

Thesis Defense

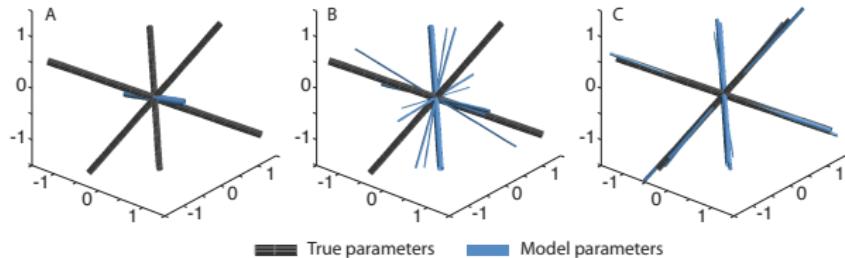
Charles Zheng

Stanford University

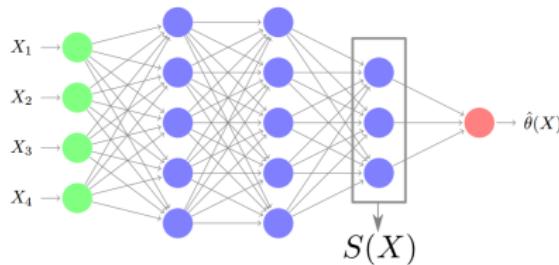
April 22, 2017

Research interests

- Statistical analysis of neuroimaging data



- Applications of machine learning in statistical inference



Functional neuroimaging

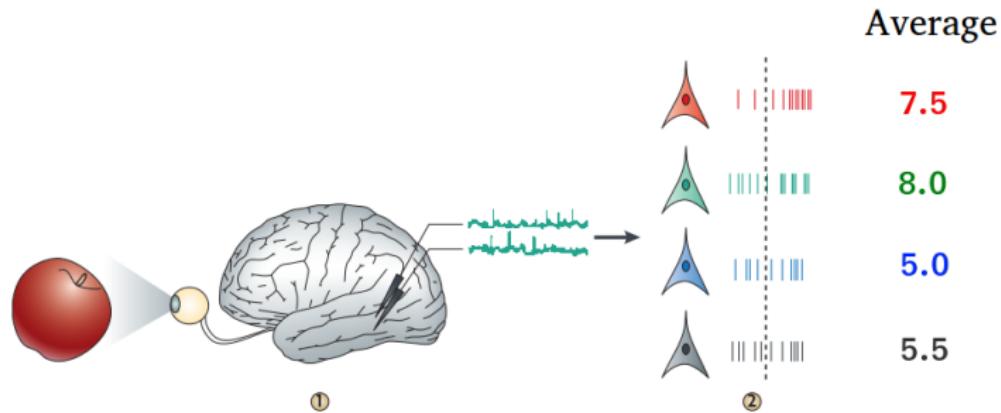
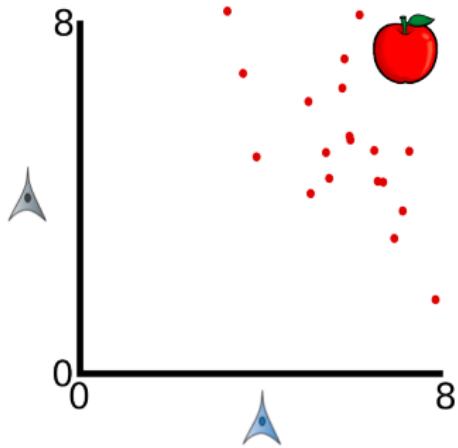
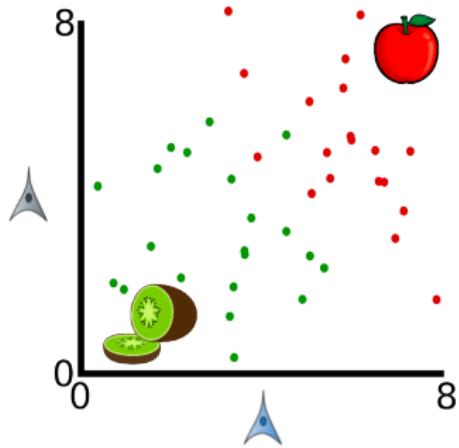


Image adapted from Quiroga et al (2009)

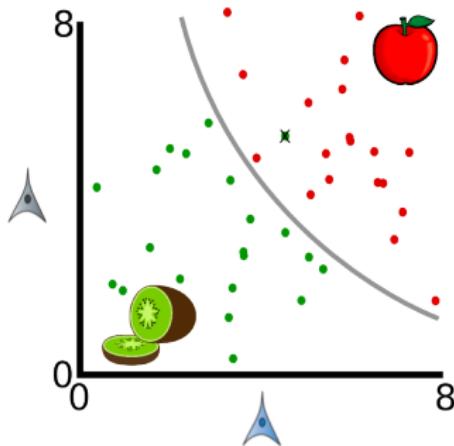
Functional neuroimaging



Functional neuroimaging



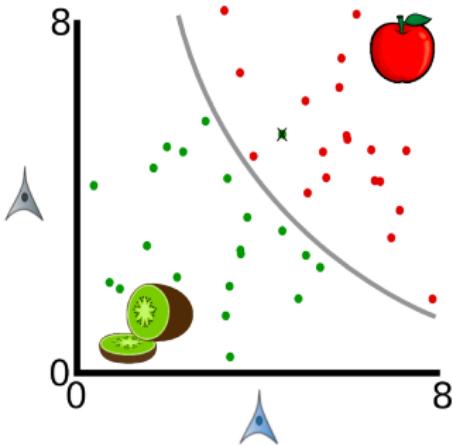
Classification/Decoding



- Response $Z = \{0 \text{ (apple)}, 1 \text{ (banana)}\}$.
- Predictors Y_1, \dots, Y_p (voxels)
- Classifier $f : (Y_1, \dots, Y_p) \rightarrow \{0, 1\}$ guesses the class.
- Generalization accuracy

$$A(f) = \Pr[f(Y_1, \dots, Y_p) = Z].$$

What's the parameter?

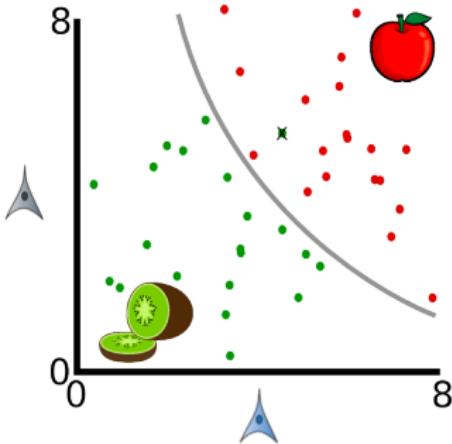


- The classifier is chosen from some class \mathcal{F} , e.g. maximizing empirical accuracy

$$\hat{f} = \sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n I\{\hat{f}(X_1^{(i)}, \dots, X_p^{(i)}) = Z^{(i)}\}.$$

- Generalization accuracy $A(\hat{f})$ varies depending on data.

What's the parameter?



- Define Bayes accuracy

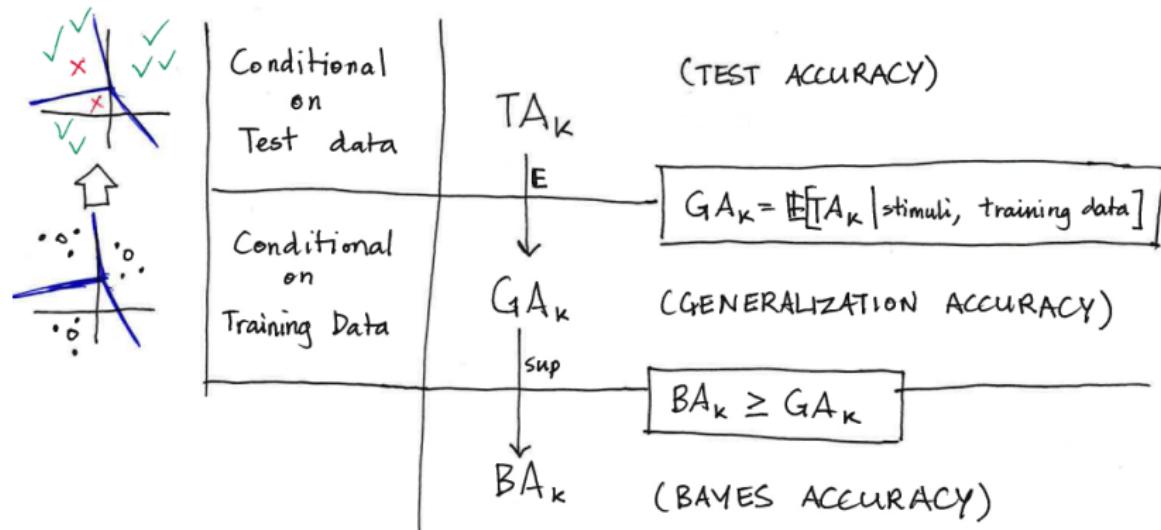
$$\text{BA} = \sup_f A(f).$$

- Under smoothness conditions on $p(x, y)$,

$$\lim_{n \rightarrow \infty} A(\hat{f}) \rightarrow \text{BA}(\hat{f})$$

for a variety of classifiers, e.g. k -nearest neighbors (Fukunaga 2009.)

