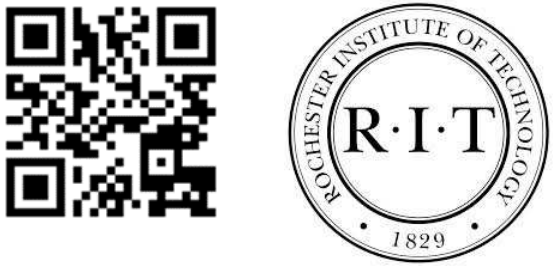
Audio Analysis of Speaker Accent using Statistical Machine Learning Methods

Fokoué Ernest
Professor of Statistics and Data Science
School of Mathematical Sciences



Lecture Delivered at the
Statistical and Applied Mathematical Sciences Institute (SAMSI)
Undergraduate Workshop 2020

February 24, 2020

This talk is dedicated to my daughter Ellie!

Happy Sweet 16^{th} *Birthday to you!*
My Precious Daughter Ellie!
With lots of love and a big daddy hug!

Acknowledgments and Grateful Thanks

Grateful thanks and sincere gratitude to SAMSI, for kindly inviting me and granting me the golden opportunity to present at this workshop. Special thanks to

- *Prof. Dr. David Banks*
- *All the SAMSI Staff*
- *All the SAMSI Postdocs*
- *All the students attending*
- *All the SAMSI Faculty Fellows*

Above all, I give thanks and praise to God Almighty for all the graces bestowed upon me throughout my whole life!!

Gratitude is the greatest force in the universe!

GRATITUDE

Within the characters for thanks
and feelings are embedded the
symbols for heart and speech.
From the heart, with feeling,
I express my gratitude.

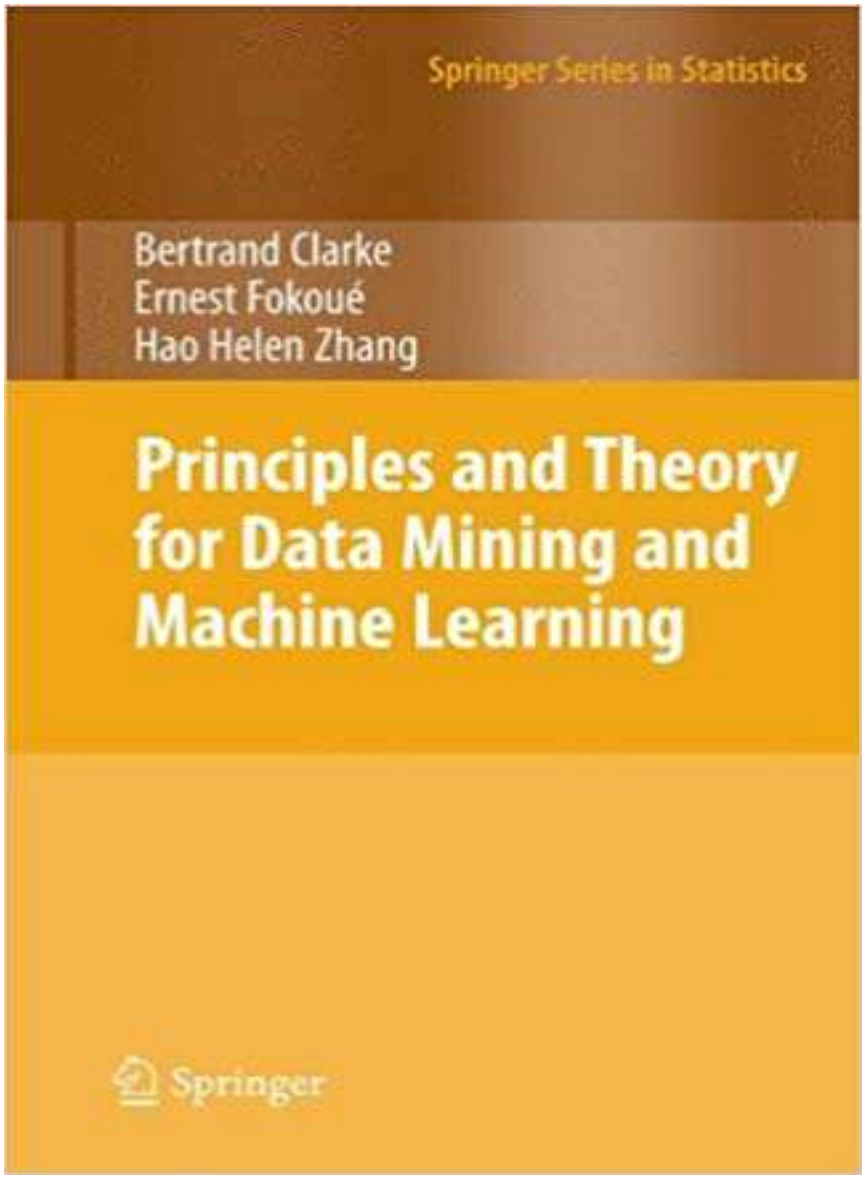
Grateful Thanks to Rochester Institute of Technology

Grateful Thanks to Duke University for my Sabbatical


THANK
YOU

Discovering Statistical Machine Learning

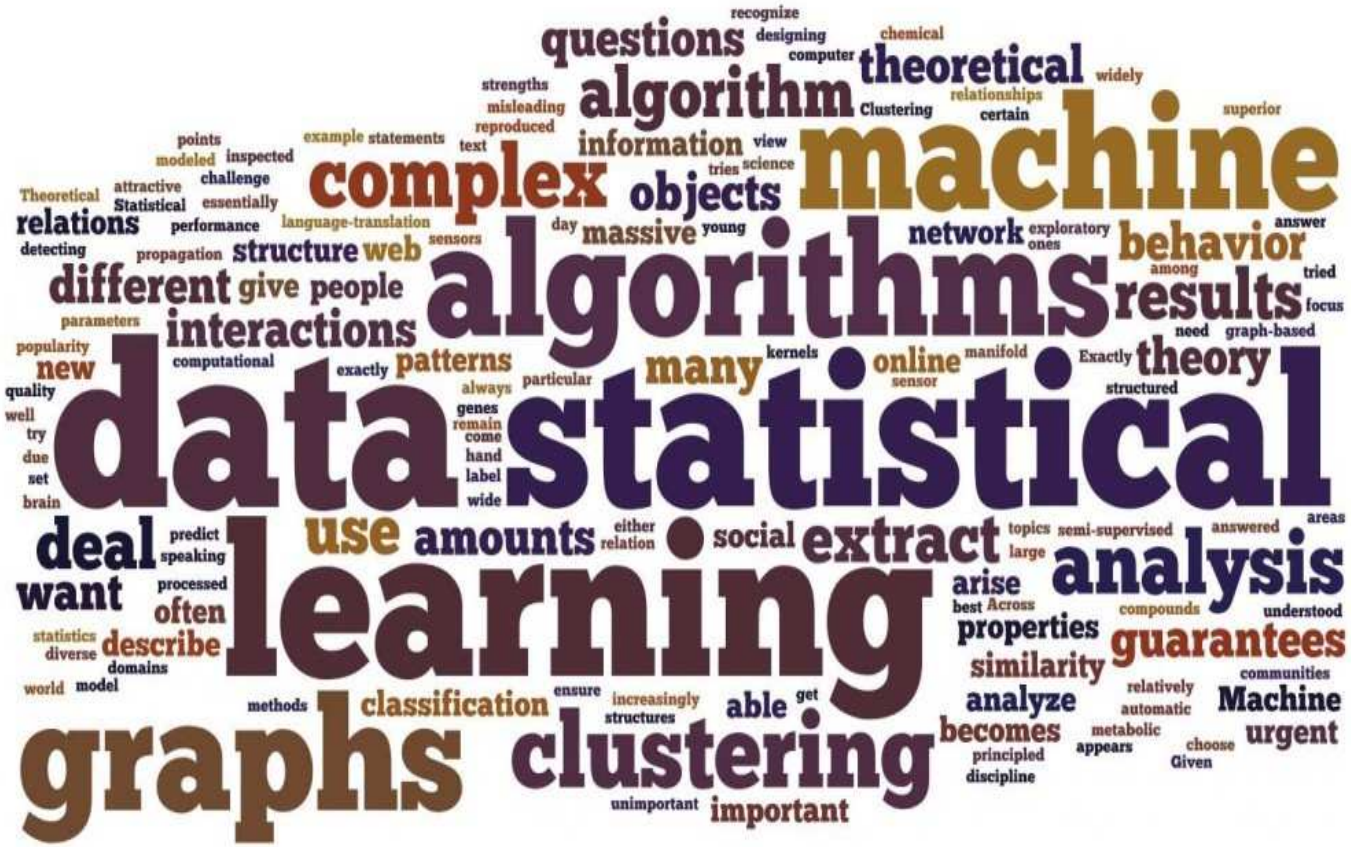
1. *What really is statistical machine learning (SML)?*
2. *Why should you care? What's interesting in it?*
3. *What on earth is accent recognition?*
4. *How can one applied SML to accent recognition?*
5. *How can you deepen your knowledge of SML?*
6. *What are the current challenges in SML?*
7. *What are the emerging trends in SML?*
8. *How can you uniquely contribute to SML?*



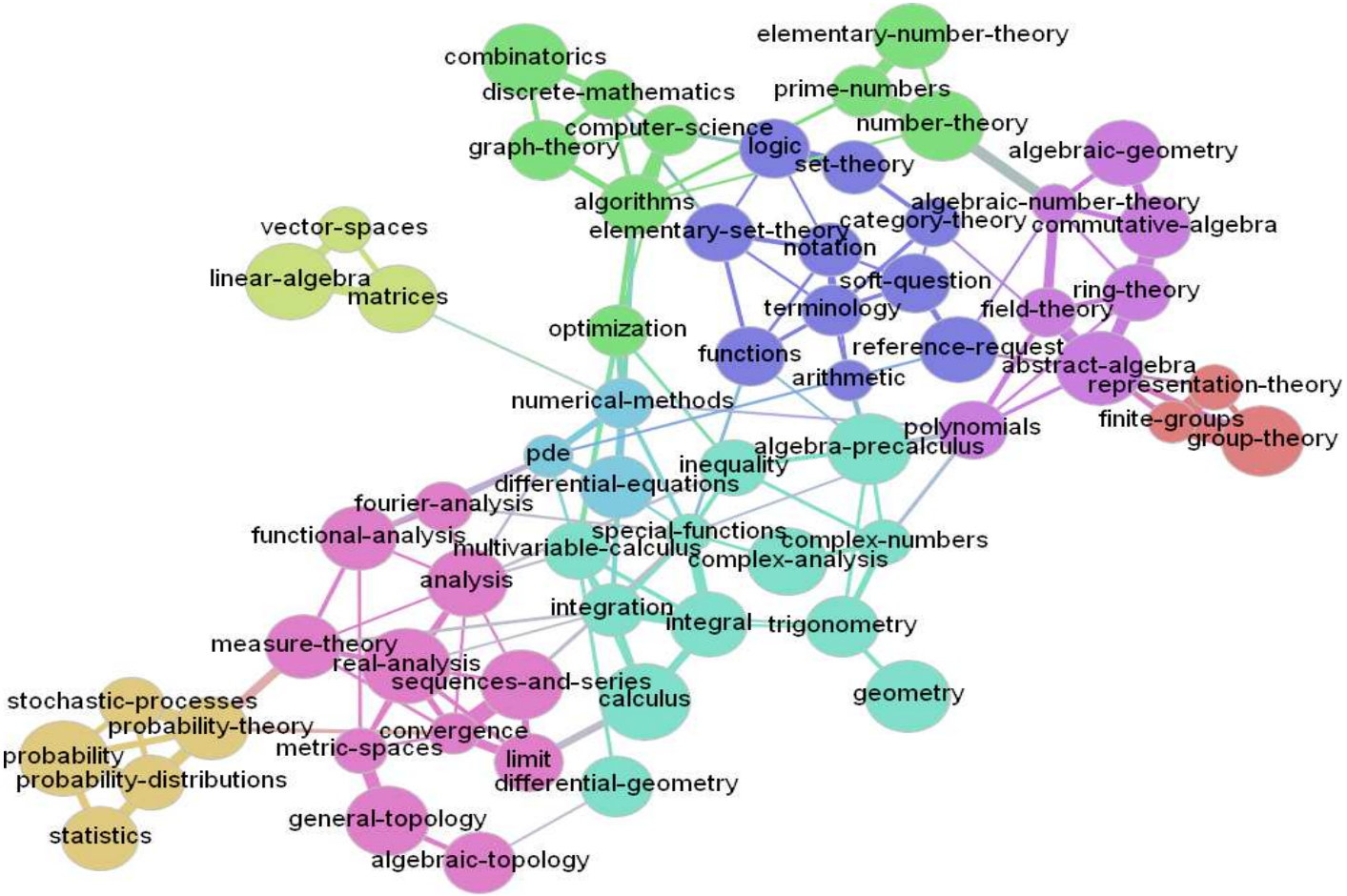
Clarke, B. and Fokoué, E. and Zhang, H. (2009). Principles and Theory for Data Mining and Machine Learning. Springer Verlag, New York, (ISBN: 978-0-387-98134-5), (2009)



How will you uniquely change the world?



How will you uniquely change the world?

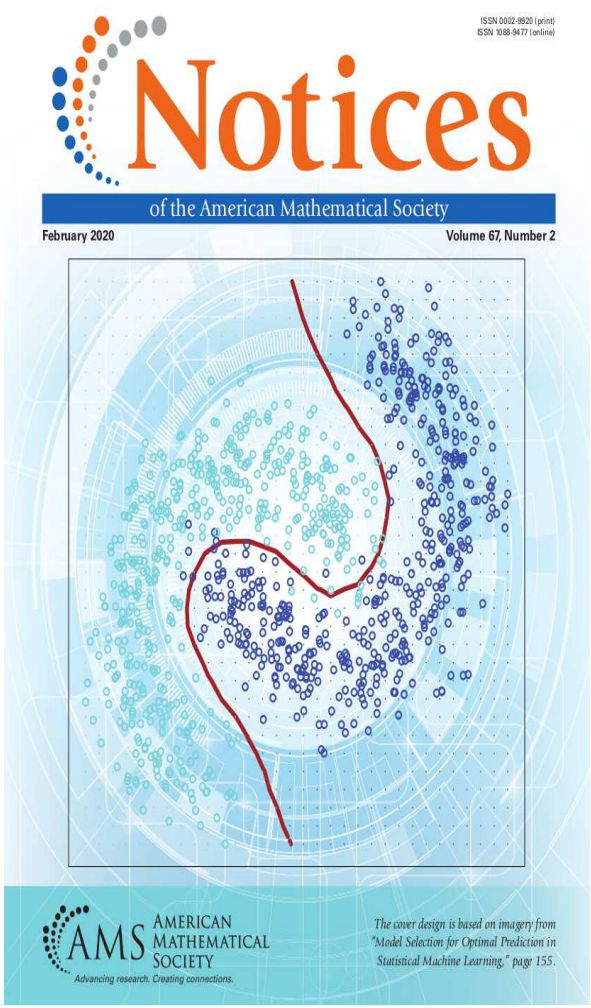


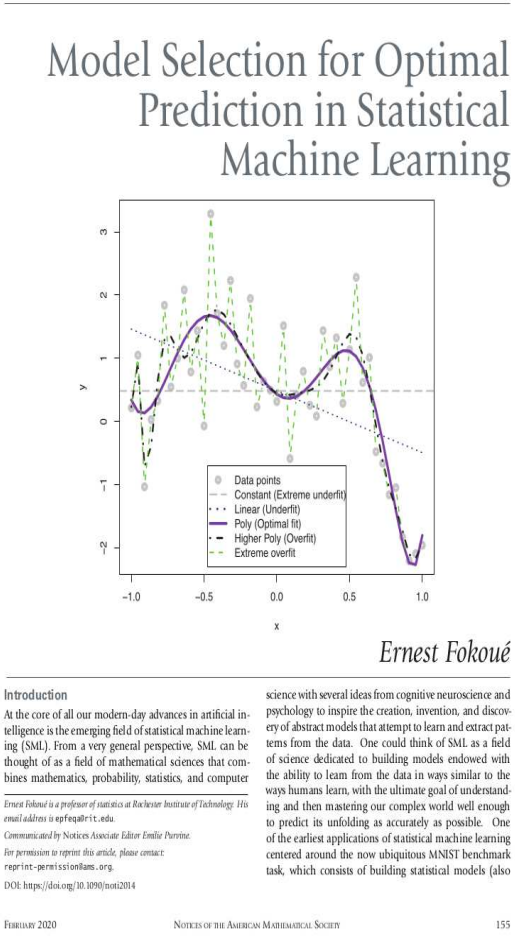
Roadmap for Statistical Machine Learning

- *Applications:* *Sharpen your intuition and your commonsense by questioning things, reading about interesting open applied problems, and attempt to solve as many problems as possible*
- *Methodology:* *Read and learn about the fundamental of statistical estimation and inference, get acquainted with the most commonly used methods and techniques, and consistently ask yourself and others what the natural extensions of the techniques could be.*
- *Computation:* *Learn and master at least two programming languages. I strongly recommend getting acquainted with R*

  http://www.r-project.org
- *Theory:* *"Nothing is more practical than a good theory" (Vladimir N. Vapnik). When it comes to data mining and machine learning and predictive analytics, those who truly understand the inner workings of algorithms and methods always solve problems better.*

Notices of the American Mathematical Society





Model Selection for Optimal Prediction in Statistical Machine Learning

Ernest Fokoué

Introduction

At the core of all our modern-day advances in artificial intelligence is the emerging field of statistical machine learning (SML). From a very general perspective, SML can be thought of as a field of mathematical sciences that combines mathematics, probability, statistics, and computer science with several ideas from cognitive neuroscience and psychology to inspire the creation, invention, and discovery of abstract models that attempt to learn and extract patterns from the data. One could think of SML as a field of science dedicated to building models endowed with the ability to learn from the data in ways similar to the ways humans learn, with the ultimate goal of understanding and then mastering our complex world well enough to predict its unfolding as accurately as possible. One of the earliest applications of statistical machine learning centered around the now ubiquitous MNIST benchmark task, which consists of building statistical models (also

Ernest Fokoué is a professor of statistics at Rochester Institute of Technology. His email address is epfeqa@rit.edu.

