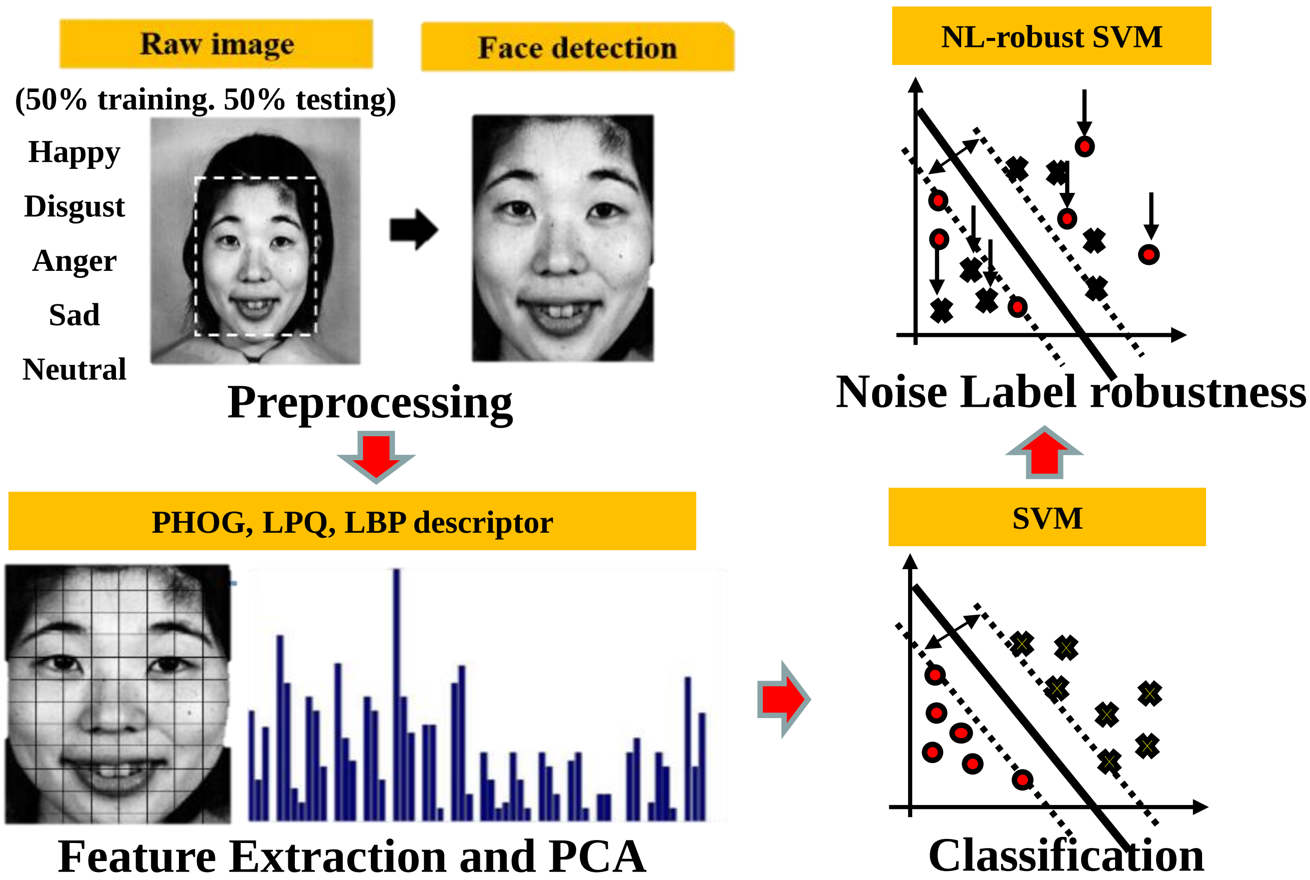
Class 4: Sad (31 images)

Class 5: Neutral (30 images)

Total: 150 images

•••

Flow Chart

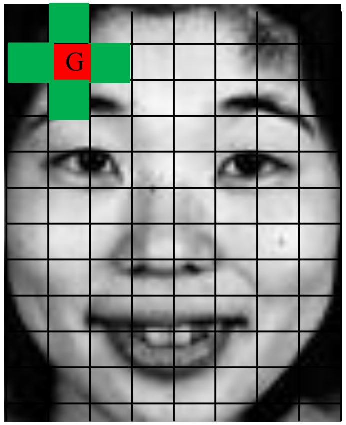

Raw image
Face detection
(50% training. 50% testing)
Happy
Disgust
Anger
Sad
Neutral
Preprocessing
NL-robust SVM
Noise Label robustness
PHOG, LPQ, LBP descriptor
Feature Extraction and PCA
SVM
Classification

Classification – without considering noise in label

- **Feature extraction**
  - Pyramid Histogram of Oriented Gradients (PHOG)
  - Local Phase Quantization (LPQ)
  - Local Binary Patterns (LBP)
- **Classification**
  - Support Vector Machines (SVM)

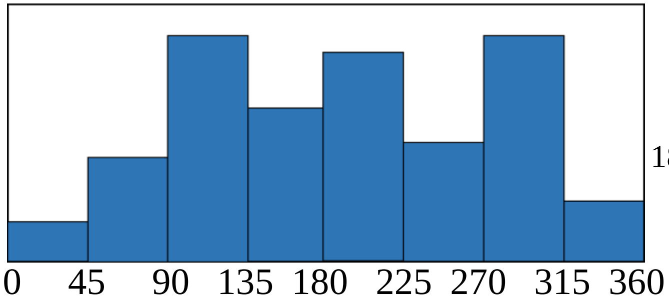
Pyramid Histogram of Gradients (PHOG)

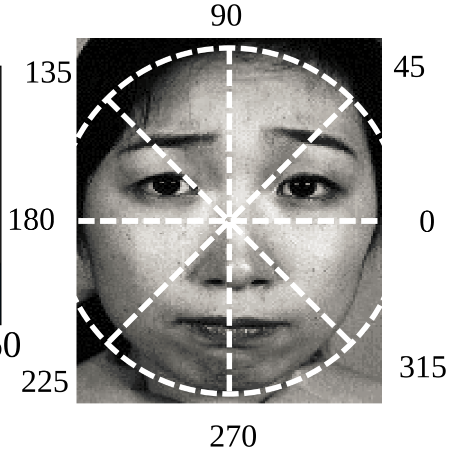
Calculating histogram information of image with kernel filters: [-1 0 1] and $[-1\ 0\ 1]^T$



- For pixel number 1 : m
  - $Horizontal\ gradient : G_x(x,y) = H(x+1,y) - H(x-1,y)$
  - $Vertical\ gradient : G_y(x,y) = H(x,y+1) - H(x,y-1)$
  - *Magnitude:* $\mathrm{G(x,y)} = \sqrt{G_x(x,y)^2 + G_y(x,y)^2}$,
  - *Angle :* $\alpha(\mathrm{x,y}) = tan^{-1}\left(\frac{G_y(x,y)}{G_x(x,y)}\right)$
  - Bin size = 8 . Range = [0-360] .
- End