Inferring Bayes accuracy

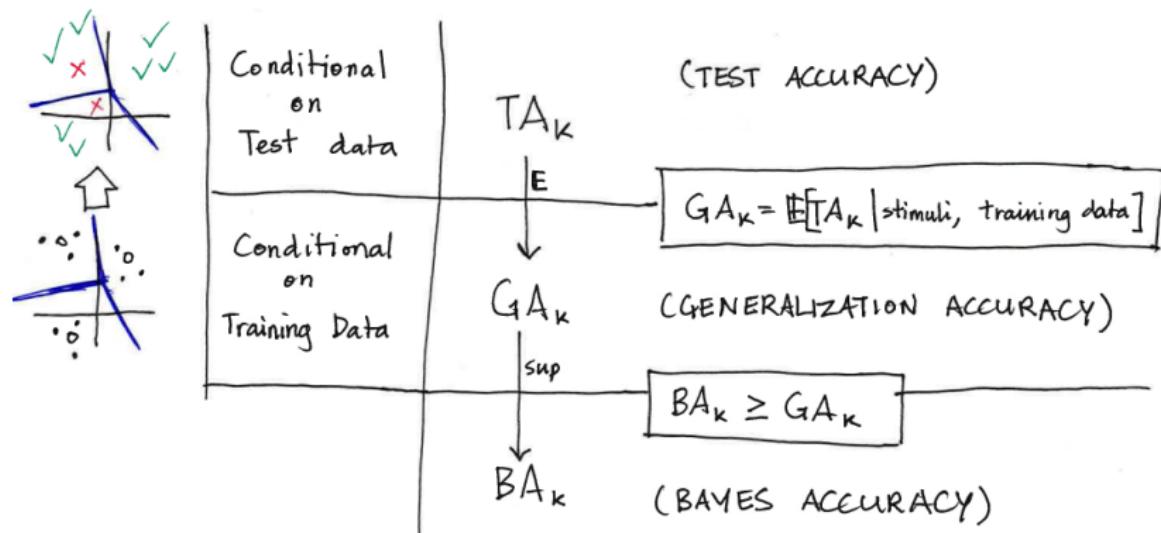


- Given m test observations,

$$GA_\alpha(\hat{f}) = TA - z_\alpha \sqrt{\frac{TA(1 - TA)}{m}}$$

is a an $(1 - \alpha)$ lower confidence bound for BA.

Inferring Bayes accuracy

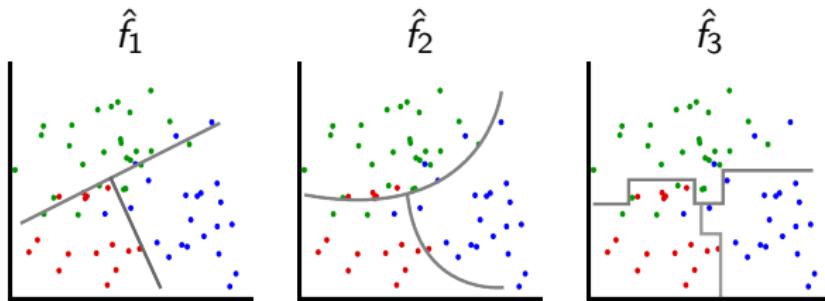


- Since $BA \geq GA$ by definition,

$$\underline{BA}_\alpha = \underline{GA}(\hat{f})$$

is an $(1 - \alpha)$ lower confidence bound for BA.

Inferring Bayes accuracy under model selection

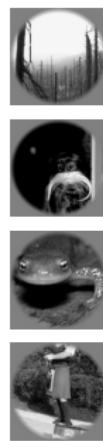
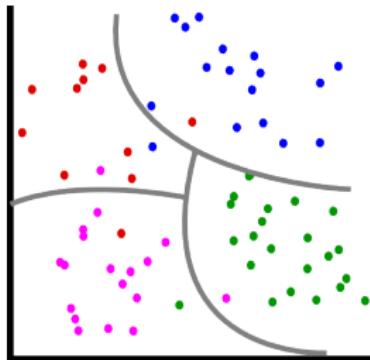
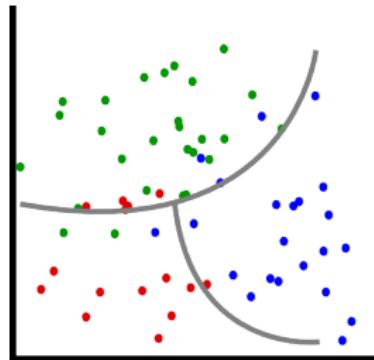


- Or, if $\hat{f}_1, \dots, \hat{f}_d$ result from d different procedures,

$$\underline{\text{BA}}_{\alpha} = \min_{i=1}^d \text{GA}_{\frac{\alpha}{d}}(\hat{f}_i)$$

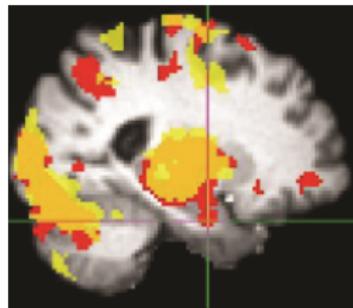
is also an $(1 - \alpha)$ lower confidence bound for BA (using Bonferroni's inequality).

Dependence of classification accuracy on stimuli



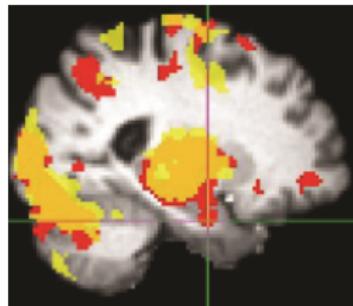
- Different stimuli sets lead to different *Bayes accuracy*.

Generalizing beyond the design



Scientists are not innately interested in the Bayes accuracy of a *particular* stimuli set, which is often chosen arbitrarily...

Generalizing beyond the design



But it would be more interesting to be able to make inferences from the data about a *larger* class of stimuli...

Section 2

Randomized classification and Average Bayes accuracy

Randomized classification

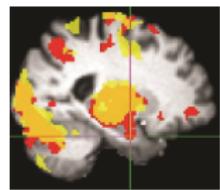
1. Population of stimuli $p(x)$



2. Subsample k stimuli



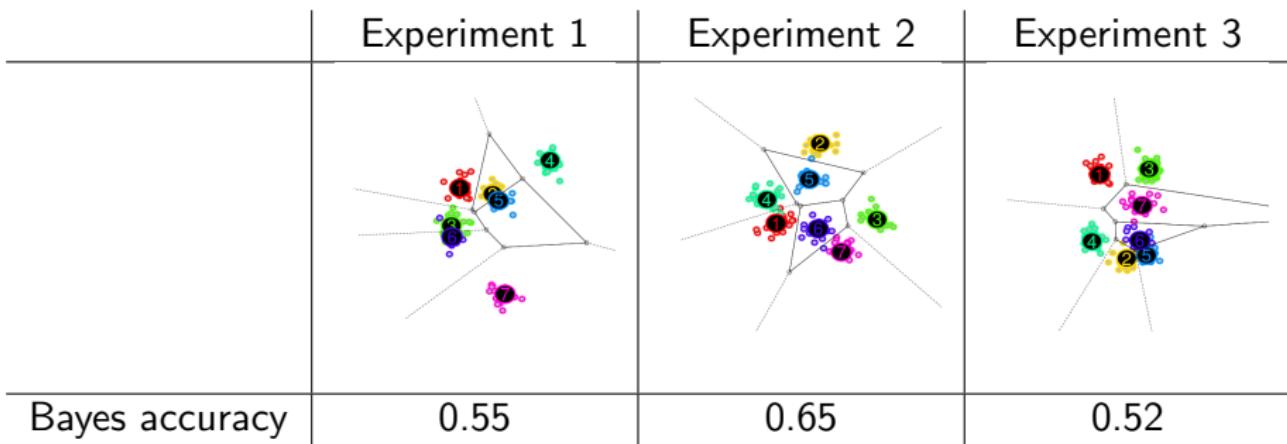
3. Data



4. Train a classifier

5. Estimate generalization accuracy (which is lower bound for the *random* Bayes accuracy BA_k)

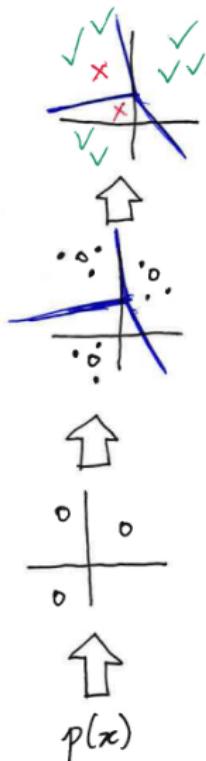
Average Bayes accuracy



- Bayes accuracy depends on the stimuli drawn.
- Therefore, define k -class *average Bayes accuracy* as the expected Bayes accuracy for $X_1, \dots, X_k \stackrel{iid}{\sim} p(x)$.

$$\text{ABA}_k = \mathbf{E}[BA(X_1, \dots, X_k)]$$

Average Bayes accuracy



Conditional on Test data	TA_k	E	(TEST ACCURACY)
Conditional on Training Data	GA_k		$GA_k = E[TA_k \text{stimuli, training data}]$
Conditional on Stimuli Set	BA_k	E \sup	(GENERALIZATION ACCURACY) $BA_k \geq GA_k$
Unconditional	ABA_k	E	(BAYES ACCURACY) $ABA_k = E[BA_k]$
			<u>AVERAGE BAYES ACCURACY</u>

Inferring average Bayes accuracy

- $\text{BA}_k \stackrel{\text{def}}{=} \text{BA}(X_1, \dots, X_k)$ is unbiased estimate of

$$\text{ABA}_k = \mathbf{E}[\text{BA}_k]$$

by definition.

