Supervised Evaluation of Representations

Charles Zheng

Stanford University

April 22, 2017

What is a representation?

- It is often hypothesized that *complex objects* (images, sounds, etc.) can be mapped into a *low-dimensional* space
- A *representation* is a nonlinear, dimensionality-reducing mapping

Why do we care?

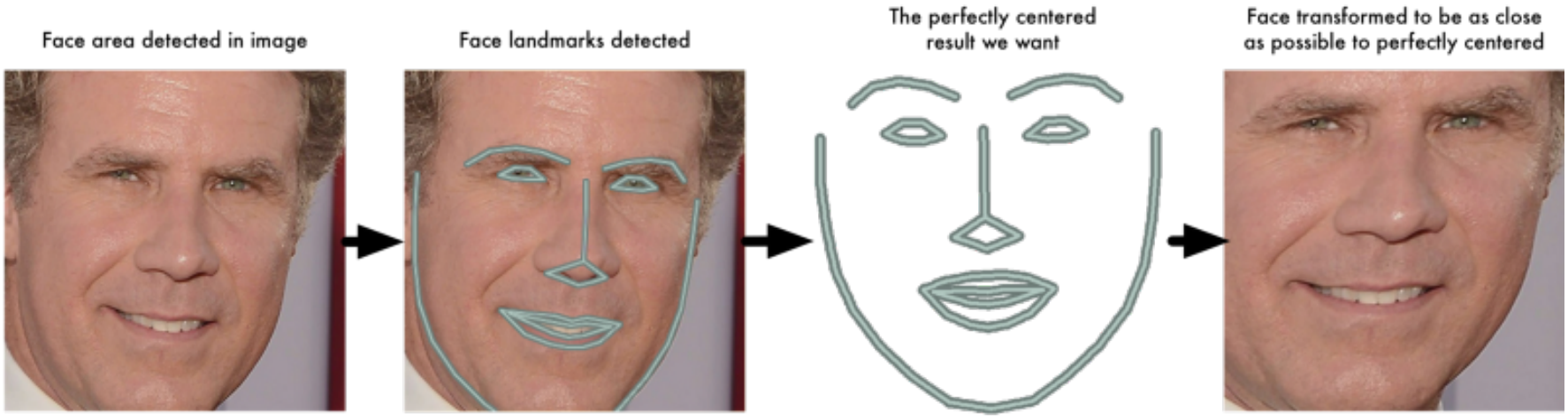
- Application 1: Neuroscience. Different regions of the brain have different representations of the same sensory stimuli.
- Application 2: Machine learning. A good representation leads to more accurate prediction.

Example: Facial recognition

- Used to tag images in software, security

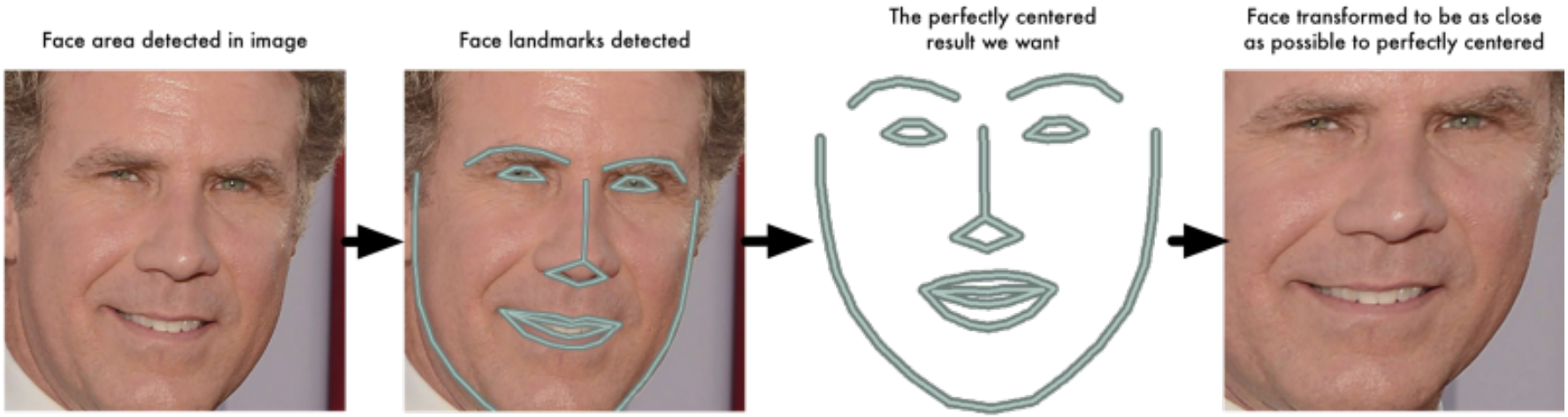
Example: Facial recognition

- Used to tag images in software, security
- Preprocessing

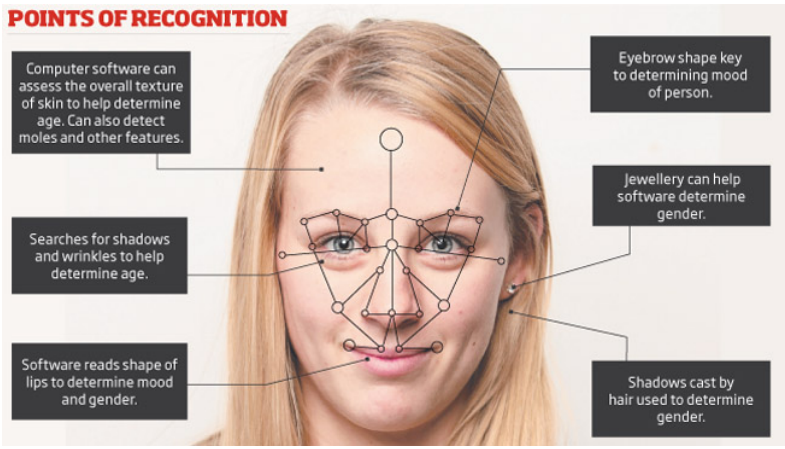


Example: Facial recognition

- Used to tag images in software, security
- Preprocessing



- Feature extraction

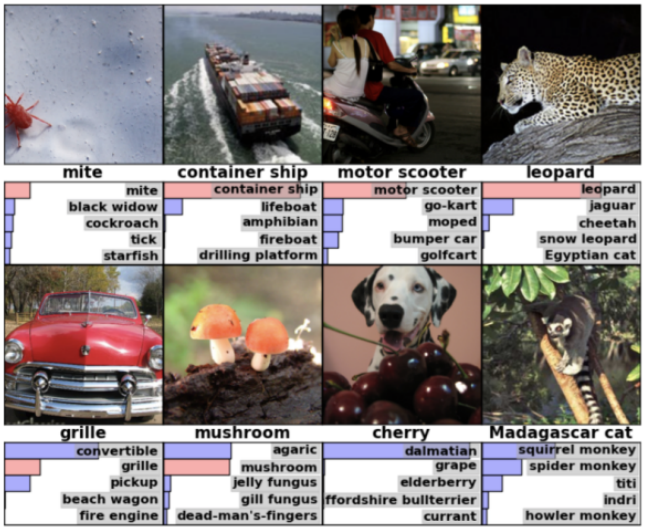


What defines a good representation?

Section 2

Randomized classification and extrapolation

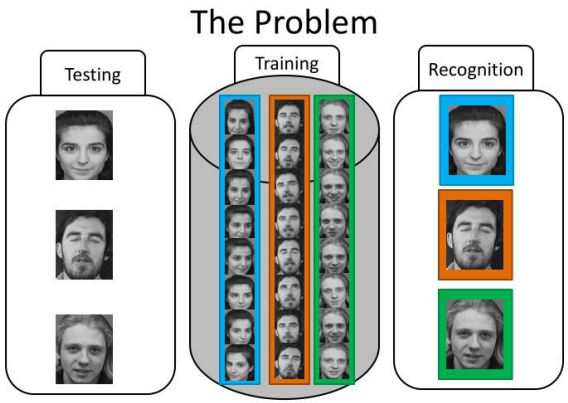
Multi-class classification



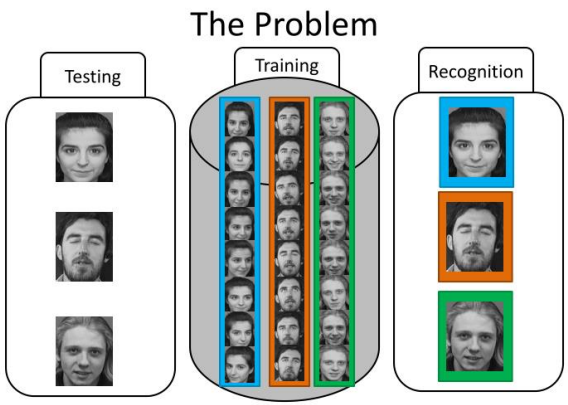
- MNIST digit recognition: 10 categories
- Human motion database: 51 categories
- ImageNet: 22,000 categories
- Wikipedia: 325,000 categories

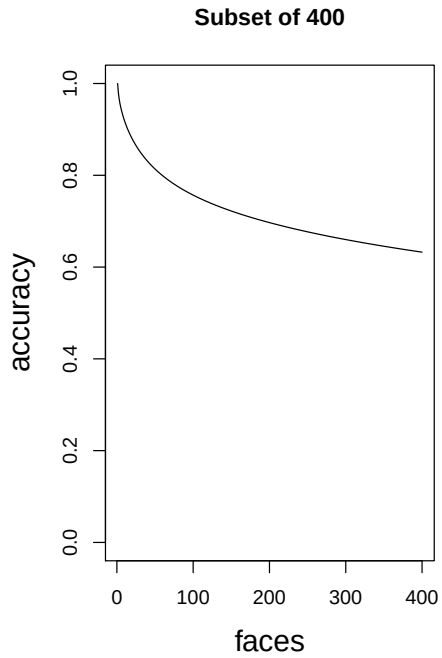
from Krizhevsky et al. 2012

Accuracy vs. number of classes

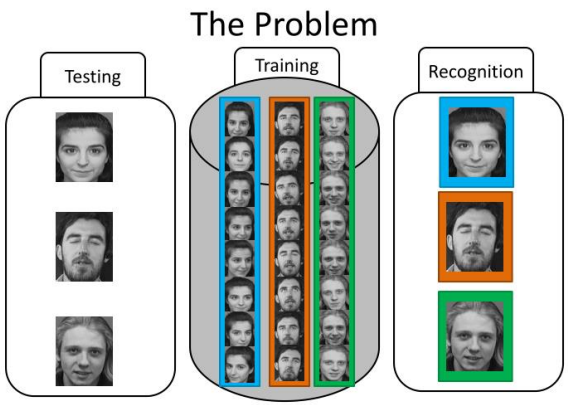


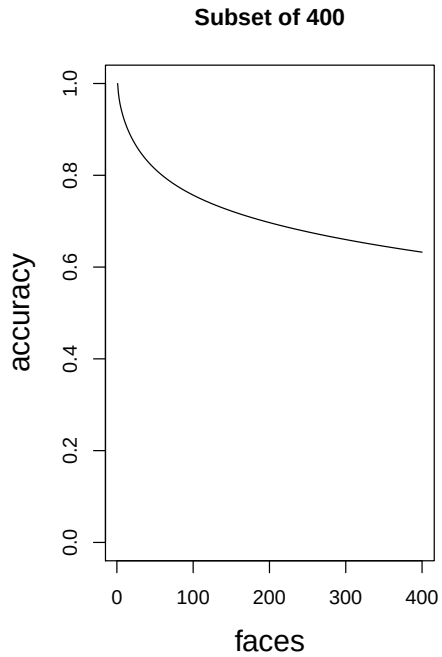
Accuracy vs. number of classes





Accuracy vs. number of classes

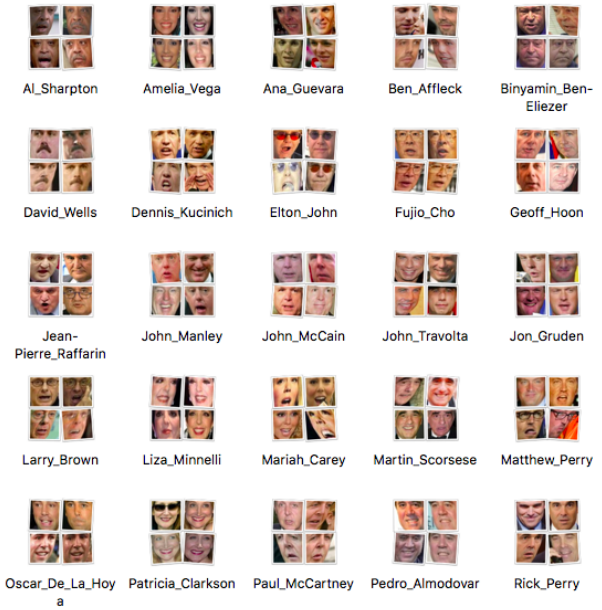




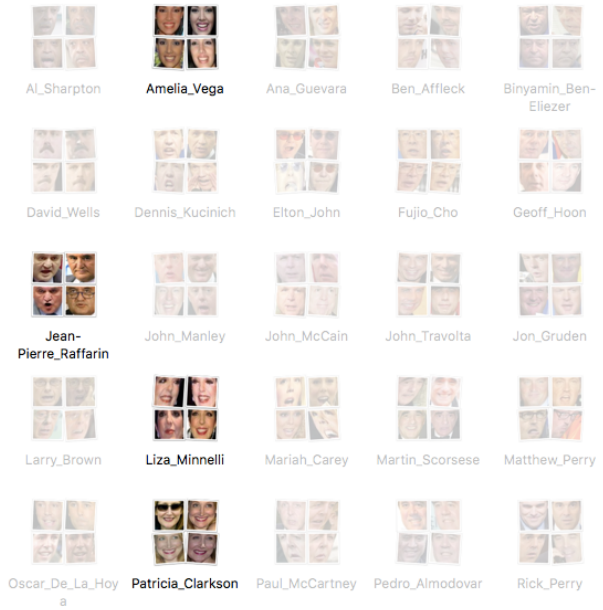
How does the accuracy scale with the number of classes (faces)?

