

Extrapolating prediction error for 'extreme' multi-class classification

Charles Zheng

Stanford University

February 22, 2017

(Joint work with Rakesh Achanta and Yuval Benjamini.)

Multi-class classification



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

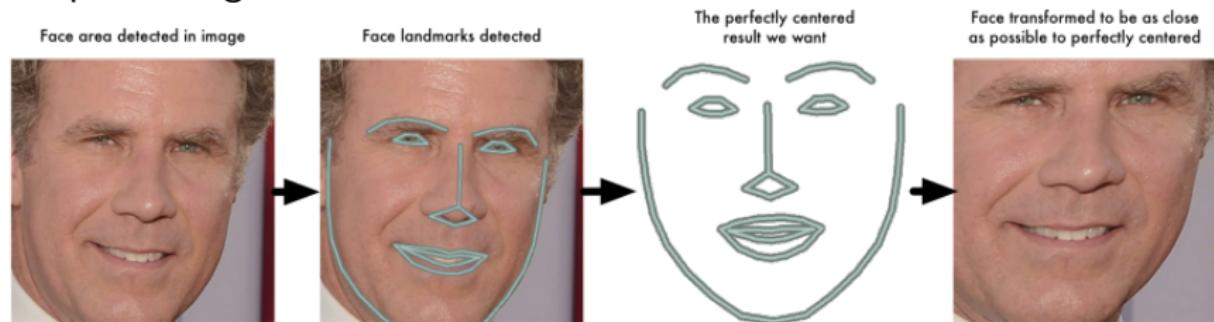
from Krizhevsky et al. 2012

Facial recognition

- Used to tag images in software, security

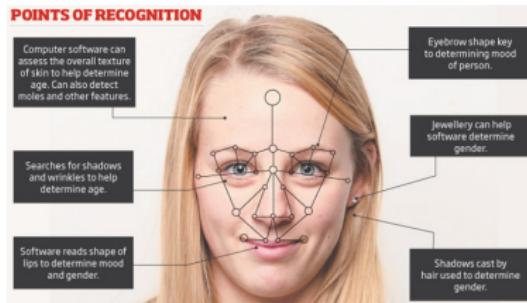
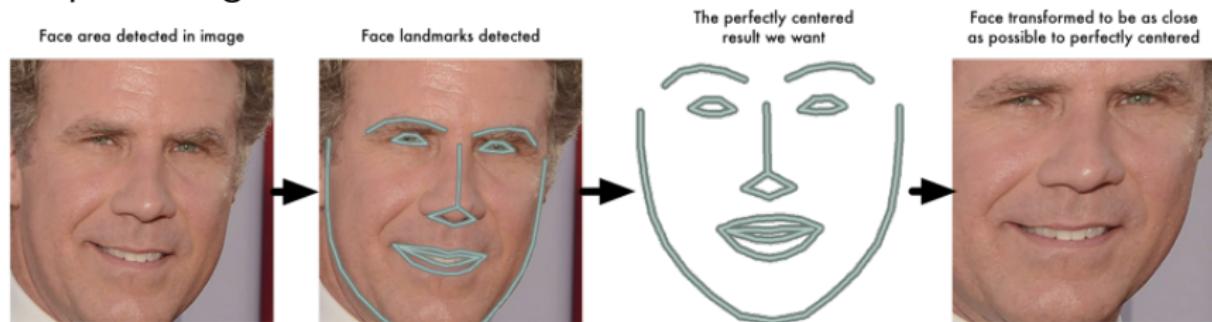
Facial recognition

- Used to tag images in software, security
- Preprocessing



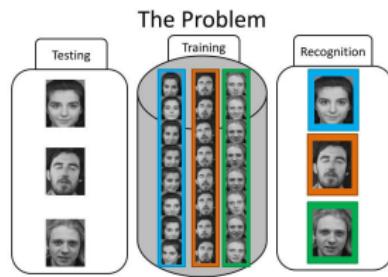
Facial recognition

- Used to tag images in software, security
- Preprocessing

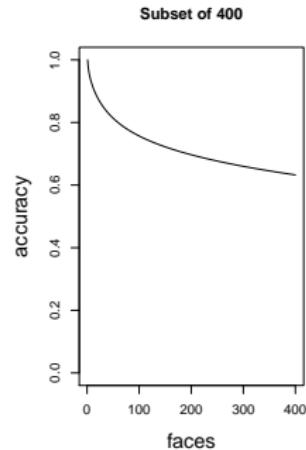
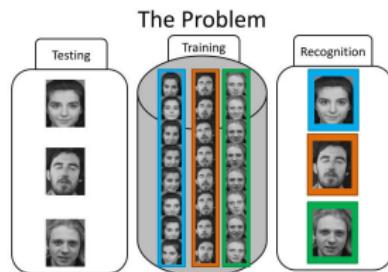


- Feature extraction

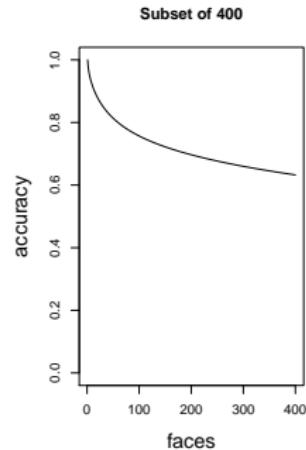
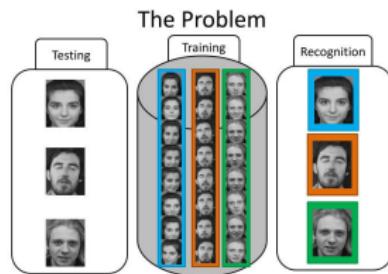
Accuracy vs. number of classes



Accuracy vs. number of classes



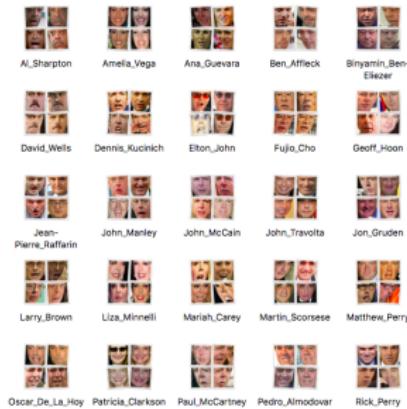
Accuracy vs. number of classes



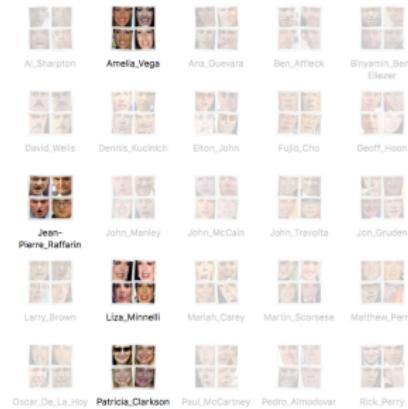
How does the accuracy scale with the number of classes (faces)?