- But what is the variance?

$$\text{Var}[\text{BA}(X_1, \dots, X_k)]$$

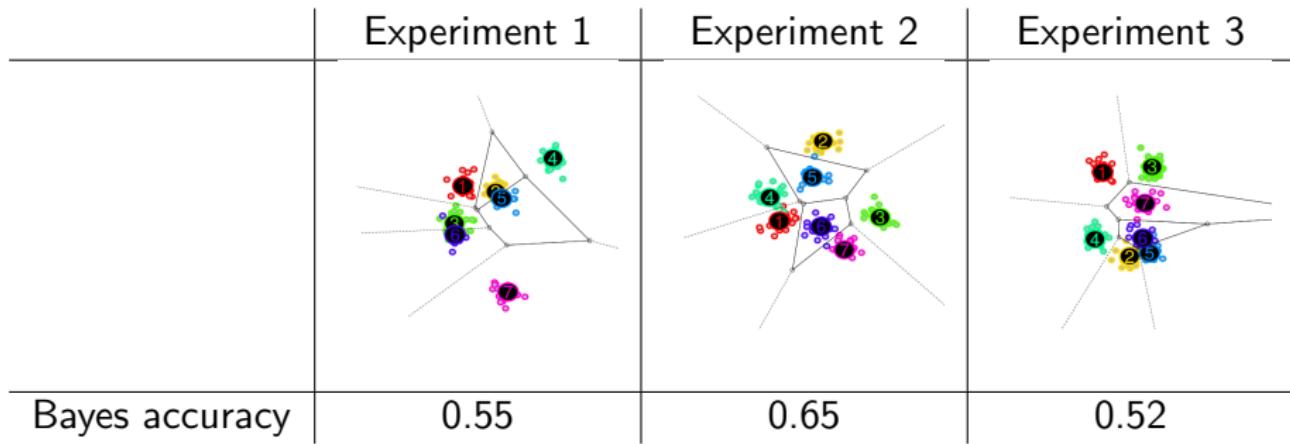
- *Theoretical result.* Maximal variability is of order $1/k$.
- Therefore, it is feasible to get a good idea of ABA_k by choosing a sufficiently large sample size k .

Two intuitions for variability result

Why does variability decrease with k ?

- 1. Bayes accuracy behaves like an average of k i.i.d random variables.
(Also gives correct $1/k$ rate.)
- 2. Bayes accuracy behaves like a max of k i.i.d. random variables.

Intuition 1: averaging



Average of k gaussian probability integrals... (which are asympt. uncorrelated.)

Intuition 2: An identity

- It is a well-known result from Bayesian inference that the optimal classifier f is defined as

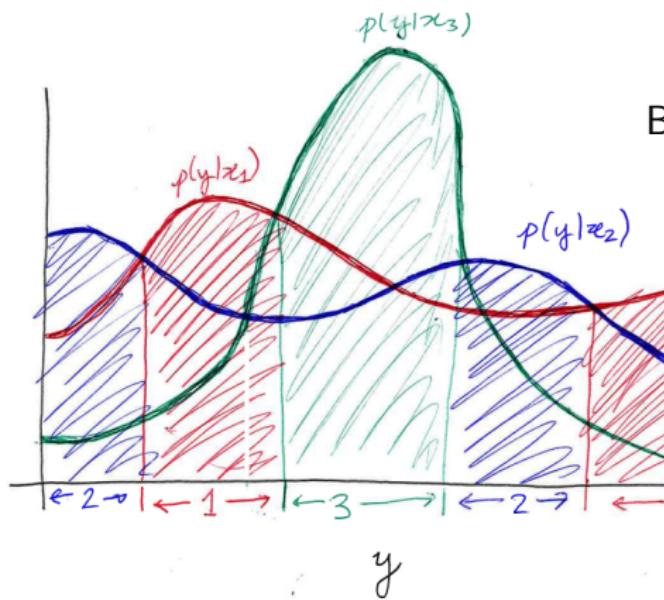
$$f(y) = \operatorname{argmax}_{i=1}^k p(y|x_i),$$

since the prior class probabilities are uniform.

- Therefore,

$$\begin{aligned}\text{BA}(x_1, \dots, x_k) &= \Pr[\operatorname{argmax}_{i=1}^k p(y|x_i) = Z | x_1, \dots, x_k] \\ &= \frac{1}{k} \int \max_{i=1}^k p(y|x_i) \prod_{i=1}^k p(x_i) dx_i dy.\end{aligned}$$

Intuition behind identity



BA(x_1, x_2, x_3)

$$= \sum_i \Pr[x_i] \Pr_{Y \sim p(y|x_i)} [Y \in \text{zone } i]$$

$$= \sum_i \frac{1}{k} \text{Area under curve } i \text{ in zone } i$$

$$= \frac{1}{k} \text{Area under } \max_{i=1}^k p(y|x_i)$$

Variability of Bayes accuracy

Theoretical result. In the max formulation of BA_k , we can apply Efron-Stein inequality to get

$$\text{sd}[\text{BA}_k] \leq \frac{1}{2\sqrt{k}}$$

Empirical results. (searching for worst-case stimuli).

k	2	3	4	5	6	7	8
$\frac{1}{2\sqrt{k}}$	0.353	0.289	0.250	0.223	0.204	0.189	0.177
Worst-case sd	0.25	0.194	0.167	0.150	0.136	0.126	0.118

Inferring average Bayes error

For now, return to the world of finite data...

- ① *Experimental design:* draw k stimuli X_1, \dots, X_k iid from $p(x)$. Then collect data (X_i, Y_i^j) .
- ② *Supervised learning:* train a classifier and obtain a test accuracy TA_k .
- ③ *Generalization accuracy:* if n_{test} is the size of the test set,

$$\underline{\text{GA}}_k = \text{TA}_k - \frac{z_{\alpha/2} \sqrt{\text{TA}_k(1 - \text{TA}_k)}}{\sqrt{n_{\text{test}}}}$$

is a lower confidence bound for GA_k

- ④ *Bayes accuracy:*

$$\underline{\text{BA}}_k = \underline{\text{GA}}_k$$

is a lower confidence bound for BA_k

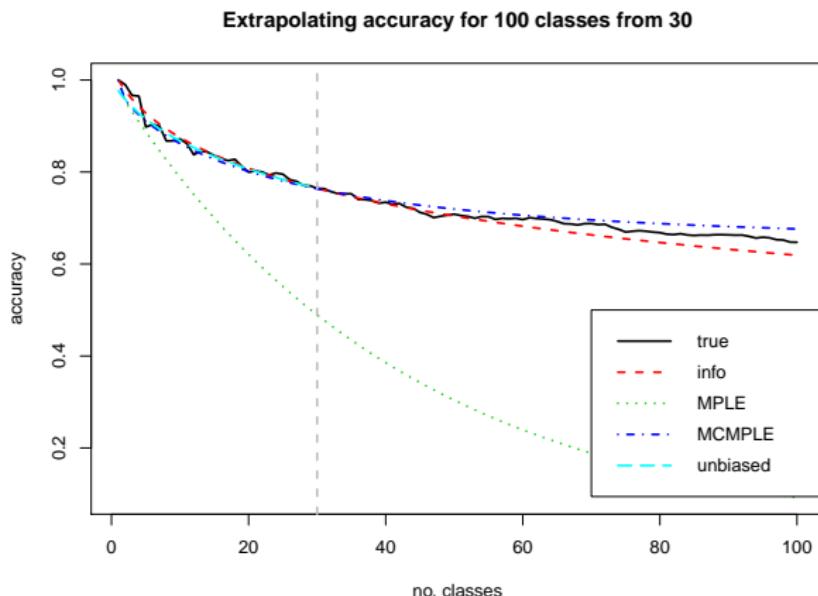
- ⑤ *Average Bayes accuracy*

$$\underline{\text{ABA}}_k = \underline{\text{BA}}_k - \frac{1}{2\sqrt{\alpha k}}$$

is a lower confidence bound for ABA_k .

Extension: Undersampled regime

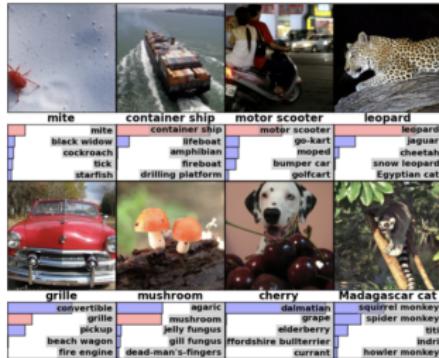
Do you actually have to collect data from k different classes to infer ABA_k ? See Z., Achanta and Benjamini (2016).