Communicated by Notices Associate Editor Emilie Purvine.
For permission to reprint this article, please contact: reprint-permission@ams.org.
DOI: https://doi.org/10.1090/noti2014

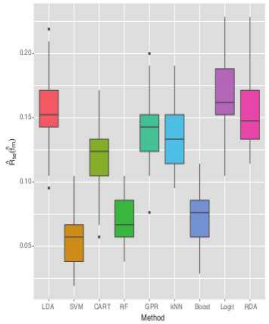


Figure 5. Predictive performances on the ionosphere data.

machines endowed with an inherent capability to predict accurately and precisely. In this paper, we have explored the niceties and subtleties of such a goal and have demonstrated that it requires a hefty dose of care and caution and definitely calls upon a solid theoretical understanding of learnability along with a lot of artlike practical common sense. Anyone who has done practical data science knows beyond a shadow of a doubt that data has a mind of its own, and tends to resist the temptation to seek a holy grail or a unified field, or any paradigm that works perfectly all the time. Practical data science almost always forces the practitioner to solve the problem at hand as thoroughly and as idiosyncratically as possible rather than seek a one-size-fits-all method that works well everywhere. At the heart of what we suggested throughout this paper is the theoretical result known as the no free lunch theorem, which reveals, both implicitly and explicitly, that the theoretical bounds studied extensively by experts do not really help much when it comes to practically selecting the optimal predictive model. Optimal predictive modelling is, and may always be, both a science and an art, requiring both mathematical and statistical rigor along with practical computational common sense.

References

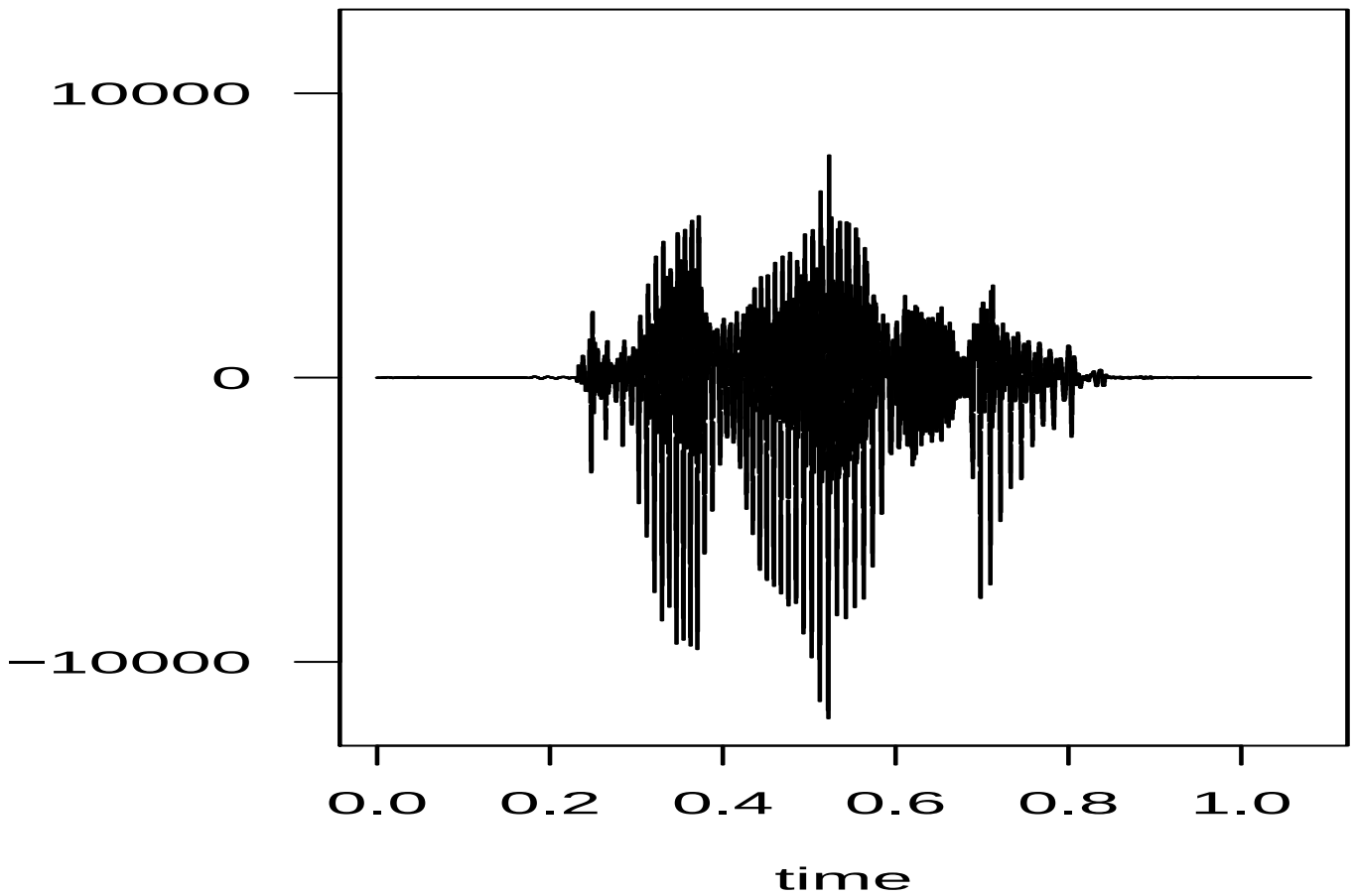
[1] Akaike H. Information theory and an extension of the maximum likelihood principle. In: *Selected Papers of Hirotugu Akaike*. Springer New York, New York, NY; 1973:199–213. MR0483125
[2] Barbieri M and Berger JO. Optimal predictive model selection, *Ann. Statist.*, (32):870–897, 2004. MR2065192
[3] Breiman L. Bagging predictors, *Machine Learning*, (24):123–140, 1996.
[4] Breiman L. Random forests, *Machine Learning*, (45):5–32, 2001. MR3874153
[5] Clarke B, Fokoue E, Zhang H. *Principles and Theory for Data Mining and Machine Learning*, first edition, Springer Texts in Statistics, Springer-Verlag, 2009. MR2839778
[6] Csató L, Fokoué E, Opper M, Schottky B, Winther O. Efficient approaches to Gaussian process classification. In: Leen TK, Solla SA, Müller K-R, eds. *Advances in Neural Information Processing Systems*, number 12 of 12. MIT Press; 2000.
[7] Devroye L, Györfi L, Lugosi G. *A Probabilistic Theory of Pattern Recognition*, Stochastic Modelling and Applied Probability, Springer New York, 1997.
[8] Domingos P. A unified bias-variance decomposition for zero-one and squared loss, *AAAI/IAAI*, AAAI Press, 564–569, 2000.
[9] Fokoué E. Estimation of atom prevalence for optimal prediction. In: *Prediction and Discovery*, Contemporary Mathematics, vol. 443. American Mathematical Society; 2007:103–129. MR2433288
[10] Hoerl A and Kennard R. Ridge regression: biased estimation for non-orthogonal problems, *Technometrics*, (12):55–67, 1970.
[11] Hoeting JA, Madigan D, Raftery AE, Volinsky CT. Bayesian model averaging: A tutorial, *Statist. Sci.*, 14(4):382–417, 1999. MR1765176
[12] Kohavi R and Wolpert DH. Bias plus variance decomposition for zero-one loss functions. In: *Machine Learning, Proceedings of the Thirteenth International Conference (ICML '96), Bari, Italy, July 3–6, 1996*. 1996:275–283.
[13] Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain, *Psychological Review*, 65–386, 1958. MR0122606
[14] Schapire RE and Freund Y. *Boosting: Foundations and Algorithms*, The MIT Press, 2012. MR2920188
[15] Schwarz G. Estimating the dimension of a model, *The Ann. Statist.*, (6):461–464, 1978. MR0468014
[16] Stone M. An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion, *J. Roy. Statist. Soc. Ser. B (Methodological)*, 39(1):44–47, 1977. MR501454
[17] Stone M. Cross-validatory choice and assessment of statistical predictions, *J. Roy. Statist. Soc. Ser. B (Methodological)*, 111–147, 1974. MR356377
[18] Tibshirani R. Regression shrinkage and selection via the lasso, *J. Roy. Statist. Soc. Ser. B (Methodological)*, 58(1):267–288, 1996. MR1379242
[19] Vapnik VN. *The Nature of Statistical Learning Theory*, Springer, 2000. MR1719582

Ernest Fokoué

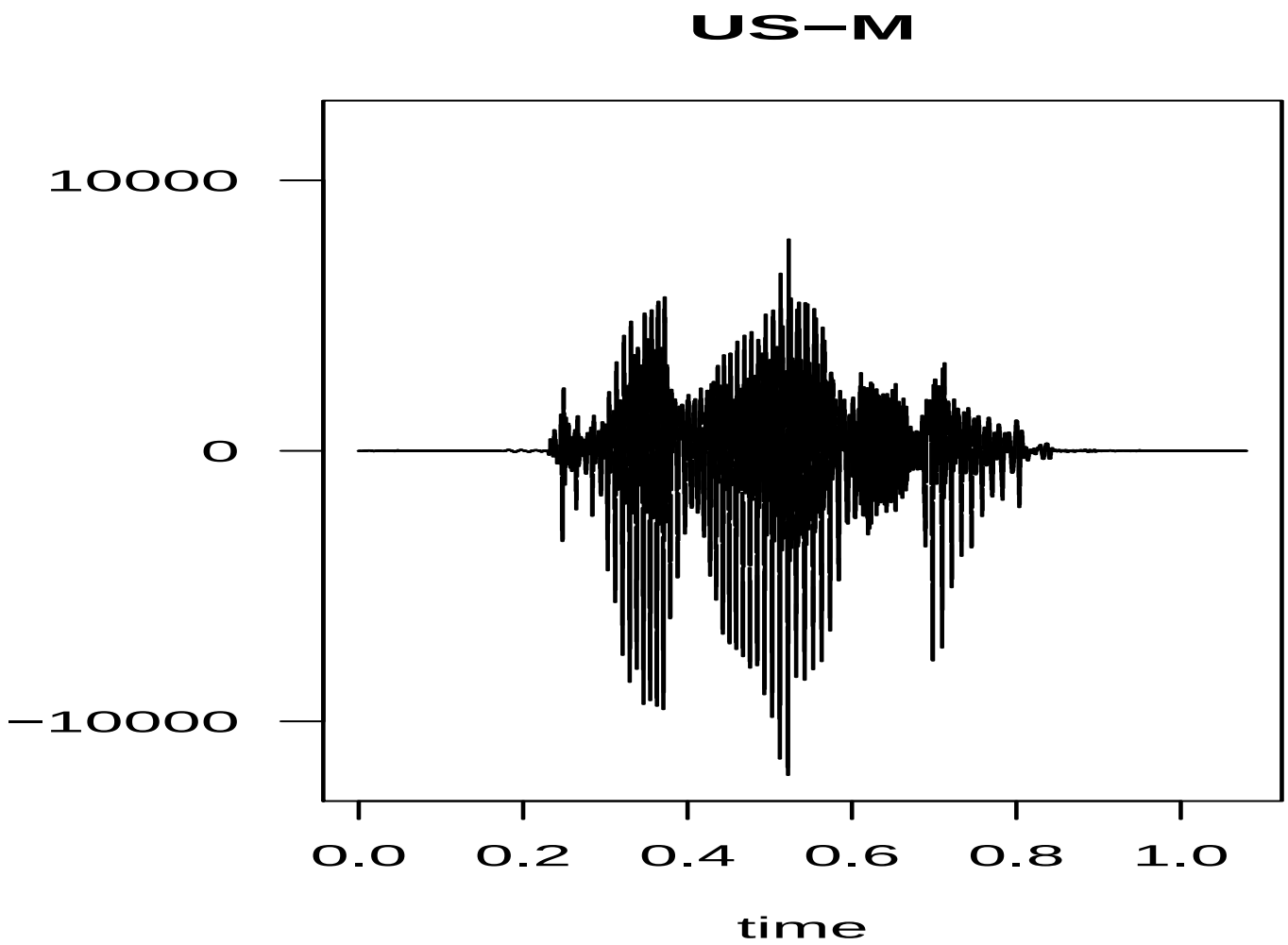
Credits
All figures are courtesy of the author.
Author photo is courtesy of Rick Scoggins.

I am infinitely grateful to God for the blessing of publishing a featured paper in the Notices of the American Mathematical Society. To God be the Glory! Amen! Alleluia!

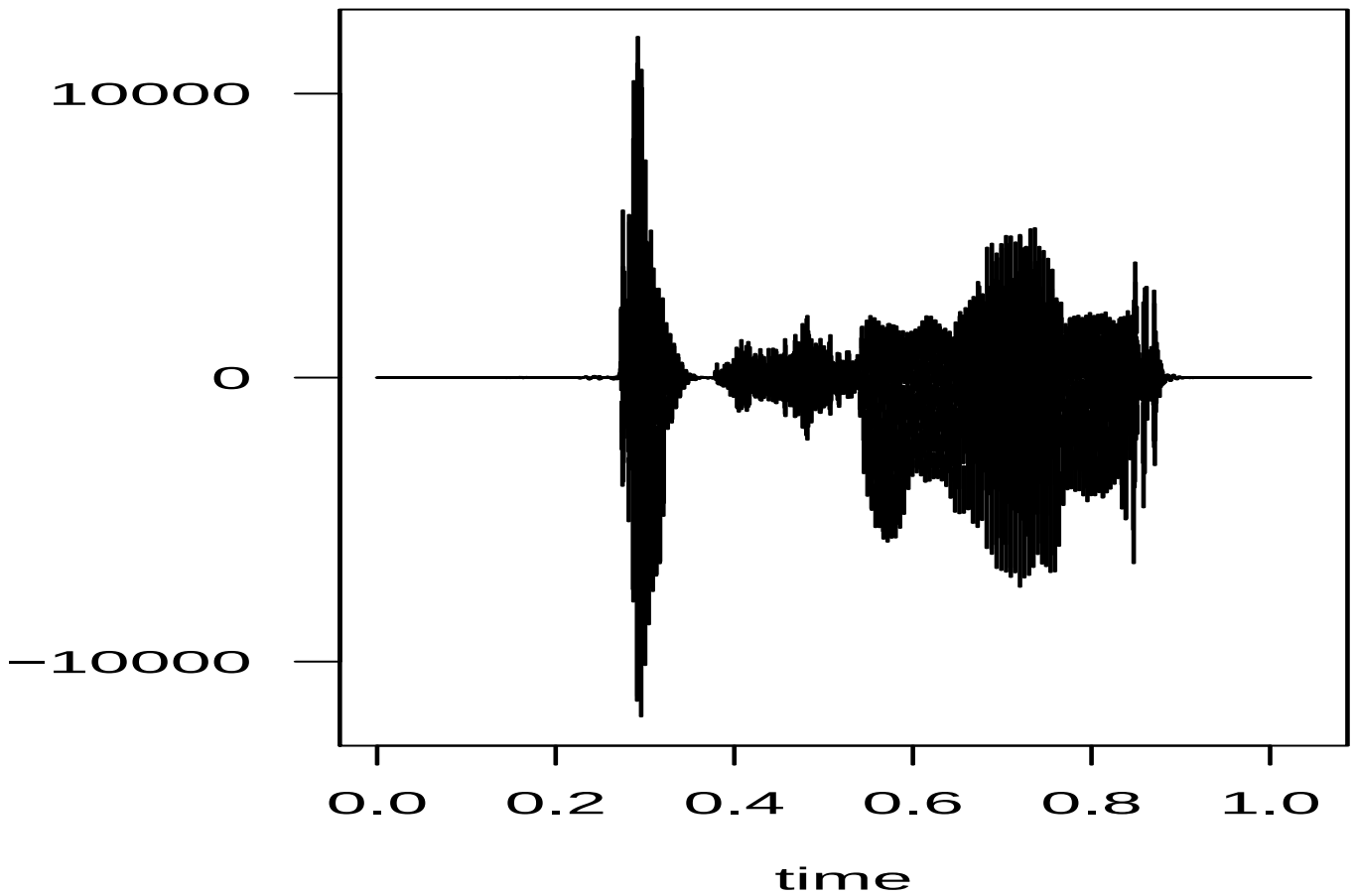
Do you detect an accent? If so, which one is it?