Setup

1. Population of categories $\pi(y)$

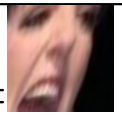
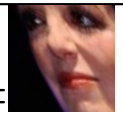
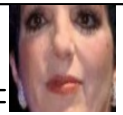


2. Subsample k labels, $y_1, \ldots, y_k$



Setup

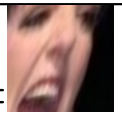
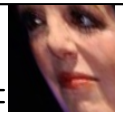
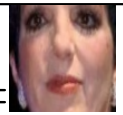
3. Collect training and test data $x_j^{(i)}$ (faces) for labels (people) $\{y_1, \ldots, y_k\}$.

Label	Training			Test
y_1=Amelia	$x_1^{(1)} =$ 	$x_1^{(2)} =$ 	$x_1^{(3)} =$ 	$x_1^* =$ 
y_2=Jean-Pierre	$x_2^{(1)} =$ 	$x_2^{(2)} =$ 	$x_2^{(3)} =$ 	$x_2^* =$ 
y_3=Liza	$x_3^{(1)} =$ 	$x_3^{(2)} =$ 	$x_3^{(3)} =$ 	$x_3^* =$ 
y_4=Patricia	$x_4^{(1)} =$ 	$x_4^{(2)} =$ 	$x_4^{(3)} =$ 	$x_4^* =$ 

4. Train a classifier and compute test error.

Setup

3. Collect training and test data $x_j^{(i)}$ (faces) for labels (people) $\{y_1, \ldots, y_k\}$.

Label	Training			Test
y_1=Amelia	$x_1^{(1)} =$ 	$x_1^{(2)} =$ 	$x_1^{(3)} =$ 	$x_1^* =$ 
y_2=Jean-Pierre	$x_2^{(1)} =$ 	$x_2^{(2)} =$ 	$x_2^{(3)} =$ 	$x_2^* =$ 
y_3=Liza	$x_3^{(1)} =$ 	$x_3^{(2)} =$ 	$x_3^{(3)} =$ 	$x_3^* =$ 
y_4=Patricia	$x_4^{(1)} =$ 	$x_4^{(2)} =$ 	$x_4^{(3)} =$ 	$x_4^* =$ 

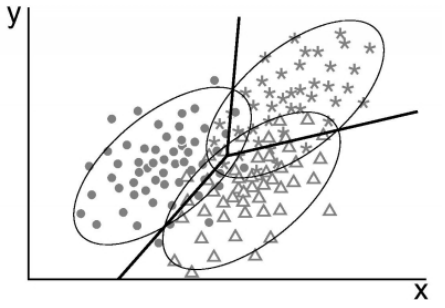
4. Train a classifier and compute test error.

Can we analyze how error depends on k?

Key assumption: marginal classifier

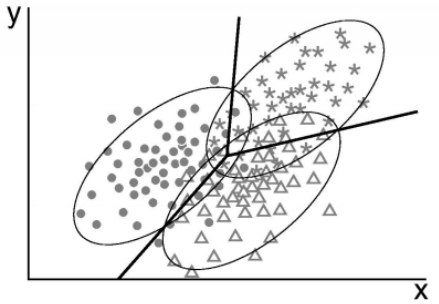
- The classifier is *marginal* if it learns a model *independently* for each class.

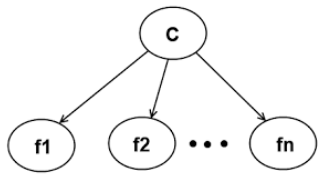
Key assumption: marginal classifier

- The classifier is *marginal* if it learns a model *independently* for each class.
- Examples: LDA/QDA 

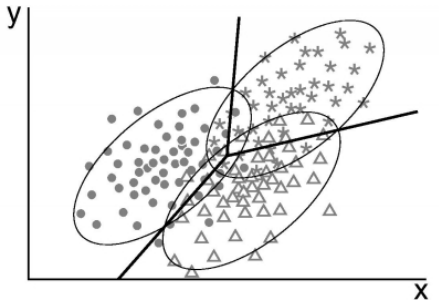
,

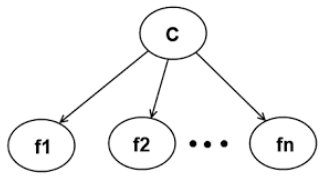
Key assumption: marginal classifier

- The classifier is *marginal* if it learns a model *independently* for each class.
- Examples: LDA/QDA 

, naïve Bayes 

Key assumption: marginal classifier

- The classifier is *marginal* if it learns a model *independently* for each class.
- Examples: LDA/QDA 

, naïve Bayes 

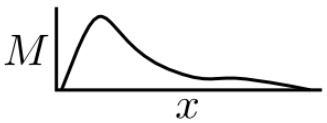
- Non-marginal classifiers: Multinomial logistic, multilayer neural networks, k-nearest neighbors

Definitions

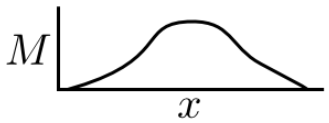
$\hat{F}_{y^{(i)}}$ is the empirical distribution obtained from the training data for label $y^{(i)}$.

Classification Rule

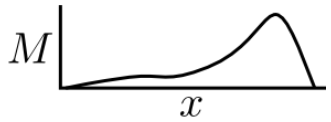
$$M_{y^{(1)}}(x) = \mathcal{M}(\hat{F}_{y^{(1)}})(x)$$



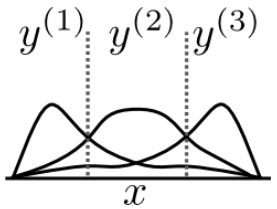
$$M_{y^{(2)}}(x) = \mathcal{M}(\hat{F}_{y^{(2)}})(x)$$



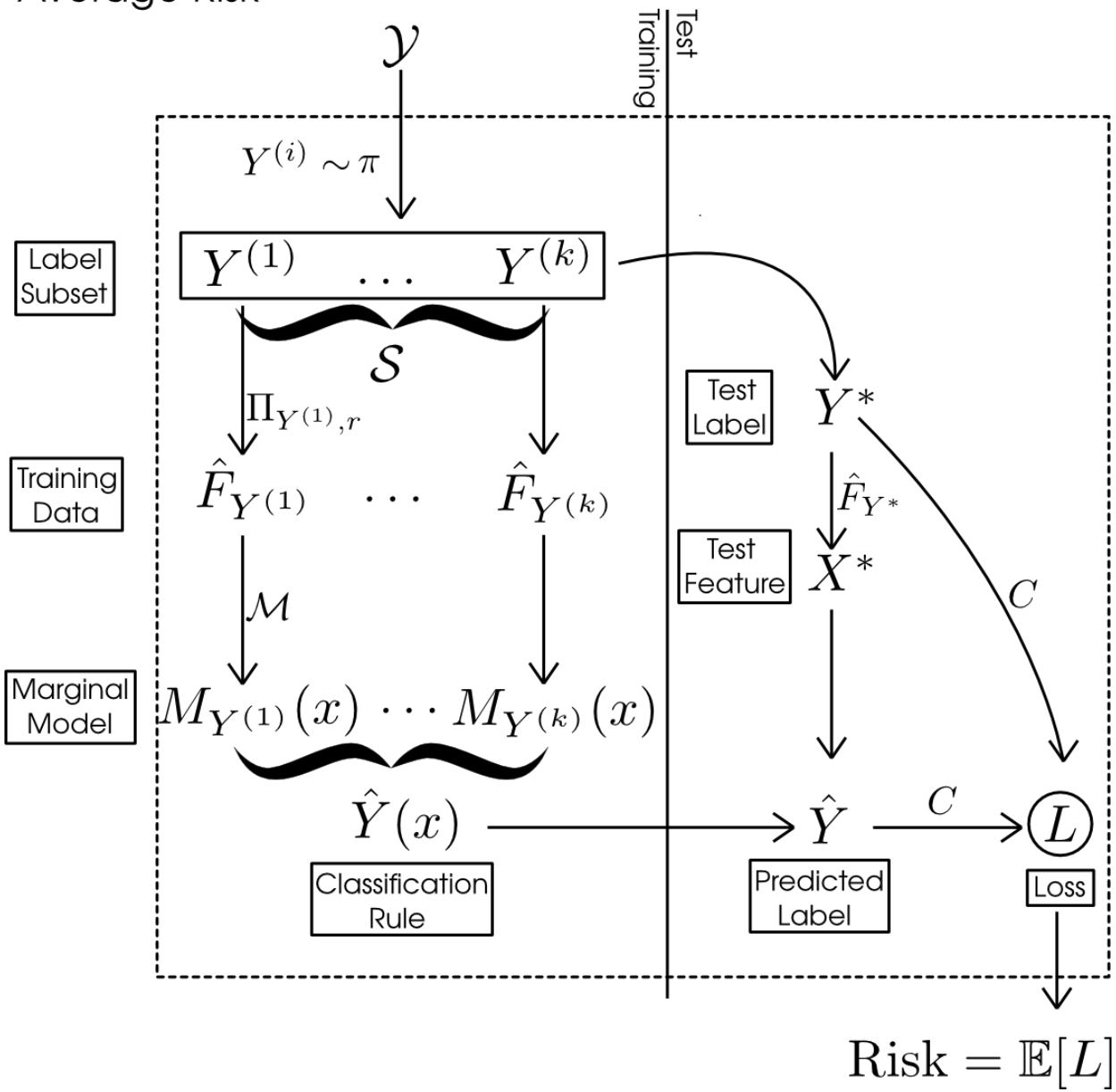
$$M_{y^{(3)}}(x) = \mathcal{M}(\hat{F}_{y^{(3)}})(x)$$



$$\hat{Y}(x) = \text{argmax}_{y \in \mathcal{S}} M_y(x)$$



Average Risk



Theoretical Result

Theorem. (Z., Achanta, Benjamini.) Suppose π, $\{F_y\}_{y \in \mathcal{Y}}$ and marginal classifier $\mathcal{F}$ satisfy *(some regularity condition)*. Then, there exists some function $\bar{D}(u)$ on $[0,1] \to [0,1]$ such that the k-class average risk is given by

$$\text{AvRisk}_k = (k-1) \int \bar{D}(u) u^{k-2} du.$$

Theoretical Result

Theorem. (Z., Achanta, Benjamini.) Suppose π, $\{F_y\}_{y\in\mathcal{Y}}$ and marginal classifier $\mathcal{F}$ satisfy *(some regularity condition)*. Then, there exists some function $\bar{D}(u)$ on $[0,1] \to [0,1]$ such that the k-class average risk is given by

$$\text{AvRisk}_k = (k-1)\int \bar{D}(u)u^{k-2}du.$$

What is this $\bar{D}(u)$ function? We will explain in the following toy example...

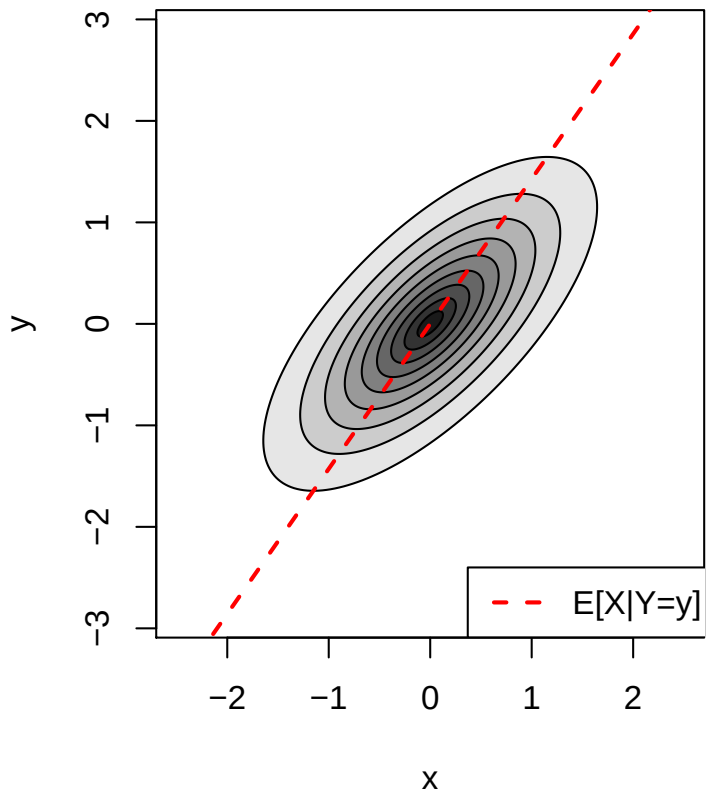
Toy example

$Y_1, \ldots, Y_k \overset{iid}{\sim} N(0, 1)$;

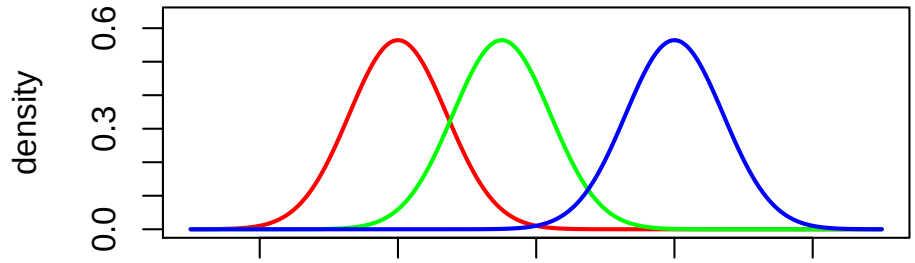
Toy example

$Y_1, \ldots, Y_k \overset{iid}{\sim} N(0,1)$;

$X|Y \sim N(\rho Y, 1-\rho^2)$ i.e. $(Y, X) \sim N(0, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix})$.



Toy example

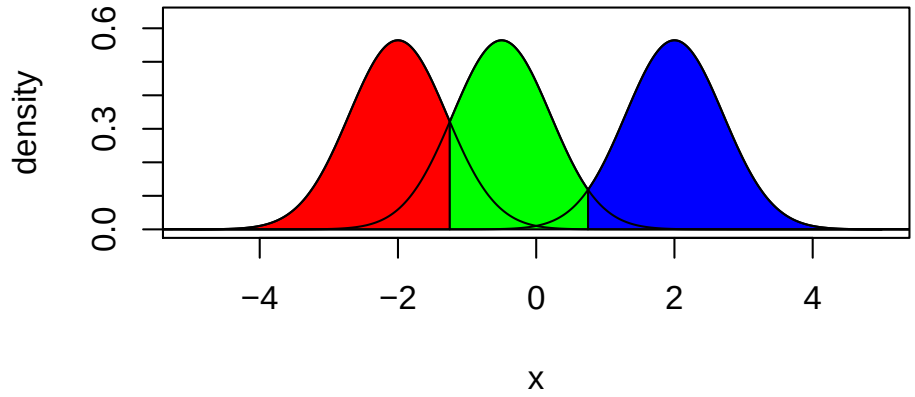


- Suppose $k = 3$, and we draw Y_1, Y_2, Y_3.
- The *Bayes rule* is the optimal classifier and depends on knowing the true densities:

$$\hat{y}(x) = \text{argmax}_{y_i} p(x|y_i)$$

- The *Bayes Risk*, which is the misclassification rate of the optimal classifier.