Section 3

Extrapolation

Multi-class classification



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

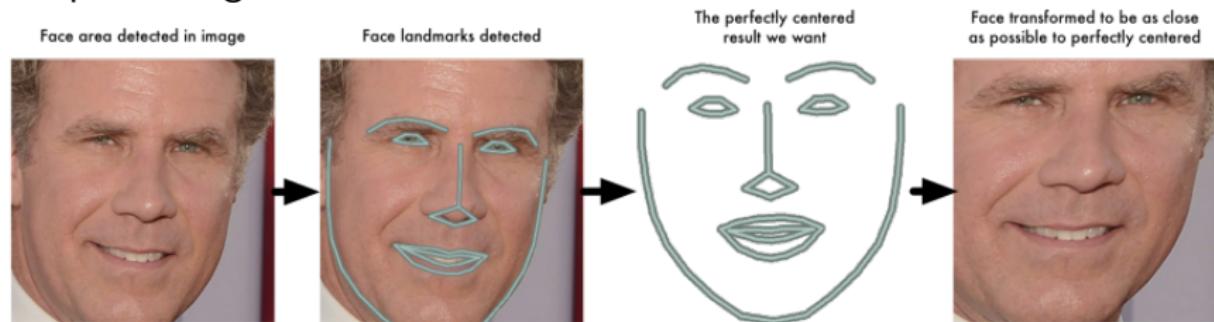
from Krizhevsky et al. 2012

Facial recognition

- Used to tag images in software, security

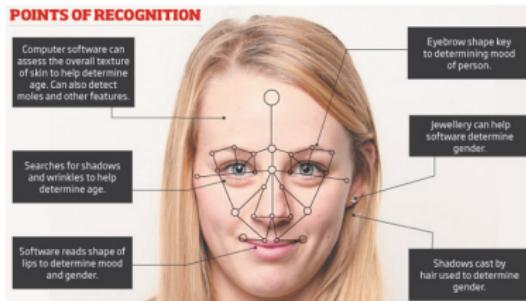
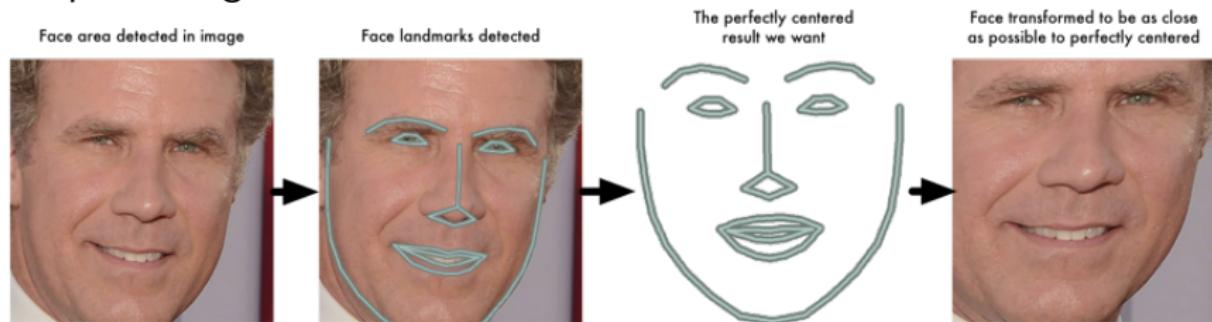
Facial recognition

- Used to tag images in software, security
- Preprocessing



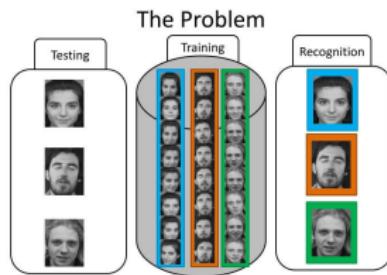
Facial recognition

- Used to tag images in software, security
- Preprocessing

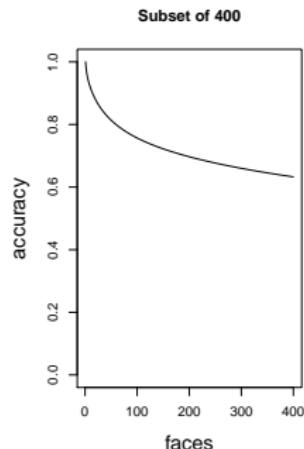
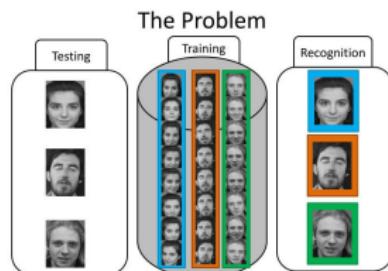


- Feature extraction

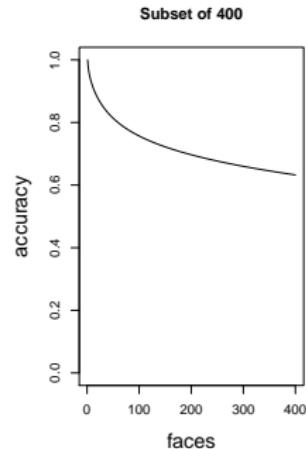
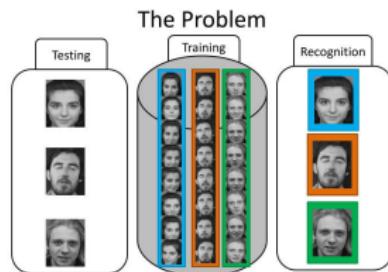
Accuracy vs. number of classes



Accuracy vs. number of classes



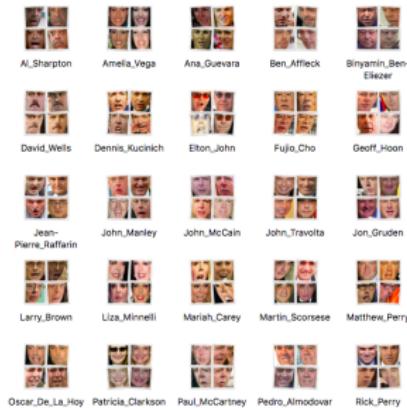
Accuracy vs. number of classes



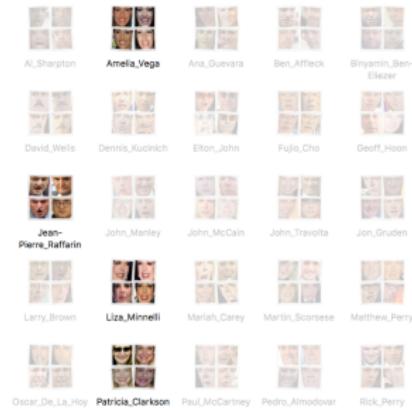
How does the accuracy scale with the number of classes (faces)?

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.

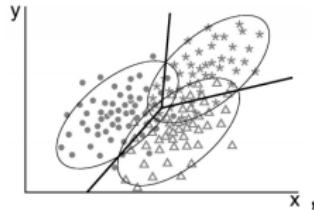
Can we analyze how error depends on k ?

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.

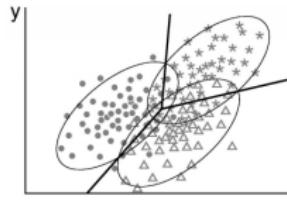


- Examples: LDA/QDA

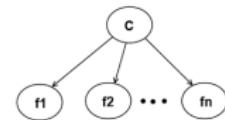
Key assumption: marginal classifier

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

- Examples: LDA/QDA

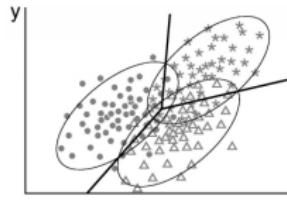


, naïve Bayes

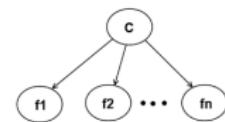


Key assumption: marginal classifier

- 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

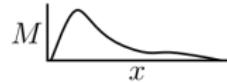


Definitions

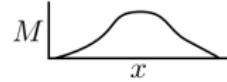
$\hat{F}_{y^{(i)}}$ is the empirical distribution obtained from the training data for label $y^{(i)}$.

Classification Rule

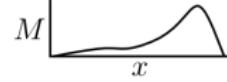
$$M_{y^{(1)}}(x) = \mathcal{M}(\hat{F}_{y^{(1)}})(x)$$



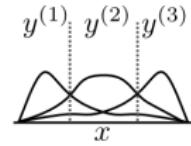
$$M_{y^{(2)}}(x) = \mathcal{M}(\hat{F}_{y^{(2)}})(x)$$



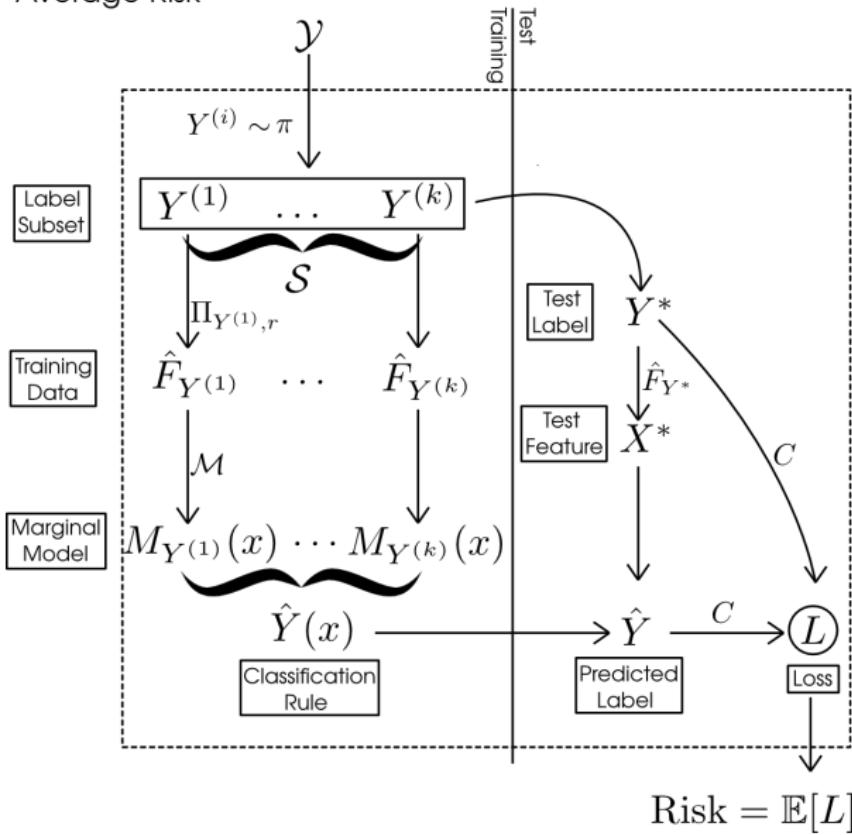
$$M_{y^{(3)}}(x) = \mathcal{M}(\hat{F}_{y^{(3)}})(x)$$



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



Average Risk



$$\text{Risk} = \mathbb{E}[L]$$

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...