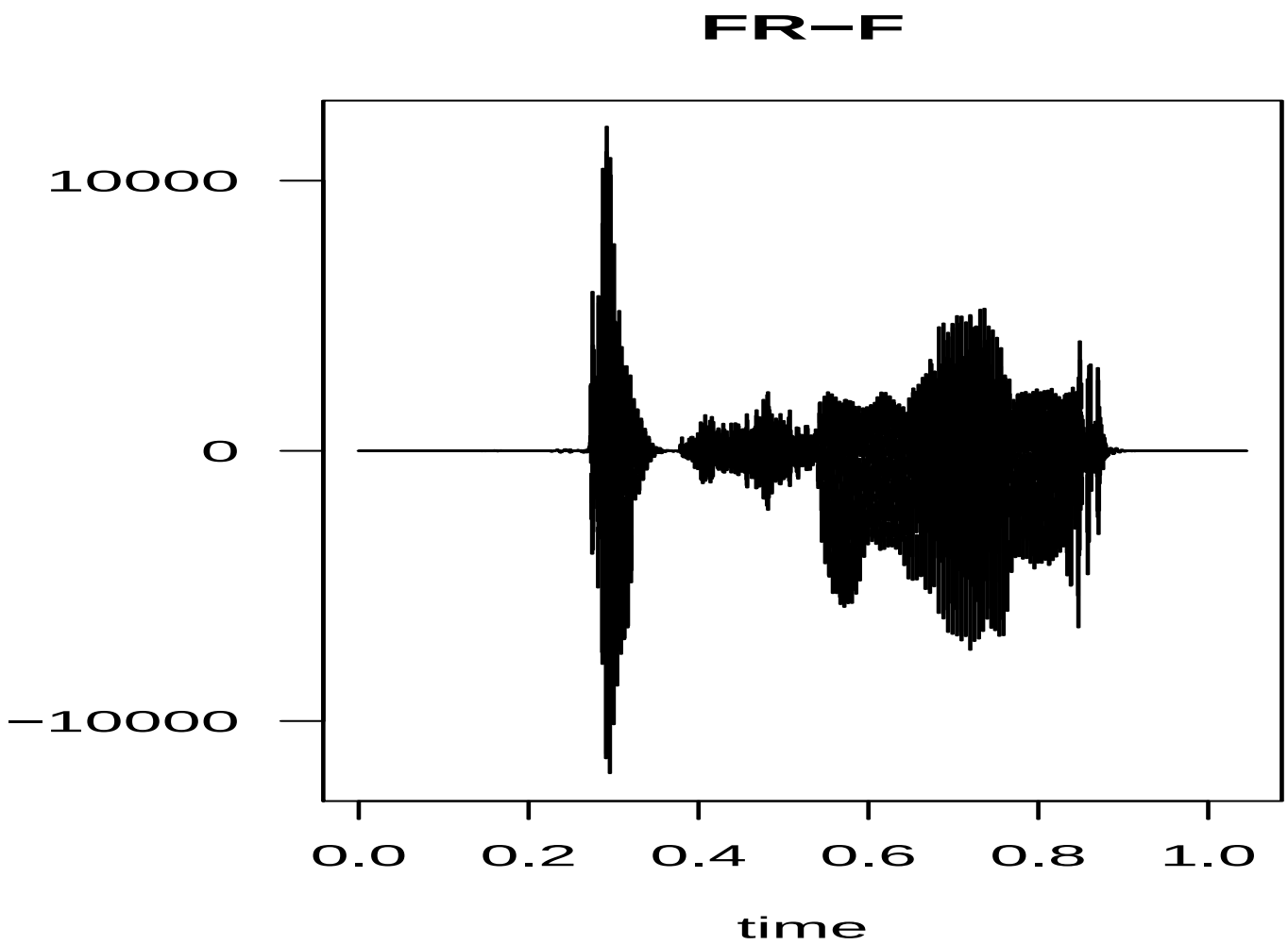
Answer: Male person from the United States



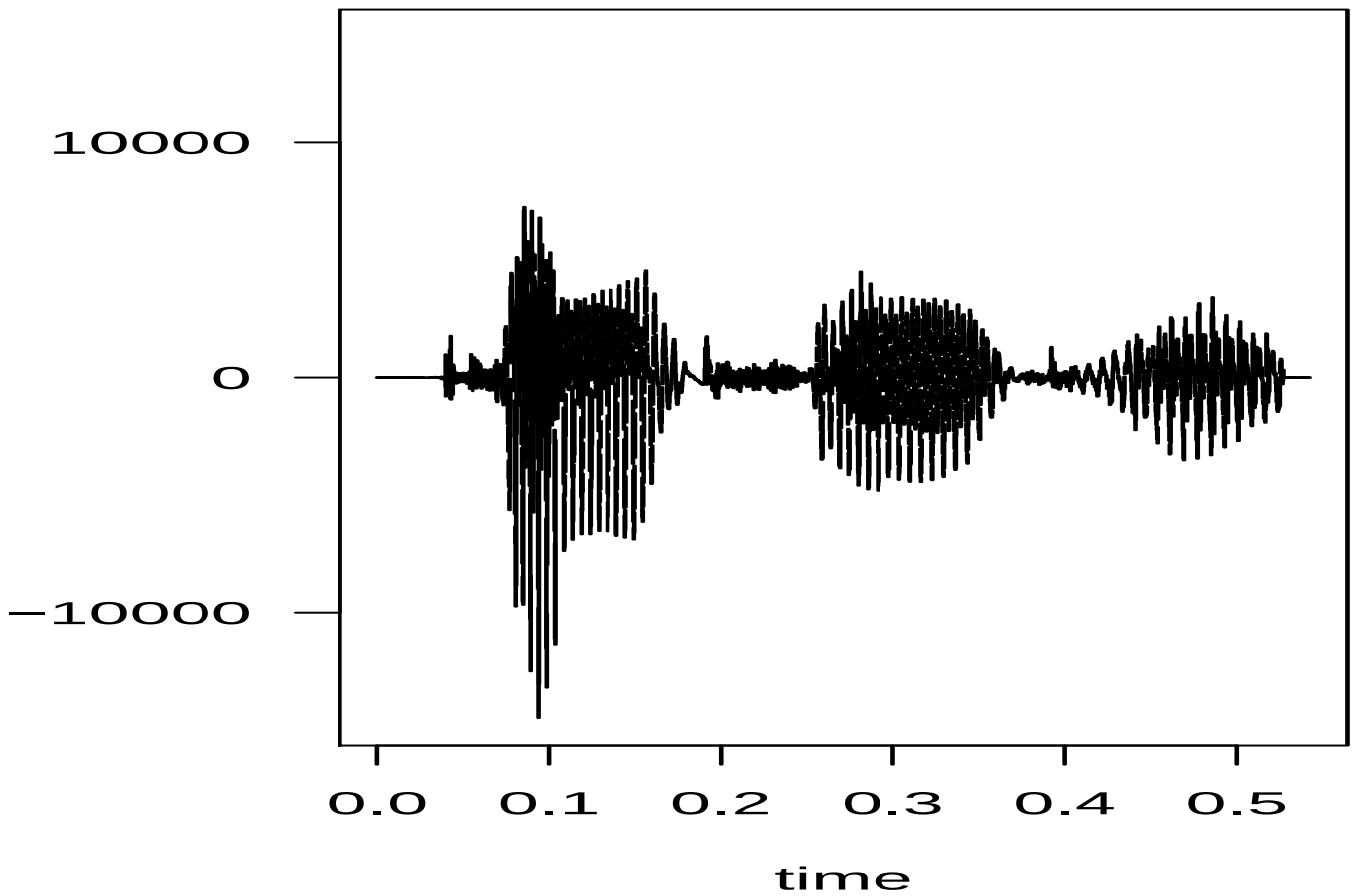
Do you detect an accent? If so, which one is it?



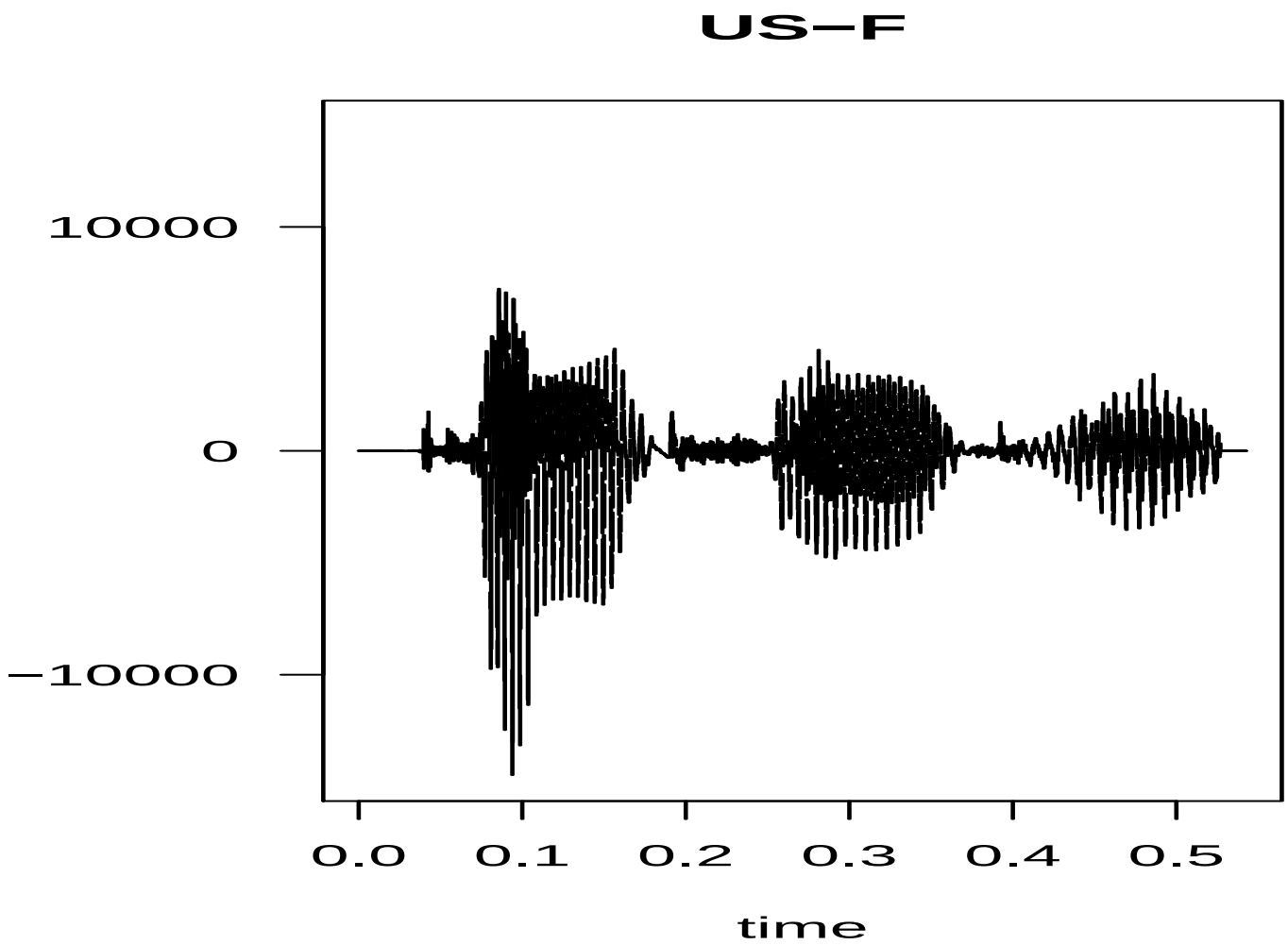
Answer: Female person from France



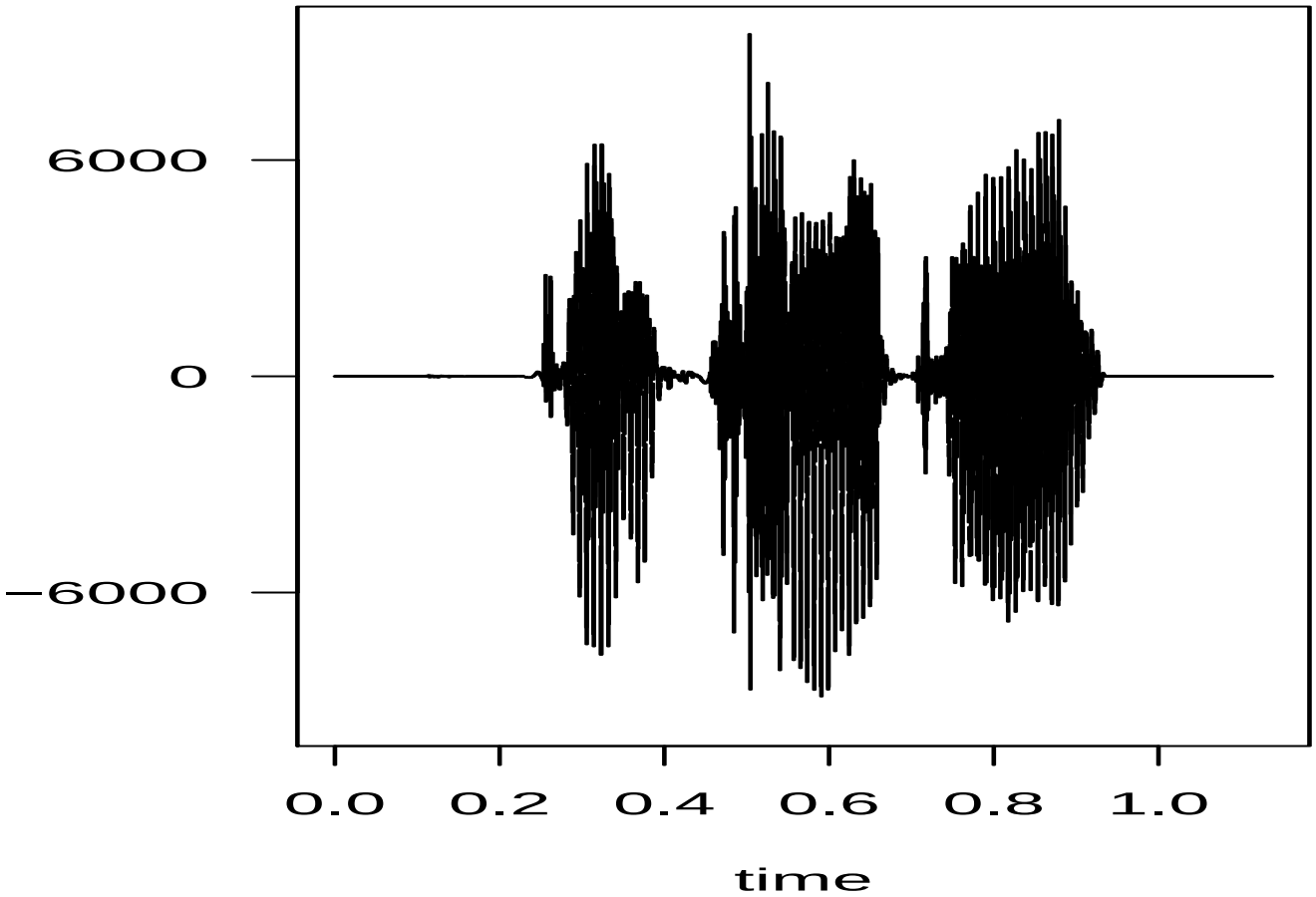
Do you detect an accent? If so, which one is it?



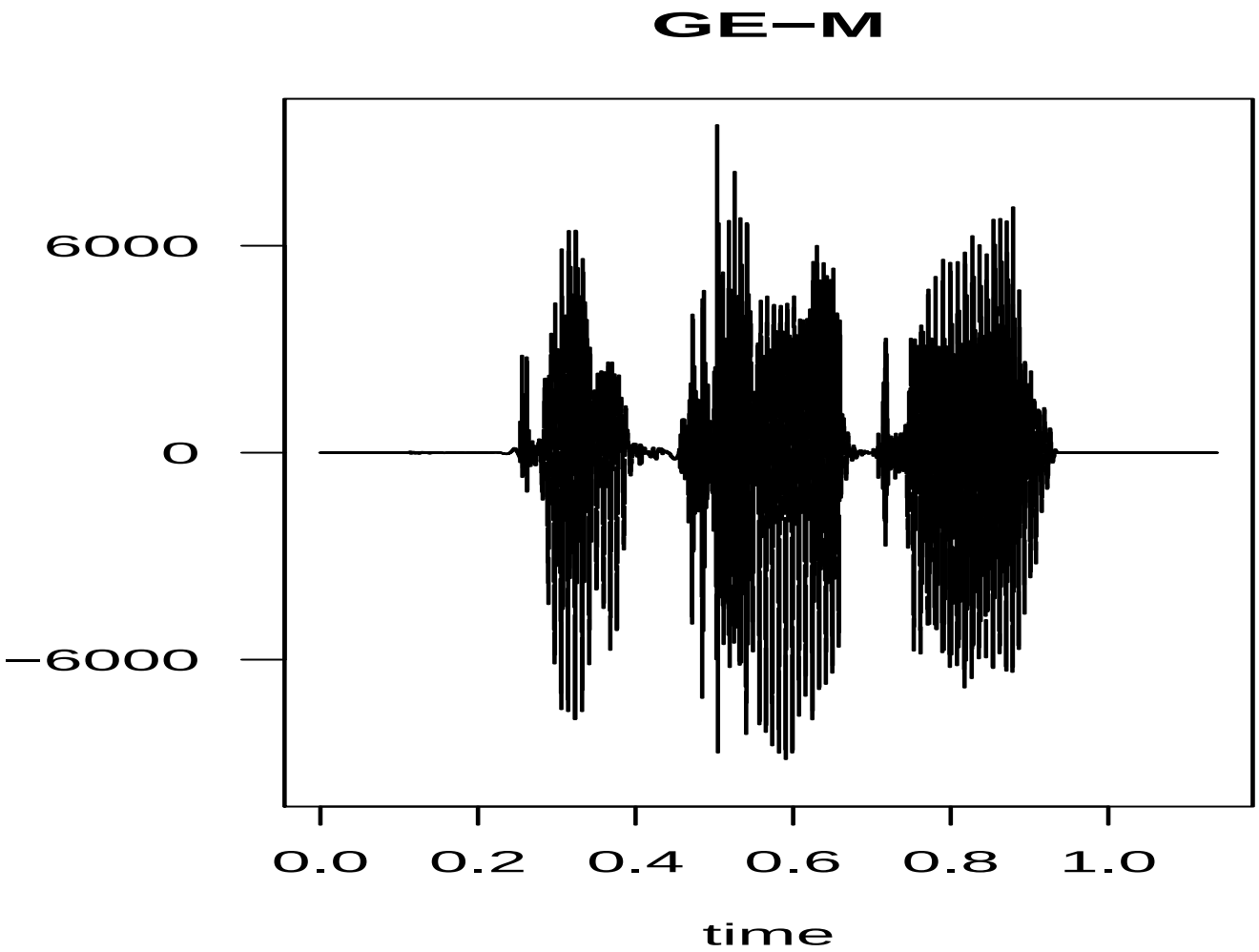
Answer: Female person from the United States