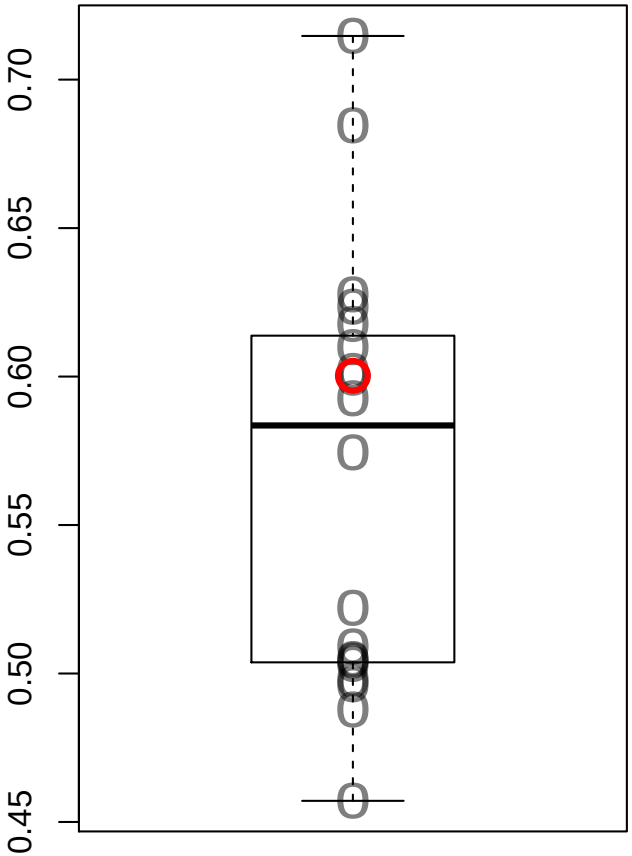
Toy example

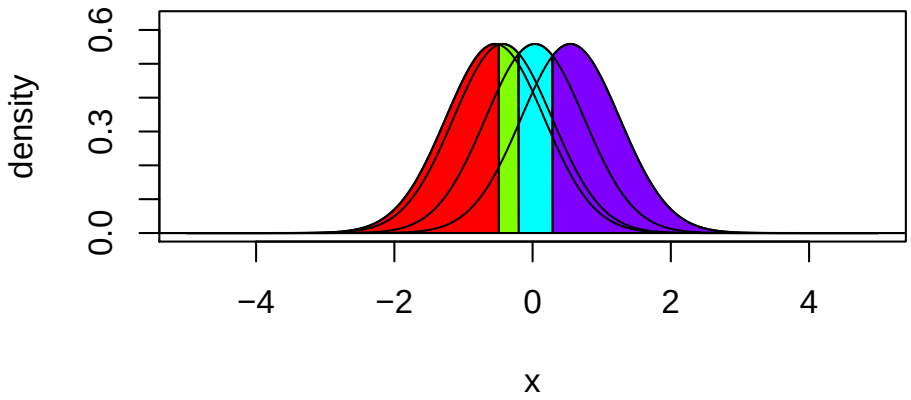


- The *Bayes Risk* is the expected test error of the Bayes rule,

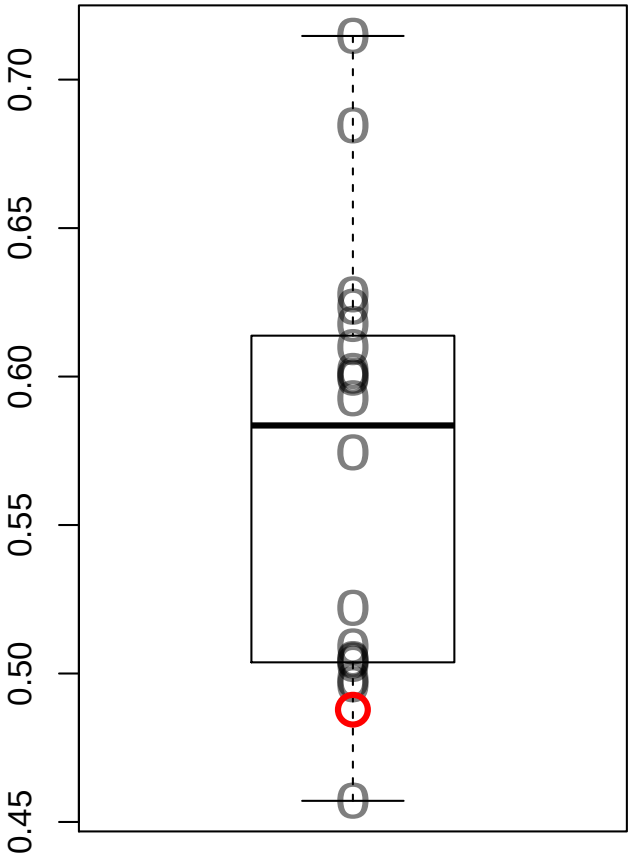
$$\frac{1}{k}\sum_{i=1}^{k}\Pr[\hat{y}(x) \neq Y | Y = y_i]$$