Count the occurrences in each of the bin

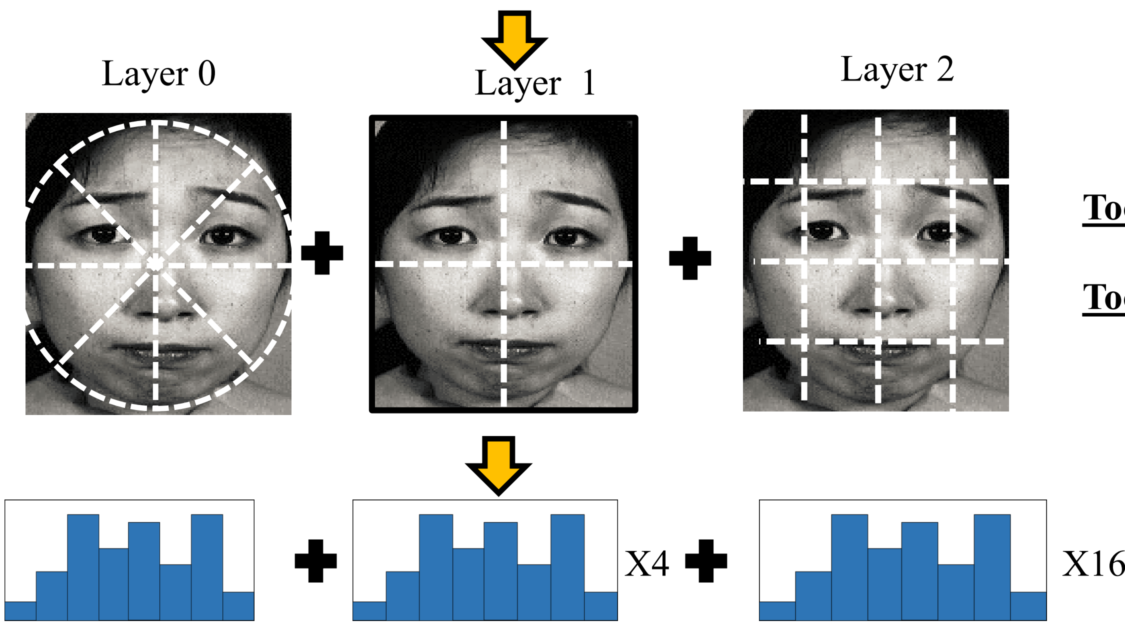




A. Bosch, A. Zisserman, and X. Munoz, “Representing shape with a spatial pyramid kernel,” in Proceedings of the ACM International Conference on Image and Video Retrieval, 2007.

Pyramid Histogram of Gradients (PHOG)

Layers (L)	0	1	2	3	4	Total
# of histogram	8	32	128	512	2048	2728



Output PHOG descriptor

Too Few layers: Information too rough

Too many layers: Information too detail

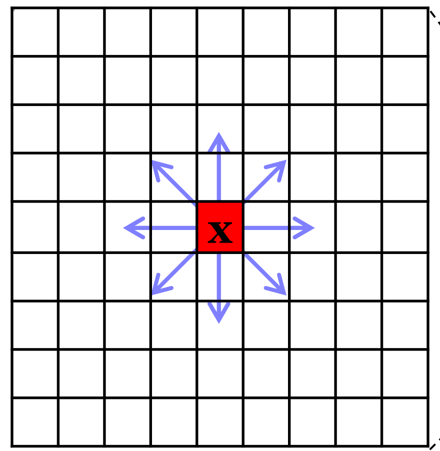
Local Phase Quantization (LPQ)

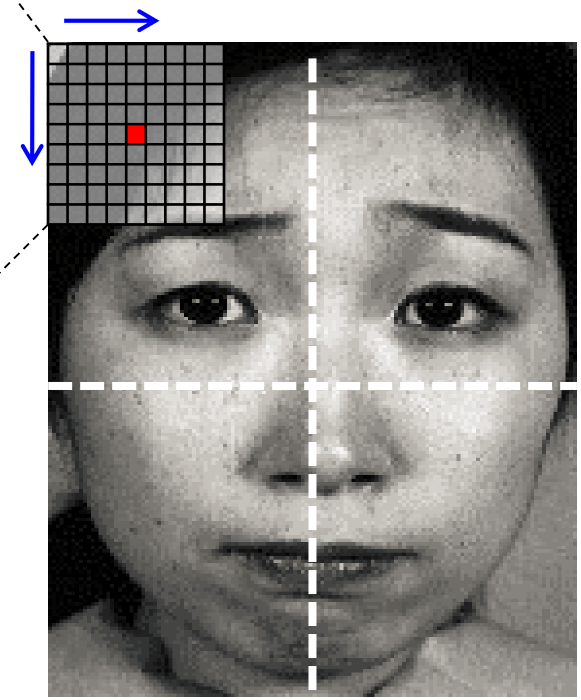
Quantize the local phase information by calculating the local Fourier transform and represent those information by the histogram distribution over the all pixels in the image.

1- Local Fourier transform

$$F(u,x) = \sum_{y \in N_x} f(x-y)e^{-j2\pi u^T y}$$

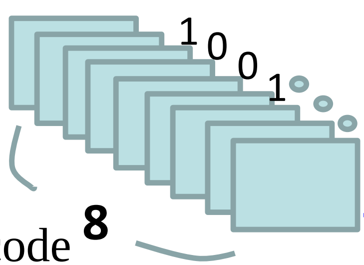
Directions: [1 0], [0 1], [1 1], [1 -1]



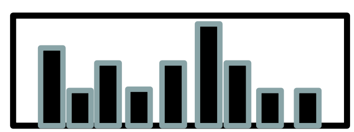


2- Calculating the histogram of phase angle

Each direction will have a ***Re***(F) and ***Im***(F) part.
We calculate the histogram of ***sign*** of these values



Binary code **8** → Decimal



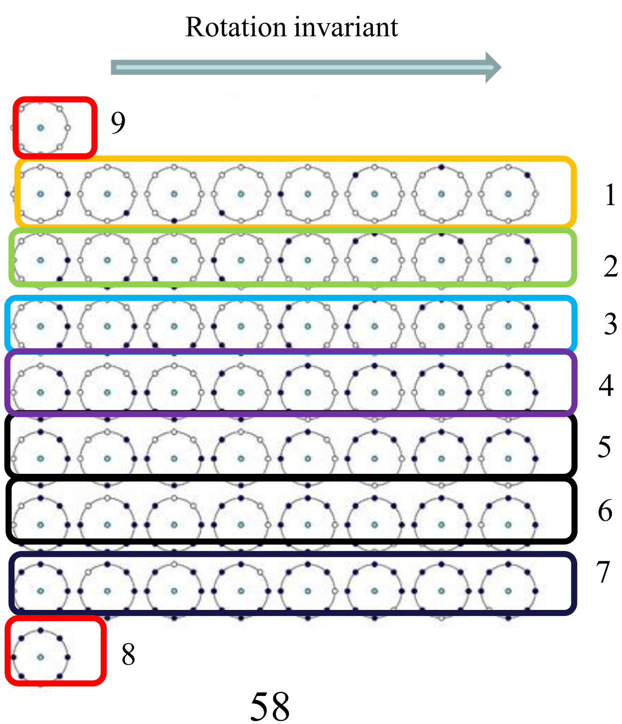
X4

Count occurrences of each possible value over all image

V. Ojansivu and J. Heikkil, "Blur insensitive texture classification using local phase quantization," in Image and Signal Processing, ser. Lecture Notes in Computer Science, 2008