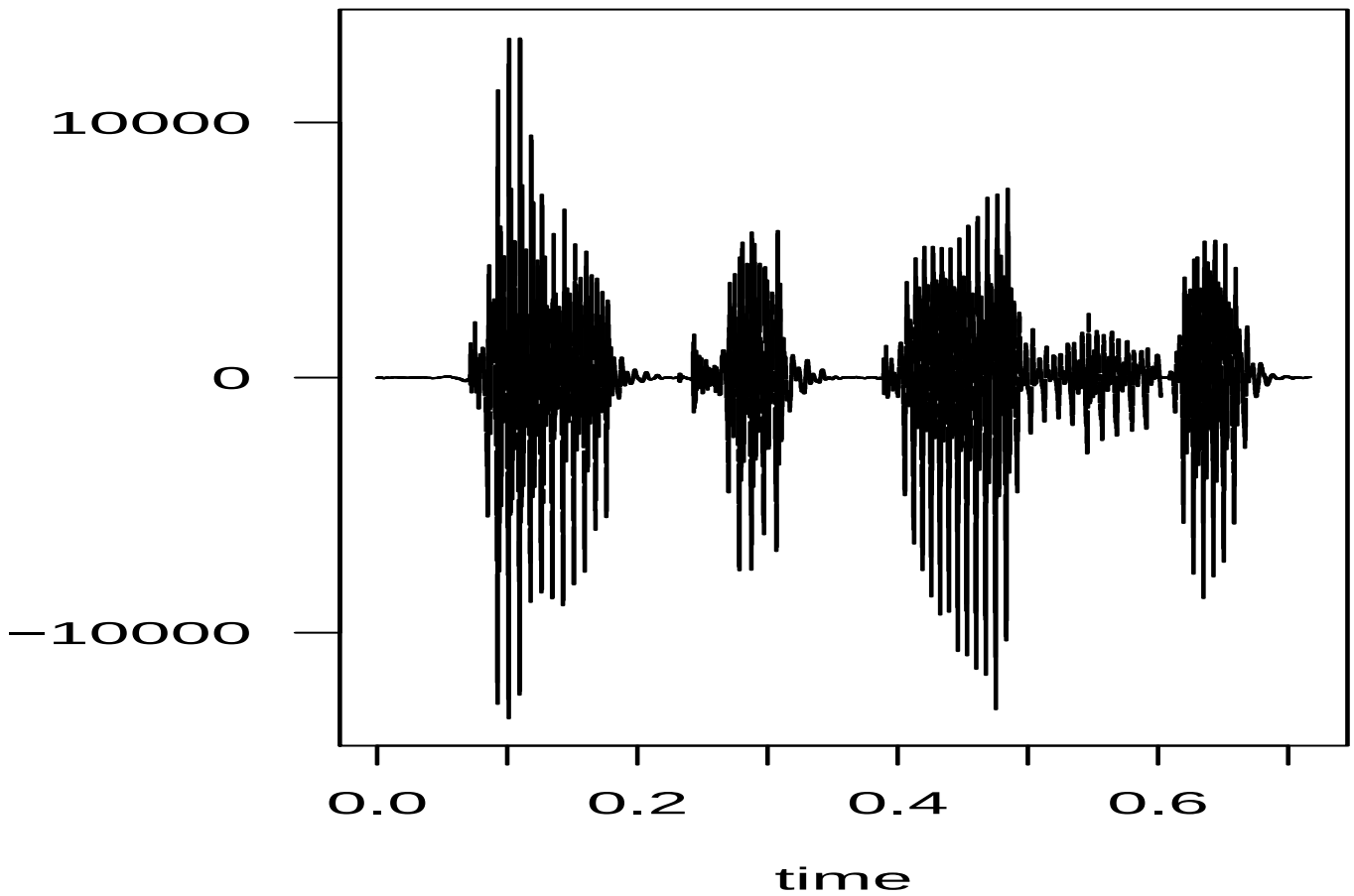
Do you detect an accent? If so, which one is it?



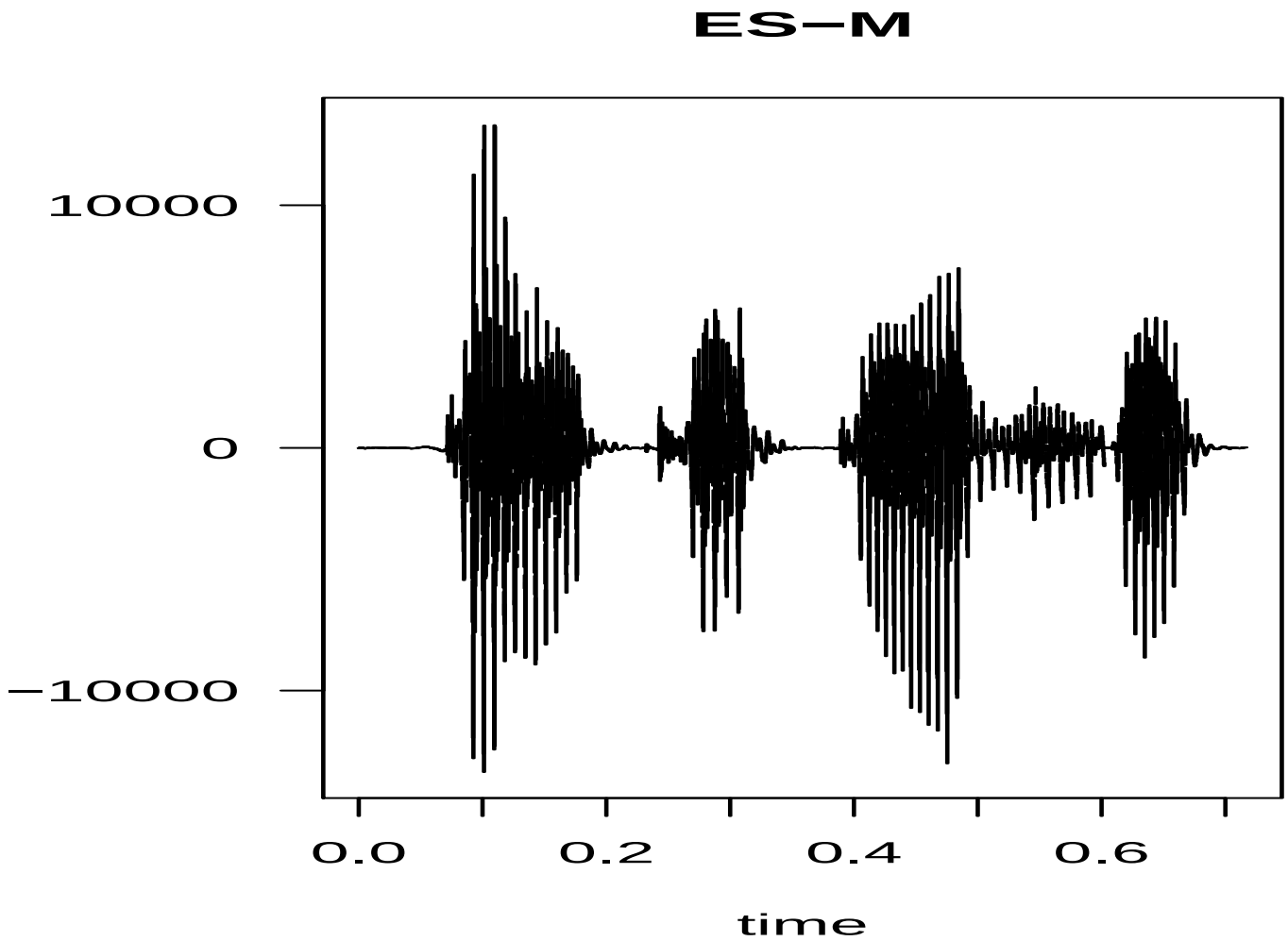
Answer: Male person from Germany



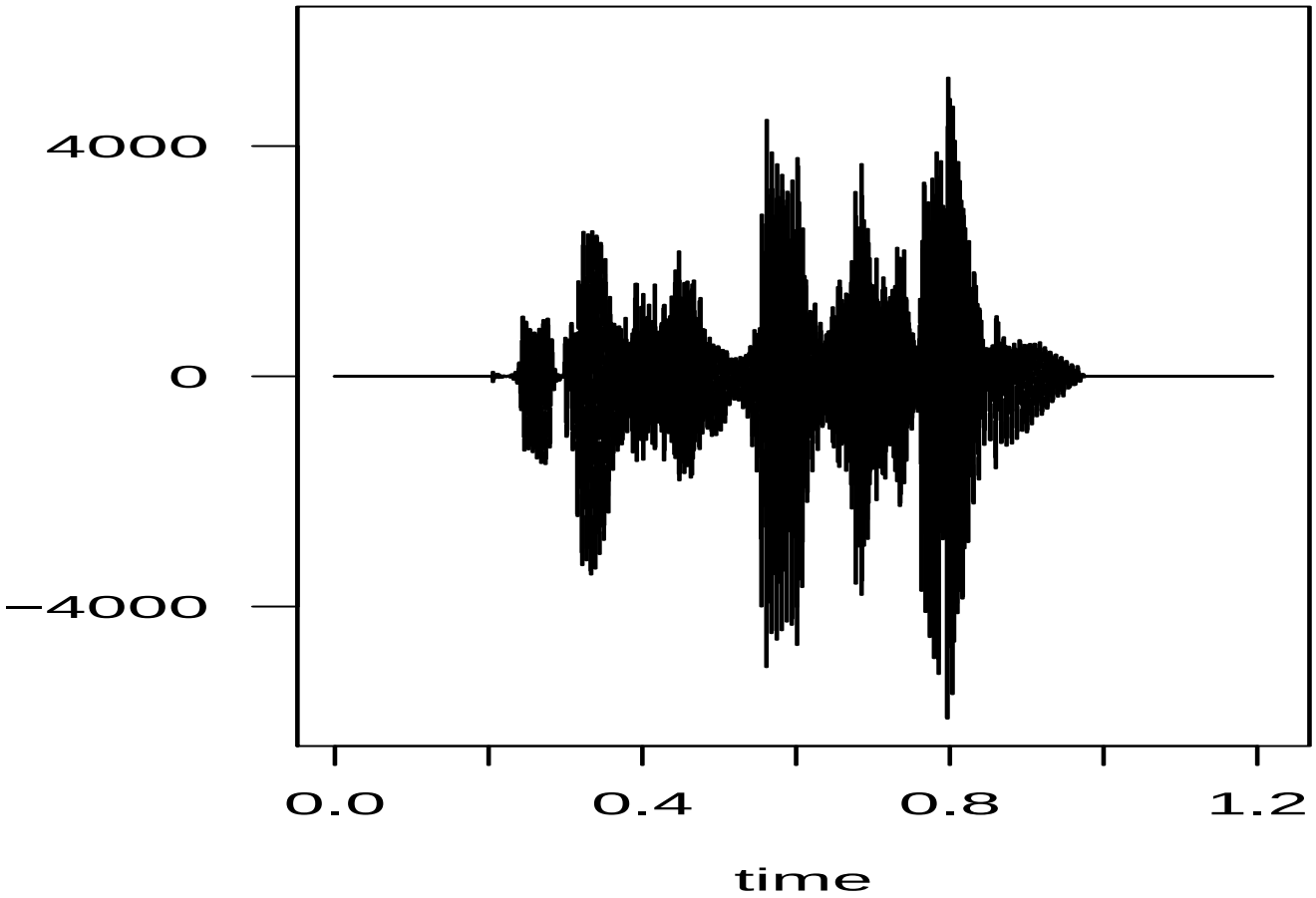
Do you detect an accent? If so, which one is it?



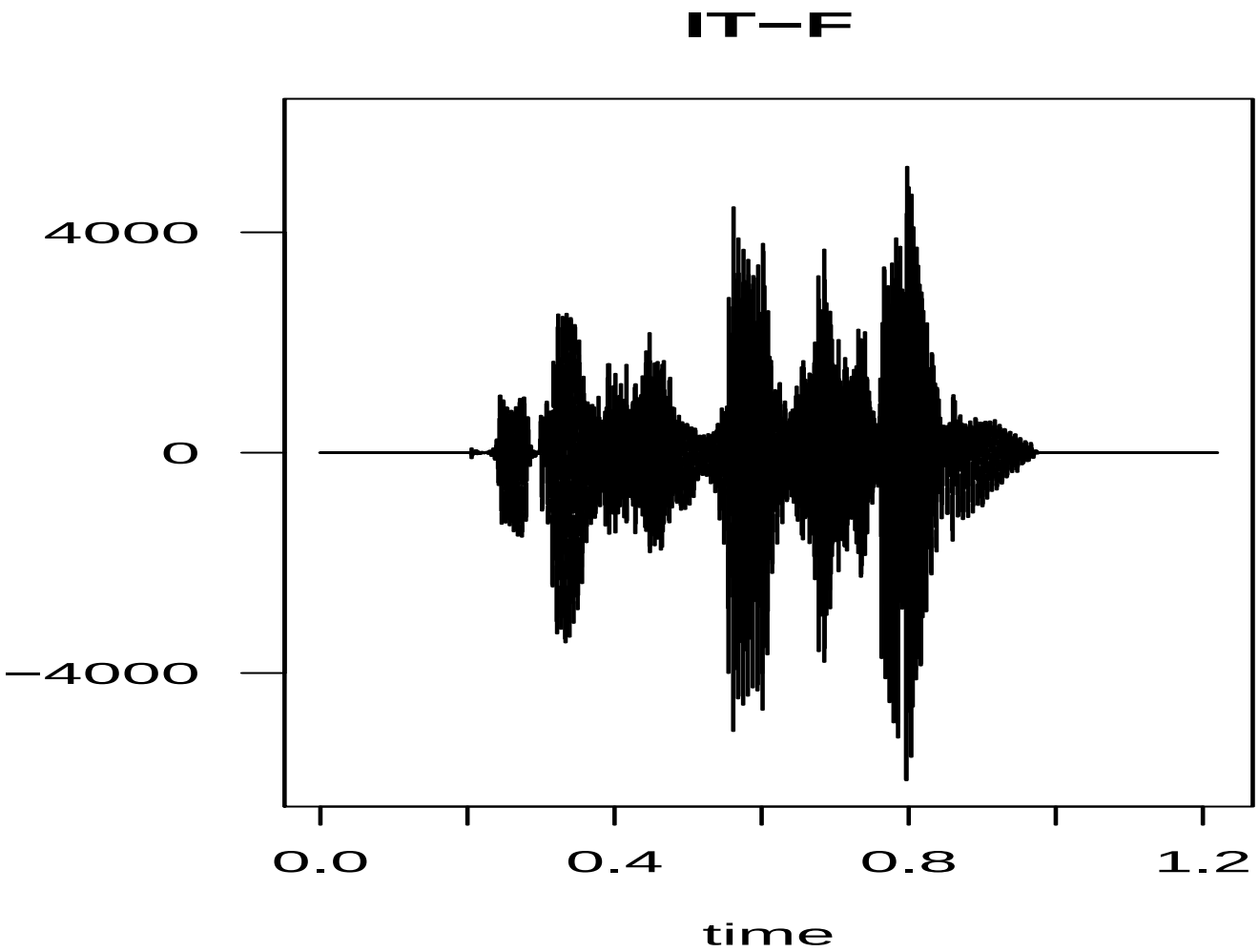
Answer: Male person from Spain



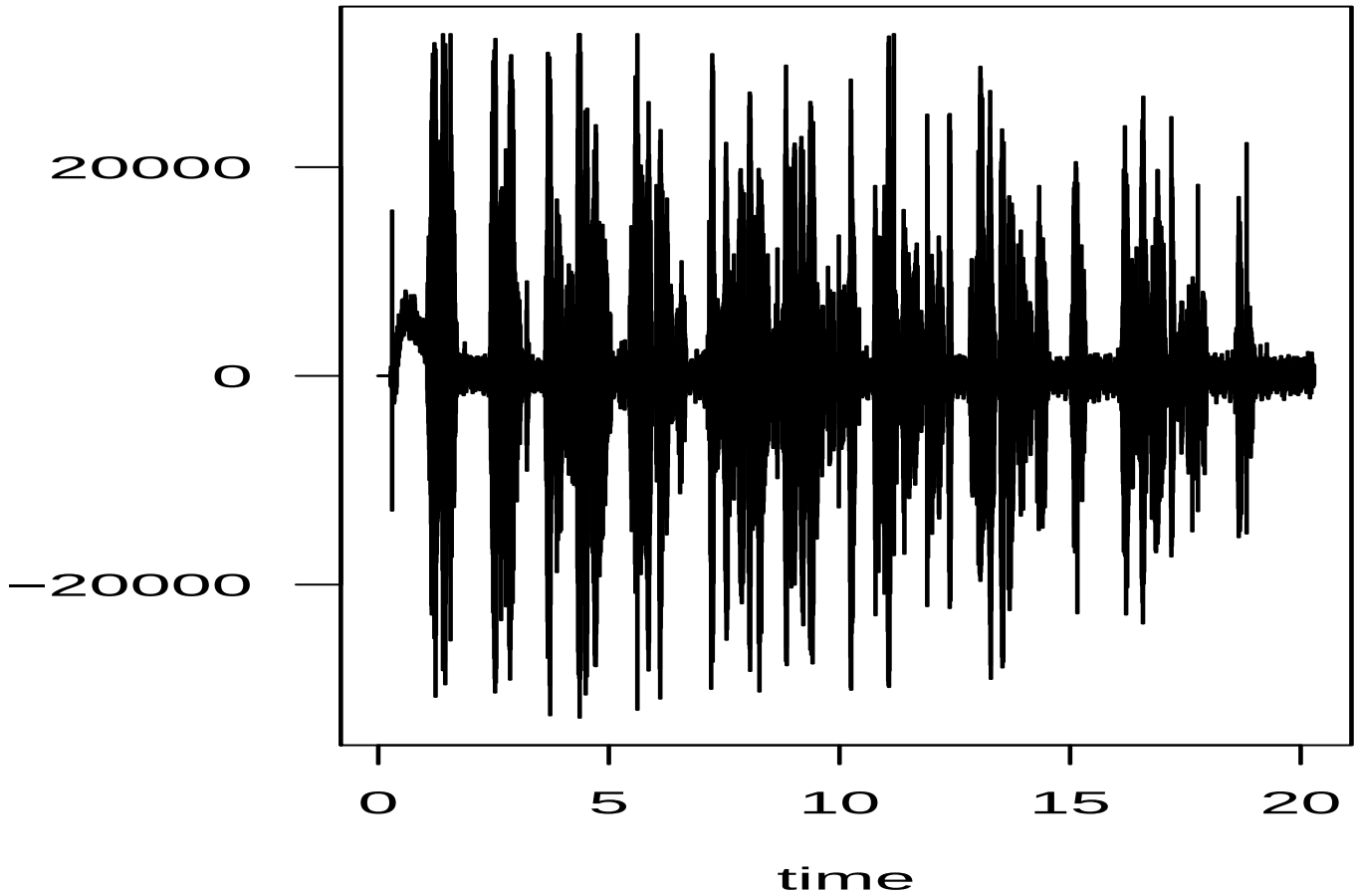
Do you detect an accent? If so, which one is it?