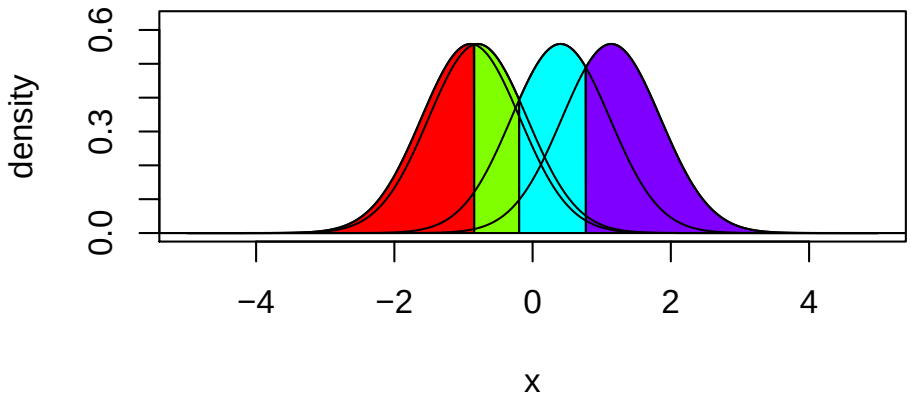
Toy example



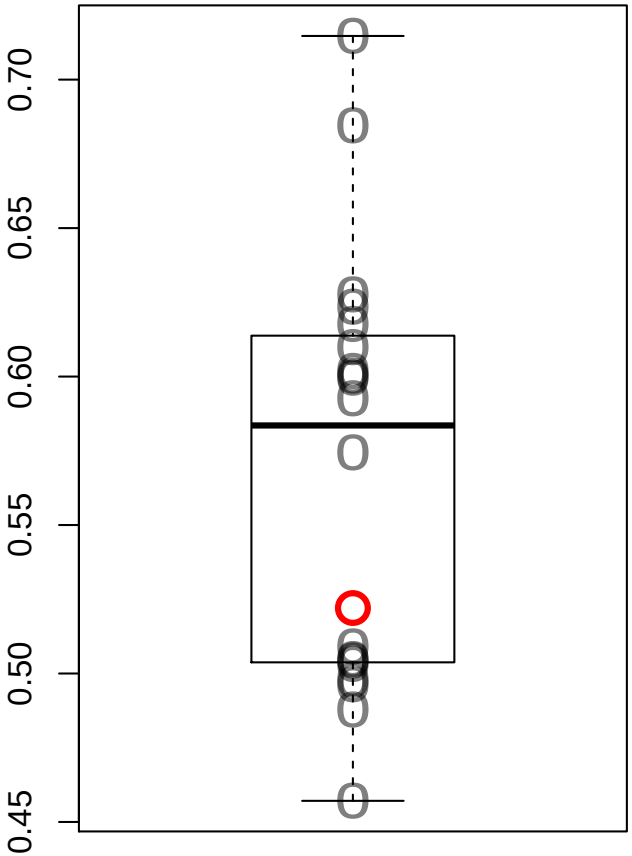


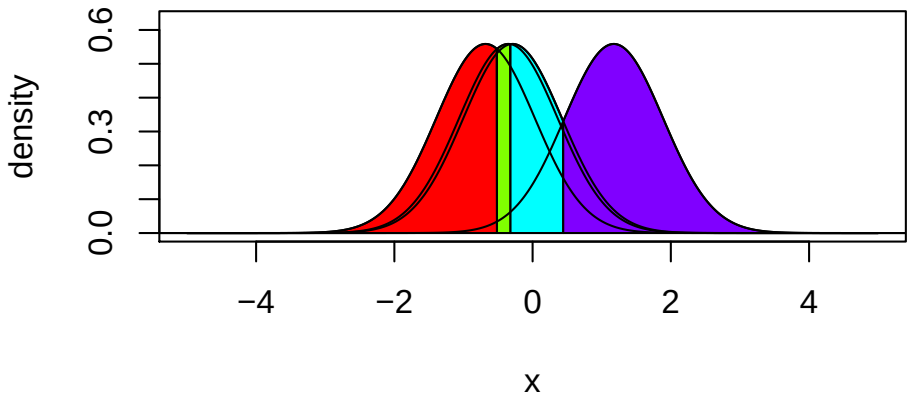
Toy example



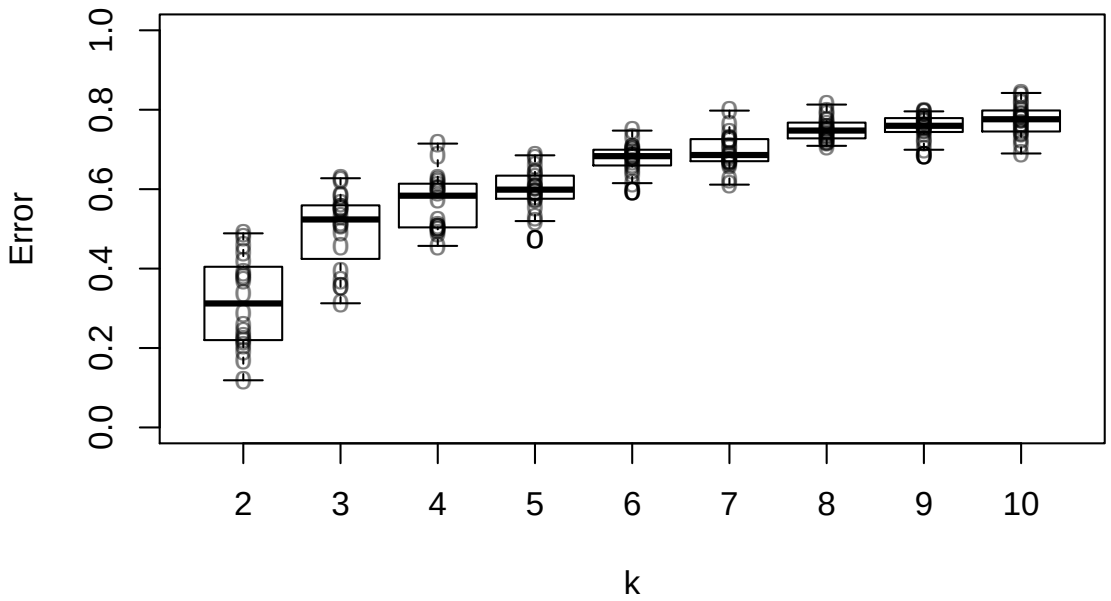


Toy example

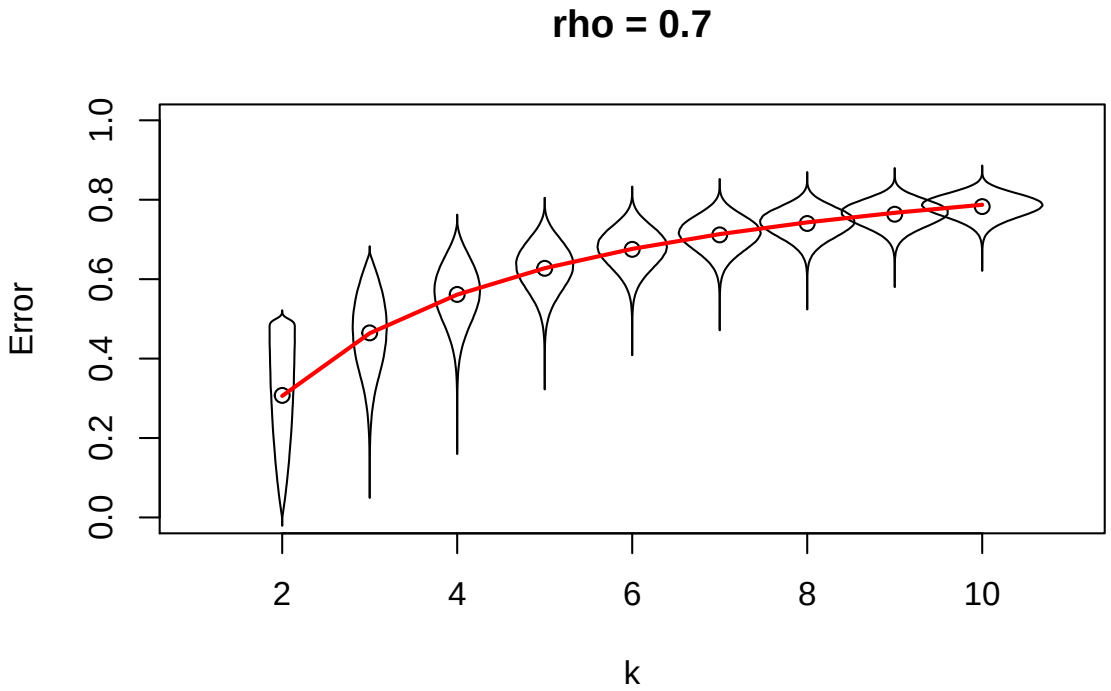




Toy example



Toy example

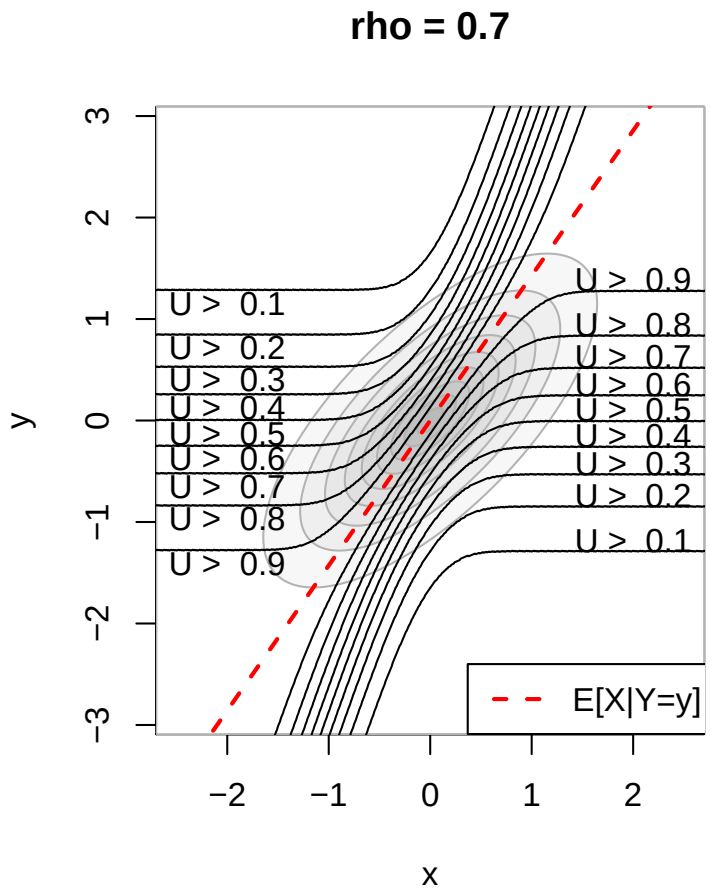


Defining the U-function

Define $U_x(y)$ as follows:

- Suppose we have test instance (face) x whose true label (person) is y.
- Let Y' be a random *incorrect* label (person).
- Use the classifier to guess whether x belongs to y or Y'.
- Define $U_x(y)$ as the probabilility of success (randomizing over training data).

Toy example



$$U_y(x) = \Pr[d(x, \rho Y') > d(x, \rho y)], \text{ for } Y' \sim N(0, 1).$$

Defining $\bar{D}(u)$

- Define random variable as $U_Y(X)$ for (Y, X) drawn from the joint distribution.

Defining $\bar{D}(u)$

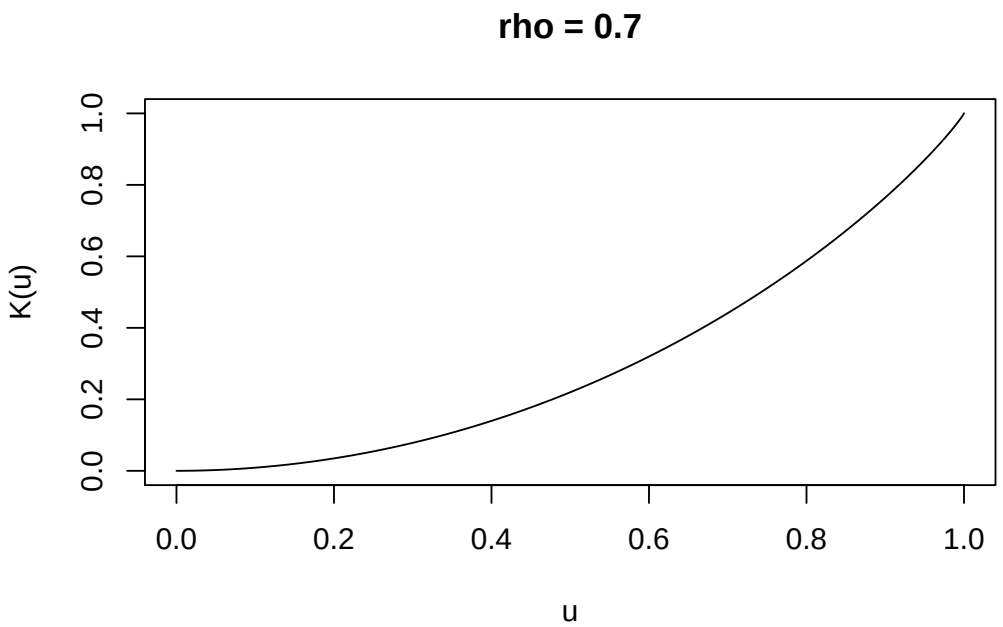
- Define random variable as $U_Y(X)$ for (Y, X) drawn from the joint distribution.
- $\bar{D}(u)$ is the cumulative distribution function of U,

$$\bar{D}(u) = \Pr[U_Y(X) \leq u].$$

Defining $\bar{D}(u)$

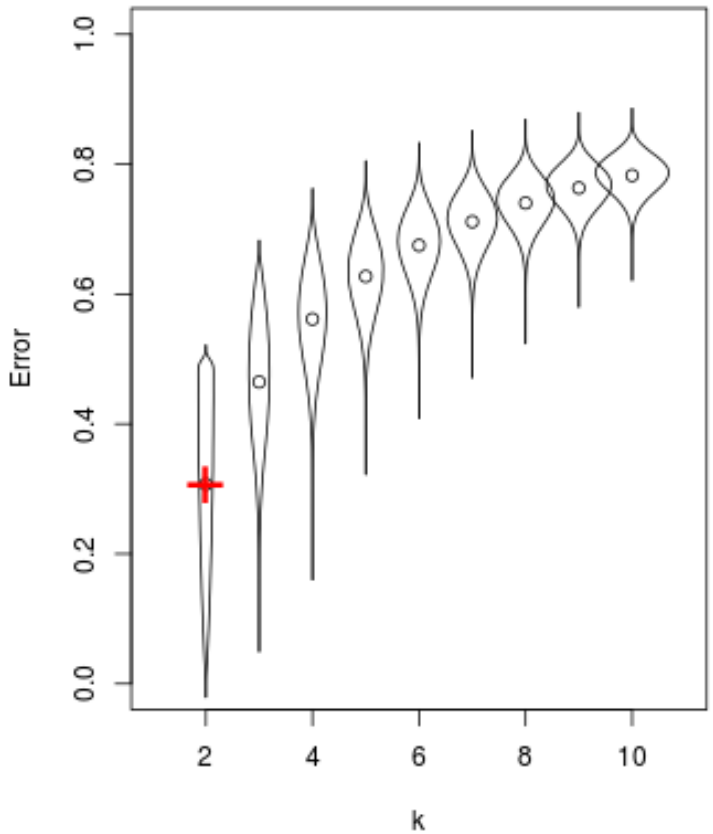
- Define random variable as $U_Y(X)$ for (Y, X) drawn from the joint distribution.
- $\bar{D}(u)$ is the cumulative distribution function of U,