Local Binary Pattern (LBP)

LBP creates the possible pattern label by thresholding neighborhood of each pixel with the intensity value



1. Uniform Patterns:

Contains, at most, two bitwise transitions

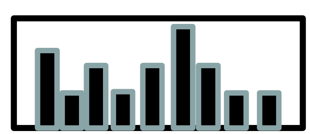
2. Rotation invariant

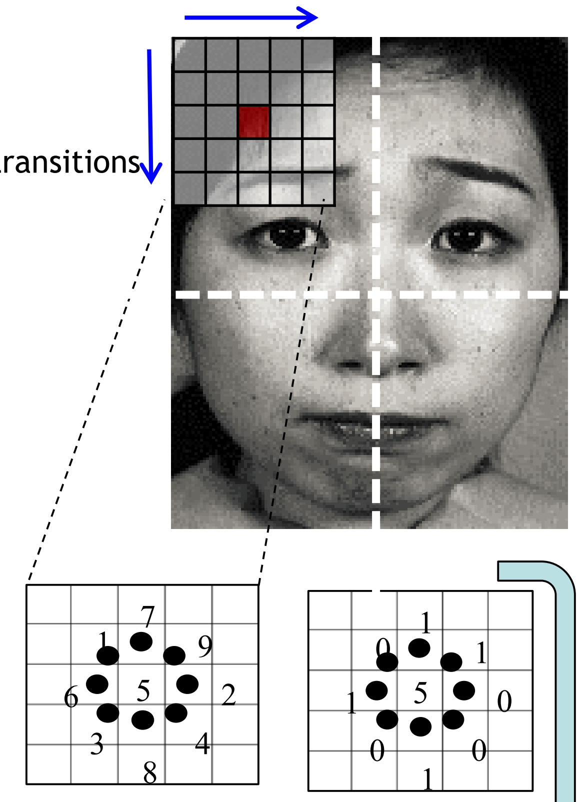
0 1 1 0 0 1 0 1

1 1 0 0 1 0 1 0

Possible labels 1:10

(1 to 9 shows left, and 10 is any other patterns except uniform patters)



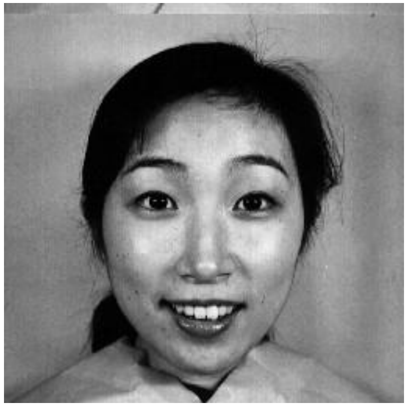


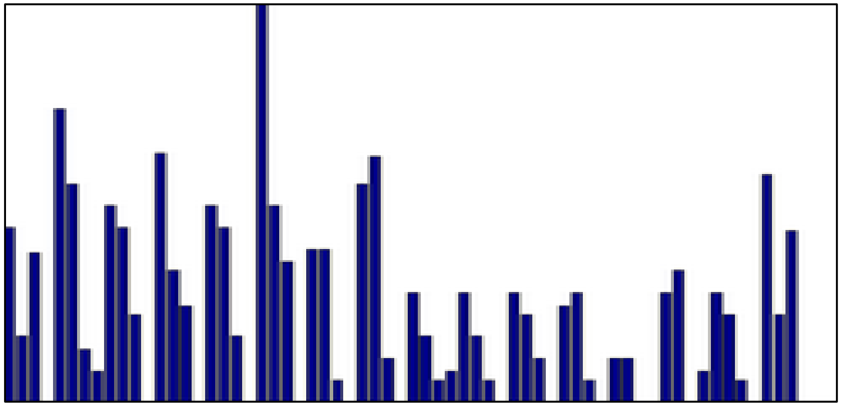
Binary: 01100101
Decimal: 101
Pattern

T. Ojala, M. Pietikainen, and D. Harwood, "A comparative study of texture measures with classification based on featured distribution," Pattern Recognit., vol. 29, no. 1, pp. 51 –59, Jan. 1996.

http://cs.stackexchange.com/questions/19844/number-of-different-output-labels-in-local-binary-pattern

Feature extraction

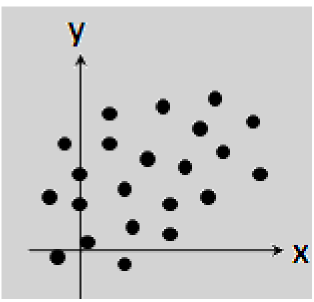




Feature Extraction	PHOG	LPQ	LBP	Total
# of features	2728	1280	3410	7418

7418 features to represent each image

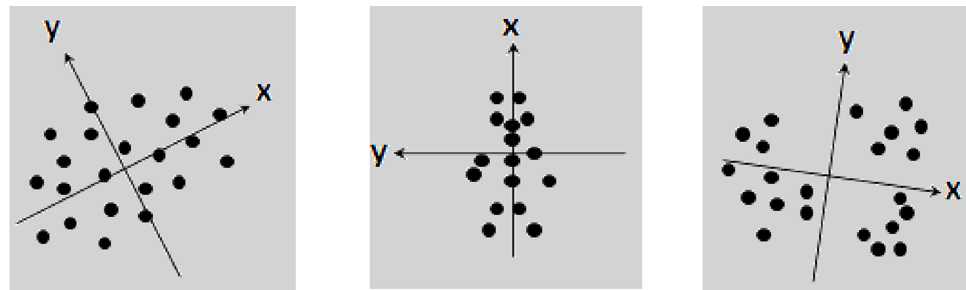
Principal Component Analysis (PCA)



Linear transformation

Project the data

PCA



Original Feature coordinate

New Feature coordinate

Figure Source: http://www.csie.ntnu.edu.tw/~u91029/Matrix.html

- So that the greatest variance by any projection of the data set comes to lie on the first axis → **First Component**
- By selecting the first few components → **Data dimension reduce without losing important information**