Setup

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



2. Subsample k labels, y_1, \dots, y_k



Setup

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

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

4. Train a classifier and compute test error.

Setup

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

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

4. Train a classifier and compute test error.

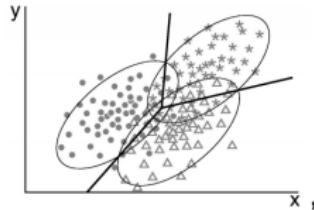
Can we analyze how error depends on k ?

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.

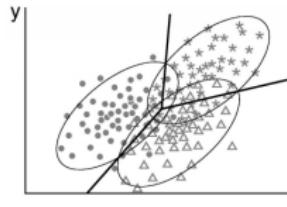


- Examples: LDA/QDA

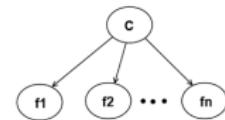
Key assumption: marginal classifier

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

- Examples: LDA/QDA

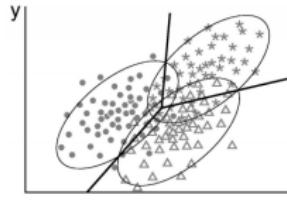


, naïve Bayes

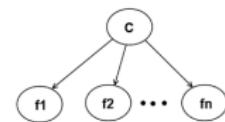


Key assumption: marginal classifier

- 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

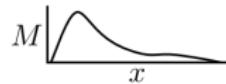


Definitions

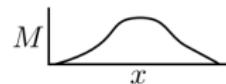
$\hat{F}_{y^{(i)}}$ is the empirical distribution obtained from the training data for label $y^{(i)}$.

Classification Rule

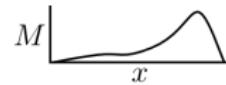
$$M_{y^{(1)}}(x) = \mathcal{M}(\hat{F}_{y^{(1)}})(x)$$



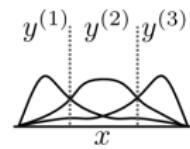
$$M_{y^{(2)}}(x) = \mathcal{M}(\hat{F}_{y^{(2)}})(x)$$



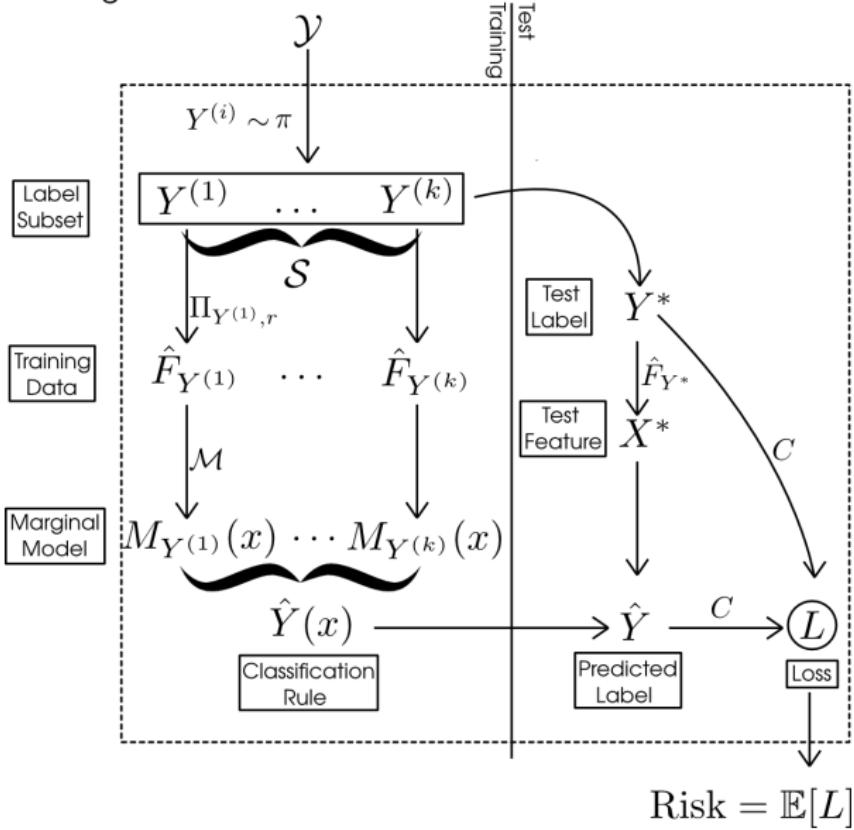
$$M_{y^{(3)}}(x) = \mathcal{M}(\hat{F}_{y^{(3)}})(x)$$



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



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 $[0, 1] \rightarrow [0, 1]$ 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 $[0, 1] \rightarrow [0, 1]$ 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...

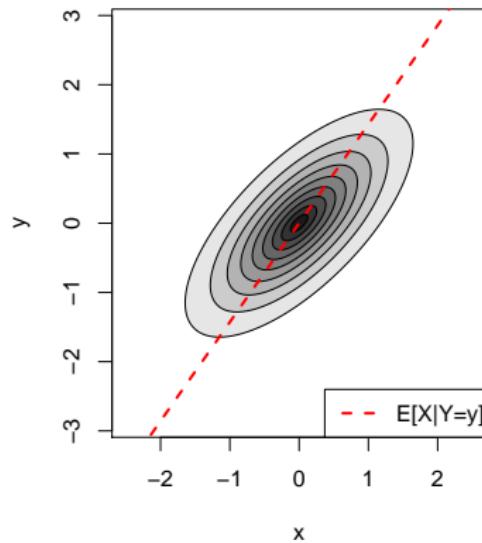
Toy example

$$Y_1, \dots, Y_k \stackrel{iid}{\sim} N(0, 1);$$

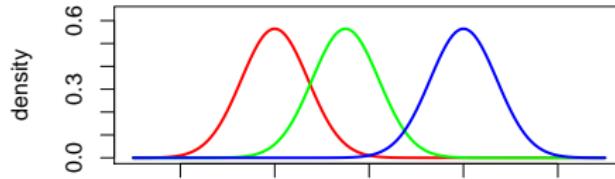
Toy example

$Y_1, \dots, Y_k \stackrel{iid}{\sim} N(0, 1);$

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

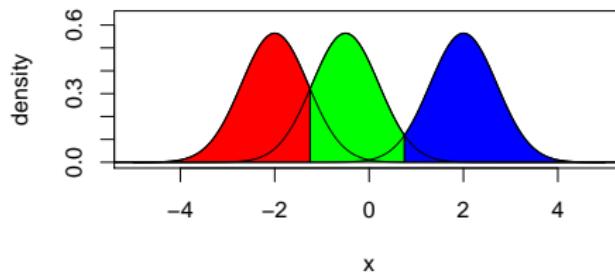


Toy example



- 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:
$$\hat{y}(x) = \operatorname{argmax}_{y_i} p(x|y_i)$$
- The *Bayes Risk*, which is the misclassification rate of the optimal classifier.