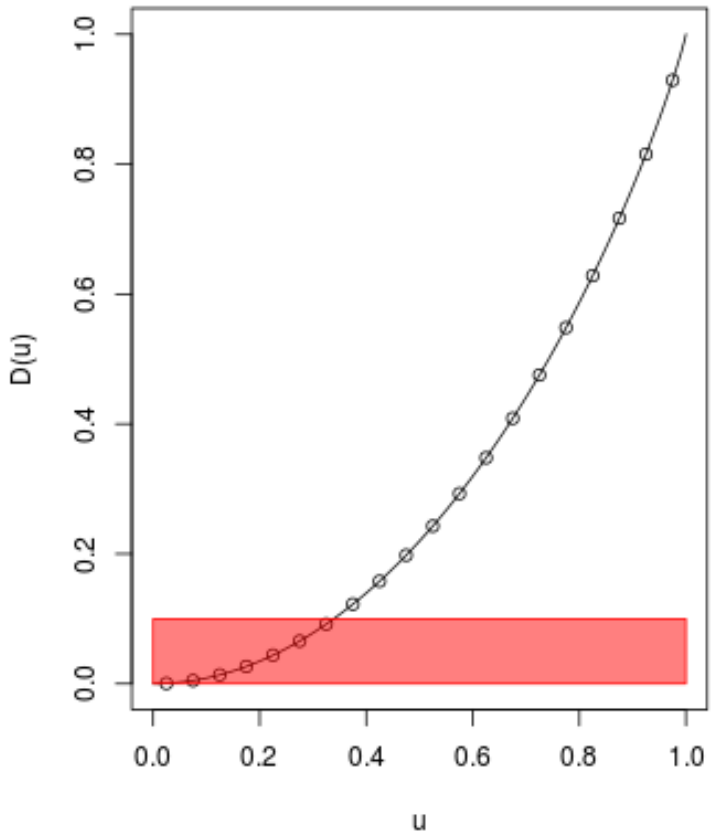
$$\bar{D}(u) = \Pr[U_Y(X) \leq u].$$



Computing average risk

$$\text{AvRisk}_k = (k-1)\int \bar{D}(u)u^{k-2}du.$$

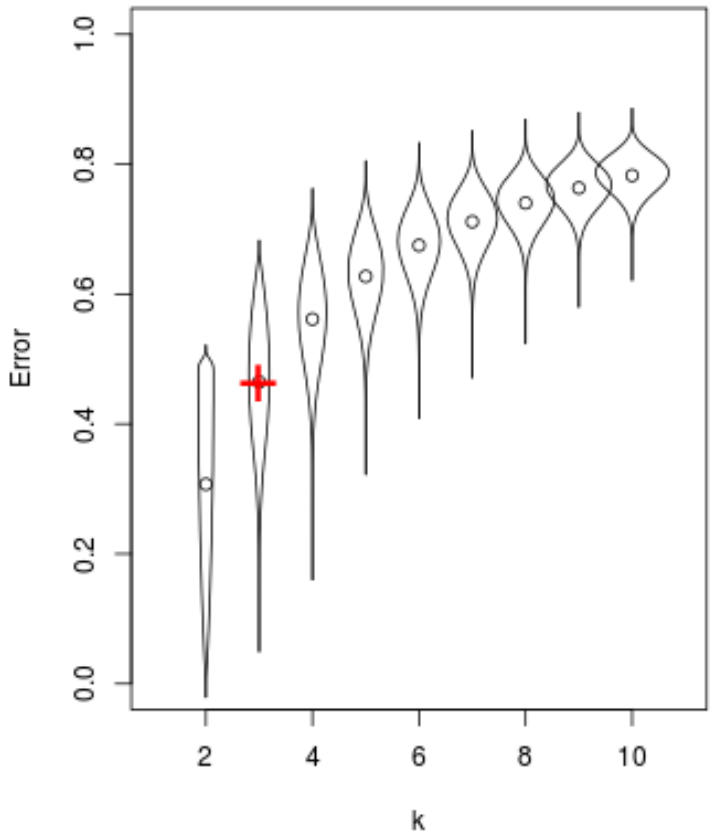


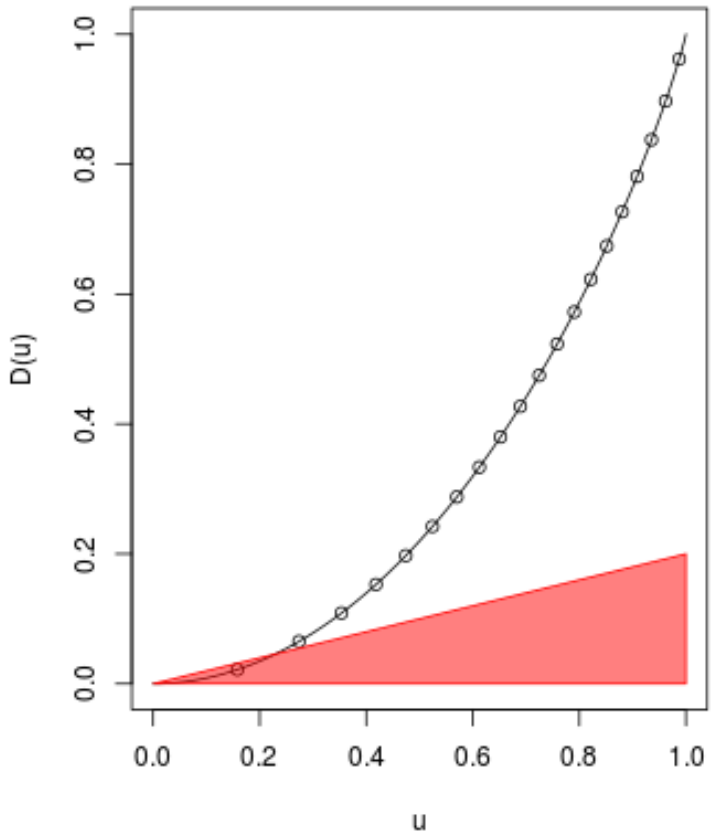


$(k = 2)$

Computing average risk

$$\text{AvRisk}_k = (k-1)\int \bar{D}(u)u^{k-2}du.$$

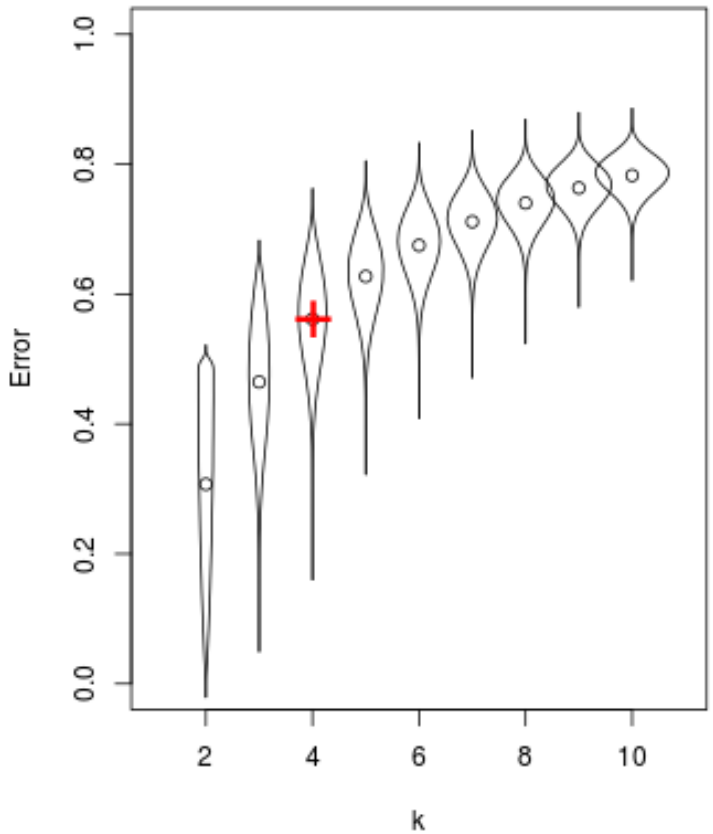


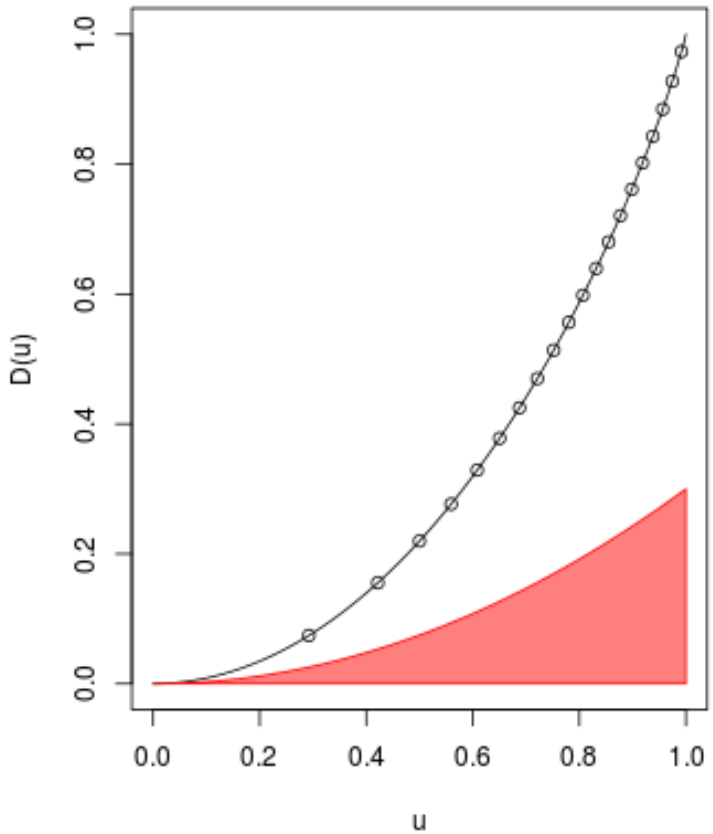


$(k = 3)$

Computing average risk

$$\text{AvRisk}_k = (k-1)\int \bar{D}(u)u^{k-2}du.$$

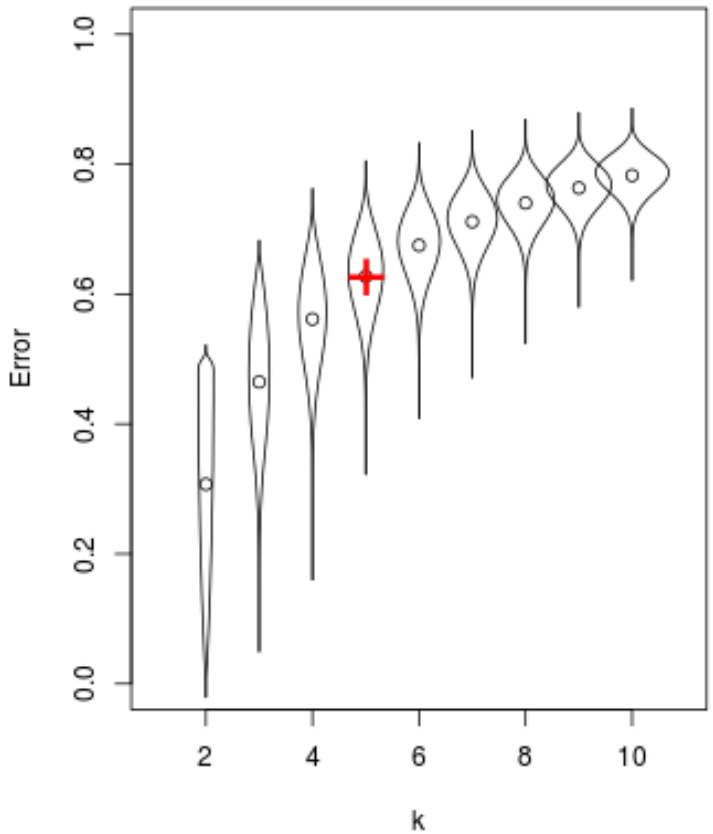


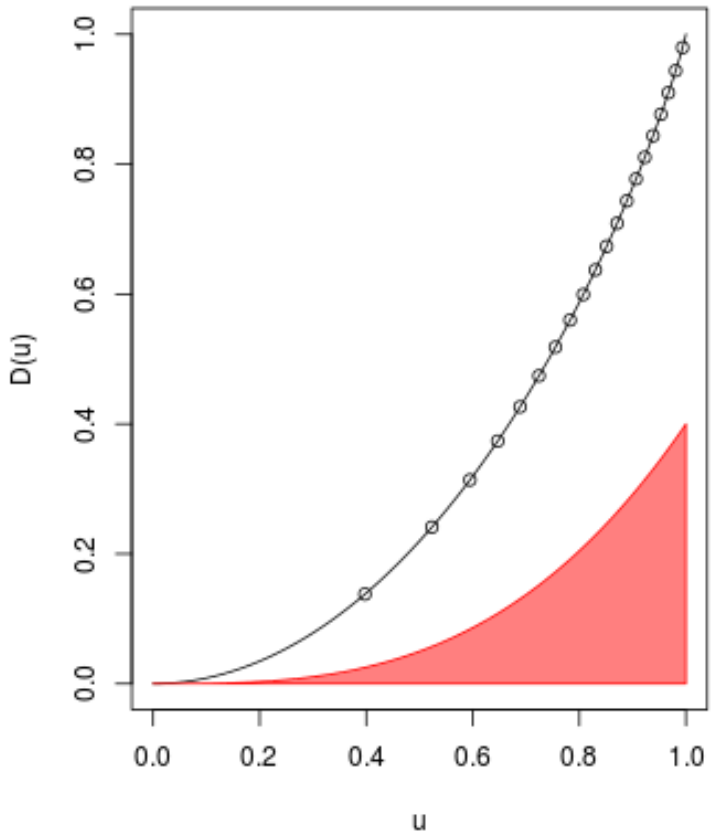


$(k = 4)$

Computing average risk

$$\text{AvRisk}_k = (k-1)\int \bar{D}(u)u^{k-2}du.$$

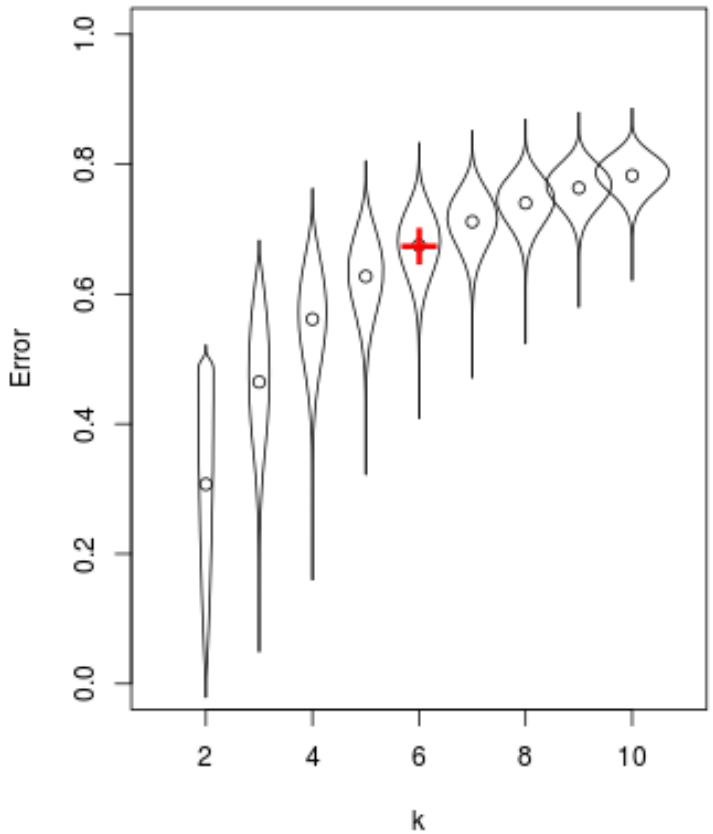


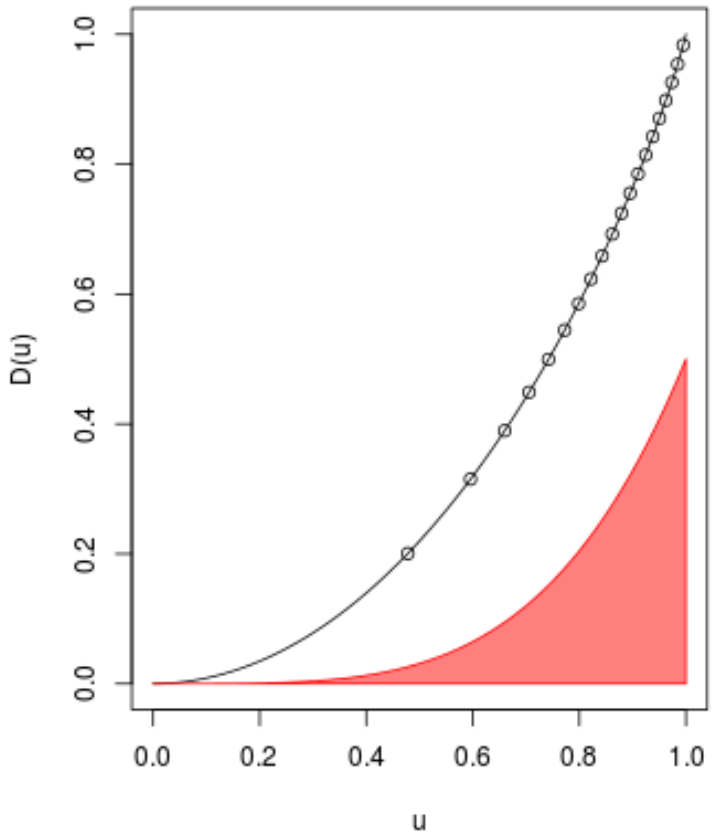


$(k = 5)$

Computing average risk

$$\text{AvRisk}_k = (k-1)\int \bar{D}(u)u^{k-2}du.$$

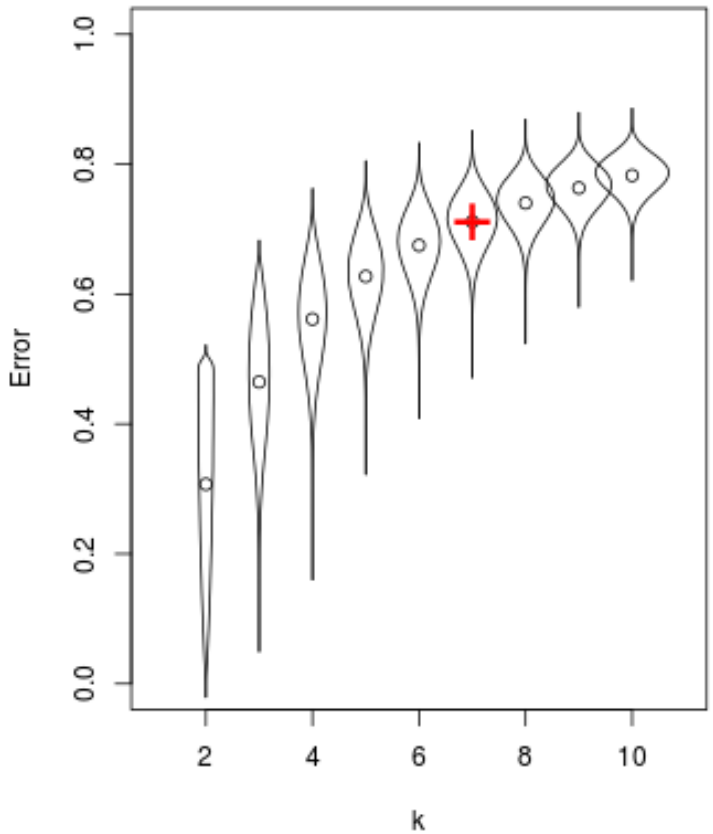


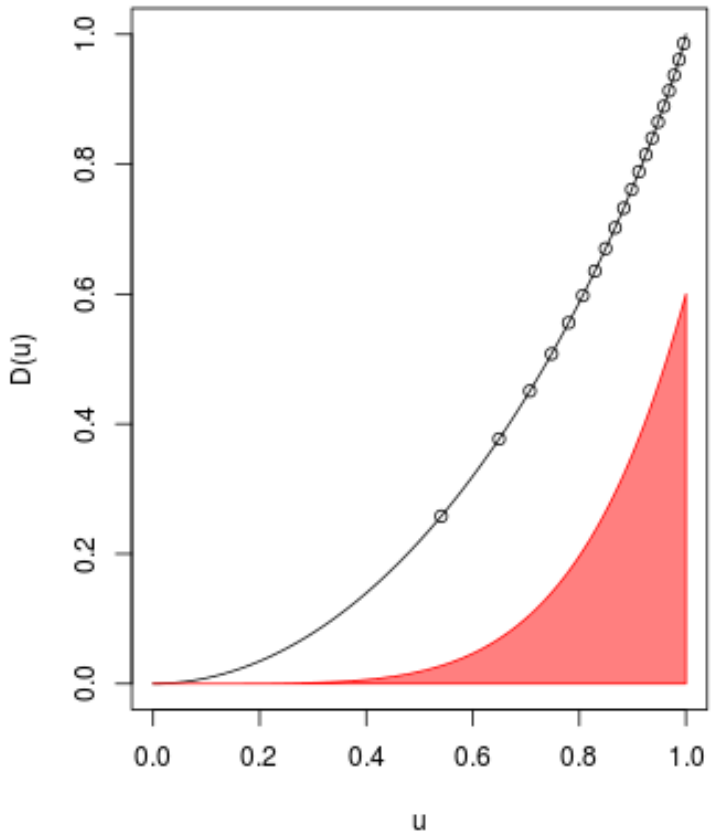


$(k = 6)$

Computing average risk

$$\text{AvRisk}_k = (k-1)\int \bar{D}(u)u^{k-2}du.$$

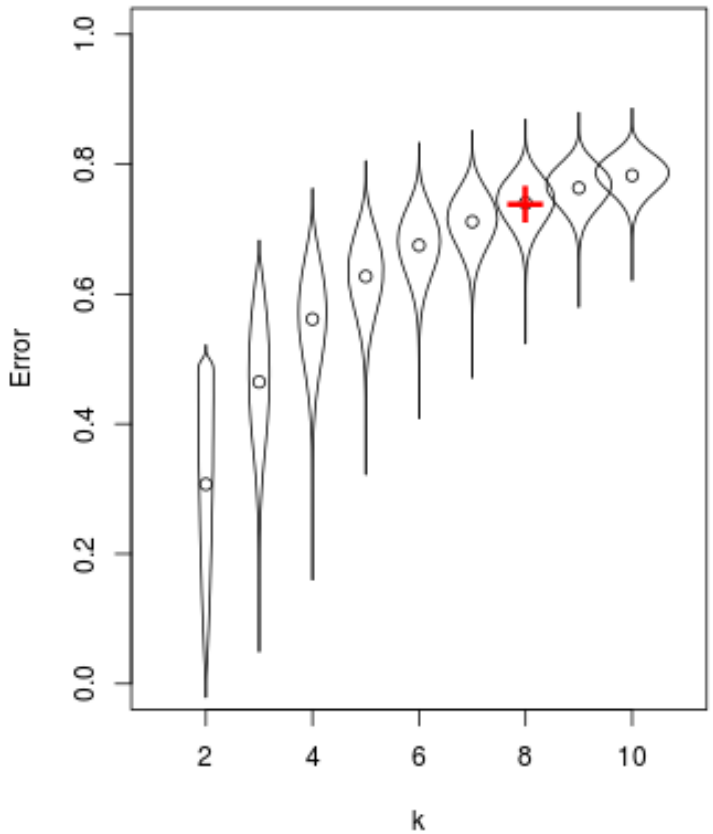