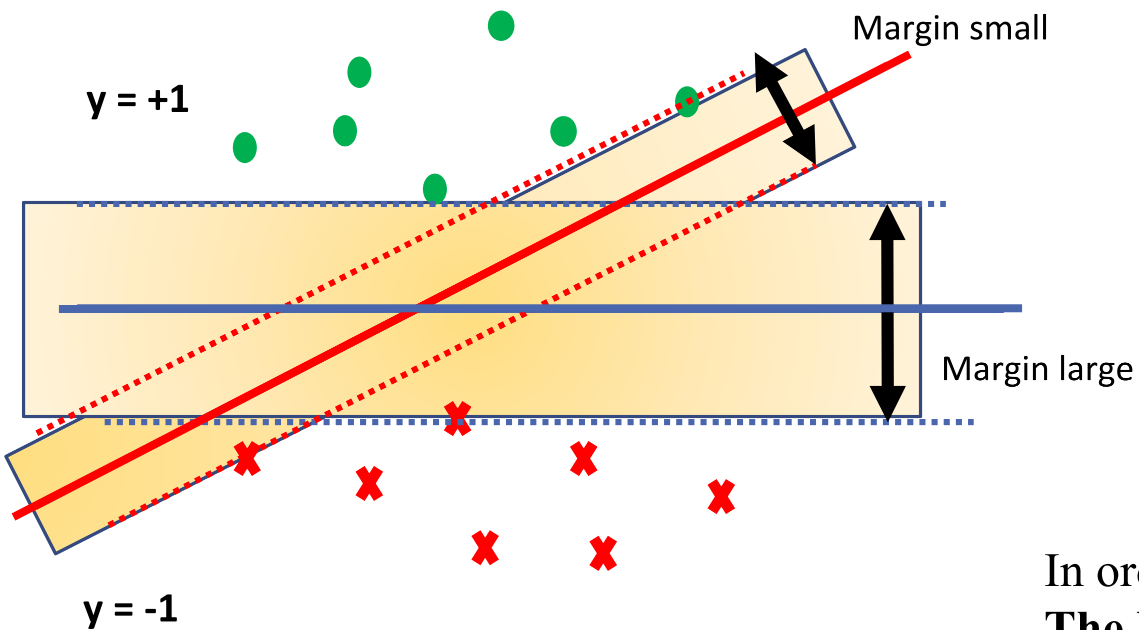
- **How many components should we choose???**

Pearson, K. (1901). "On Lines and Planes of Closest Fit to Systems of Points in Space". Philosophical Magazine 2 (11): 559–572.

Abdi. H., & Williams, L.J. (2010). "Principal component analysis.". Wiley Interdisciplinary Reviews: Computational Statistics, 2: 433–459

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology, 24, 417–441, and 498–520.
Hotelling, H. (1936). Relations between two sets of variates.Biometrika, 27, 321–77

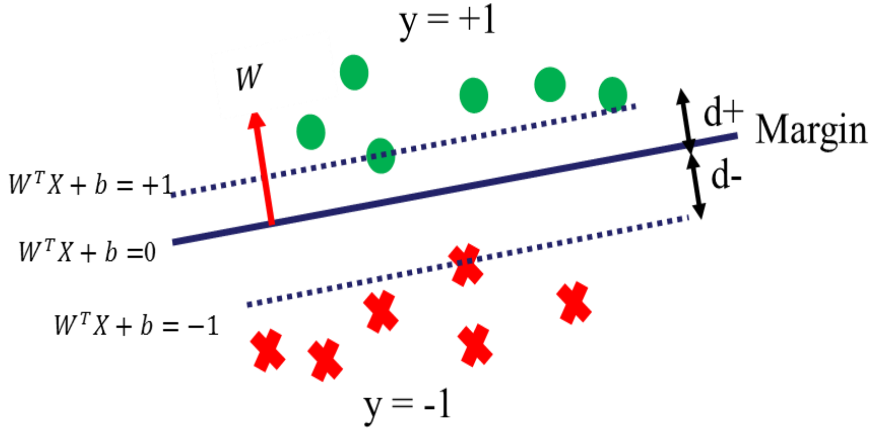
Classification-
Support Vector Machine (SVM)



In order to separate the data most:
The larger margin → the better

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.

Support Vector Machine (SVM)



Goal: Hyperplane with max margin

$$d(+) + d(-) = \frac{1}{||w||} + \frac{1}{||w||} = \frac{2}{\sqrt{w^T w}}$$

$$\text{s.t.} \begin{cases} W^T X + b \geq 1 & if\ y_i = 1 \\ W^T X + b \leq -1 & if\ y_i = -1 \end{cases}$$

W : normal vector
b : offset

Primal Optimization Problem

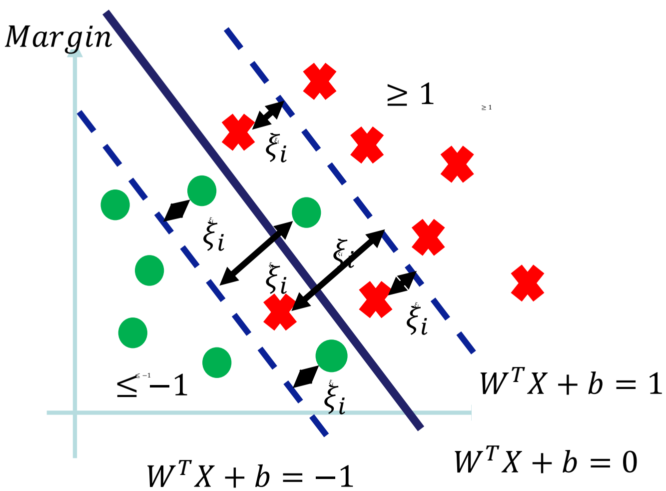
$$\boldsymbol{Minimize} \quad \frac{1}{2} \boldsymbol{w^T w}$$

$$\boldsymbol{s.t.} \quad \boldsymbol{y_i(w^T x_i + b) \geq 1 \quad i = 1, \ldots. n}$$

$$discriminant\ function: \ y_{new} = sign(w^T x + b)$$

C

Support Vector Machine (SVM)



Linear nseparable problems :
Soft Margin Hyperplane

Introduce a non-negative slack variable ξ_i for each data
Allow “error” ξ_i in classification

$$\begin{cases} W^TX + b \geq 1 & if\ y_i = 1 \\ W^TX + b \leq -1 & if\ y_i = -1 \end{cases} \Rightarrow \begin{cases} W^TX + b \geq 1 - \xi_i & if\ y_i = 1 \\ W^TX + b \leq -1 + \xi_i & if\ y_i = -1 \end{cases}$$

$$\text{Minimize} \quad \frac{1}{2} w^T w \qquad s.t. \quad y_i(w^T x_i + b) \geq 1 \quad i = 1, \ldots. n$$

$$\Rightarrow$$

$$\text{Minimize} \quad \frac{1}{2} w^T w + \boldsymbol{C} \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{\xi_i}$$

$$s.t. \quad y_i(w^T x_i + b) \geq 1 - \xi_i \quad i = 1, \ldots. n, \qquad \xi_i \geq 0, i = 1, \ldots \ldots, n.$$

Parameters

C

Kernel type,

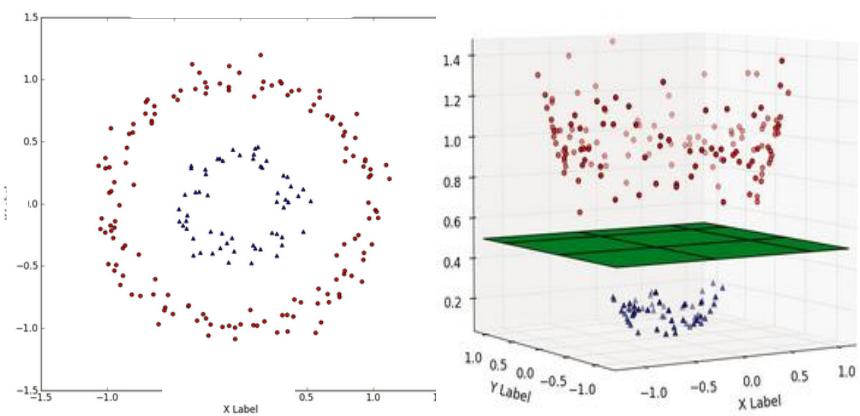
Kernel Parameter

Support Vector Machine (SVM)