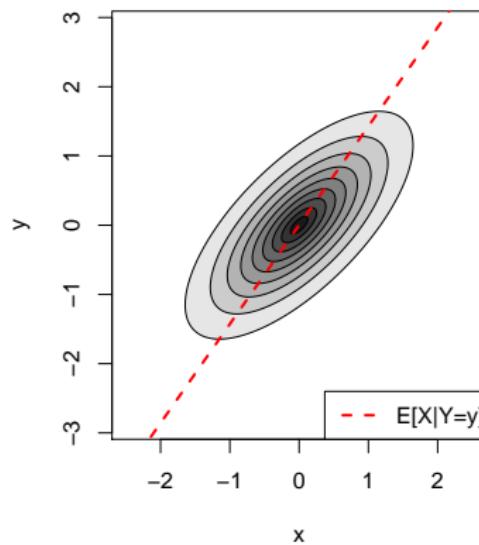
Toy example

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

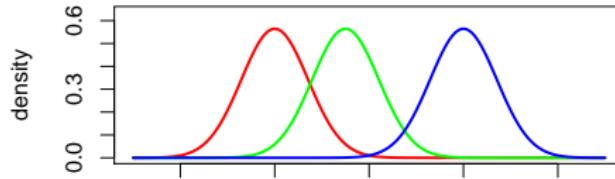
Toy example

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

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

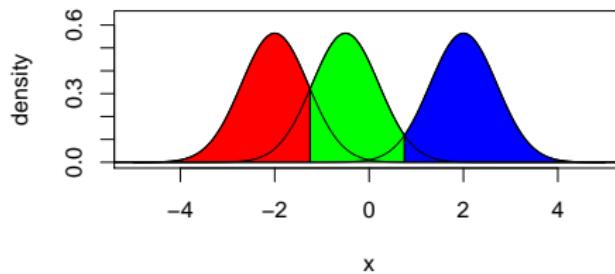


Toy example



- 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.

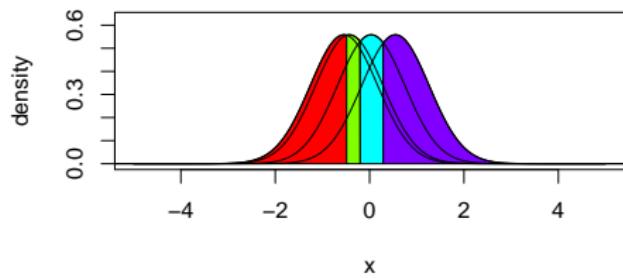
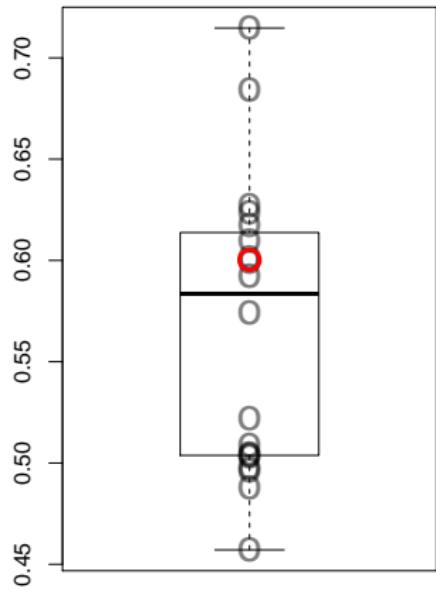
Toy example



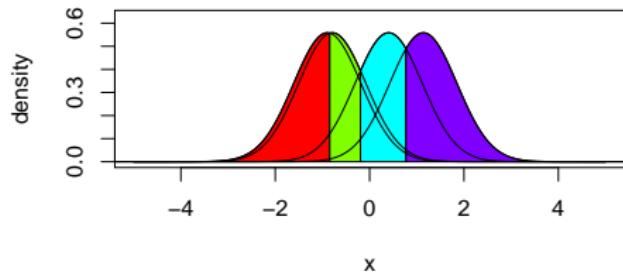
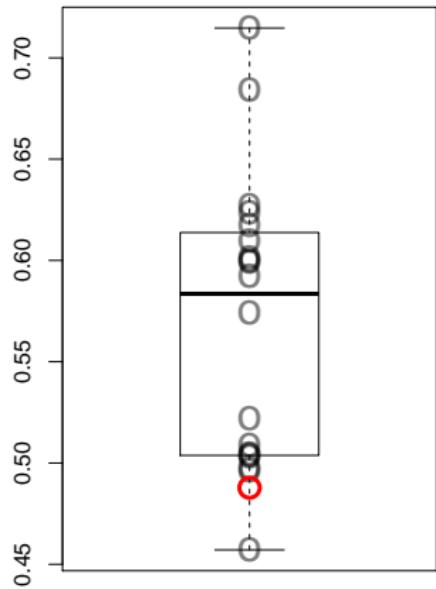
- The *Bayes Risk* is the expected test error of the Bayes rule,

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

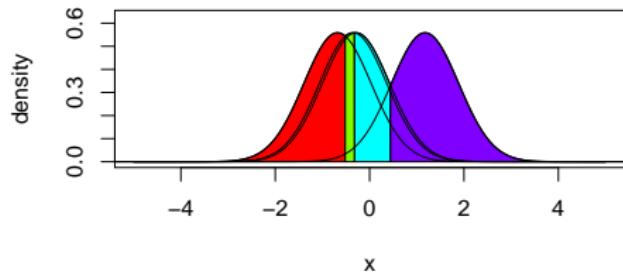
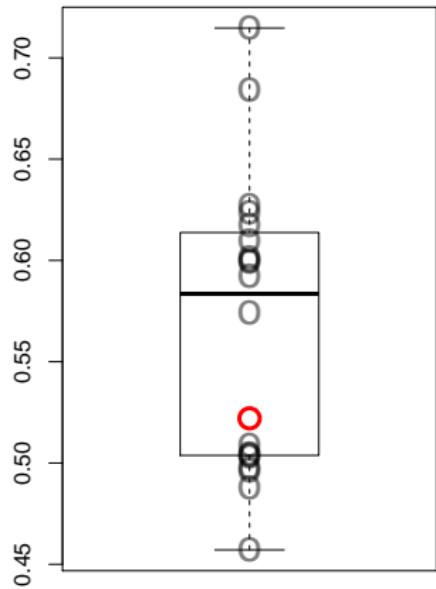
Toy example



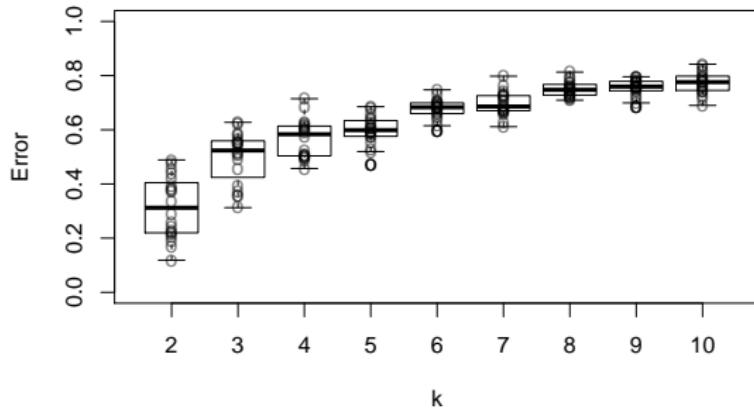
Toy example



Toy example

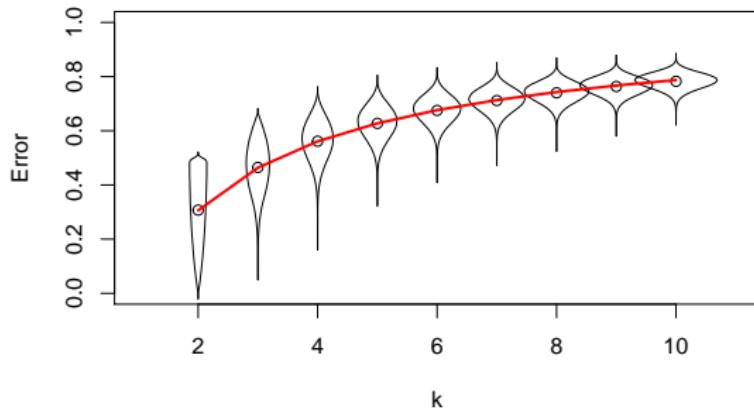


Toy example



Toy example

$\rho = 0.7$

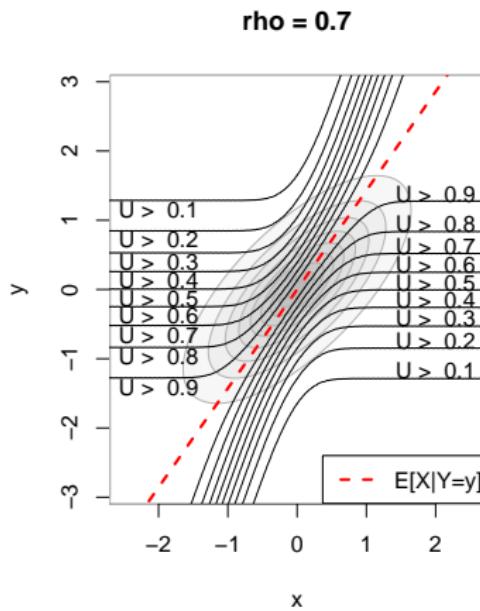


Defining the U -function

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



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

Defining $\bar{D}(u)$

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

Defining $\bar{D}(u)$

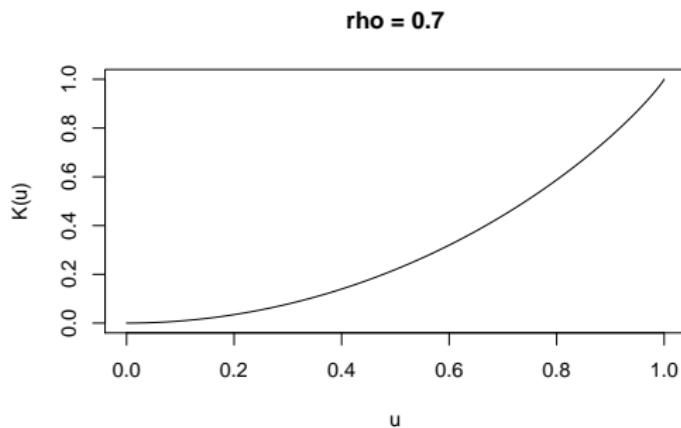
- Define random variable as $U_Y(X)$ for (Y, X) drawn from the joint distribution.
- $\bar{D}(u)$ is the cumulative distribution function of U ,