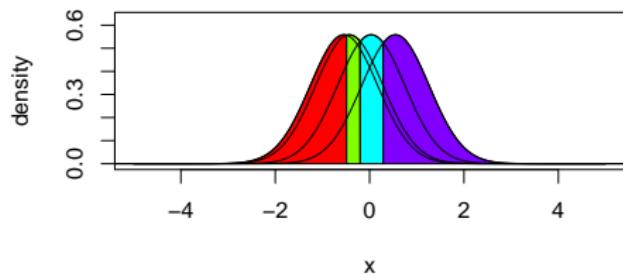
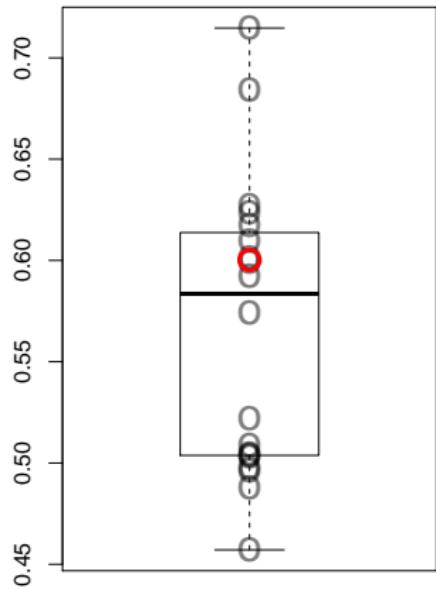
Toy example



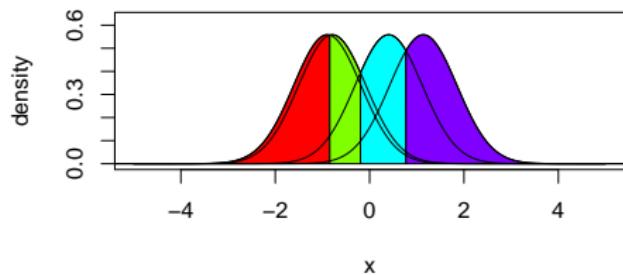
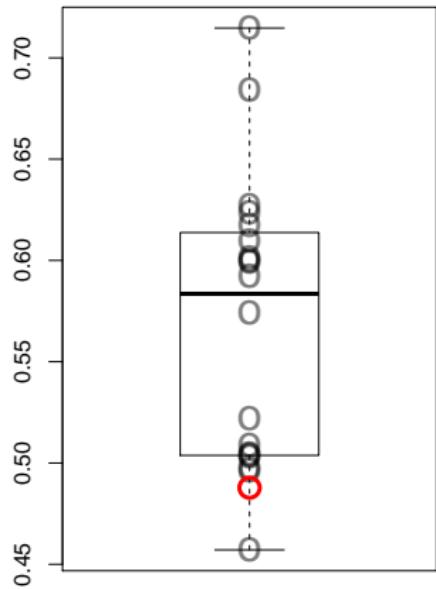
- The *Bayes Risk* is the expected test error of the Bayes rule,

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

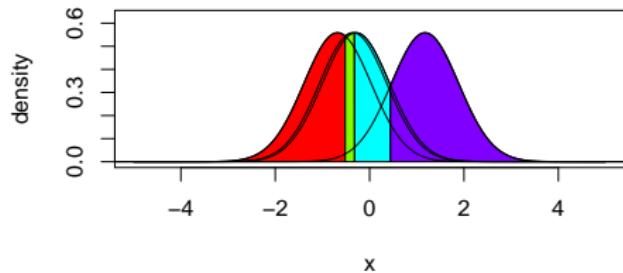
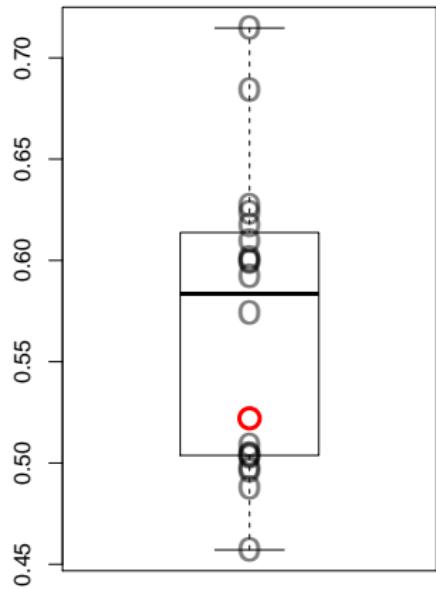
Toy example