Extension to **Non-linear decision boundary**

Key idea: Transform data by **kernel function** to higher dimensional space to **make it linear**

$R^2 dimension$ $R^3 dimension$



Kernel function Feature mapping

Linear Kernel : $\boldsymbol{K(x, y) = x^T y}$

Polynomial Kernel : $\boldsymbol{K(x, y) = (x^T y)^r}$

Radial Basis Function Kernel (RBF) : $\mathbf{K}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{exp}(-\frac{||x-y||^2}{2\sigma^2})$

Sigmoid Kernel : $\mathbf{K(x, y) = tanh(\alpha x^T y - 1)}$

Parameter

Source: http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html

C

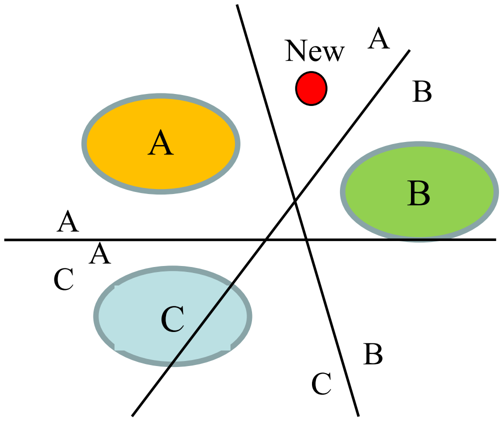
Kernel type,

Kernel Parameter

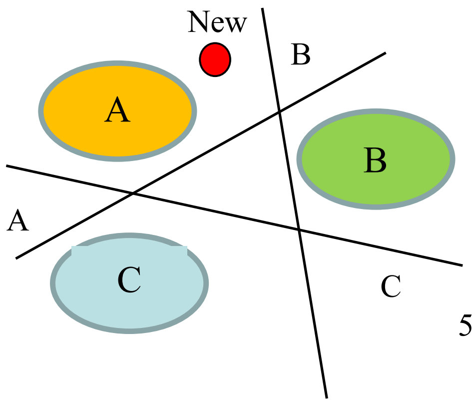
Support Vector Machine (SVM)

SVM multi-class classification → Combine a lot of binary classifications!

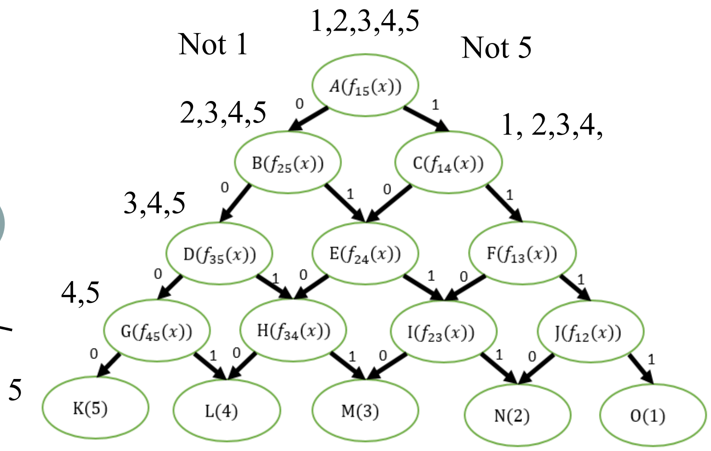
- One vs One
- One vs Others
- Directed Acyclic Graph (DAGSVM)



Train $k(k-1)/2$ binary SVMs
Largest vote from all the classifiers



Train k binary SVMs



Train k classifier

C

Kernel type,

Kernel Parameter

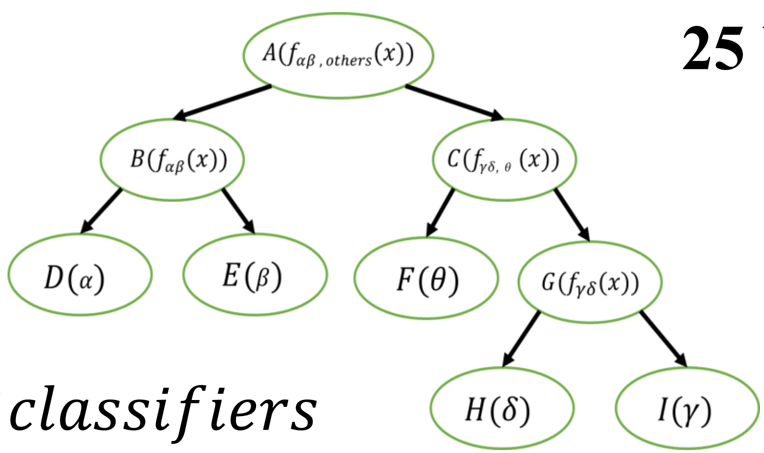
Support Vector Machine (SVM)

Strategy type	Classes combination
One-against-One	1 vs 2
	1 vs 3
	1 vs 4
	1 vs 5
	2 vs 3
	2 vs 4
	2 vs 5
	3 vs 4
	3 vs 5
	4 vs 5

Strategy type	Classes combination
One-against-All	1 vs All
	2 vs All
	3 vs All
	4 vs All
	5 vs All

Strategy type	Classes combination
Two-against-All	{1,2} vs others
	{1,3} vs others
	{1,4} vs others
	{1,5} vs others
	{2,3} vs others
	{2,4} vs others
	{2,5} vs others
	{3,4} vs others
	{3,5} vs others
	{4,5} vs others