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

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 ,

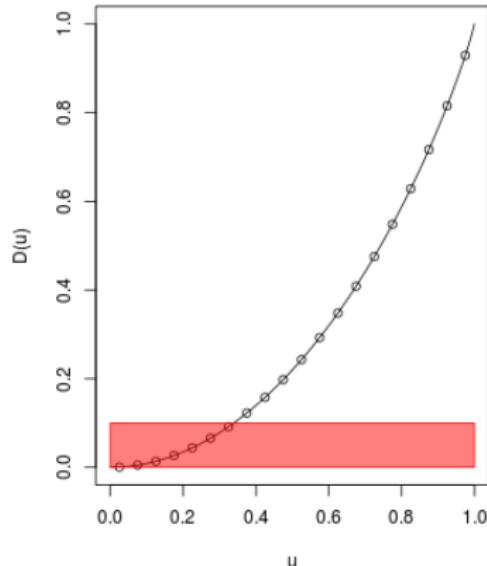
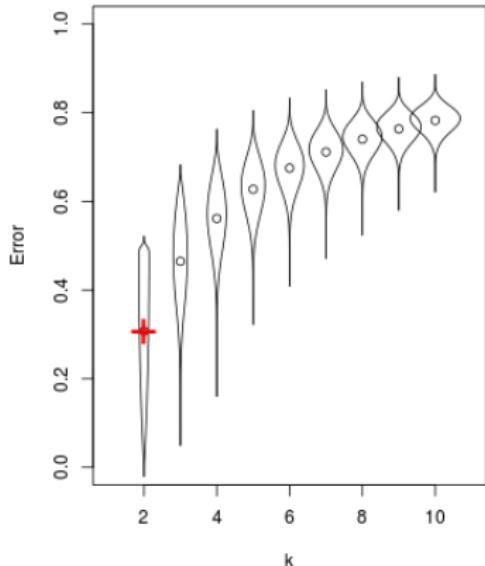
$$\bar{D}(u) = \Pr[U_Y(X) \leq u].$$



Computing average risk

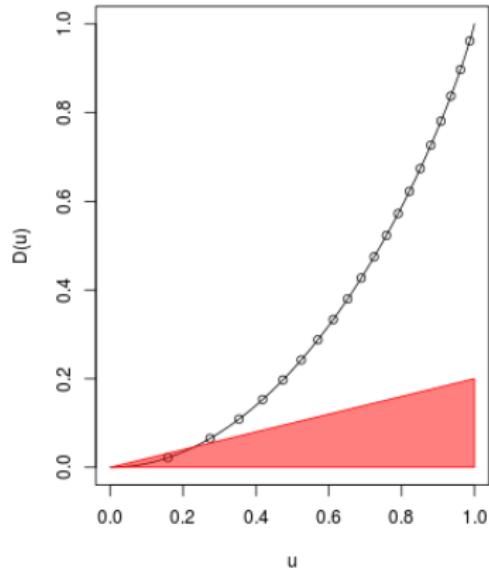
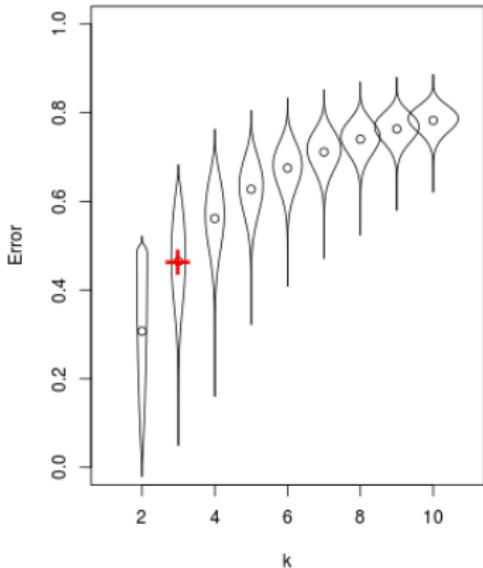
$$\text{AvRisk}_k = (k - 1) \int \bar{D}(u) u^{k-2} du.$$

$(k = 2)$



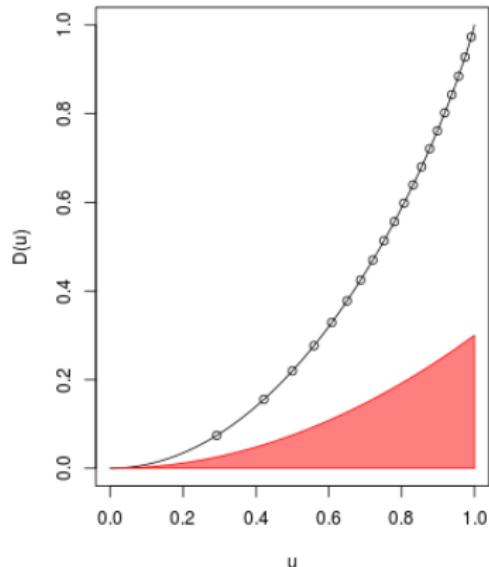
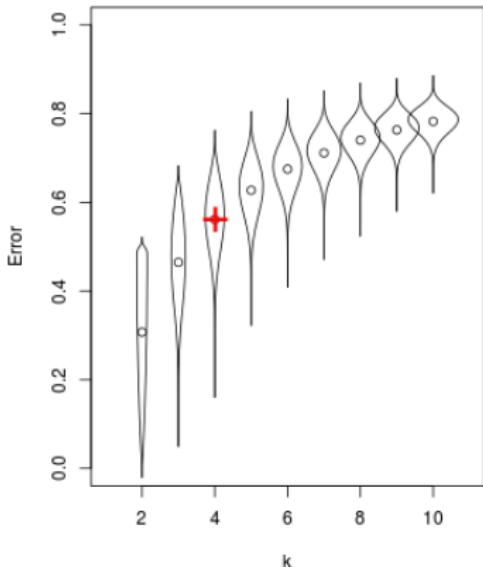
Computing average risk

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



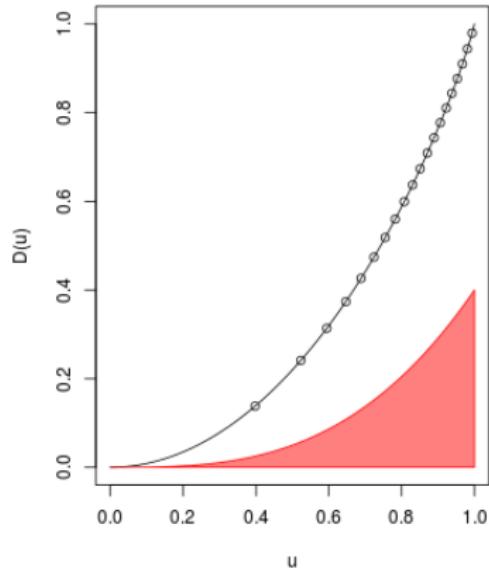
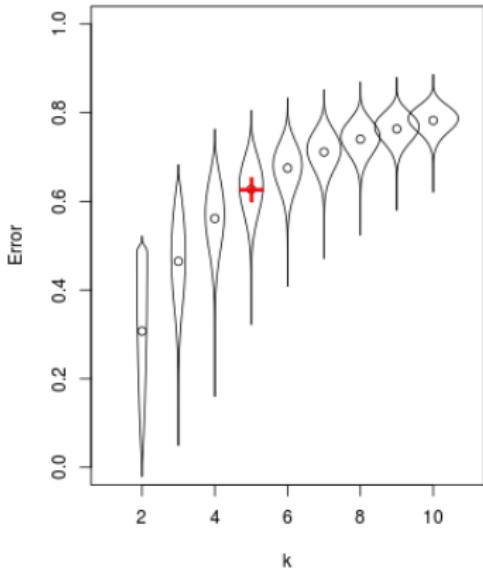
Computing average risk

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



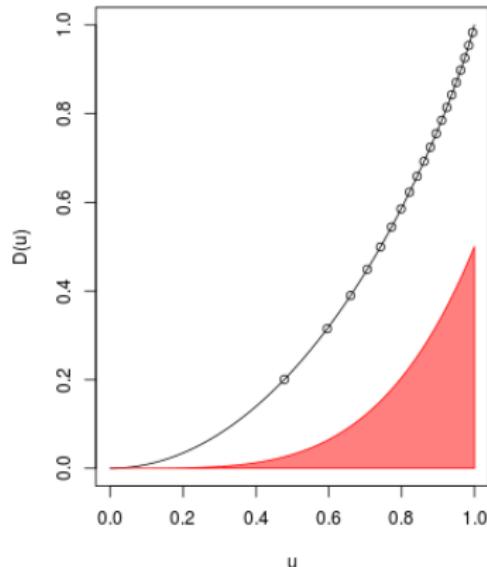
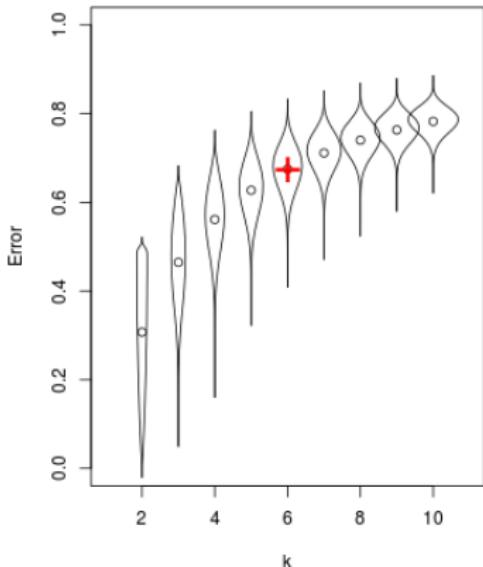
Computing average risk