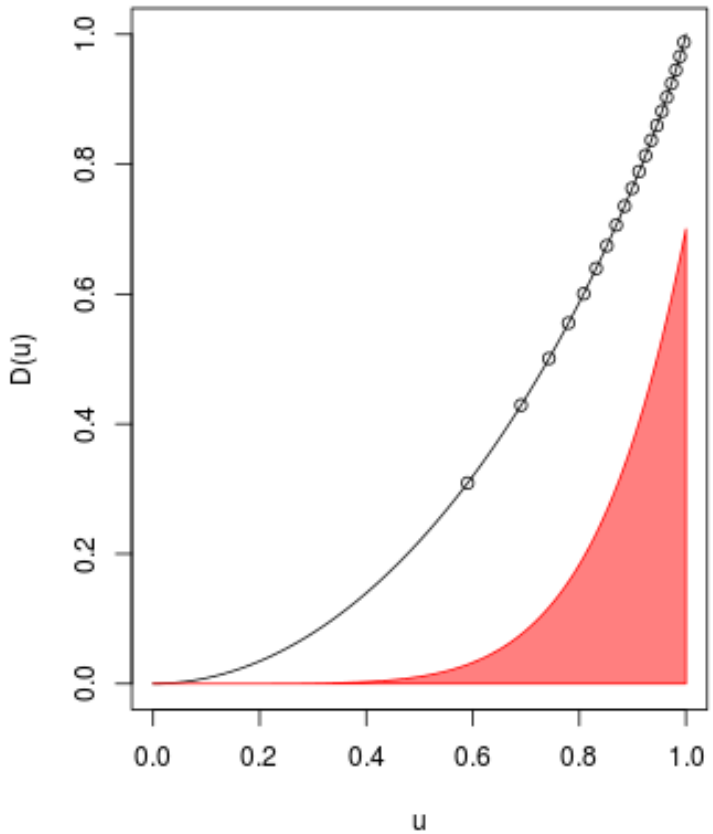


$(k = 7)$

Computing average risk

$$\text{AvRisk}_k = (k-1)\int \bar{D}(u)u^{k-2}du.$$

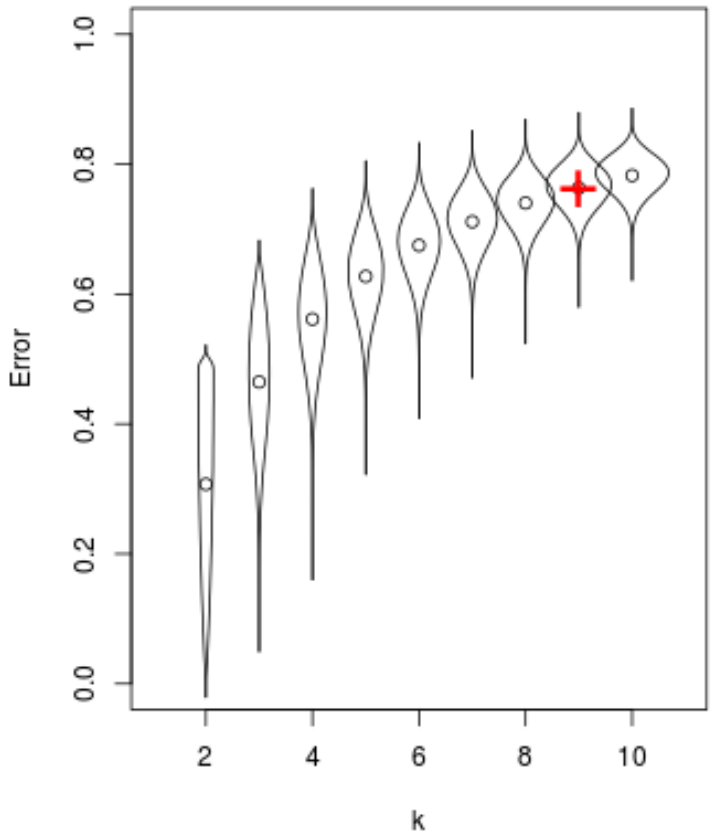


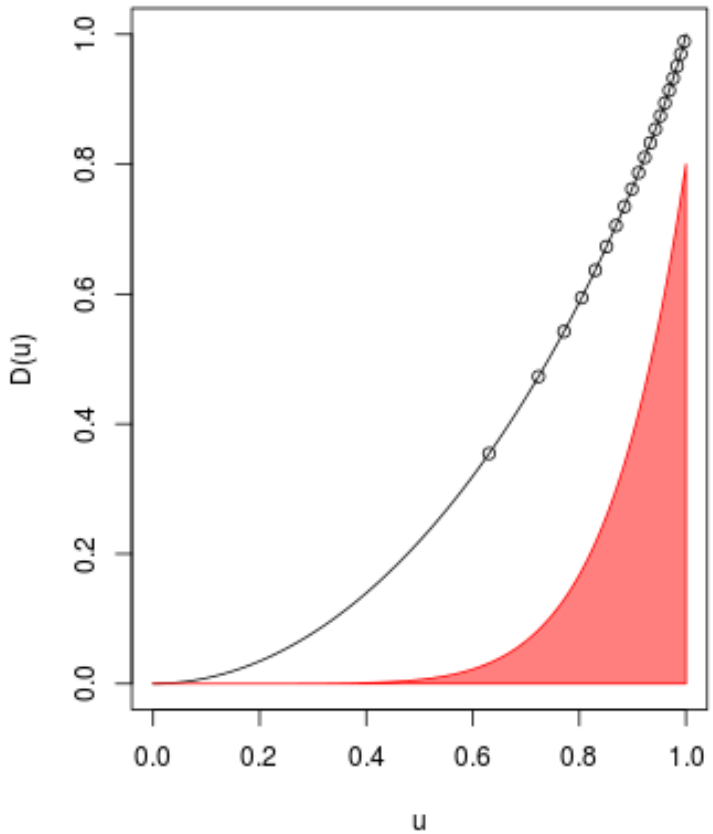


$(k = 8)$

Computing average risk

$$\text{AvRisk}_k = (k-1)\int \bar{D}(u)u^{k-2}du.$$

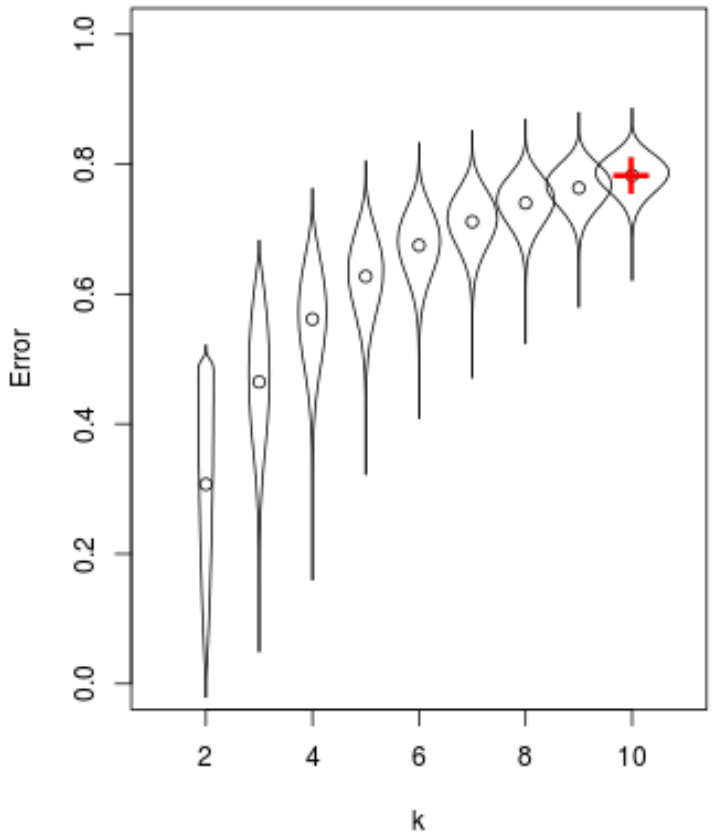


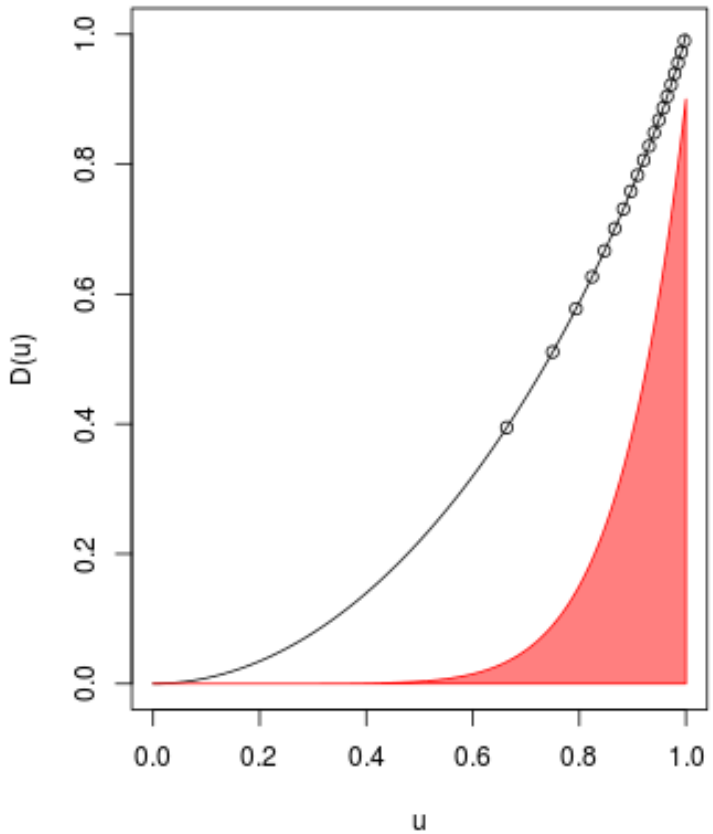


$(k = 9)$

Computing average risk

$$\text{AvRisk}_k = (k-1)\int \bar{D}(u)u^{k-2}du.$$





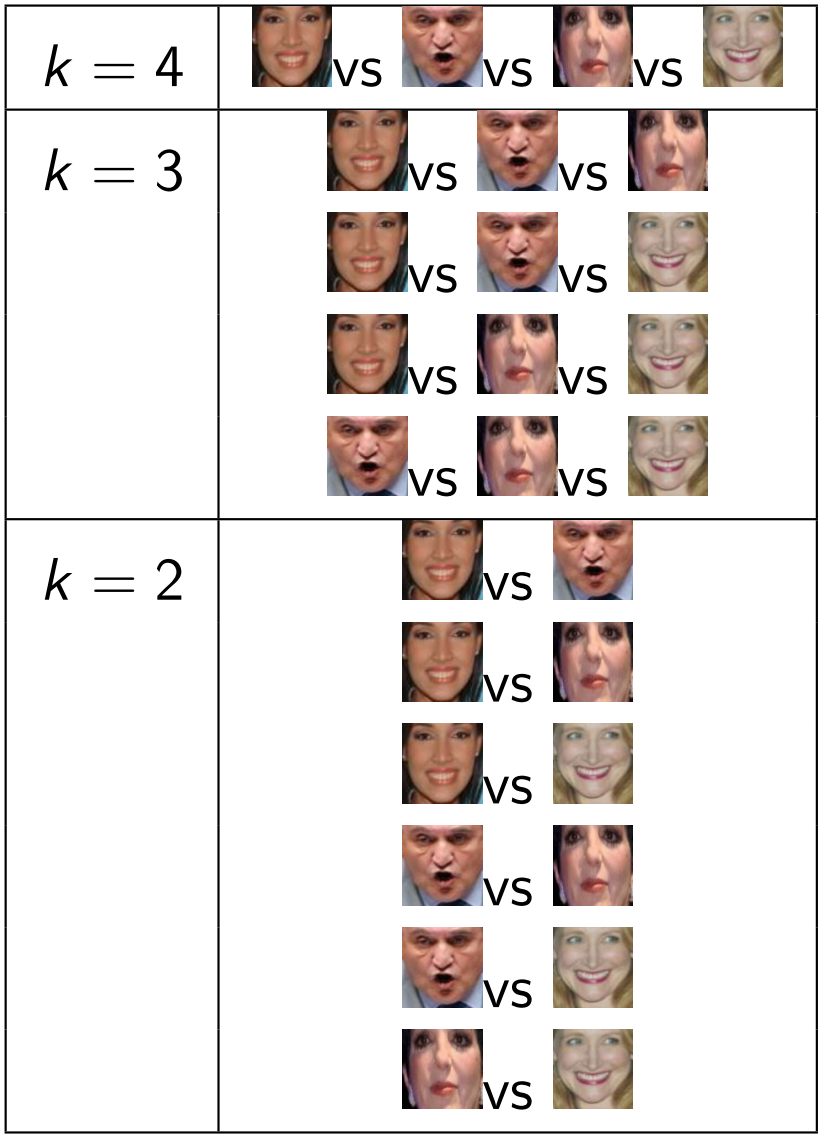
$(k = 10)$

Implication: estimate $\bar{D}(u)$ to predict risk

- Theoretical result links k-class average risk to $\bar{D}(u)$ function
- In real data, we do not know $\bar{D}(u)$ since it depends on the unknown joint distribution
- However, given a model, we can estimate $\bar{D}(u)$

Subsampled risk estimates

- Suppose we have data for k classes (subsampled from π)
- The test error TestErr_k is an unbiased estimate for AvRisk_k
- For any $\ell < k$, we can estimate AvRisk_ℓ by *subsampling* the k classes, and taking the average test risk, AvTestErr_ℓ.