25 binary classification



k^2 *classifiers*

C

Kernel type,

Kernel Parameter

Classifier Selection

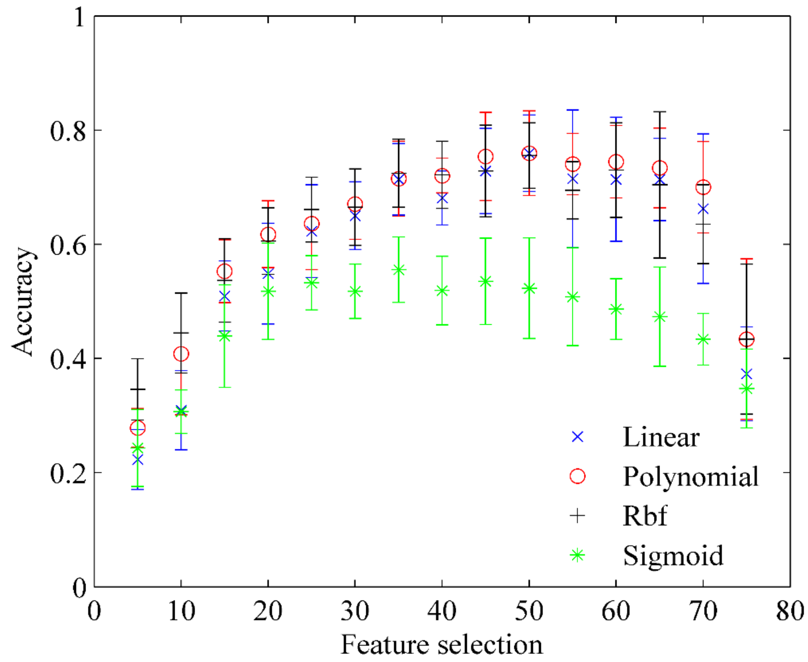
Optimizing the type of **Kernel**

Optimizing the **kernel parameter** and **C value** : **2^(-6)~2^(6)**

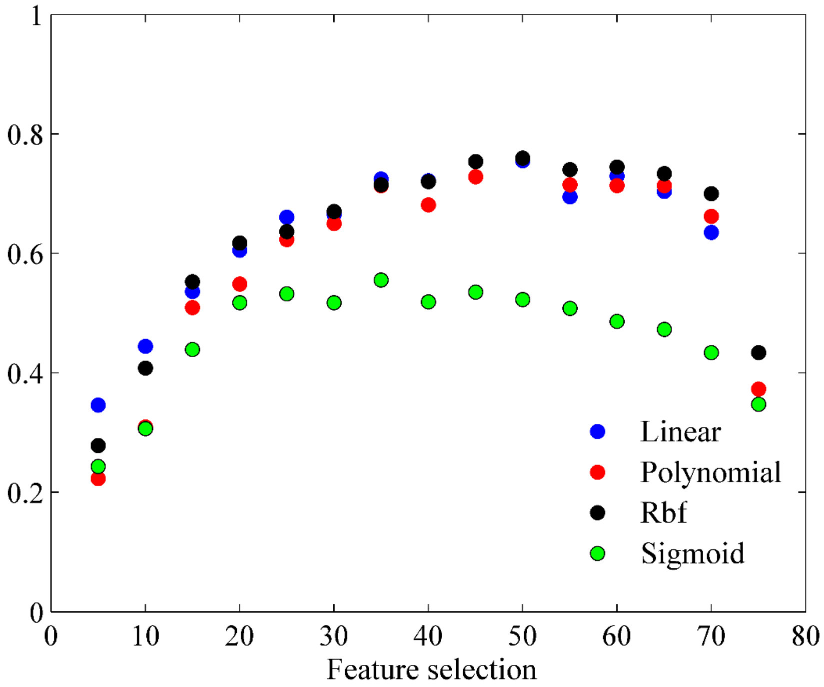
Optimizing the **number of features**

- For Feature number 1:5:75
  - For classifiers number 1:10
    - For kernel parameter 1:m
      - 10 random folds with 50%-50 classification (2 cross validation)
    - End
  - End
- Select the best parameter for each classifier
- End

Effects of number of features and Kernel type/parameters on classification accuracy



Mean and deviation of 10 random fold



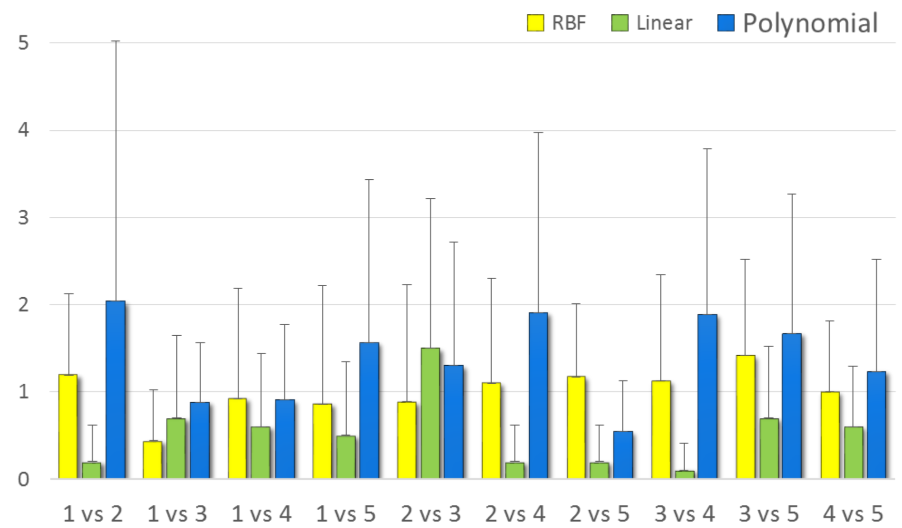
Mean of 10 random fold.

Sensitivity Analysis

- For each kernel type, compare the optimum parameter of each random fold to the optimum parameter of overall random folds

  (10 fold cross validation is performed)

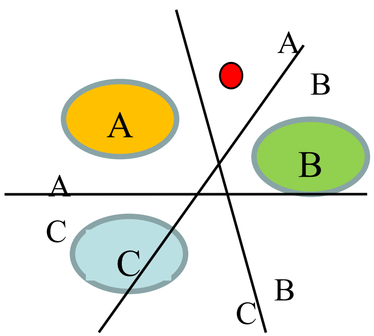
- The smaller the distance: the more robust the system



Binary classifier	1 vs 2	1 vs 3	1 vs 4	1 vs 5	2 vs 3	2 vs 4	2 vs 5	3 vs 4	3 vs 5	4 vs 5
Most stable kernel	Linear	RBF	Linear	Linear	RBF	Linear	Linear	Linear	Linear	Linear

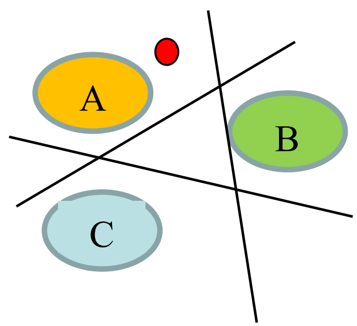
Final Classifier Selection

One vs One

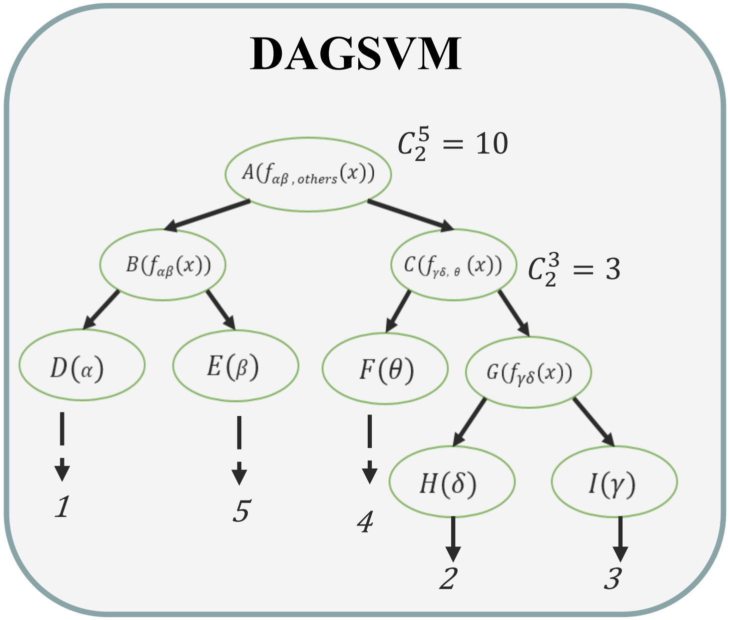


76%

One vs Others



75%



81%

81% Accuracy – without considering the noise in label

Label Noise (LN) Robust SVM[1]