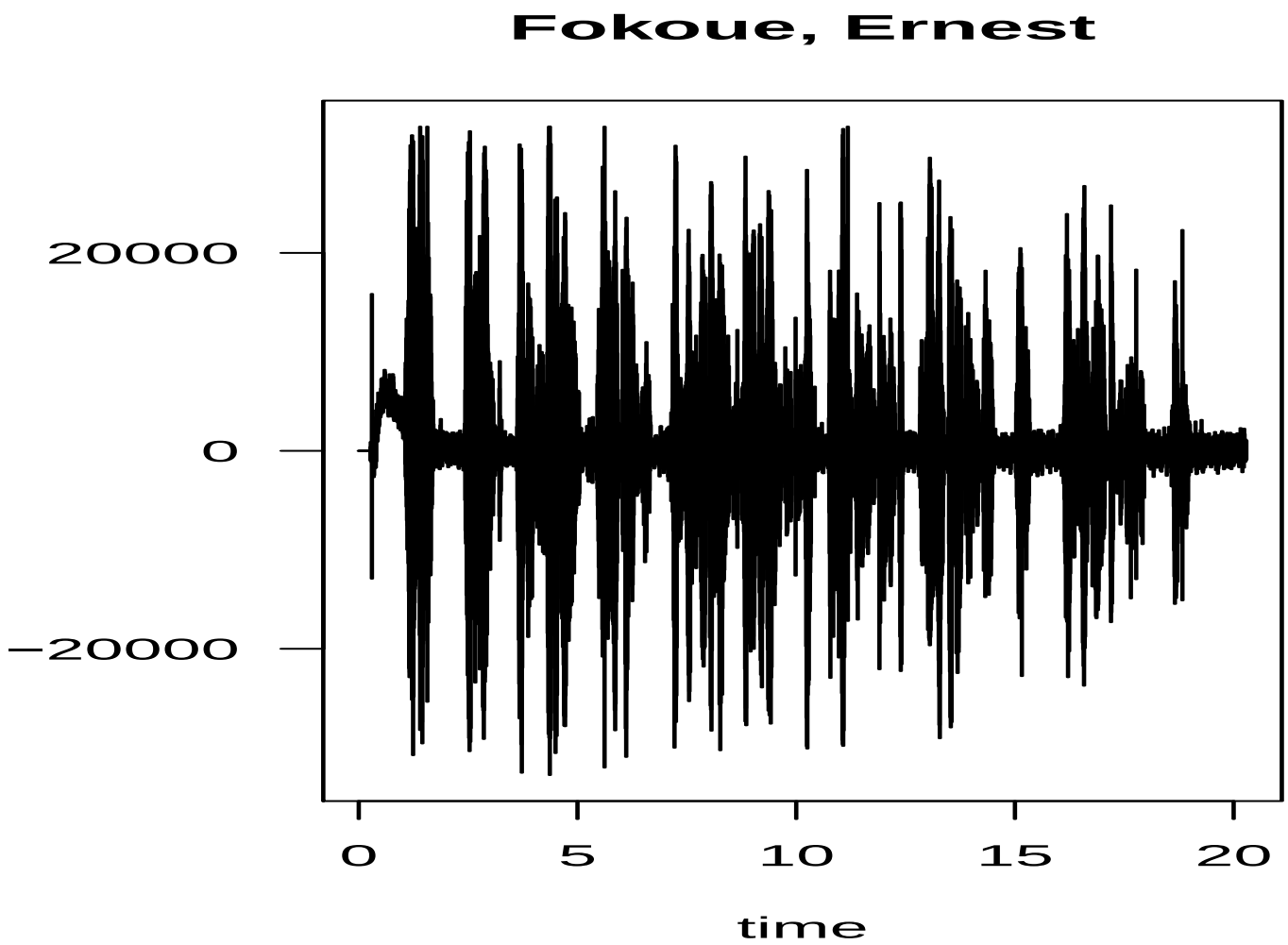
Answer: Female person from Italy



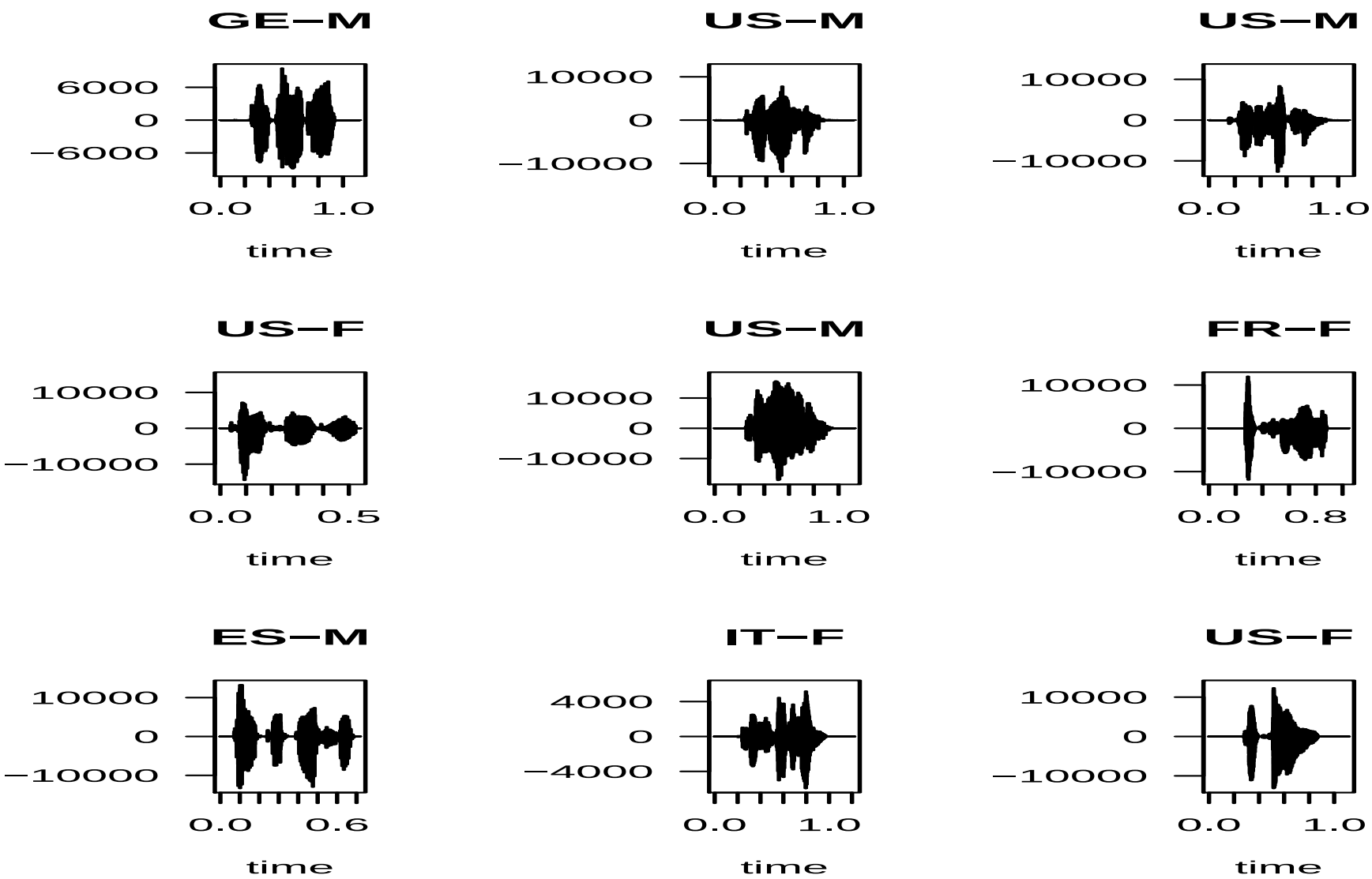
Question: Who is this guy talking?



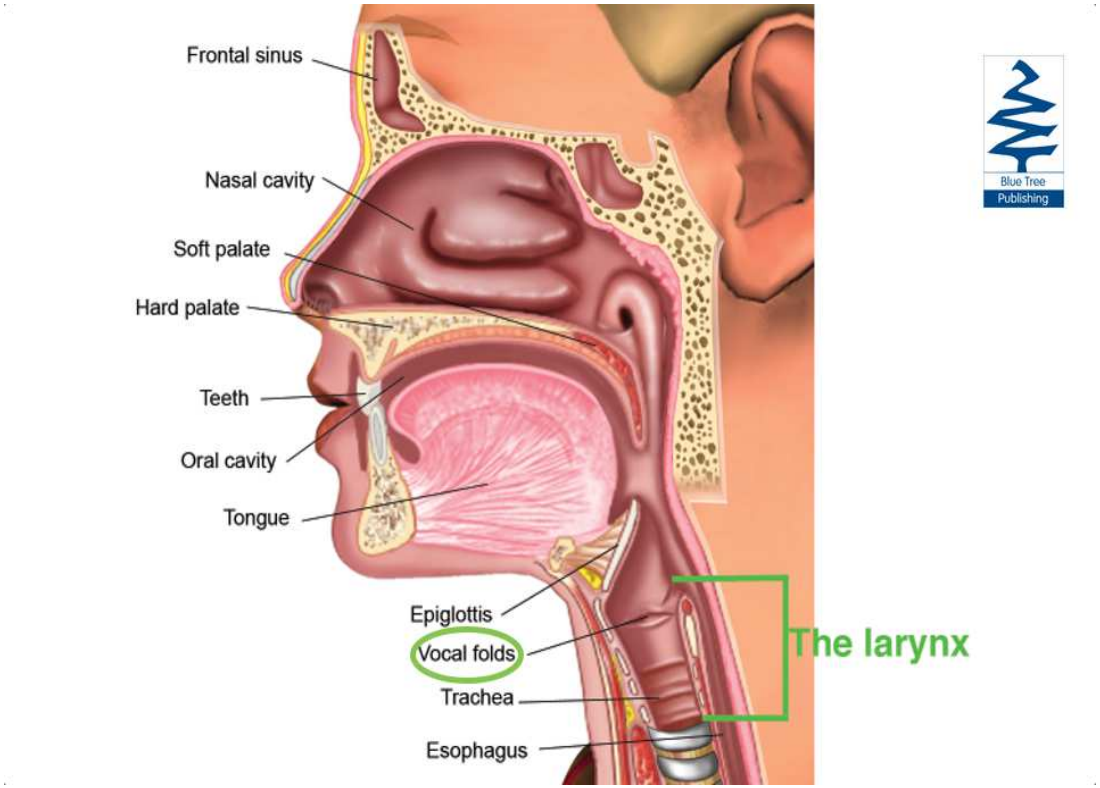
Answer: Male person named Fokoue, Ernest



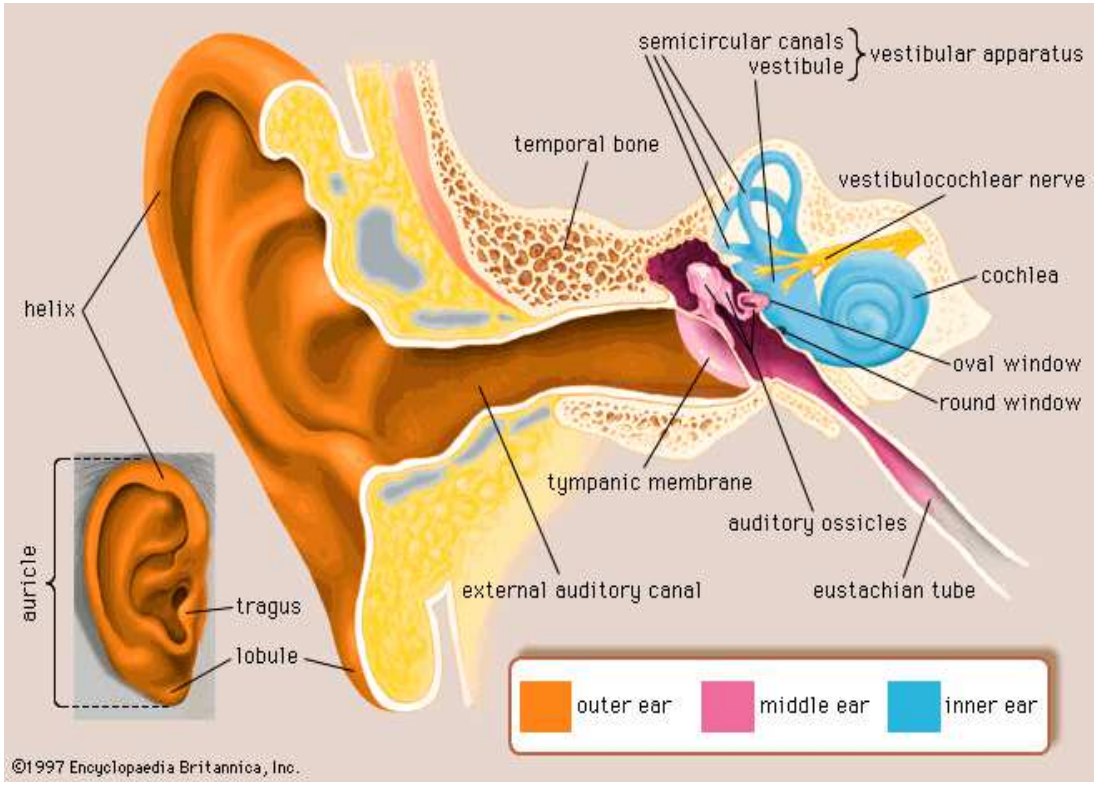
Time domain plot of audio files of nine speakers



How do we speak and/or sing?



How do we hear?