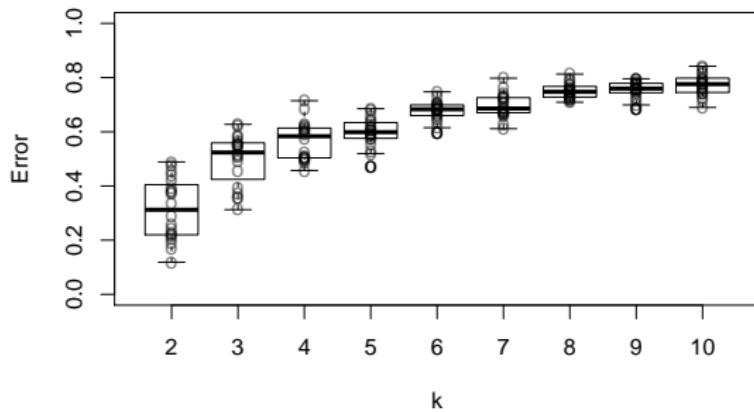
Toy example



Toy example

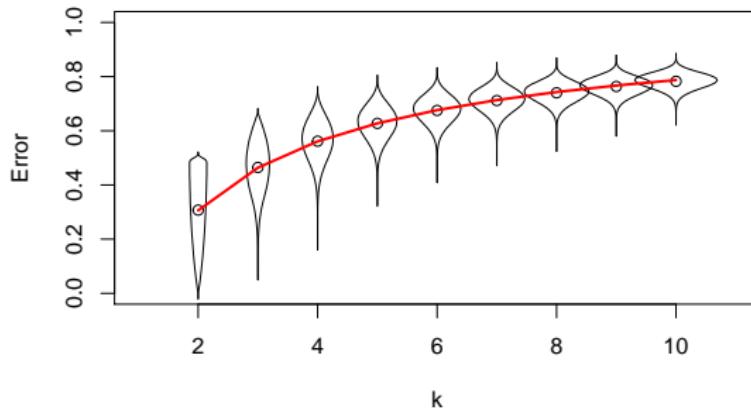


Toy example



Toy example

$\rho = 0.7$



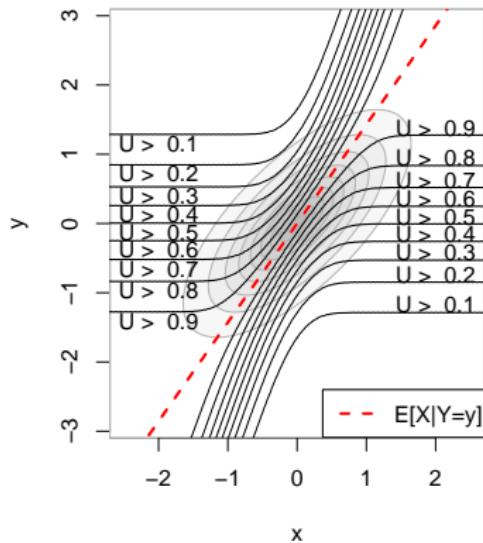
Defining the U -function

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 probability of success (randomizing over training data).

Toy example

rho = 0.7



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

Defining $\bar{D}(u)$

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

Defining $\bar{D}(u)$

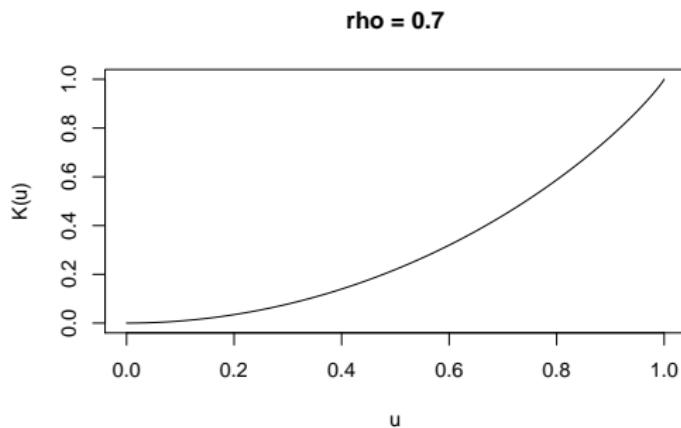
- Define random variable as $U_Y(X)$ for (Y, X) drawn from the joint distribution.
- $\bar{D}(u)$ is the cumulative distribution function of U ,

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

Defining $\bar{D}(u)$

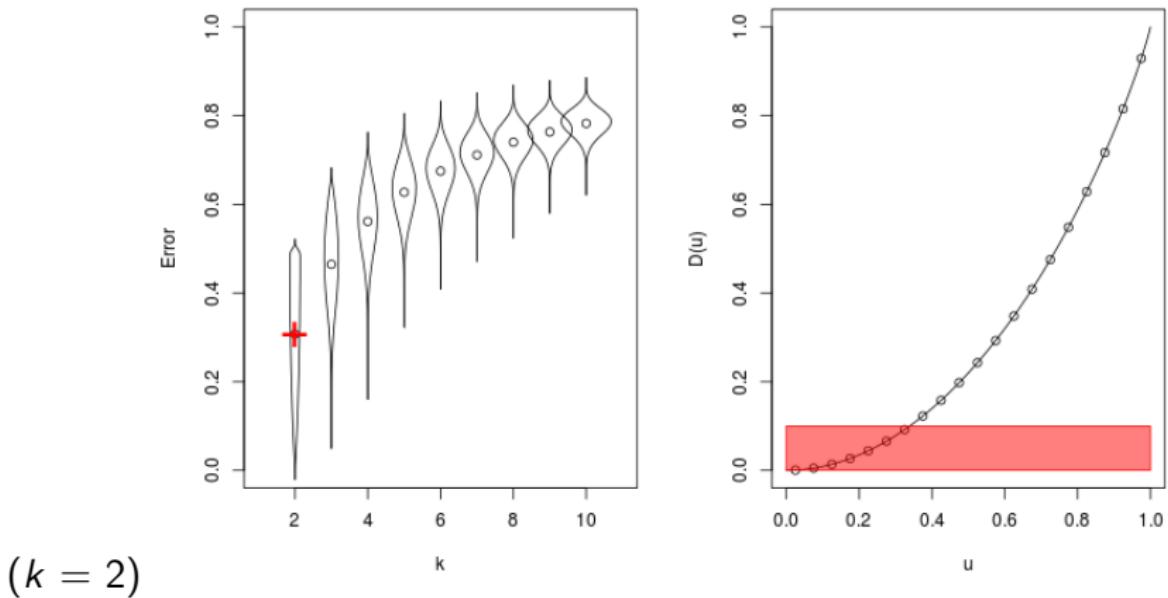
- Define random variable as $U_Y(X)$ for (Y, X) drawn from the joint distribution.
- $\bar{D}(u)$ is the cumulative distribution function of U ,

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



Computing average risk

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

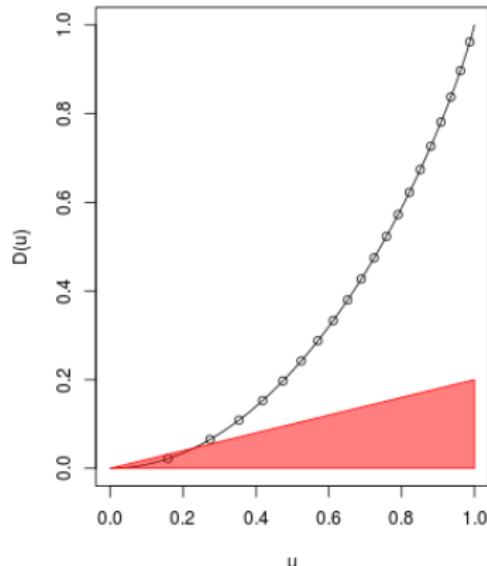
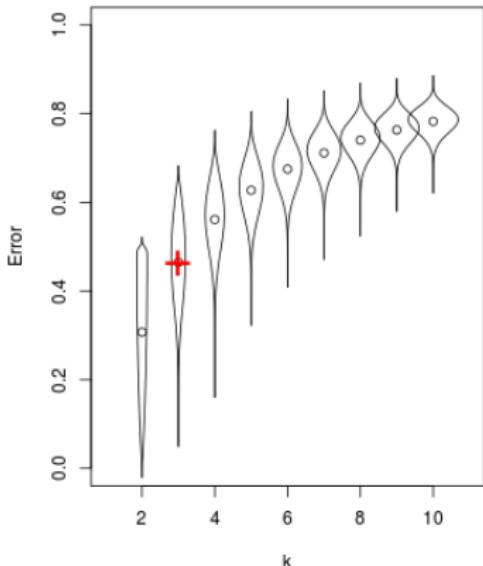