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



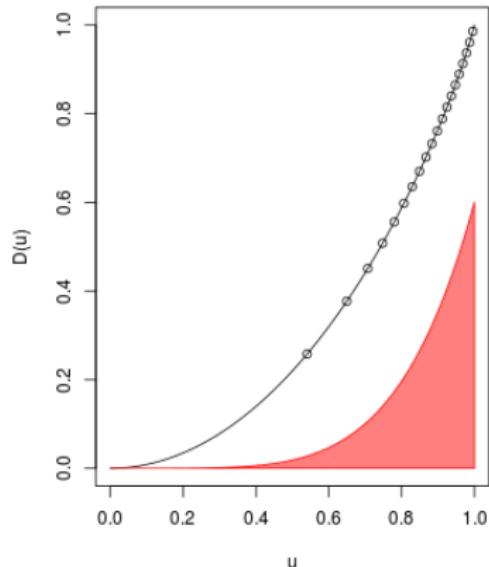
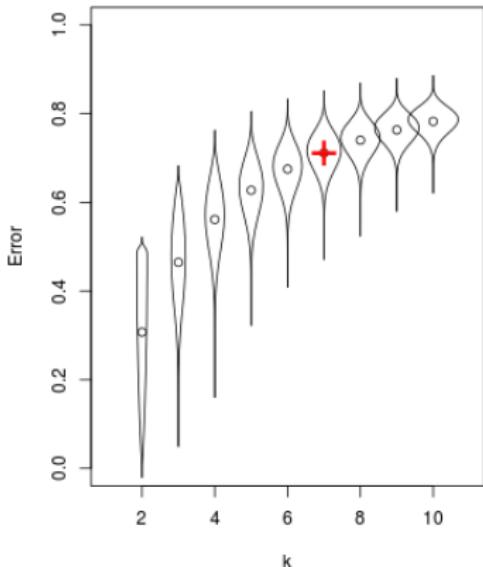
Computing average risk

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



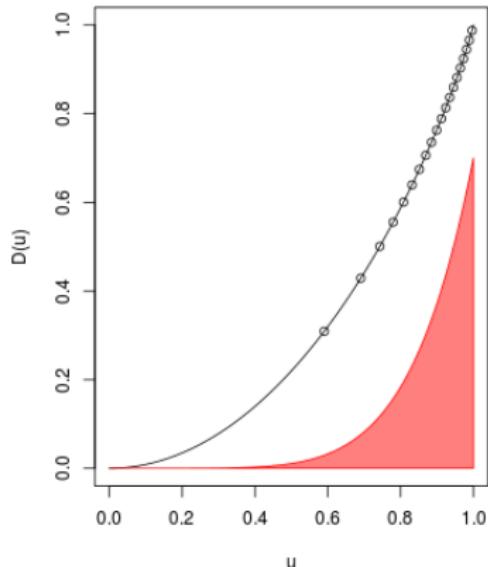
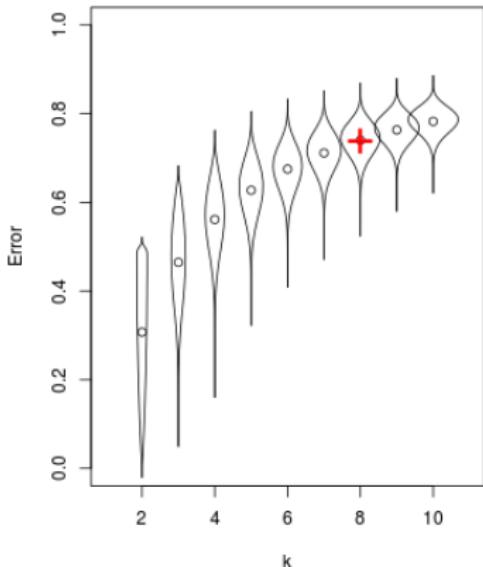
Computing average risk

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



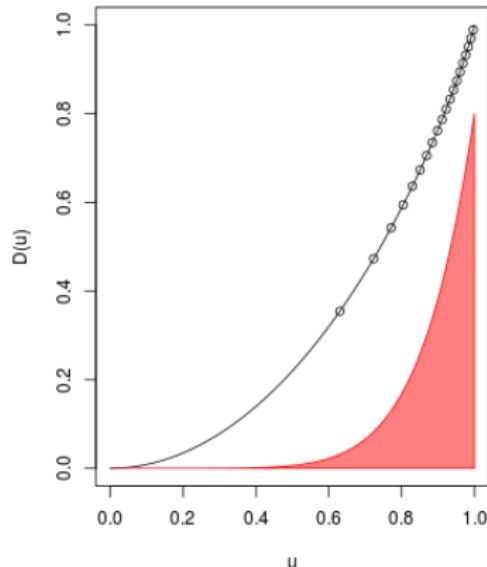
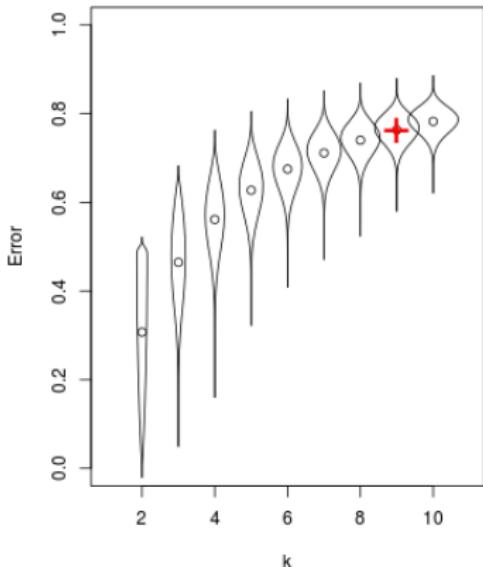
Computing average risk

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



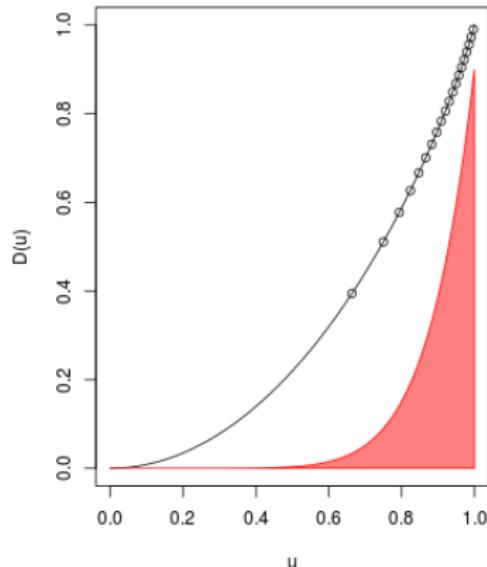
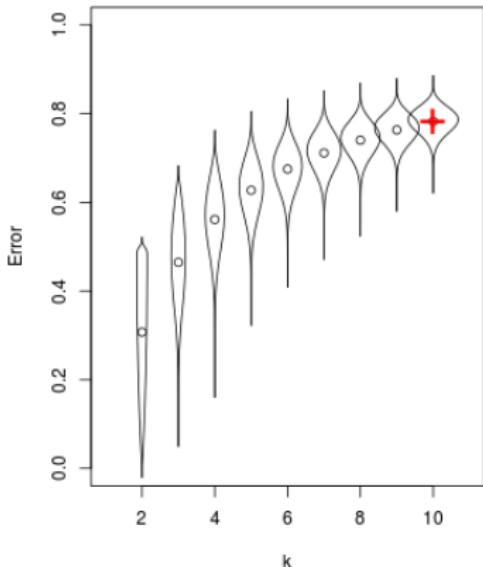
Computing average risk

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



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 .

$k = 4$	
$k = 3$	A grid of four rows and two columns. The first three rows each contain two small square images of faces, separated by the word "vs". The fourth row contains three such pairs.
$k = 2$	A grid of three rows and two columns. Each row contains two small square images of faces, separated by the word "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, $[x - t_\ell]_+$

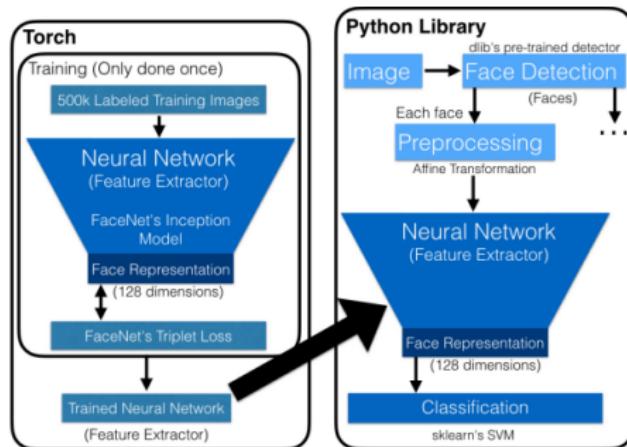
Optional constraint: assume β is *non-negative*. In the case of linear splines, this results in a *convex* fit.