Parametric modelling approach

Assume that for set of basis functions $h_1, \ldots, h_\ell$, we have

$$\bar{D}(u) = \sum_{\ell=1}^{m} \beta_\ell h_\ell(u).$$

Then

$$\text{AvRisk}_k = \sum_{\ell=1}^{m} \beta_\ell H_{\ell,k} = \beta^T \vec{H}_k$$

where

$$H_{\ell,k} = (k-2) \int_0^1 h_\ell(u) u^{k-2} du.$$

and $\vec{H}_k = (H_{1,k}, \ldots, H_{\ell,k})$

Prediction extrapolation as regression

1. Choose basis $h_1, \ldots, h_\ell$

$$\bar{D}(u) = \sum_{\ell=1}^{m} \beta_\ell h_\ell(u).$$

Prediction extrapolation as regression

1. Choose basis $h_1, \ldots, h_\ell$

$$\bar{D}(u) = \sum_{\ell=1}^{m} \beta_\ell h_\ell(u).$$

2. Obtain subsampled test errors $\text{AvTestErr}_2, \ldots, \text{AvTestErr}_k$.

Prediction extrapolation as regression

1. Choose basis $h_1, \ldots, h_\ell$

$$\bar{D}(u) = \sum_{\ell=1}^{m} \beta_\ell h_\ell(u).$$

2. Obtain subsampled test errors $\text{AvTestErr}_2, \ldots, \text{AvTestErr}_k$.
3. Fit regression model

$$\hat{\beta} = \text{argmin}_\beta \sum_{i=2}^{k} \left(\text{AvTestErr}_i - \vec{H}_i^T \beta_\ell \right)^2$$

Prediction extrapolation as regression

1. Choose basis $h_1, \ldots, h_\ell$

$$\bar{D}(u) = \sum_{\ell=1}^{m} \beta_\ell h_\ell(u).$$

2. Obtain subsampled test errors $\text{AvTestErr}_2, \ldots, \text{AvTestErr}_k$.
3. Fit regression model

$$\hat{\beta} = \text{argmin}_\beta \sum_{i=2}^{k} \left(\text{AvTestErr}_i - \vec{H}_i^T \beta_\ell \right)^2$$

4. For $K > k$, predict AvRisk_K as

$$\widehat{\text{AvRisk}}_K = \vec{H}_K^T \beta_\ell.$$

Examples of basis functions

- Polynomials, $1, x, x^2, \ldots$
- Cubic splines
- Linear splines, $[x - t_\ell]_+$

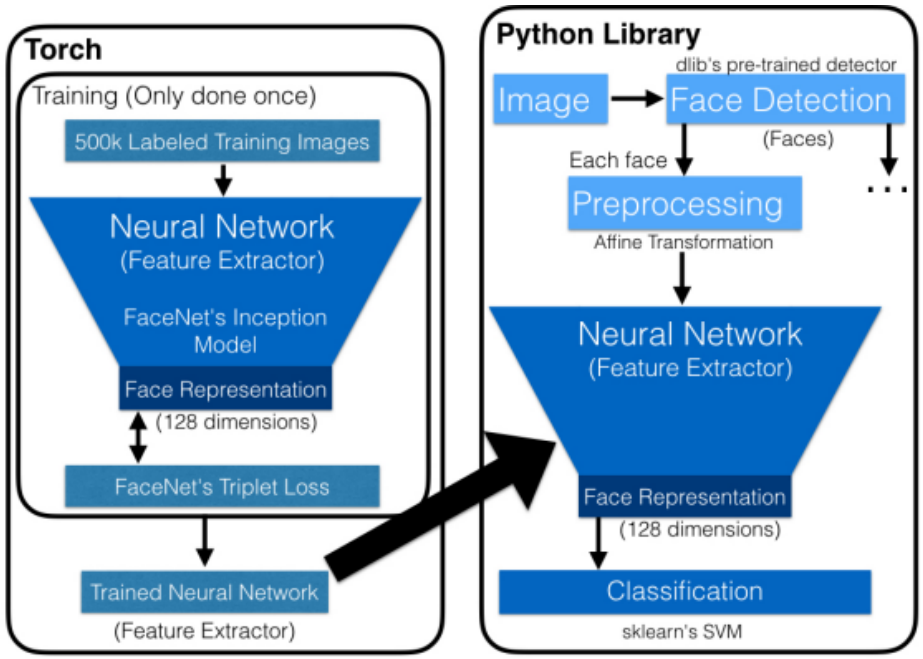
Optional constraint: assume β is *non-negative*. In the case of linear splines, this results in a *convex* fit.

Section 3

Applications

Facial recognition example

- Data: faces from "Labeled Faces in the Wild."
- 1672 people with at least 2 photos
- Featurization: trained neural network from OpenFace

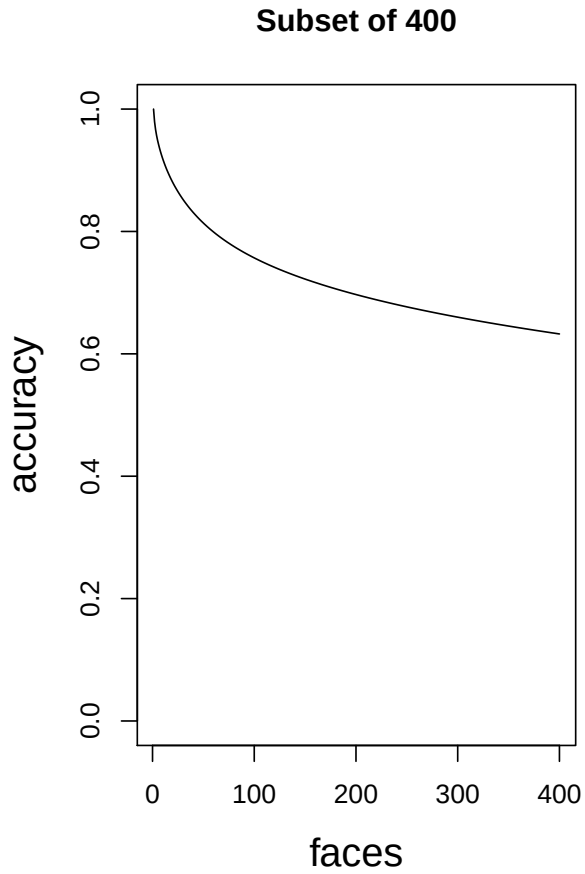


Facial recognition example

- Let us first subsample 400 faces (out of 1672)
- Randomly choose 1 face as training and 1 as test for each person
- Use 1-nearest neighbor.
  - NOTE: 1-NN with 1 example/class is equivalent to LDA with $\Sigma = I$: this fits marginal classifier assumption!

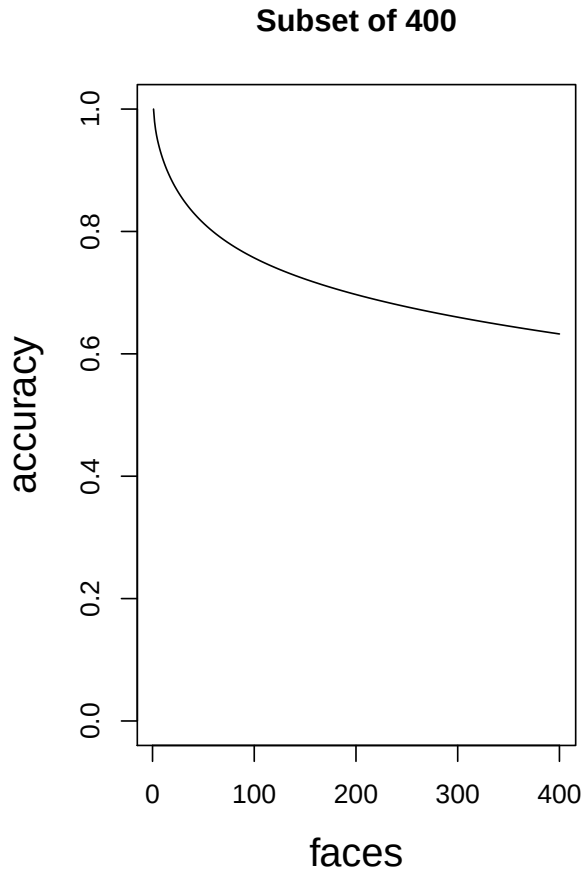
Facial recognition example

- Let us first subsample 400 faces (out of 1672)
- Randomly choose 1 face as training and 1 as test for each person
- Use 1-nearest neighbor.
  - NOTE: 1-NN with 1 example/class is equivalent to LDA with $\Sigma = I$: this fits marginal classifier assumption!



Facial recognition example

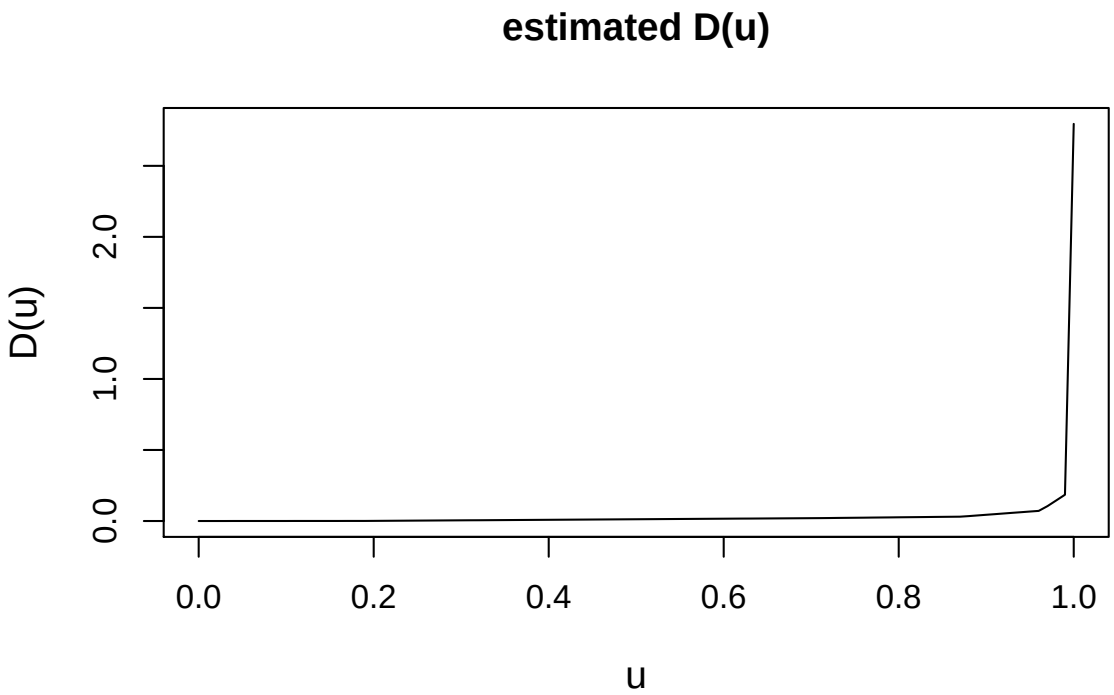
- Let us first subsample 400 faces (out of 1672)
- Randomly choose 1 face as training and 1 as test for each person
- Use 1-nearest neighbor.
  - NOTE: 1-NN with 1 example/class is equivalent to LDA with $\Sigma = I$: this fits marginal classifier assumption!



Can we predict the accuracy on the full set of 1672?