$(k = 2)$

Computing average risk

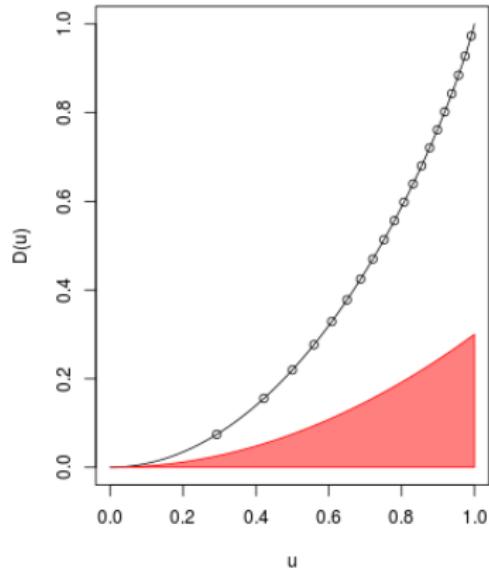
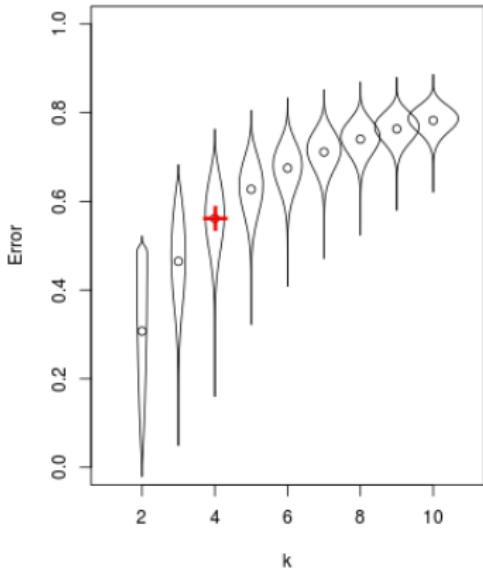
$$\text{AvRisk}_k = (k - 1) \int \bar{D}(u) u^{k-2} du.$$

$(k = 3)$



Computing average risk

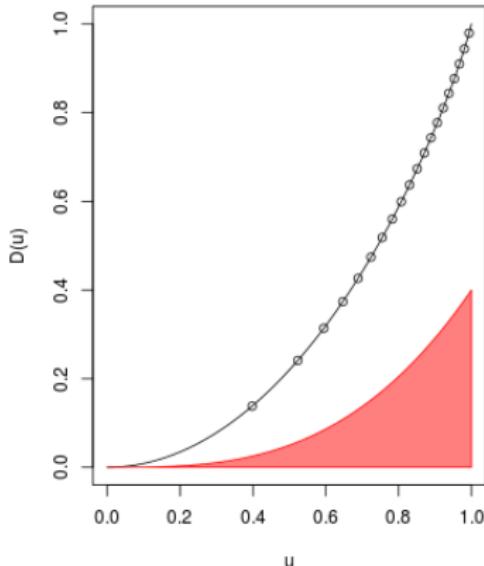
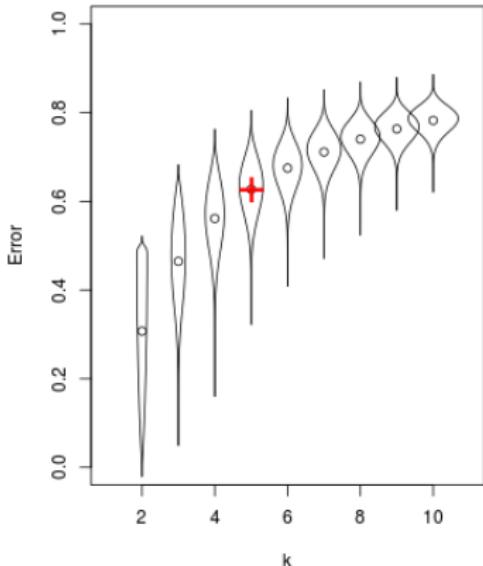
$$\text{AvRisk}_k = (k - 1) \int \bar{D}(u) u^{k-2} du.$$



Computing average risk

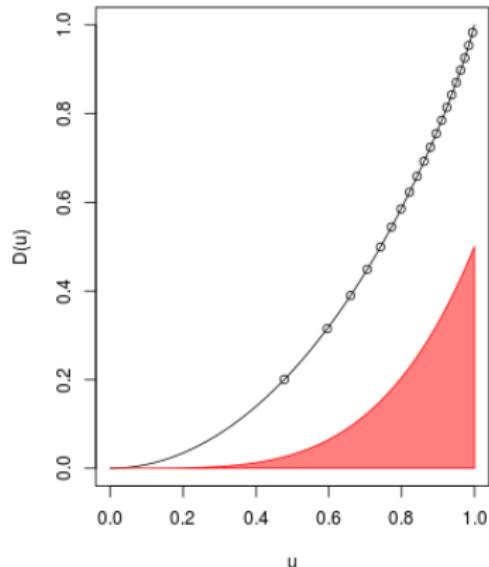
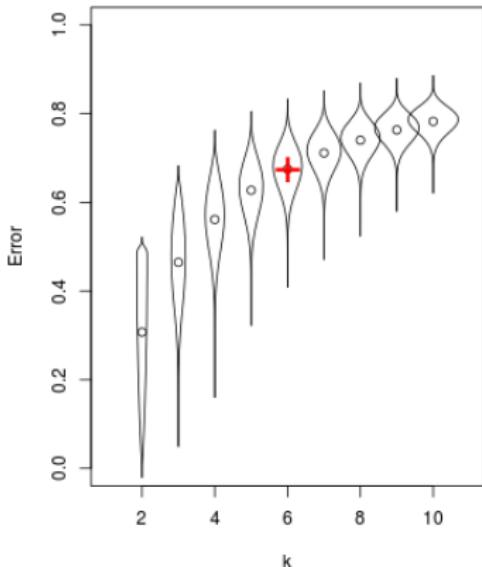
$$\text{AvRisk}_k = (k - 1) \int \bar{D}(u) u^{k-2} du.$$