Section 4

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

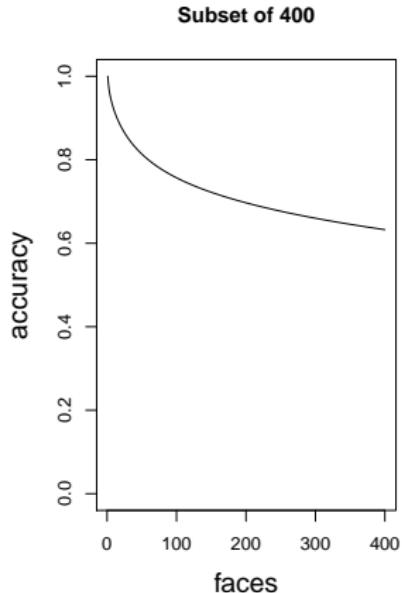


Facial recognition example

- 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!

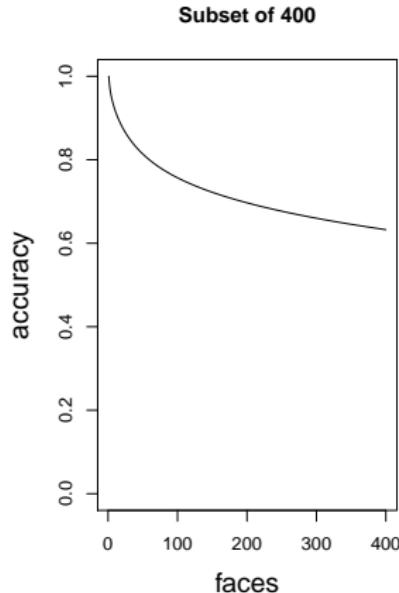
Facial recognition example

- 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

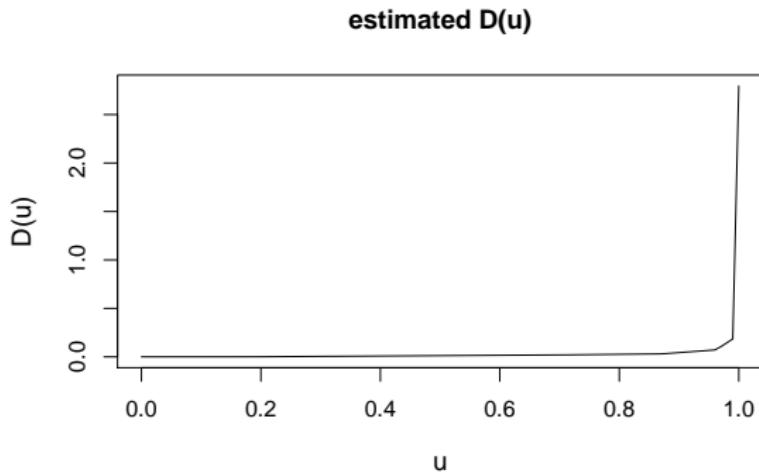
- Let us first subsample 400 faces (out of 1672)
- Randomly choose 1 face as training and 1 as test for each person
- Use 1-nearest neighbor.
 - NOTE: 1-NN with 1 example/class is equivalent to LDA with $\Sigma = I$: this fits marginal classifier assumption!



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

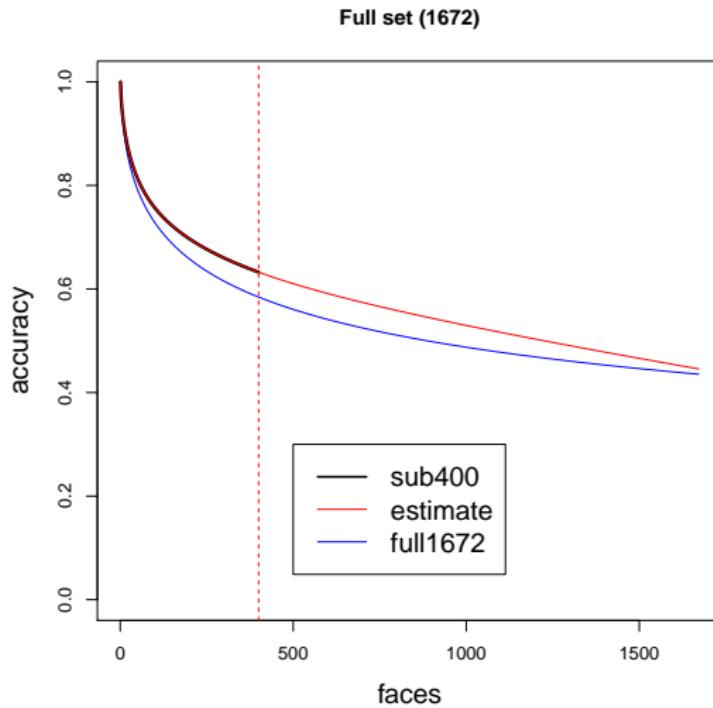
Estimated $\bar{D}(u)$

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

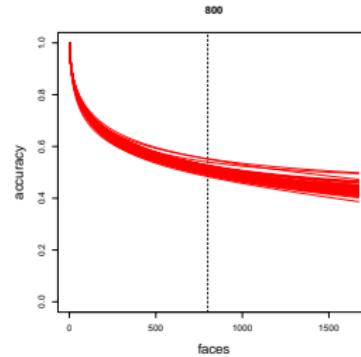
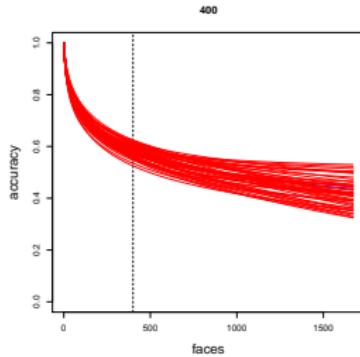
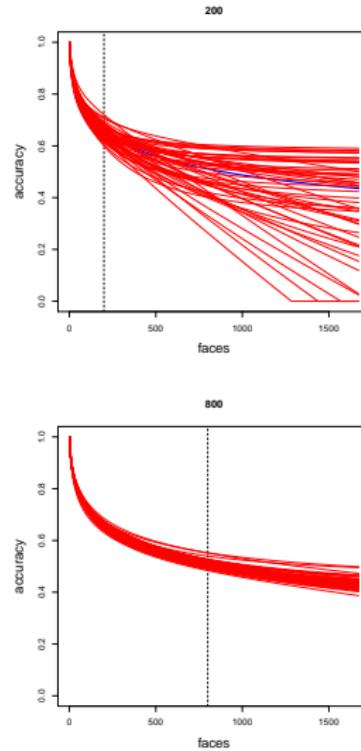
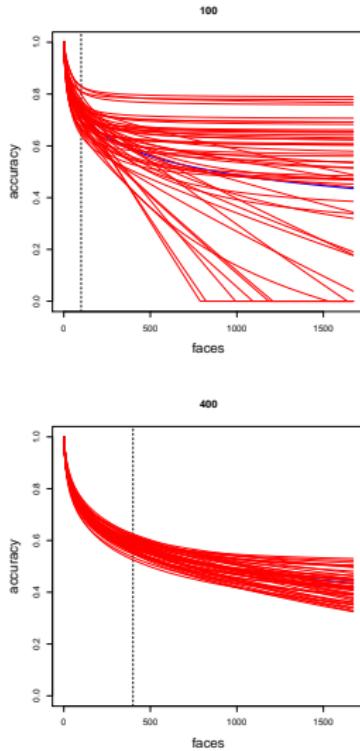


Estimated risk

Compare to test risk at $K = 1672$

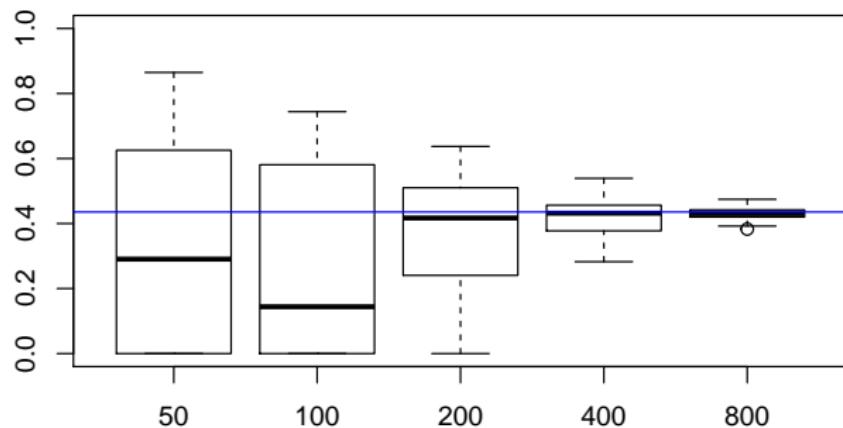


Estimated risk: more experiments



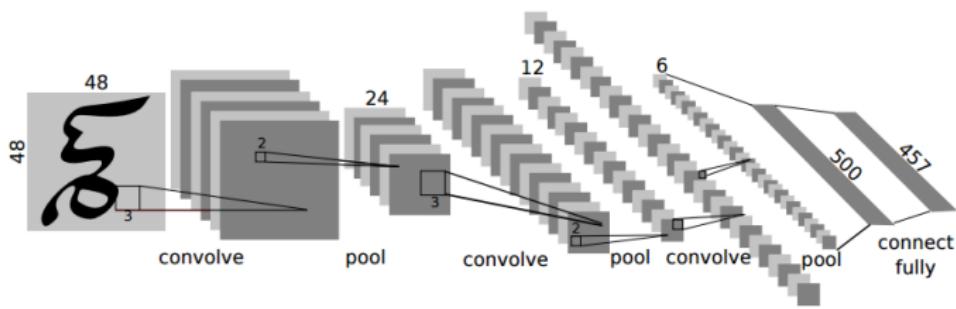
Estimated risk: more experiments

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 5

Acknowledgements







Section 6

The end