- **Label Flip → Noise label in the training data :**
  Label of each data i will flip with some certain probability u

- **Kernel matrix becomes:**

$$Q_{ij} = y_i y_j K(x_i, x_j)(1 - 2\varepsilon_i)(1 - 2\varepsilon_j)$$

i

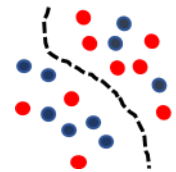
$\vec{x}_i$

y_i

$y'_i = y_i(1\text{-}2\varepsilon_i)$

- **To be less sensitive to label flips, we learn an SVM using the expected kernel matrix :**

$$Q = \begin{cases} y_i y_j K(x_i, x_j)(1 - 4\sigma^2), & if\ i \neq j \\ y_i y_j K(x_i, x_j), & othrwise \end{cases}$$

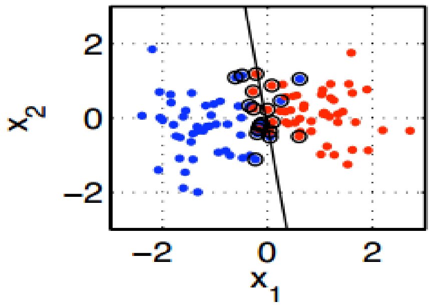


Where μ *is the mean value of* ε, $\sigma^2 = \mu(1 - \mu)$

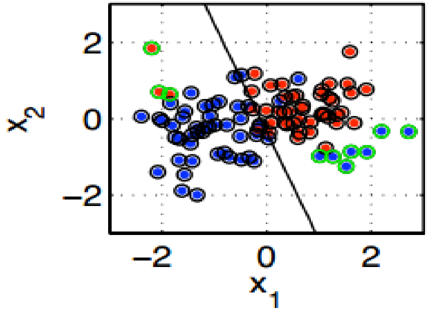
1. Biggio, B., Nelson, B., & Laskov, P. (2011). Support Vector Machines Under Adversarial Label Noise. In ACML (pp. 97-112).

Label Noise (LN) Robust SVM[1]

Untainted data
Trained on Standard SVM



Tainted data (10 % flipped)
Trained on LN-robust SVM



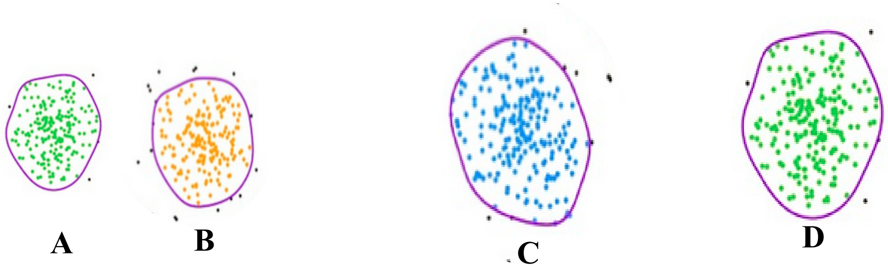
Binary classification NL-robust SVM

⇓

?? How to extend the binary classification to multiple classes ??
?? Does the separability of data affect the performance??

⇓

Preliminary data simulation
Separability of data may affect the performance of NL-robust SVM

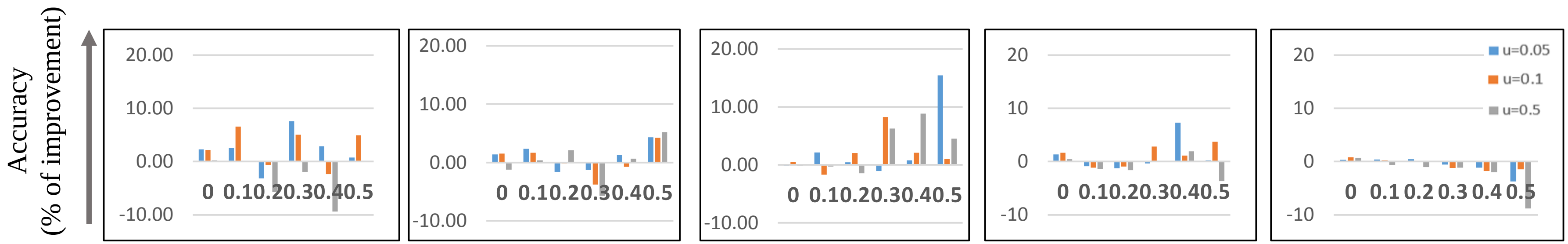


1. Biggio, B., Nelson, B., & Laskov, P. (2011). Support Vector Machines Under Adversarial Label Noise. In ACML (pp. 97-112).

Label Noise (LN) Robust *SVM* -- Separability investigation on modeling data

Generate different separability of data

Data cases	(1)	(2)	(3)	(4)	(5)
Separability[2]	1.35%	2.77%	6.99%	13.2%	32.2%



(1) (2) (3) (4) (5)

- Label flipping: 10%, 20%, 30%, 40%, 50%
- Apply μ : 0.05, 0.1, 0.5
- Standard SVM V.S. LN Robust SVM on different separability cases

- Modeling Result :
  1. Case (3) is the best
  2. Much spearability → Decreasing the performance
  3. Performance decrease suddenly in Case (2)
- **Separability affects the performance of LN Robust SVM**

2. Sihouette value is used to measure the seperability of two clusters of data

Emotion classification – Consider noise in labels

- **Human Subject Study:**

  5 number of subjects (4 male and 1 female) to classify the images to one or more appropriate emotional classes.

- **Confusion matrix and voting probability**
  - Higher ambiguity images → Training set
  - Less ambiguity images → Testing set

Noise label dataset (NL-dataset)

	Happy	Disgust	Anger	Sad	Neutral
Happy	**92.6**	0	0	1	6.3
Disgust	1.3	**46**	12.5	35.2	4.8
Anger	0.3	11.2	**45.6**	27.8	14.8
Sad	3.87	2.9	2.9	**79**	11.2
Neutral	8.3	0	0.6	4.3	**86.6**