$(k = 5)$



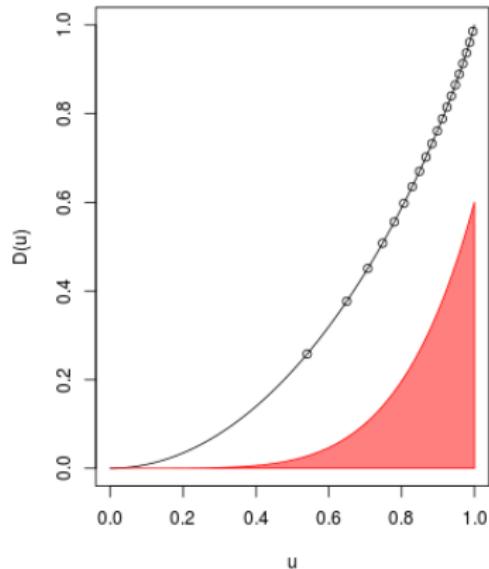
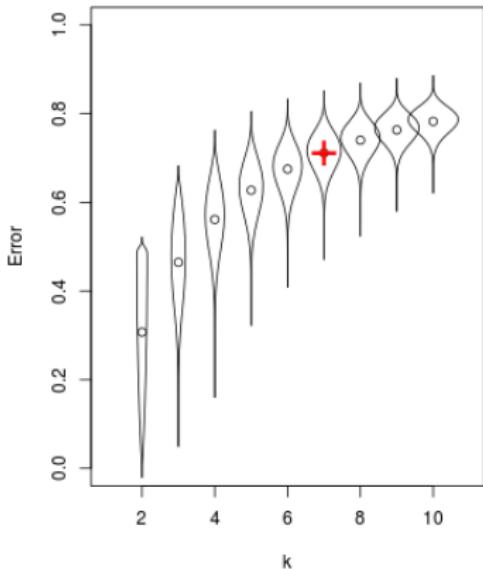
Computing average risk

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



Computing average risk

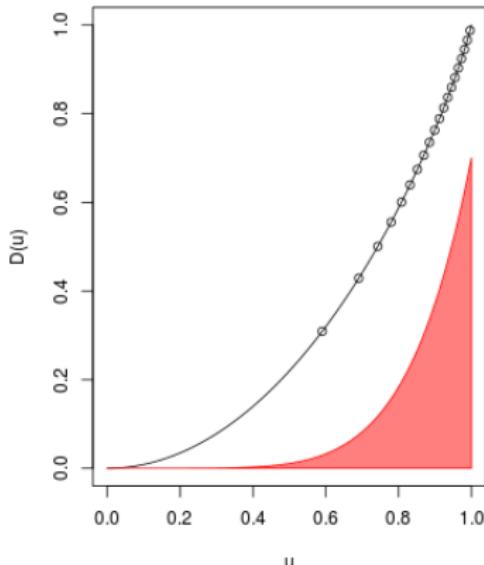
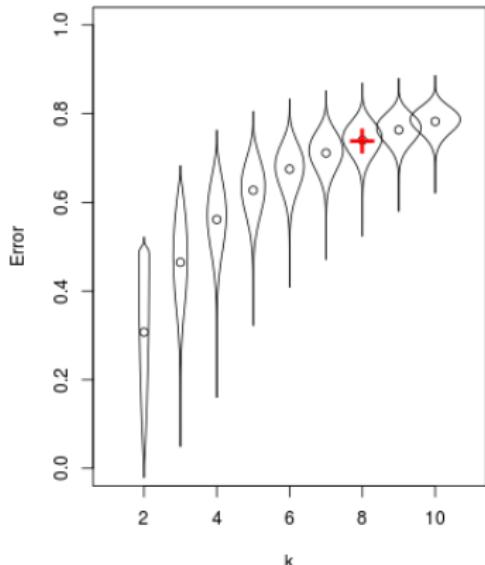
$$\text{AvRisk}_k = (k - 1) \int \bar{D}(u) u^{k-2} du.$$



Computing average risk

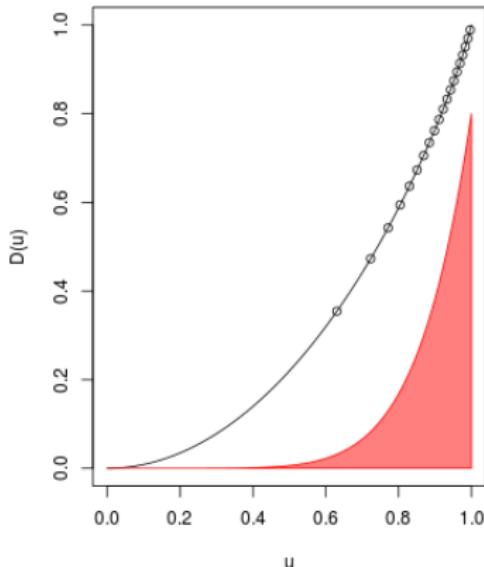
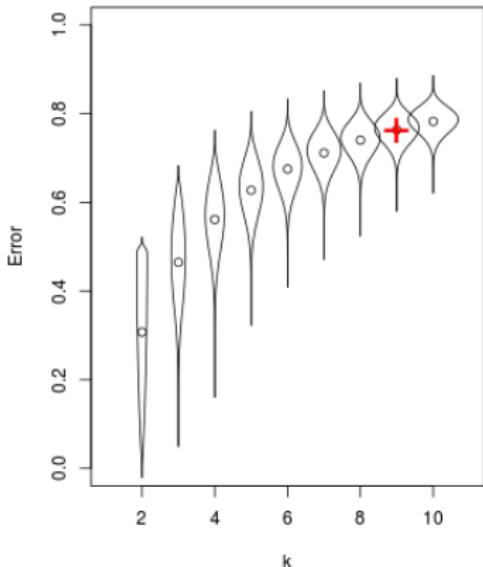
$$\text{AvRisk}_k = (k - 1) \int \bar{D}(u) u^{k-2} du.$$