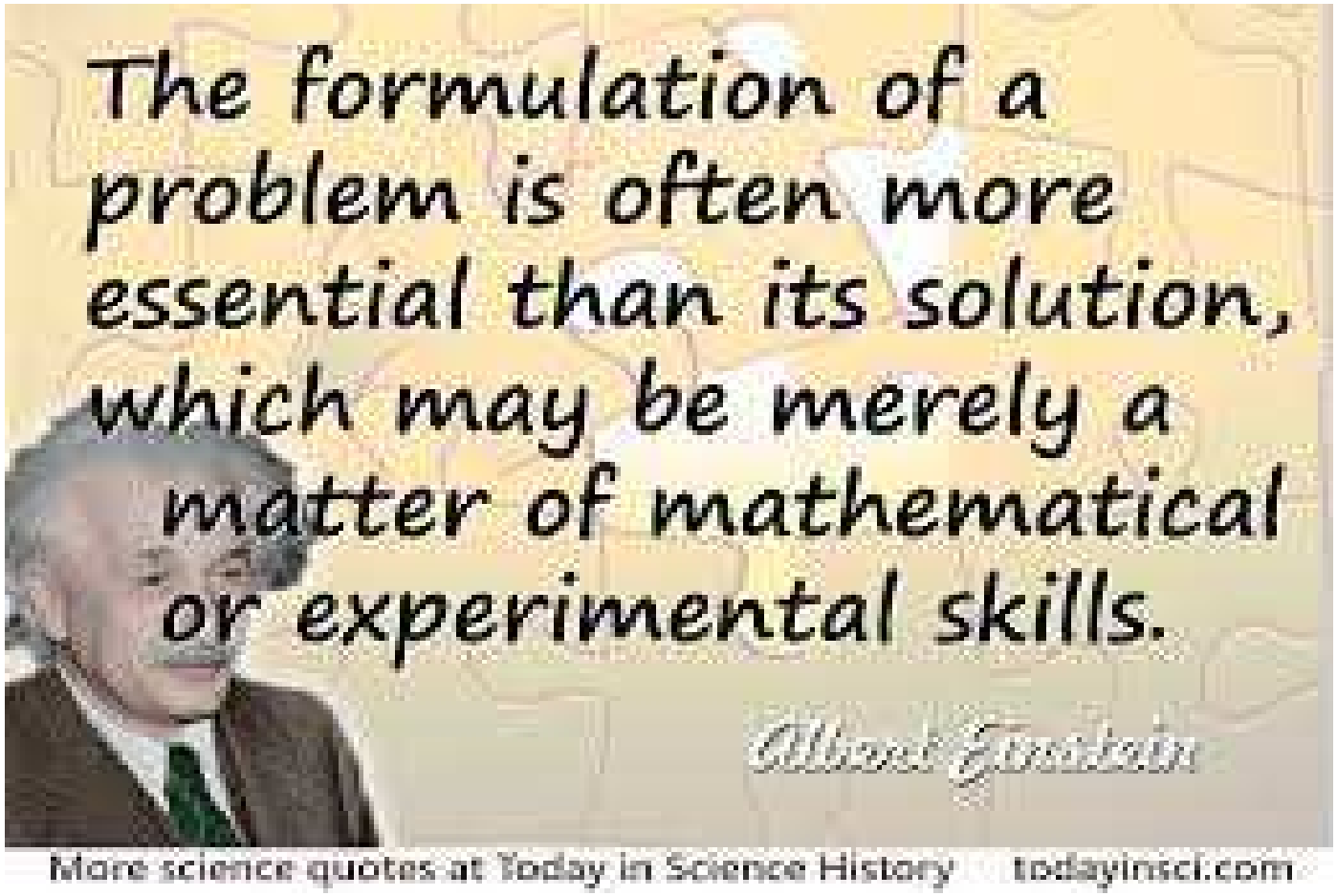


Learning Problem Formulation

What is the clear formulation and conceptualization of the problem to be solved?

Importance of Clear Formulation

Importance of Clear Formulation



Statistical Recognition of the Accent of a Speaker

- *Consider* $X_i = (\mathrm{x}_{i1}, \cdots, \mathrm{x}_{ip})^\top \in \mathbb{R}^p$ *and* $Y_i \in \{-1, +1\}$, *and the set*

$$\mathcal{D} = \Big\{(X_1, Y_1), (X_2, Y_2), \cdots, (X_n, Y_n)\Big\}$$

where

$$Y_i = \begin{cases} +1 & \textit{if person } i \textit{ is a Native US} \\ -1 & \textit{if person } i \textit{ is a Non Native US} \end{cases}$$

and $X_i = (\mathrm{x}_{i1}, \cdots, \mathrm{x}_{ip})^\top \in \mathbb{R}^p$ *is the time domain representation of his/her reading of an English sentence. The design matrix is*

$$\boldsymbol{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & \cdots & \cdots & \cdots & x_{1j} & \cdots & x_{1p} \\ \vdots & \vdots & \ddots & \ddots & \cdots & \cdots & \cdots & \cdots & \vdots \\ x_{i1} & x_{i2} & \cdots & \cdots & \cdots & \cdots & x_{ij} & \cdots & x_{ip} \\ \vdots & \vdots & \ddots & \ddots & \cdots & \cdots & \cdots & \cdots & \vdots \\ x_{n1} & x_{n2} & \cdots & \cdots & \cdots & \cdots & x_{nj} & \cdots & x_{np} \end{bmatrix}$$

Statistical Recognition of the Accent of a Speaker

- *Consider this design matrix*

$$\boldsymbol{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & \cdots & \cdots & \cdots & x_{1j} & \cdots & x_{1p} \\ \vdots & \vdots & \ddots & \ddots & \cdots & \cdots & \cdots & \cdots & \vdots \\ x_{i1} & x_{i2} & \cdots & \cdots & \cdots & \cdots & x_{ij} & \cdots & x_{ip} \\ \vdots & \vdots & \ddots & \ddots & \cdots & \cdots & \cdots & \cdots & \vdots \\ x_{n1} & x_{n2} & \cdots & \cdots & \cdots & \cdots & x_{nj} & \cdots & x_{np} \end{bmatrix}$$

- *We recently collected $n = 329$ voices of people reading words.*
- *Each word required less than 1 second to be read.*
- *At a sampling rate of 441000 Hz, each word requires a vector of dimension at most p=44100 in the time domain.*
- *We therefore have a gravely underdetermined system with $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ where $n \lll p$. Here, n=329 and p=44100.*

Elements of Statistical Machine Learning

- *All the datasets shown are samples from a wider population*
- *Ideally* **random samples**
- *Randomness is central for needed* impersonal chance
- *Randomness carries* **uncertainty**
- *Estimation (learning) from random data carries* **Estimation Error**
- *Learning requires* **probabilistic modeling**
- *Modeling requires measures and concepts like distribution, likelihood and loss functions, entropy and information*
- *Modeling requires* **Approximation**
- **Approximation Error** *is pervading in machine learning*
- *Data science must quantify* **Estimation and Approximation Error**
- *Probabilistic inequalities*
- *Interval estimation and learning bounds determination*
- *Hypothesis Testing in the classical and modern sense*

Introduction to the basic idea of pattern recognition

1. *Data: Given a data set* $\left\{(\mathbf{x}_1, \mathrm{y}_1), (\mathbf{x}_2, \mathrm{y}_2), \cdots, (\mathbf{x}_n, \mathrm{y}_n)\right\}$ *where* $\mathbf{x}_i = (\mathrm{x}_{i1}, \cdots, \mathrm{x}_{iq})^\top \in \mathbb{R}^q$ *and* $\mathrm{y}_i \in \{1, 2, \cdots, G\}$. *Here,* y_i *is simply the label of the class.*
2. *Goal: The immediate aspect of the pattern recognition (classification) problem deals with building a classification rule that assigns vectors (representing a collection of characteristics of entities under consideration) to* G *different categories*
3. *Generalization: Given the data, the goal in PR is to find the best classifier among all classifiers* f*, not just on the present data set, but also for all future entities generated by the same population.*

Question: When one says best classifier, it is foundational important to specify the criterion by which the best is determined.

Introduction to the basic idea of pattern recognition

- *Loss function:* *Best!! With respect to what criterion? This is central to machine learning. One needs to define a loss function that measure the goodness of the classifier.*

$$\ell(Y, f(X)) = \text{loss incurred from using } f(X) \text{ in place of } Y$$

For instance, in binary classification (K=2), an intuitively appealing and mathematically sound loss function is the 0-1 loss function

$$\ell(Y, f(X)) = \begin{cases} 1 & \text{if } Y \neq f(X) \\ 0 & \text{if } Y = f(X) \end{cases}$$

which says that you incur no loss if you correctly classify and a loss of 1 is you misclassify. In matrix or table format, it is

		$f(X)$	
		0	1
Y	0	0	1
	1	1	0

Introduction to the basic idea of pattern recognition

- *Symmetric Loss function:* *The zero-one loss function*

		$f(X)$	
		0	1
Y	0	0	1
	1	1	0

implicitly assumes that the cost of false positive is the same as the cost of false negative, $\ell(0,1) = \ell(1,0)$.

- *Asymmetric Loss function:* *This equal cost assumption may be wrong in some applications. For such applications, a so-called non-symmetric loss function must be used is*

		$f(X)$	
		0	1
Y	0	0	a
	1	b	0

Introduction to the basic idea of pattern recognition

- *Cost function (Risk functional):* *The objective function is therefore the expected loss also known as risk*

$$R(f) = \mathbb{E}[\ell(Y, f(X))] = \int_{\mathcal{X}\times\mathcal{Y}} \ell(\mathrm{y}, f(\mathrm{x}))dP(\mathrm{x}, \mathrm{y})$$

- *Cost function as misclassification rate:* *It is interesting to see that the objective function (risk functional) is simply the probability of misclassification rate*

$$R(f) = \mathbb{E}[\ell(Y, f(X))] = \Pr[Y \neq f(X)]$$

Note: It turns out that in practice, the above risk functional cannot be obtained in closed-form, because clearly the joint cdf $P(\mathrm{x}, \mathrm{y})$ is not known. If it were, all would be easy.

Introduction to the basic idea of pattern recognition

- *Cost function (Risk functional):* *The risk functional* $R(f) = \mathbb{E}[\ell(Y, (X))] = \Pr[Y \neq f(X)]$ *confirms our intuition because it is estimated in practice by simply computing the proportion of misclassified entities. We are basically saying that*
- *The universally best classifier:*

  the best classifier f^* *is the one that minimizes the rate of misclassifications.*

$$f^* = \arg\min_f \mathbb{E}[\ell(Y, f(X))] = \arg\min_f \Pr[Y \neq f(X)]$$

- *Note of generalization:*

  The minimization must be achieved on the whole population of entities to be classified, not just the ones found in some random sample from that population.

Introduction to the basic idea of pattern recognition

- *Approximation:* *Since searching all possible functions in the universe in order to find the one that best explains our data is clearly a daunting task, it is usually the case in ML and Data mining to approximate, i.e. choose a particular class of function.*
- *Linear classifiers:* *Search for the best among all linear classifiers.*

$$f^+ = \arg\min_{f \in \mathcal{L}} \mathbb{E}[\ell(Y, f(X))] = \arg\min_{f \in \mathcal{L}} \Pr[Y \neq f(X)]$$

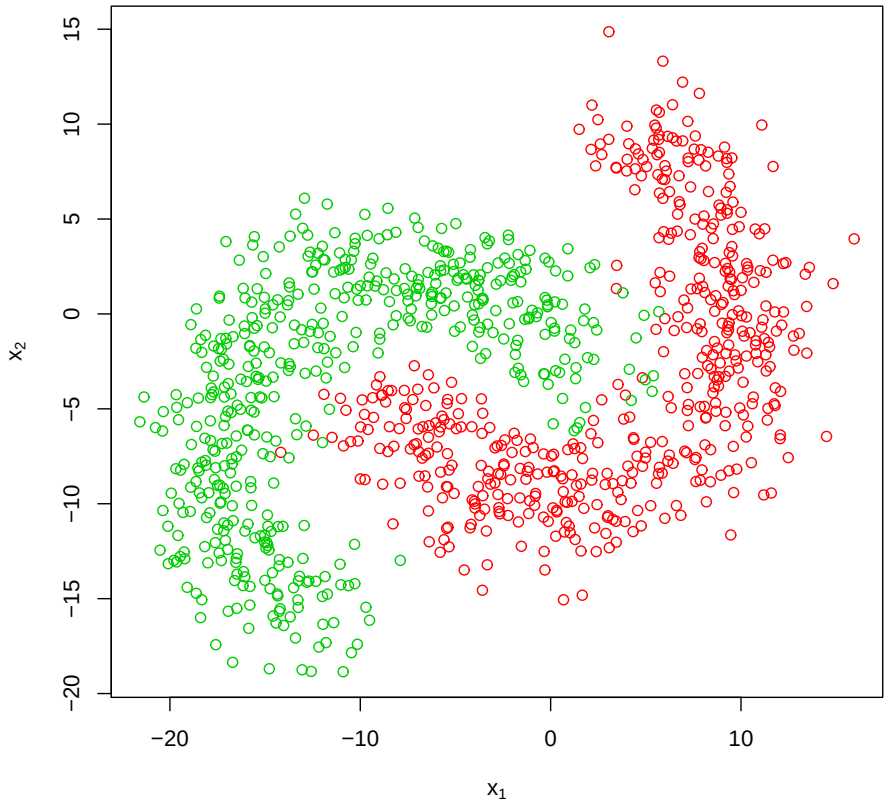
- *Arbitrary set $\mathcal{H}$ of classifiers:* *Search for the best among all the classifiers from set $\mathcal{H}$.*

$$f^+ = \arg\min_{f \in \mathcal{H}} \mathbb{E}[\ell(Y, f(X))] = \arg\min_{f \in \mathcal{H}} \Pr[Y \neq f(X)]$$

The set $\mathcal{H}$ above could be finite or infinite. All the algorithms and methods of classification studied in this course search such a class as most start by assuming a certain form for the classifier.

Binary Classification in the Plane, $\mathscr{X} \subset \mathbb{R}^2$

- $\mathscr{D}_n = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$, *with* $\mathbf{x}_i \in \mathscr{X} \subset \mathbb{R}^2$ *and* $y_i \in \{-1, +1\}$



- *What is the "best" classifier $f^\star$ that separates the red from the green?*

Binary Classification in the Plane

Learning tasks like the banana shaped classification problem have the distinct advantage that thanks to the ability to visualize in the plan, we get to gain deeper insights into some of the concepts inherent in the methods prior to generalizing to higher dimensions. Binary classification in 2D classification can be formalized and formulated as follows:

- *A collection* $\mathscr{D}_n = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$ *of i.i.d. observations*
  - $\mathbf{x}_i \in \mathscr{X} \subset \mathbb{R}^2, i = 1, \cdots, n$. $\mathscr{X}$ *is the* input *space.*
  - $y_i \in \{-1, +1\}$. $\mathscr{Y} = \{-1, +1\}$ *is the* output *space.*
- *What is the probability law that governs the* $(\mathbf{x}_i, y_i)$*'s?*
- *What is the functional relationship between* $\mathbf{x}$ *and* y*?*
- What is the "best" approach to determining from the available observations, the relationship between $\mathbf{x}$ and y in such a way that, *given a new (unseen) observation* $\mathbf{x}^{\text{new}}$*, its class* y^{new} *can be predicted as accurately as possible.*

Basic Remarks on Classification

- *Finding an automatic classification rule that achieves the absolute very best on the present data is not enough since infinitely many more observations can be generated by* $\mathbb{P}(\mathbf{x}, y)$ *for which good classification will be required.*
- *Even the universally best classifier will make mistakes.*
- *Of all the functions in* $\mathscr{Y}^{\mathscr{X}}$*, it is reasonable to assume that there is a function* $f^\star$ *that maps any* $\mathbf{x} \in \mathscr{X}$ *to its corresponding* $y \in \mathscr{Y}$*, i.e.,*

$$\begin{array}{rccc} f^\star: & \mathscr{X} & \to & \mathscr{Y} \\ & \mathbf{x} & \mapsto & f^\star(\mathbf{x}), \end{array}$$

with the minimum number of mistakes.

Loss and Risk in Pattern Recognition

For this classification/pattern recognition, the so-called 0-1 *loss function defined below is used. More specifically,*

$$\ell(y, f(\mathbf{x})) = \mathbb{1}\{y \neq f(\mathbf{x})\} = \begin{cases} 0 & \text{if} \quad y = f(\mathbf{x}), \\ 1 & \text{if} \quad y \neq f(\mathbf{x}). \end{cases} \tag{1}$$

The corresponding risk functional is

$$R(f) = \int \ell(y, f(\mathbf{x})) d\mathbb{P}(\mathbf{x}, y) = \mathbb{E}\left[\mathbb{1}\{Y \neq f(X)\}\right] = \underset{(X,Y)\sim\mathbb{P}}{\Pr}[Y \neq f(X)].$$

The minimizer of the 0-1 *risk functional over all possible classifiers is the so-called* Bayes classifier *which we shall denote here by* $f^\star$ *given by*

$$f^\star = \arg\inf_f \left\{ \underset{(X,Y)\sim\mathbb{P}}{\Pr}[Y \neq f(X)] \right\}.$$

Specifically, the Bayes' classifier $f^\star$*, whose risk is* $R^\star = R(f^\star)$*, is given by the posterior probability of class membership, namely*

$$f^\star(\mathbf{x}) = \arg\max_{y \in \mathcal{Y}} \left\{ \Pr[Y = y | \mathbf{x}] \right\}.$$

How does one go about building learning machines?

Multitude of approaches to Building Learning Machines

- *Implicit approximation of $f^\star$ via intuitive creation and development of algorithms and methods*
  - *Nearest Neighbors Learning Paradigm*
  - *Classification Trees*
- *Explicit approximation of $f^\star$ via analytic and computational methods for search suitable chosen function spaces*
  - *Support Vector Machines*
  - *Gaussian Processes*
  - *Neural Networks*
- *Direct Estimation of $f^\star$ via specific assumptions on the measure from which the data originates*
  - *Linear and Quadratic Discriminant Analyzers*
  - *Naive Bayes Classifiers*

How intuition helps create learning machines!!

"When the character of a man is not clear to you, look at his friends"
Japanese Proverb

How intuition help create learning machines!!

"Tell me who your friends are, and I will tell you who you are."
Mexican Proverb

"When it comes to skills or character, you often tend to be as good or as bad as your constant companions are"

Unknown

k-Nearest Neighbors (kNN) Learning Machine

$\mathscr{D}_n = \Big\{(\mathbf{x}_1, \mathrm{y}_1), \cdots, (\mathbf{x}_n, \mathrm{y}_n)\Big\}$, *with* $\mathbf{x}_i \in \mathscr{X}$, $\mathrm{y}_i \in \mathscr{Y}$.