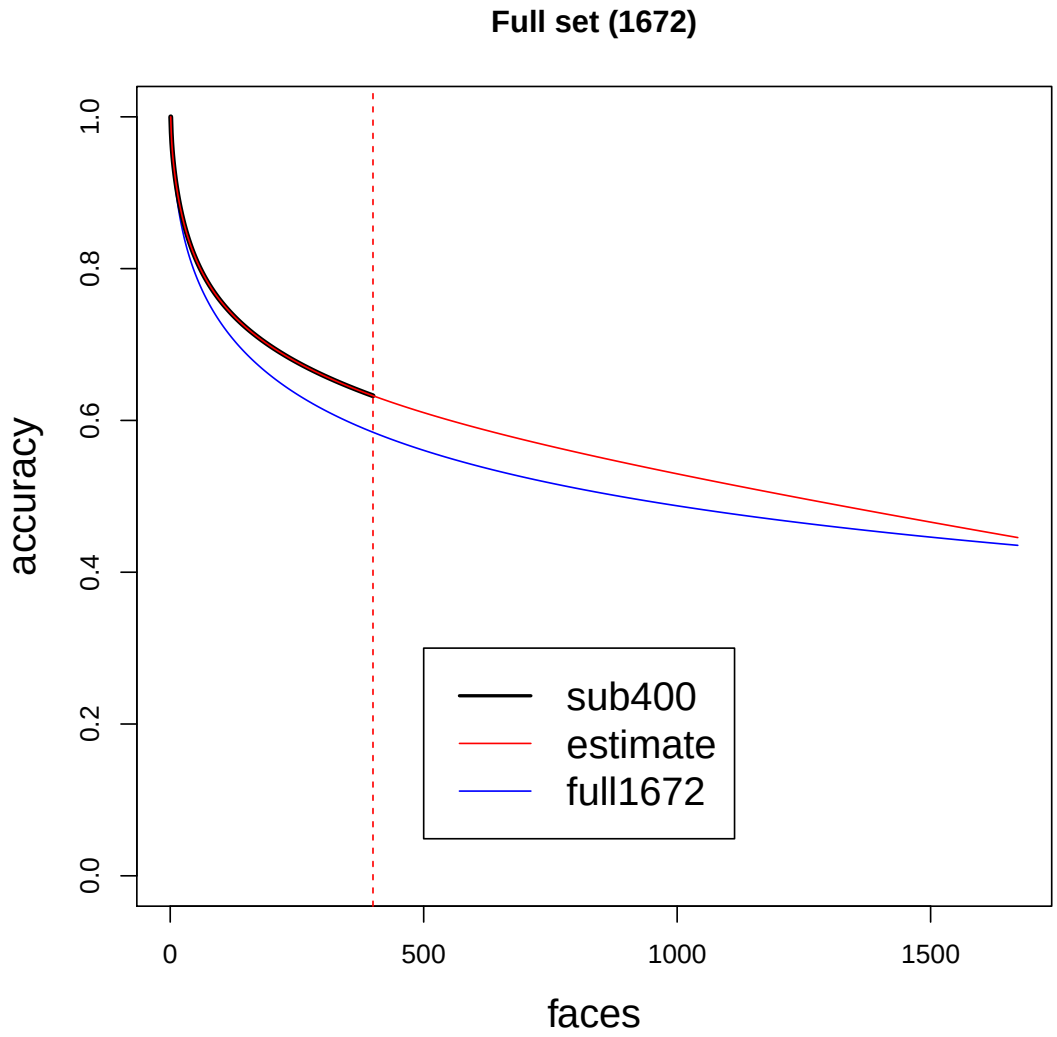
Estimated $\bar{D}(u)$

Using linear spline basis ($p = 10000$) and nonnegativity constraint.

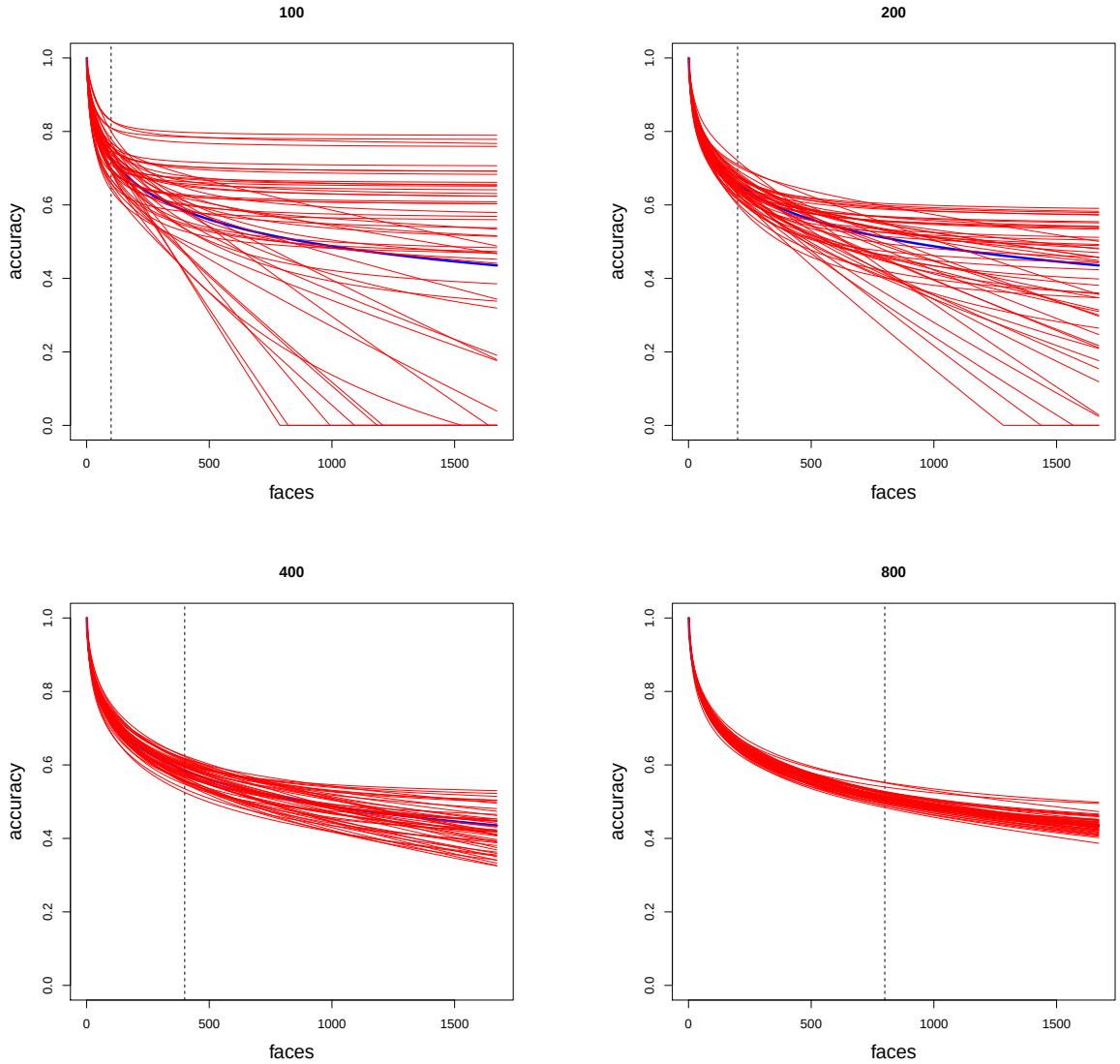


Estimated risk

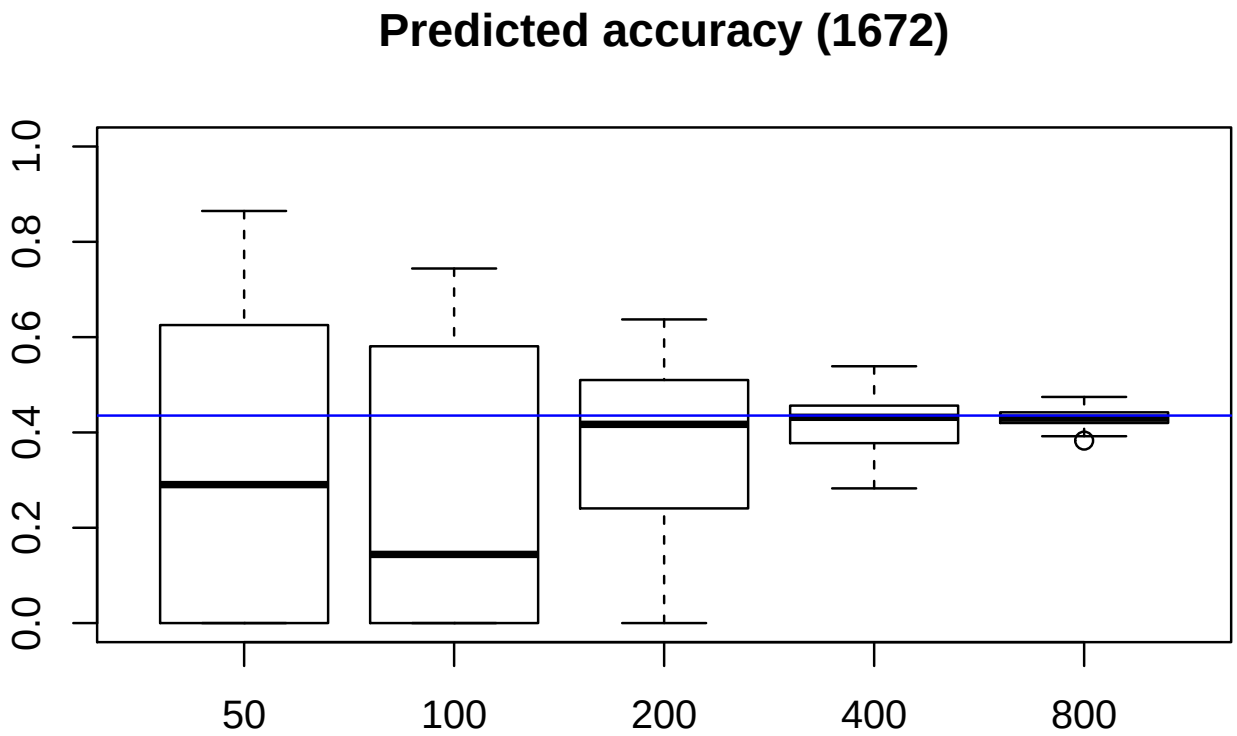
Compare to test risk at $K = 1672$



Estimated risk: more experiments

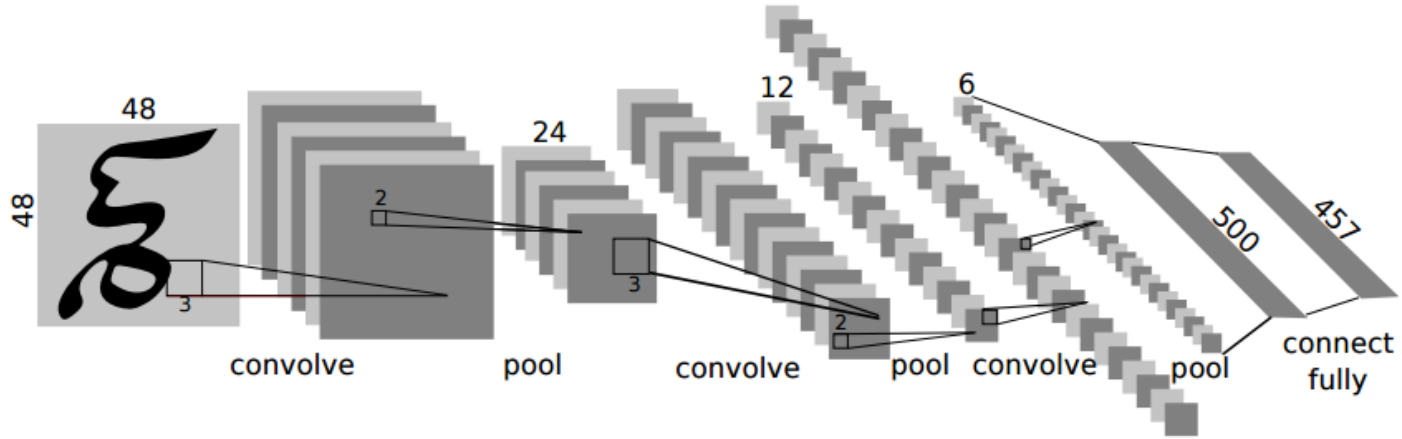


Estimated risk: more experiments



Telugu OCR example

కమ్మనిలతాంతముల కుమ్మినసివచ్చు మధు
పమ్మిల సుగీతనినదమ్మిలెగసెంజూ
తమ్మిల లసత్కిసలయమ్మిల సుగంధిముకు
ళమ్మిలను నానుచు ముదమ్మినర వాచా



Telugu OCR example

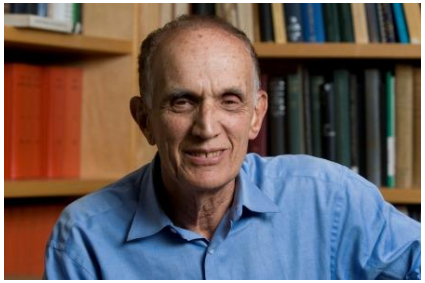
Classifier	Test err$^{(20)}$	Test err$^{(400)}$	$\hat{p}_{400}^{EXP}$	$\hat{p}_{400}^{POS}$	$\hat{p}_{400}^{(5)}$
Naive Bayes	0.049	0.399	0.108	**0.142**	0.079
Logistic	0.078	0.289	0.166	**0.188**	0.130
SVM	0.140	0.455	0.299	**0.313**	0.227
ϵ-NN	0.049	0.409	0.084	**0.590**	0.102
Deep neural net	0.005	0.014	**0.011**	0.093	0.010

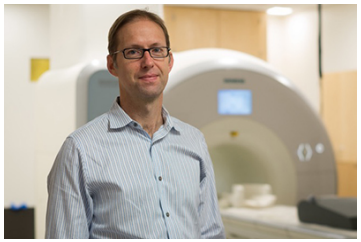
Performance extrapolation: predicting the accuracy on 400 classes using data from 20 classes on a Telugu character dataset. $\epsilon = 0.002$ for ϵ-nearest neighbors.

Section 4

Acknowledgements


CHALLENGER






Year 1
Year 2
Year 3
Year 4
Year 5 +
Fang Cai
Michael Sklar
Mona Azadkia
Jing Miao
Yu Bai
Jelena Markovich
Rakesh Achanta
Cyrus DiCiccio
Kelvin Gu
Minyong Lee
Michael Celentano
Youngtak Sohn
Stephen Bates
Evan Rosenman
Alex Chin
Paulo Orenstein
Joey Arthur
Snigdha Panigrahi
Edgar Dobriban
Ruojun Huang
Haben Michael
Dylan Greaves
Kenneth Tay
Claire Donnat
Matteo Sesia
Wenfei Du
Evan Patterson
Nan Bi
Scott Powers
Julia Fukuyama
Lucas Janson
Amir Sepehri
Xiaoyan Han
Xiaowei Wang
Rina Friedberg
Andrew Tsao
Leying Guan
Junyang Qian
Zhou Fan
Kris Sankaran
Katelyn Gao
Bai Jiang
Xiaoying Tian
Nikolaos Ignatiadis
Zhang Yiguang
Nima Hamidi
Jun Yan
Eugene Katsevich
Feng Ruan
Jessica Hwang
Subhabrata Sen
Jack Gorham
Ya Le
Charles Zheng
Zhimei Ren
Jaime Roquero Gimenez
Qian Zhao
Keli Liu
Pragya Sur
Jeha Yang
Jiyao Kou
David Walsh
Murat Erdogdu
Chaojung Wang

Section 5

The end