Accuracy: 69%

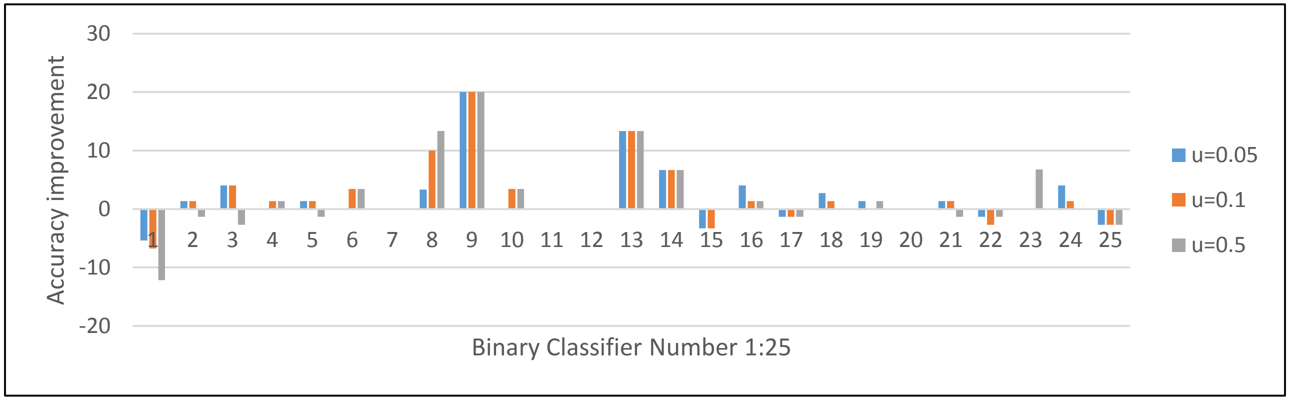
Noise information

Image No.	Happy	Disgust	Anger	Sad	Neutral
1	0	0	0.2	0.3	0.5
2	0.3	0.2	0	0	0.5
3	0	0	0	0.6	0.4
4	0	0	0	0.8	0.2
5	1	0	0	0	0
6	0.4	0	0	0	0.6
		……..			
150	0	0.2	0.4	0.2	0.2

Voting probability

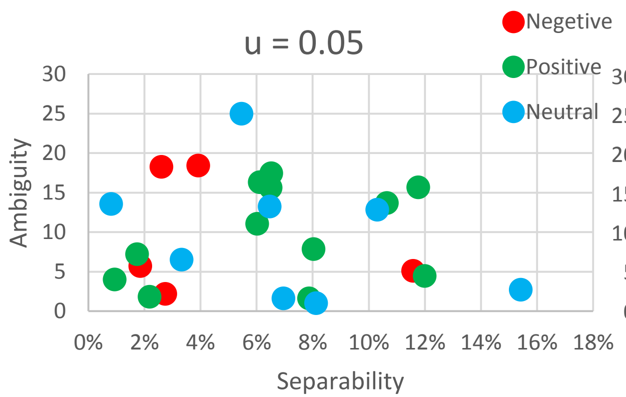
Emotion classification – Consider noise in labels

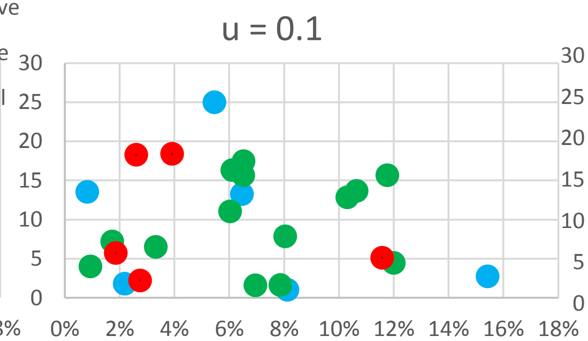
- For binary classifier number 1:25
  - Apply u =0.05, 0.1 and 0.5 on LN-Robust SVM
    - Calculate the separability and ambiguity[3]
    - Accuracy improvement compared with Standard SVM
  - End
- End

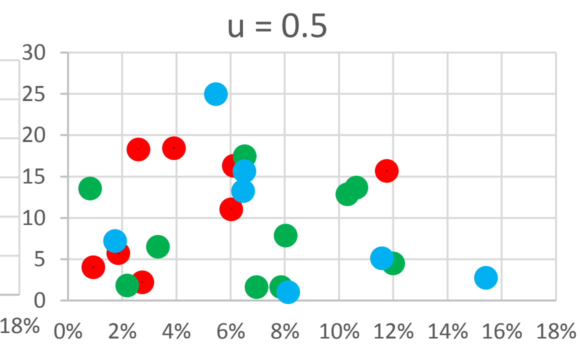


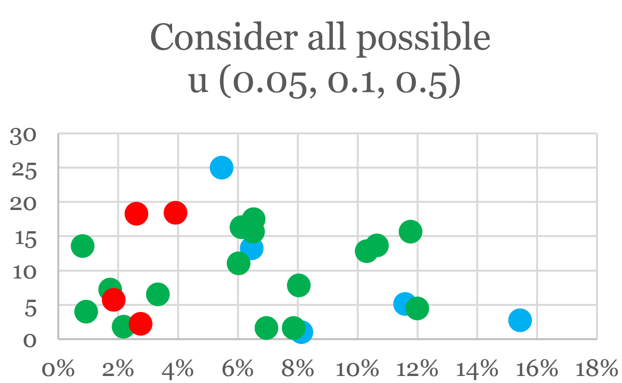
3. Ambiguity is measured on human subject voting probability :
if $i\ v.s\ j$, then only consider data with true label is i and j.
$if\ true\ is\ i, the\ ambiguity\ is\ \frac{Probability\ (j)}{Probability(i)+probability\ (j)}$

Emotion classification – Consider noise in labels









Conclusion:

1. Each point stand for one of the classifiers
2. No matter what u value, there are still 4 classifiers reduce the accuracy
3. **<u>NL-robust SVM may not be effective for all classifiers because of separability criteria</u>**
4. **Red points are almost located in the range of 2%-4%, which is similar to the modeling result**

Emotion classification – Consider noise in labels

Multiple Classes LN-Robust SVMs Framework:
Only apply NL-Robust SVMs on classifiers, **whose corresponding data separability is higher than 5 %**

	Standard SVMs	LN-Robust SVMs	Total
number of classifiers	9	16	25

NL-dataset	u value	Accuracy
Apply to all classifiers	u=0 (Standard SVMs)	64.86%
	u=0.05	66.21%
	u=0.1	66.21%
	u=0.5	66.21%
Apply Framework	u=0.05	67.60%
	u=0.1	67.56%
	u=0.5	67.56%

Random images - dataset	u value	Accuracy
Apply to all classifiers	u=0 (Standard SVMs)	81.08%
	u=0.05	83.78%
	u=0.1	83.78%
	u=0.5	82.43%
Apply Framework	u=0.05	83.78%
	u=0.1	85.13%
	u=0.5	83.78%

Conclusion:

Dataset

Noise in the label

Clean data

Training data

Testing data

With noise in the label

Binary classification NL-robust SVM:

1. More features → Better performance
2. More data → Better performance

?? How to extend the binary classification to multiple classes ??

?? Does the separability of data affect the performance??

Multi-classes classification NL-robust SVMs:

1. Preliminary data simulation
   - Separability of data may affect the performance of NL-robust SVM
2. Case study human emotion classification (5 classes)
   - Classification tree **81%** with Standard SVMs
   - Classification tree **83%** with NL-robust SVMs at all binary classification
   - Classification tree **85%** with NL-robust SVMs at some binary classification (based on separability criteria)

Question

Thank You !

Dr. Ehsan Tarkesh Esfahani

Dr Rahul Rai

Dr. Amin Karami

Lab member

And all the audience !