1. *Choose the value of* k *and the distance to be used*
2. *Let* $\mathbf{x}$ *be a new point. Compute*

$$d_i = d(\mathbf{x}, \mathbf{x}_i) \qquad i = 1, \cdots, n$$

3. *Rank all the distances* d_i *in increasing order*

$$d_{(1)} \leq d_{(2)} \leq \cdots \leq d_{(k)} \leq d_{(k+1)} \leq \cdots \leq d_{(n)}$$

4. *Form* $\mathcal{V}_k(\mathbf{x})$, *the* k*-Neighborhood of* $\mathbf{x}$

$$\mathcal{V}_k(\mathbf{x}) = \Big\{\mathbf{x}_i : \ d(\mathbf{x}, \mathbf{x}_i) \leq d_{(k)}\Big\}$$

5. *Compute the predicted response* $\widehat{f}^{(\mathtt{kNN})}(\mathbf{x})$ *as*

$$\widehat{f}^{(\mathtt{kNN})}(\mathbf{x}) = \textit{Most common tendency in } \mathcal{V}_k(\mathbf{x})$$

Algorithm 1 k-Nearest Neighbors (kNN) Learning Machine

Input: $\mathscr{D}_n = \{\mathbf{z}_i = (\mathbf{x}_i, \mathrm{y}_i),\ \mathbf{x}_i \in \mathscr{X}, \mathrm{y}_i \in \mathscr{Y},\ i = 1, \cdots, n\}$, neighborhood size k, sample size n, new test point $\mathbf{x}$, distance $d(\cdot, \cdot)$.

Output: Prediction $\widehat{f}^{(\texttt{kNN})}(\mathbf{x})$

for $i = 1, \cdots, n$ **do**

  Compute $d_i = d(\mathbf{x}, \mathbf{x}_i)$

Rank all the distances d_i in increasing order: $d_{(1)} \leq \cdots \leq d_{(k)} \leq \cdots \leq d_{(n)}$

Form $\mathcal{V}_k(\mathbf{x})$, the k-Neighborhood of $\mathbf{x}$: $\mathcal{V}_k(\mathbf{x}) = \left\{\mathbf{x}_i :\ d(\mathbf{x}, \mathbf{x}_i) \leq d_{(k)}\right\}$

Compute the predicted response $\widehat{f}^{(\texttt{kNN})}(\mathbf{x})$ as

$$\widehat{f}^{(\texttt{kNN})}(\mathbf{x}) = \begin{cases} \underset{g \in \mathscr{Y}}{\texttt{argmax}} \left\{ \dfrac{1}{k} \displaystyle\sum_{i=1}^{n} \mathbb{1}(y_i = g) \mathbb{1}(\mathbf{x}_i \in \mathcal{V}_k(\mathbf{x})) \right\} & \texttt{Classification} \\ \dfrac{1}{k} \displaystyle\sum_{i=1}^{n} y_i \mathbb{1}(\mathbf{x}_i \in \mathcal{V}_k(\mathbf{x})) & \texttt{Regression} \end{cases}$$

Basic Properties of a Distance (Metric)

A distance on an input space $\mathscr{X}$ is a bivariate function $d(\cdot, \cdot)$, that is

$$d : \mathscr{X} \times \mathscr{X} \longrightarrow [0, +\infty)$$

with the following properties

1. $d(\mathbf{x}, \mathbf{y}) \geq 0$ *(Nonnegativity)*
2. $d(\mathbf{x}, \mathbf{y}) = 0$ *if and only if* $\mathbf{x} = \mathbf{y}$ *(identity of indiscernibles)*
3. $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$ *(Symmetry)*
4. $d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y})$ *(Subadditivity or triangle inequality)*

Some examples of metrics (distances) include

- $d(\mathrm{x}, \mathrm{y}) = |\mathrm{x} - \mathrm{y}|$ *(univariate $\mathcal{L}_1$ distance)*
- $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_1 = \sum_{j=1}^{p} |\mathrm{x}_j - \mathrm{y}_j|$ *(multivariate $\mathcal{L}_1$ distance)*

Distances as Similarity Measures

Given two vectors $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^p$, The most common distances are

- *Euclidean distance: $\mathcal{L}_2$ distance*

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{\ell=1}^{p} (\mathrm{x}_{i\ell} - \mathrm{x}_{j\ell})^2} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$$

- *Manhattan distance: city block or $\mathcal{L}_1$ distance*

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{\ell=1}^{p} |\mathrm{x}_{i\ell} - \mathrm{x}_{j\ell}| = \|\mathbf{x}_i - \mathbf{x}_j\|_1$$

- *Maximum distance: infinity or Supremum distance*

$$d(\mathbf{x}_i, \mathbf{x}_j) = \max_{\ell=1,\cdots,p} |\mathrm{x}_{i\ell} - \mathrm{x}_{j\ell}| = \|\mathbf{x}_i - \mathbf{x}_j\|_\infty$$

Other common distances: (a) Minkowski*; (b)* canberra*; (c)* binary*.*

Distances as Similarity Measures

Given two vectors $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^p$, The most common distances are

- *Minkowski distance: $\mathcal{L}_q$ distance*

$$d(\mathbf{x}_i, \mathbf{x}_j) = \left\{ \sum_{\ell=1}^{p} |\mathrm{x}_{i\ell} - \mathrm{x}_{j\ell}|^q \right\}^{1/q}$$

- *Canberra distance:*

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{\ell=1}^{p} \frac{|\mathrm{x}_{i\ell} - \mathrm{x}_{j\ell}|}{|\mathrm{x}_{i\ell} + \mathrm{x}_{j\ell}|}$$

- *Jaccard/Tanimoto distance: For binary vectors ie $\mathbf{x}_i \in \{0,1\}^p$*

$$d(\mathbf{x}_i, \mathbf{x}_j) = 1 - \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{|\mathbf{x}_i|^2 + |\mathbf{x}_j|^2 - \mathbf{x}_i \cdot \mathbf{x}_j}$$

$\mathbf{x}_i \cdot \mathbf{x}_j = \sum_{\ell=1}^{p} \mathrm{x}_{i\ell}\mathrm{x}_{j\ell} = \sum_{\ell=1}^{p} \mathrm{x}_{i\ell} \wedge \mathrm{x}_{j\ell}$ *and* $|\mathbf{x}_i|^2 = \sum_{\ell=1}^{p} \mathrm{x}_{i\ell}^2$

Challenges of Model selection

Given the data

$$\mathscr{D}_n = \Big\{(\mathbf{x}_i, y_i) \overset{iid}{\sim} p_{\mathbf{xy}}(\mathbf{x}, y),\ i = 1, \cdots, n\Big\}, \tag{3}$$

One seeks to use a suitably defined model selection criterion, to search $\mathscr{H}$ and select the best function.

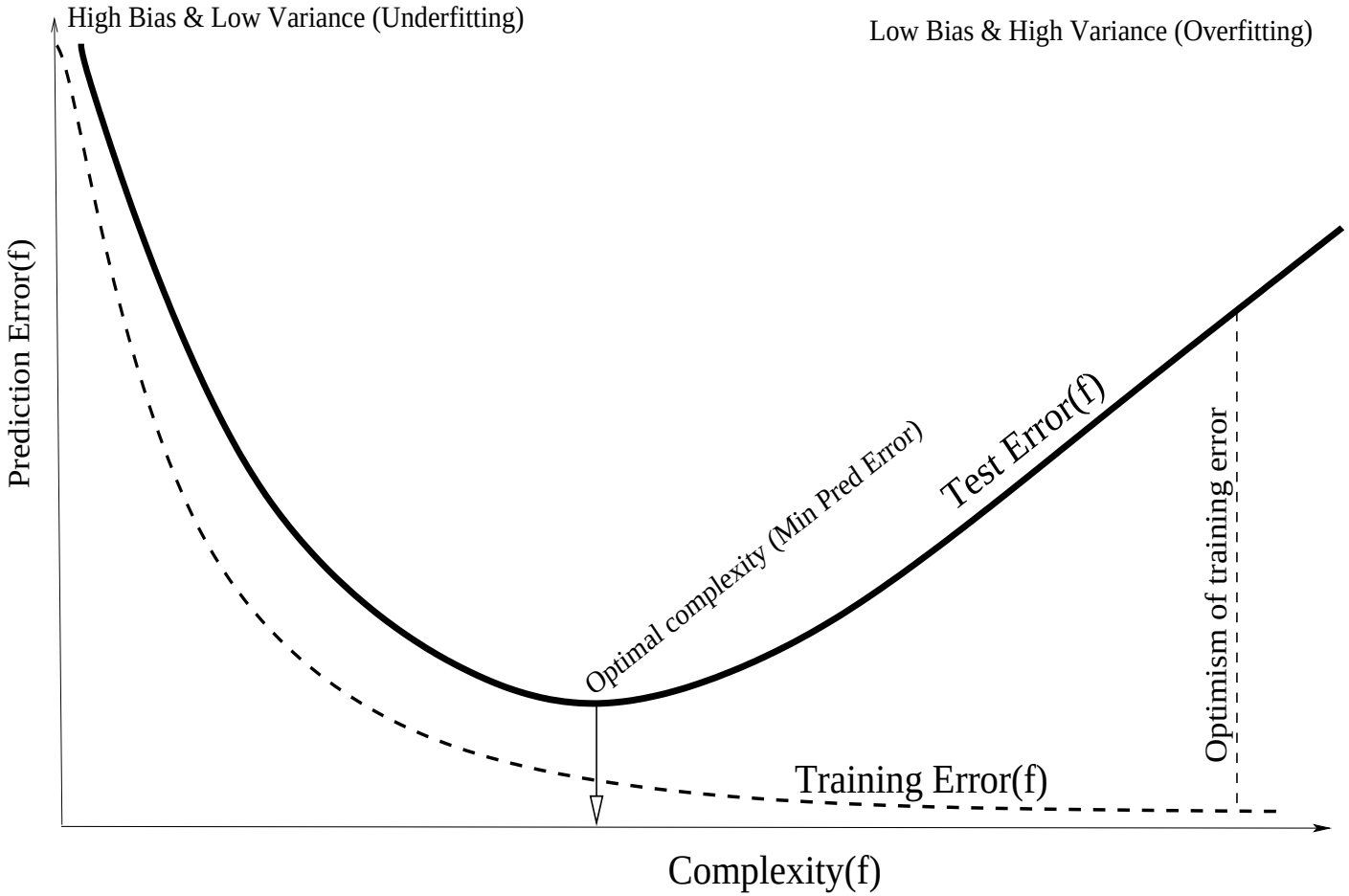
$$\widehat{f}_{\mathscr{H},\lambda,n} = \underset{f \in \mathscr{H}}{\texttt{argmin}} \left\{ \widehat{R}_{\mathscr{H},n}(f) + \lambda \Omega_{\mathscr{H}}(f) \right\}, \tag{4}$$

where λ controls the bias-variance trade-off. Typically, $\lambda > 0$ and determined by cross validation. Cross validation for determining λ proceeds by defining a grid $\Lambda \subset \mathbb{R}_+^\star = (0, +\infty)$ of possible values of λ. Sometimes, based on intuition or experience, it could just be $\Lambda = [\lambda_{\texttt{min}}, \lambda_{\texttt{max}}]$, then

$$\widehat{\lambda}^{(opt)} = \underset{\lambda \in \Lambda}{\texttt{argmin}} \left\{ \texttt{CV}(\widehat{f}_\lambda) \right\}. \tag{5}$$

Note: $\widehat{f}_{\mathscr{H},\widehat{\lambda}^{(opt)},n}$ is usually not unique

The Ubiquitous Bias-Variance Trade-off



Algorithm 2 $V-$fold Cross Validation

Input: Training data $\mathscr{D}_n = \{\mathbf{z}_i = (\mathbf{x}_i^\top, \mathrm{y}_i)^\top,\ i = 1, \cdots, n\}$, where $\mathbf{x}_i^\top \in \mathscr{X}$ and $\mathrm{y}_i \in \mathscr{Y}$, and the function of interest is denoted by g, sample size n, number of folds V

Output: *Cross Validation score* $\mathtt{CV}(\widehat{\mathrm{g}})$

for $\mathrm{v} = 1$ *to* V **do**

Extract the validation set $\mathscr{D}_\mathrm{v} = \left\{\mathbf{z}_i \in \mathscr{D}_n :\ i \in [1+(\mathrm{v}-1)\times m, \mathrm{v}\times m]\right\}$

Extract the training set $\mathscr{D}_\mathrm{v}^c := \mathscr{D}_n \backslash \mathscr{D}_\mathrm{v}$

Build the estimator $\widehat{f}^{(-\mathscr{D}_\mathrm{v})}(\cdot)$ using $\mathscr{D}_\mathrm{v}^c$

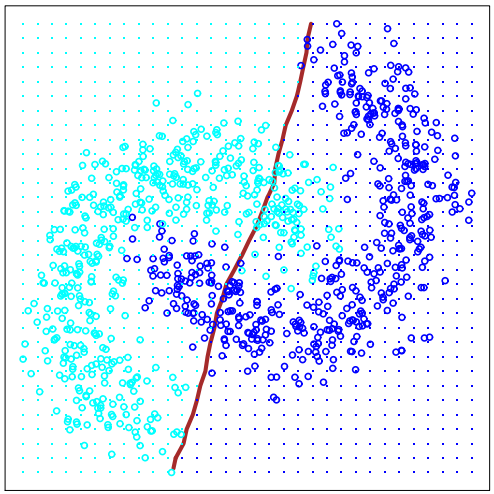
Compute predictions $\widehat{f}^{(-\mathscr{D}_\mathrm{v})}(\mathbf{x}_i)$ `for` $\mathbf{z}_i \in \mathscr{D}_\mathrm{v}$

Compute the validation error for the v^{th} chunk

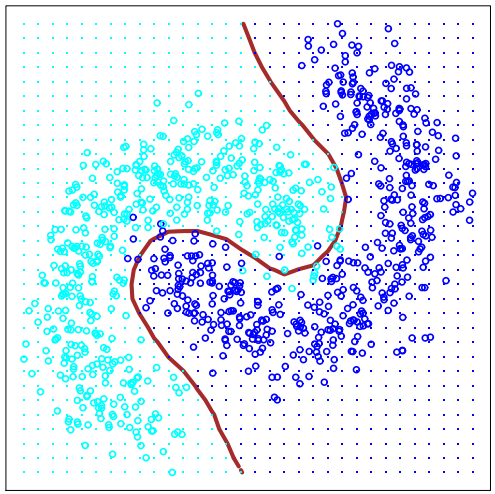
$$\widehat{\varepsilon}_\mathrm{v} = \frac{1}{|\mathscr{D}_\mathrm{v}|}\sum_{i=1}^{n} \mathbb{1}(\mathbf{z}_i \in \mathscr{D}_\mathrm{v})\mathcal{L}(\mathrm{y}_i, \widehat{f}^{(-\mathscr{D}_\mathrm{v})}(\mathbf{x}_i))$$

Compute the $\mathtt{CV}$ score $\mathtt{CV}(\widehat{\mathrm{g}}) = \frac{1}{V}\sum_{\mathrm{v}=1}^{V} \hat{\varepsilon}_\mathrm{v}$

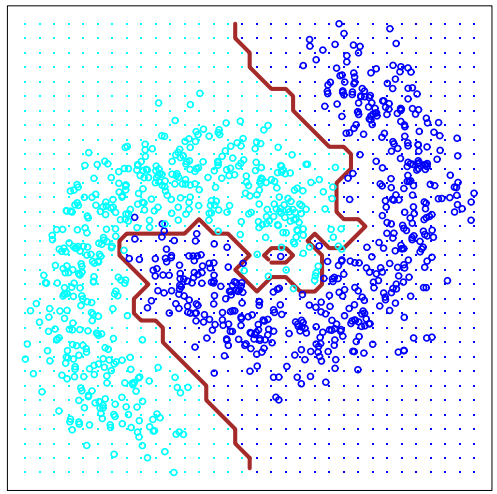
Decision Boundaries from kNearest Neighbors



(a) Underfitting .