$(k = 8)$



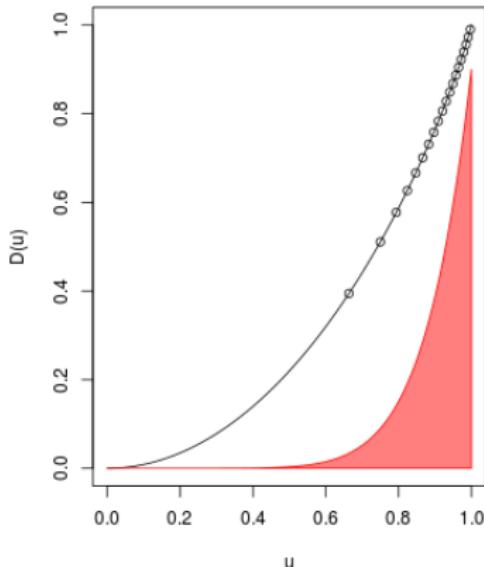
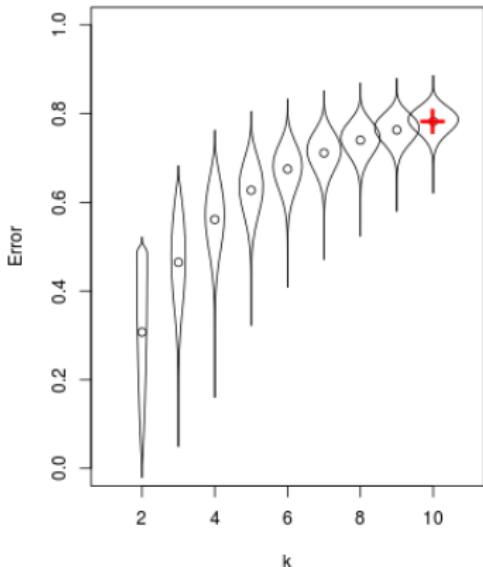
Computing average risk

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



Computing average risk

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



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 .

| | |
|---------|--|
| $k = 4$ |  vs  vs  vs  vs  |
| $k = 3$ |  vs  vs   vs  vs   vs  vs   vs  vs  |
| $k = 2$ |  vs   vs   vs   vs   vs   vs  |

Parametric modelling approach

Assume that for set of basis functions h_1, \dots, h_ℓ , 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 $\vec{H}_k = (H_{1,k}, \dots, H_{\ell,k})$

Prediction extrapolation as regression

- ① Choose basis h_1, \dots, h_ℓ

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

Prediction extrapolation as regression

- ① Choose basis h_1, \dots, h_ℓ

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

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

Prediction extrapolation as regression

- ① Choose basis h_1, \dots, h_ℓ

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

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

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

Prediction extrapolation as regression

- ① Choose basis h_1, \dots, h_ℓ

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

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

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

- ④ For $K > k$, predict AvRisk_K as

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

Examples of basis functions

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

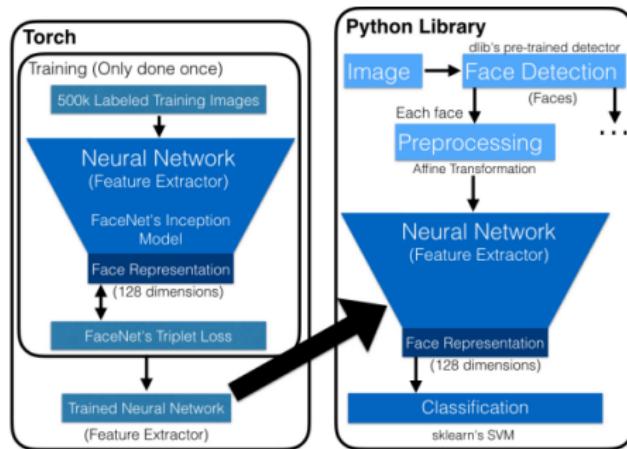
Optional constraint: assume β is *non-negative*. In the case of linear splines, this results in a *convex* fit.

Section 1

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

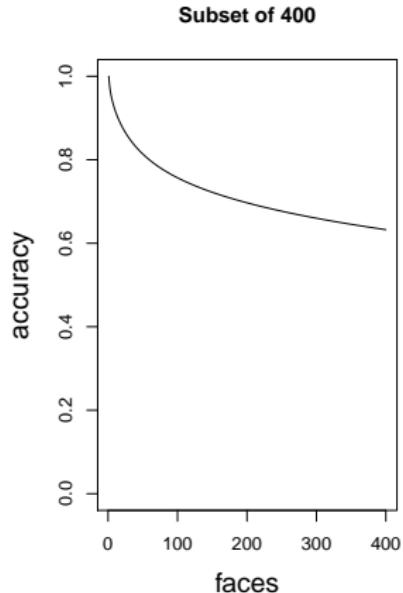


Facial recognition example

- 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!

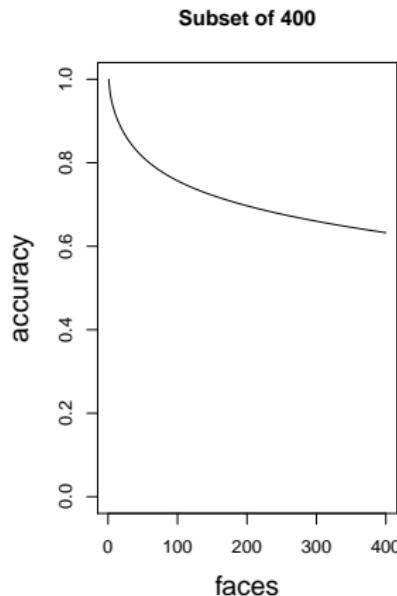
Facial recognition example

- 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

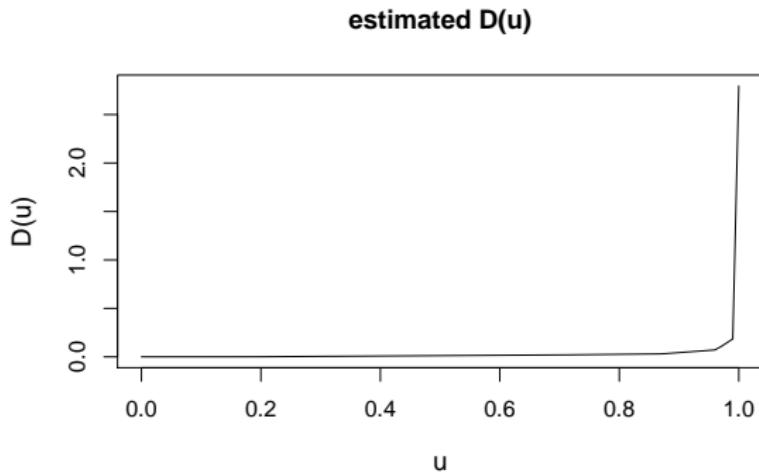
- Let us first subsample 400 faces (out of 1672)
- Randomly choose 1 face as training and 1 as test for each person
- Use 1-nearest neighbor.
 - NOTE: 1-NN with 1 example/class is equivalent to LDA with $\Sigma = I$: this fits marginal classifier assumption!



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

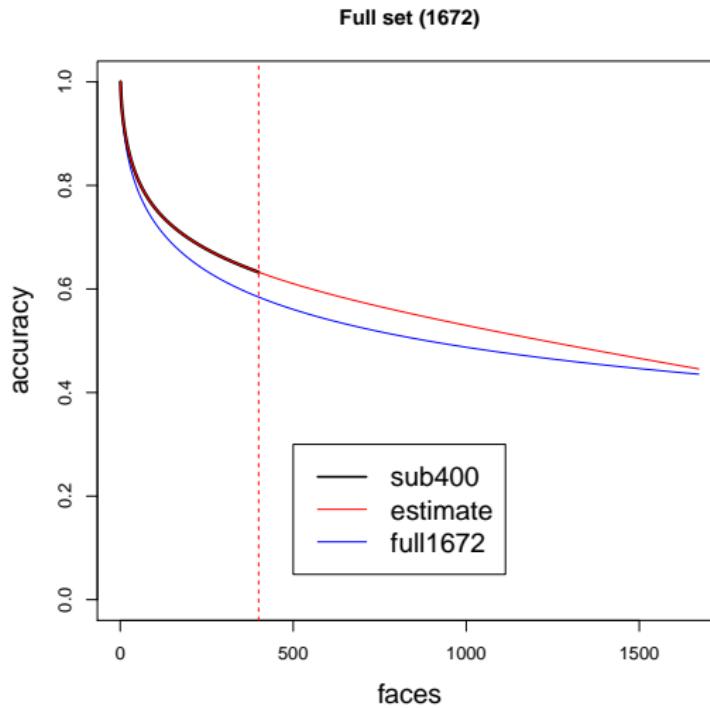
Estimated $\bar{D}(u)$

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

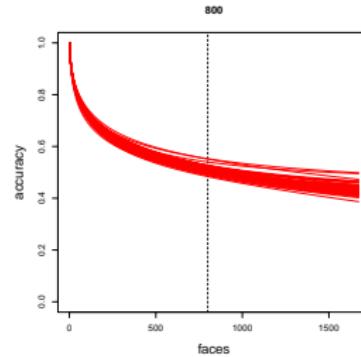
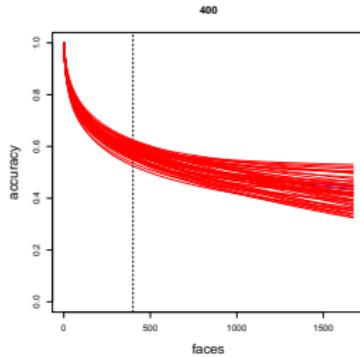
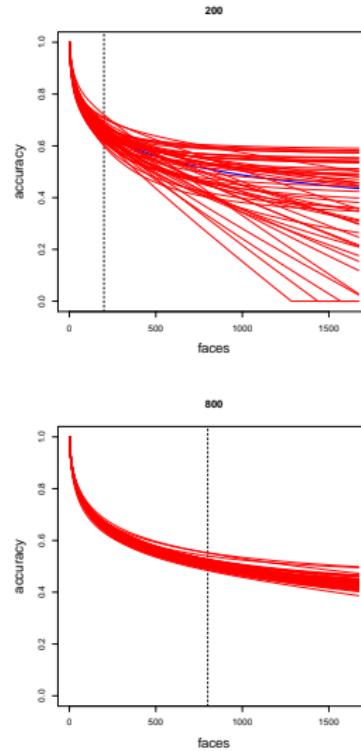
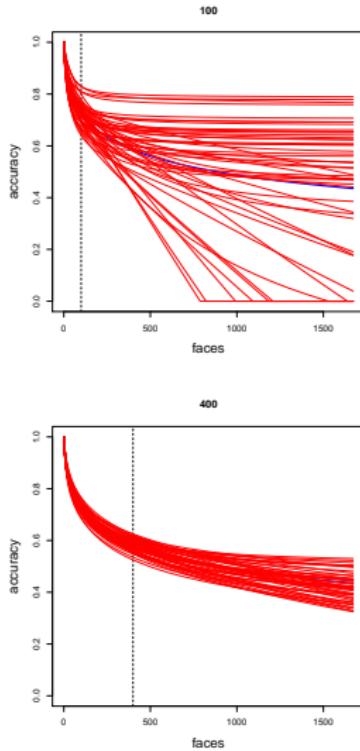


Estimated risk

Compare to test risk at $K = 1672$

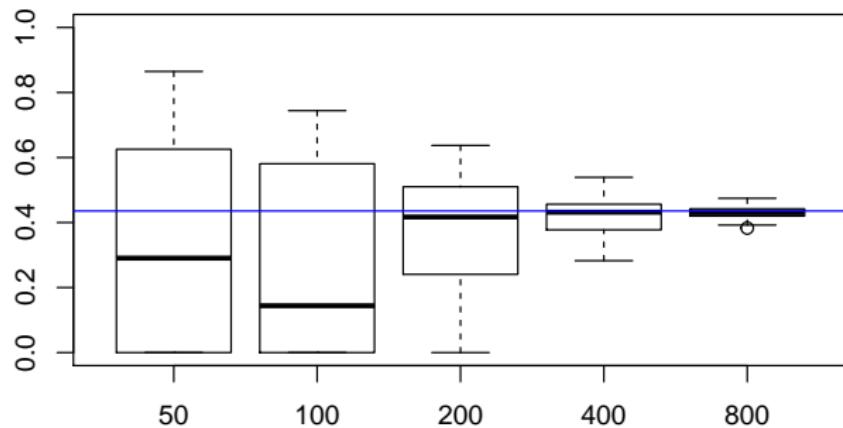


Estimated risk: more experiments



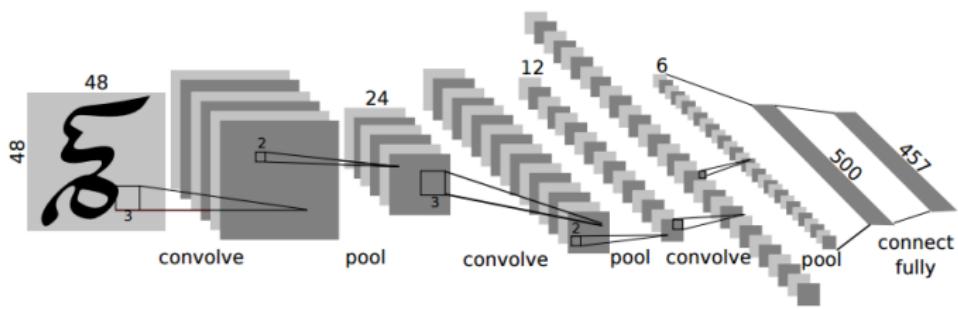
Estimated risk: more experiments

Predicted accuracy (1672)



Telugu OCR example

కమ్మనిలతాంతముల కుమ్మెన్నసవచ్చ) మధు
పమ్ముల సుగీతనినదమ్ములెగెసింజా
తమ్ముల లసత్కిసలయమ్ముల సుగంధిముకు
తమ్ములను నానుచు ముదమ్మెనర 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 2

The end