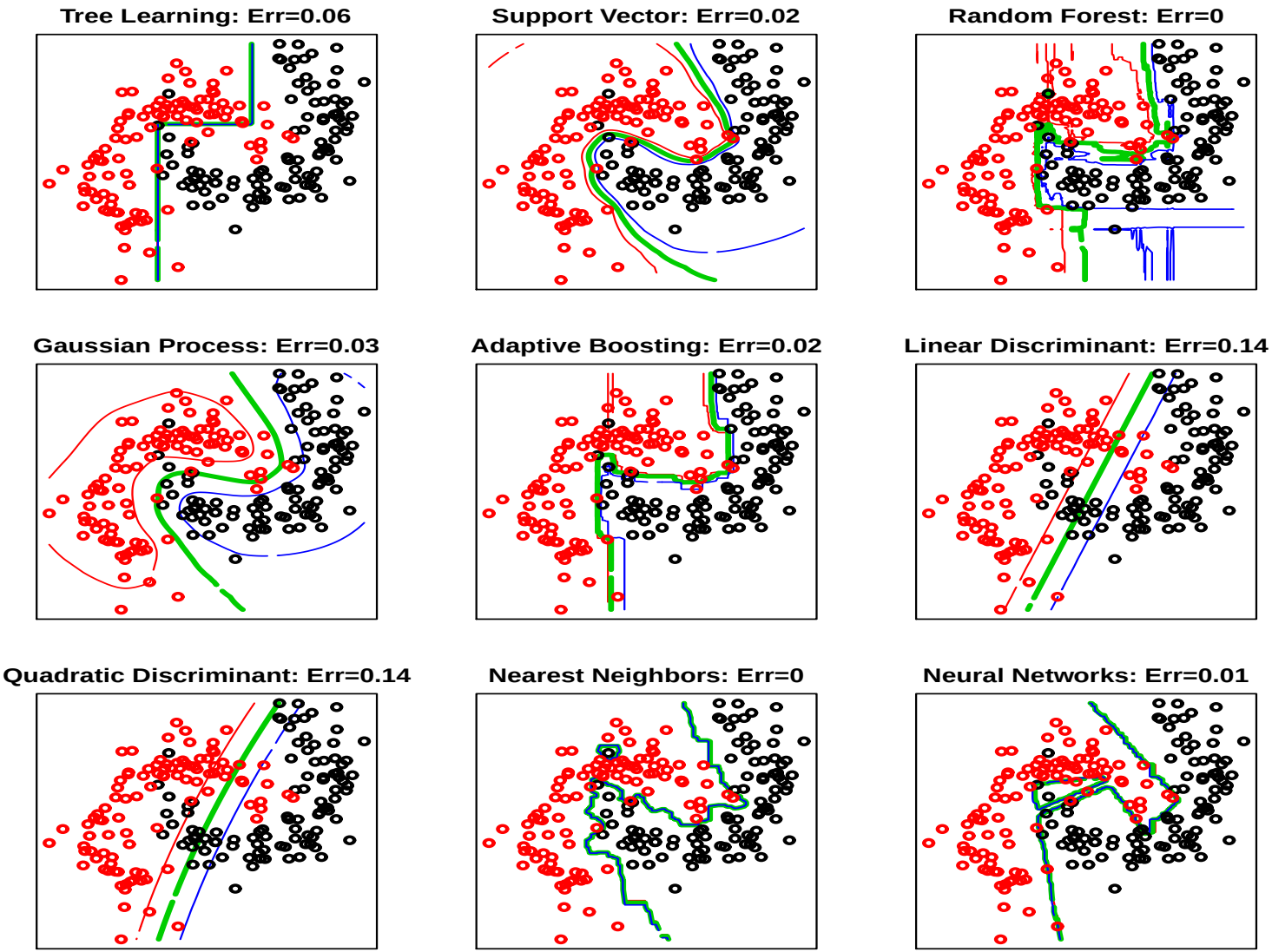
(b) Optimally



(c) Overfitting

Figure: *Depicted above are kNN decision boundaries for the banana classification task. Nearest Neighbors learning machines are referred to as lazy learners because they do not yield models constructed in the typical sense, which makes kNN purely predictive. However, implicitly, kNN does correspond to an underlying model with varying degrees of complexity, herein captured via probabilities that help build the decision boundaries.*

kNN boundaries compared to other decision boundaries



Assessing the Predictive Performance of the Classifier

- *[True Positives (TP)]: The True Positives (TP) count is the number of positives correctly classified as positives.*
- *[False Positives (FP)]: The False Positives (FP) count is means the number of negatives incorrectly classified as positives.*
- *[True Negatives (TN)]: The True Negatives (TN) count is the number of negatives correctly classified as negatives.*
- *[False Negatives (FN)]: The False Negatives (FN) count is the number of positives incorrectly classified as negatives.*

Assessing the Predictive Performance of the Classifier

		Prediction	
		Negative	*Positive*
Actual	*Negative*	TN	FP
	Positive	FN	TP

Table: *Counts of possible outcomes of binary classification*

With the $0-1$ *coding, our software delivers a confusion matrix of the form*

		Prediction	
		0	1
Actual	0	TN	FP
	1	FN	TP

Elements of the Performance the Classifier

Let f denote our binary classifier. Then f is a mapping from the input space $\mathcal{X}$ (usually a subset of $\mathbb{R}^p$) to the indicator set $\{0, 1\}$. In other words, to each $\mathrm{x} \in \mathcal{X}$, the function f assigns 0 or 1.

$$f : \mathcal{X} \to \{0, 1\}$$

The estimated accuracy of the classifier (estimated percentage of correct classification) is the most common measure of performance.

$$\widehat{\texttt{accuracy}}(f) = \Pr(\widehat{Y = f(X)}) = \frac{\textit{number of correctly classified}}{\textit{sample size}} = \frac{\texttt{TP} + \texttt{TN}}{\texttt{TP} + \texttt{TN} + \texttt{FP} + \texttt{FN}}$$

The number of correctly classified is the sum of the diagonal elements of the confusion matrix. The False Positive Rate (FPR) and the True Positive Rate are also very important measures.

$$\texttt{FPR} = \frac{\texttt{FP}}{\texttt{FP} + \texttt{TN}} \qquad \textit{and} \qquad \texttt{TPR} = \frac{\texttt{TP}}{\texttt{TP} + \texttt{FN}}$$

Elements of the Performance the Classifier

For reasons that will be made clear later, the accuracy of a classifier alone does not tell enough of the story. Consider the following two measures, namely precision *and* recall*. Precision in this context measures the proportion of estimated positives that are truly positives.*

$$\text{precision} = \frac{\textit{true positives}}{\textit{true positives} + \textit{false positives}}$$

In other words, of all those that were classified as positive, which proportion was indeed positive? Now, a 100% *precision says that the classifier all those declared positive where indeed positive. However, it does not necessarily account for all the positive that are there. On the other hand,*

$$\text{recall} = \frac{\textit{true positives}}{\textit{true positives} + \textit{false negatives}}$$

In other words, of all those that are actually positive, what proportion did the classifier succeed at classifying correctly. Here a 100% *recall says that all positives are classified as positives. However, it does not account for the goodness of all the positive test results.*

Elements of the Performance the Classifier

Another important measure is specificity *defined by*

$$\text{Specificity} = \frac{\textit{true negatives}}{\textit{true negatives} + \textit{false positives}}$$

`The` f`-measure:` *Clearly, there is a dilemma between* Precision *and* Recall. *It seems therefore reasonable to the solve this dilemma by combining* precision *and* recall *is such a way that a trade-off is achieved. Hence the use of a much more elaborate measure of goodness that combines both* precision *and* recall. *This measure is known as the so-called* f *measure defined simply by*

$$f = 2\frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} == \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}}$$

The famous concept of Receiver Operating Characteristic (ROC) curve plots the The False Positive Rate (FPR) and the True Positive Rate (TPR) as a way to assess the quality of a classification technique.

Loss functions

1. *When $f(X)$ is used to predict Y, a loss is incurred.*
   - *Question: How is such a loss quantified?*
   - *Answer: Define a suitable loss function.*
2. *Common loss functions in regression*
   - *Squared error loss or (ℓ_2) loss*

$$\ell(Y, f(X)) = (Y - f(X))^2$$

   ℓ_2 is by far the most used (prevalent) because of its differentiability. Unfortunately, not very robust to outliers.
   - *Absolute error loss or (ℓ_1) loss*

$$\ell(Y, f(X)) = |Y - f(X)|$$

   ℓ_1 is more robust to outliers, but not differentiable at zero.
3. *Note that $\ell(Y, f(X))$ is a random variable.*

Risk Functionals and Cost Functions

1. *Definition of a risk functional,*

$$R(f) = \mathbb{E}[\ell(Y, f(X))] = \int_{\mathcal{X}\times\mathcal{Y}} \ell(\mathrm{y}, f(\mathrm{x}))p_{XY}(\mathrm{x}, \mathrm{y})d\mathrm{x}d\mathrm{y}$$

   $R(f)$ is the expected loss over all pairs of the cross space $\mathcal{X} \times \mathcal{Y}$.

2. *Ideally, one seeks the best out of all possible functions, i.e.,*

$$f^*(X) = \underset{f}{\mathtt{arg\,min}}\, R(f) = \underset{f}{\mathtt{arg\,min}}\, \mathbb{E}[\ell(Y, f(X))]$$

   $f^*(\cdot)$ is such that

$$R^* = R(f^*) = \min_f R(f)$$

3. *This ideal function cannot be found in practice, because the fact that the distributions are unknown, make it impossible to form an expression for $R(f)$.*

Cost Functions and Risk Functionals

- *Theorem: Under regularity conditions,*

$$f^*(X) = \mathbb{E}[Y|X] = \underset{f}{\texttt{arg min}}\, \mathbb{E}[(Y - f(X))^2]$$

Under the squared error loss, the optimal function f^* that yields the best prediction of Y given X is no other than the expected value of Y given X.

- *Since we know neither $p_{XY}(\mathrm{x}, \mathrm{y})$ nor $p_X(\mathrm{x})$, the conditional expectation*

$$\mathbb{E}[Y|X] = \int_{\mathcal{Y}} \mathrm{y} p_{Y|X}(\mathrm{y})(d\mathrm{y}) = \int_{\mathcal{Y}} \mathrm{y} \frac{p_{XY}(\mathrm{x}, \mathrm{y})}{p_X(\mathrm{x})} d\mathrm{y}$$

cannot be directly computed.

Empirical Risk Minimization

- *Let* $\mathcal{D} = \Big\{(X_1, Y_1), (X_2, Y_2), \cdots, (X_n, Y_n)\Big\}$ *represent an iid sample*
- *The empirical version of the risk functional is*

$$\widehat{R}(f) = \widehat{\texttt{MSE}}(f) = \mathbb{E}[\widehat{(Y - f(X))^2}] = \frac{1}{n}\sum_{i=1}^{n}(Y_i - f(X_i))^2$$

It turns out that $\widehat{R}(f)$ *provides an unbiased estimator of* $R(f)$*.*

- *We therefore seek the best by empirical standard,*

$$\hat{f}^*(X) = \underset{f}{\texttt{arg min}}\, \widehat{\texttt{MSE}}(f) = \underset{f}{\texttt{arg min}} \left\{\frac{1}{n}\sum_{i=1}^{n}(Y_i - f(X_i))^2\right\}$$

Since it is impossible to search all possible functions, it is usually crucial to choose the "right" function space.

Function spaces

For the function estimation task for instance, one could assume that the input space $\mathcal{X}$ is a closed and bounded interval of $\mathbb{R}$, i.e. $\mathcal{X} = [a, b]$, and then consider estimating the dependencies between $\boldsymbol{x}$ and y from within the space $\mathcal{F}$ all bounded functions on $\mathcal{X} = [a, b]$, i.e.,

$$\mathcal{F} = \{f : \mathcal{X} \to \mathbb{R} | \ \exists B \geq 0, \ \text{such that } |f(\boldsymbol{x})| \leq B, \quad \text{for all } \boldsymbol{x} \in \mathcal{X}\}.$$

One could even be more specific and make the functions of the above $\mathcal{F}$ continuous, so that the space to search becomes

$$\mathcal{F} = \{f : [a, b] \to \mathbb{R} | \quad f \ \text{ is continuous}\} = C([a, b]),$$

which is the well-known space of all continuous functions on a closed and bounded interval $[a, b]$. This is indeed a very important function space.

Space of Univariate Polynomials

In fact, polynomial regression consists of searching from a function space that is a subspace of $C([a,b])$. *In other words, when we are doing the very common polynomial regression, we are searching the space*

$$\mathcal{P}([a,b]) = \{f \in C([a,b])|\ f \text{ is a polynomial with real coefficients}\}.$$

It is interesting to note that Weierstrass did prove that $\mathcal{P}([a,b])$ *is dense in* $C([a,b])$. *One considers the space of all polynomial of some degree p, i.e.,*

$$\mathcal{F} = \mathcal{P}^p([a,b]) = \left\{ f \in C([a,b])|\ \exists \beta_0, \beta_1, \cdots, \beta_p \in \mathbb{R}| \right.$$
$$\left. f(\boldsymbol{x}) = \sum_{j=0}^{p} \beta_j \boldsymbol{x}^j,\ \forall \boldsymbol{x} \in [a,b] \right\}$$

Empirical Risk Minimization in $\mathcal{F}$

- *Having chosen a class $\mathcal{F}$ of functions, we can now seek*

$$\hat{f}(X) = \underset{f \in \mathcal{F}}{\texttt{arg min}}\, \widehat{\texttt{MSE}}(f) = \underset{f \in \mathcal{F}}{\texttt{arg min}} \left\{ \frac{1}{n} \sum_{i=1}^{n} (Y_i - f(X_i))^2 \right\}$$

We are seeking the best function in the function space chosen.

- *For instance, if the function space in the space of all polynomials of degree p in some interval $[a, b]$, finding $\hat{f}$ boils down to estimating the coefficients of the polynomial using the data, namely*

$$\hat{f}(\boldsymbol{x}) = \hat{\beta}_0 + \hat{\beta}_1 \boldsymbol{x} + \hat{\beta}_2 \boldsymbol{x}^2 + \cdots + \hat{\beta}_p \boldsymbol{x}^p$$

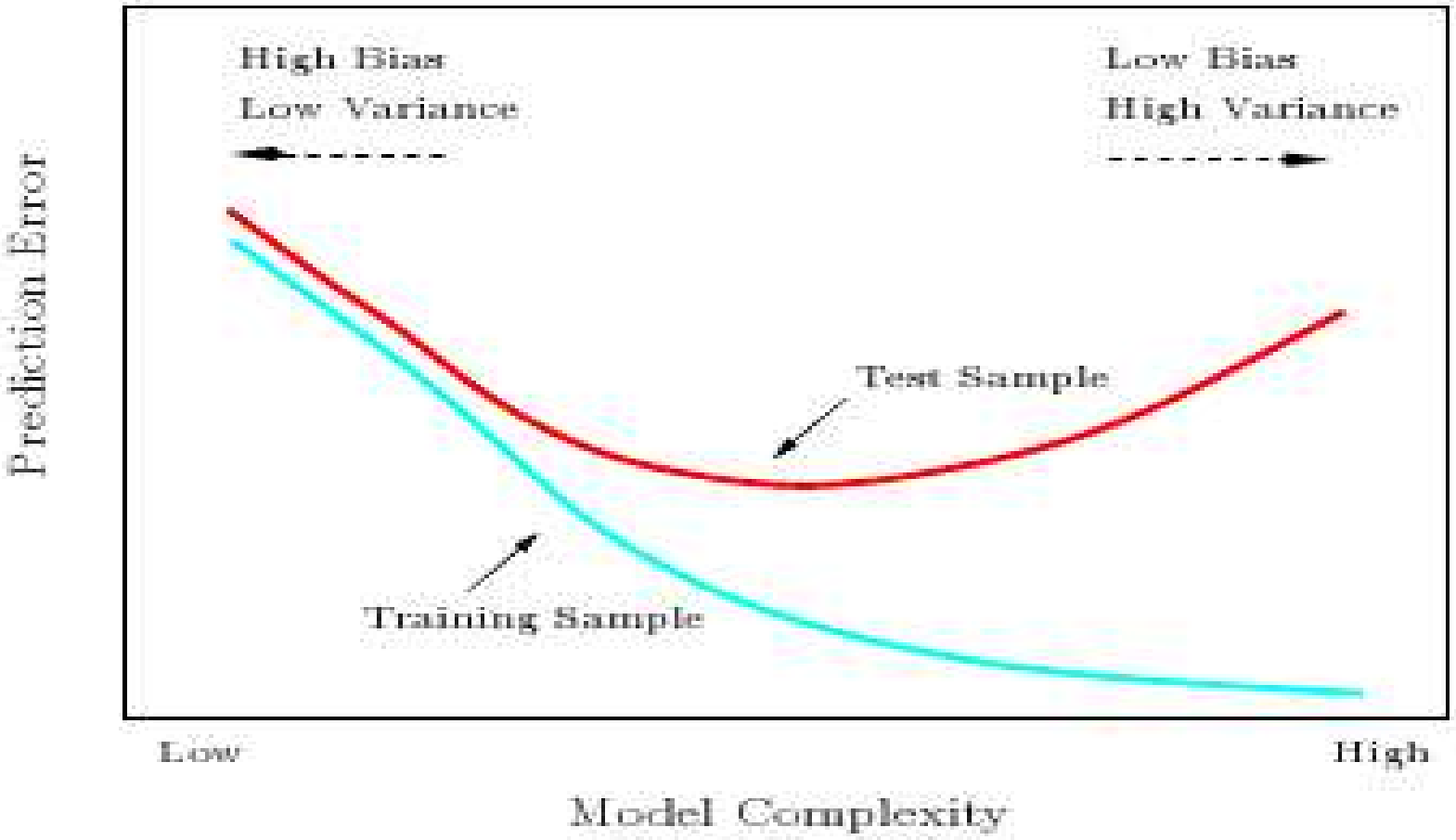
where using $\boldsymbol{\beta} = (\beta_0, \beta_1, \cdots, \beta_p)^\top$, we have

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\texttt{arg min}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(Y_i - \sum_{j=0}^{p} \beta_j \boldsymbol{x}_i^j \right)^2 \right\}$$

Important Aspects of Statistical Learning

- *It is very tempting at first to use the data at hand to find/build the* $\hat{f}$ *that makes* $\widehat{\texttt{MSE}}(\hat{f})$ *is the smallest. For instance, the higher the value of p, the smaller* $\widehat{\texttt{MSE}}(\hat{f}(\cdot))$ *will get.*
- *The estimate* $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \cdots, \hat{\beta}_p)^\top$ *of* $\boldsymbol{\beta} = (\beta_0, \beta_1, \cdots, \beta_p)^\top$*, is a random variable, and as a result the estimate* $\hat{f}(\boldsymbol{x}) = \hat{\beta}_0 + \hat{\beta}_1 \boldsymbol{x} + \hat{\beta}_2 \boldsymbol{x}^2 + \cdots + \hat{\beta}_p \boldsymbol{x}^p$ *of* $f(\boldsymbol{x})$ *is also a random variable.*
- *Since* $\hat{f}(\boldsymbol{x})$ *is random variable, we must compute important aspects like its bias* $\mathbb{B}[\hat{f}(\boldsymbol{x})] = \mathbb{E}[\hat{f}(\boldsymbol{x})] - f(\boldsymbol{x})$ *and its variance* $\mathbb{V}[\hat{f}(\boldsymbol{x})]$*.*
- *We have a dilemma: If we make* $\hat{f}$ *complex (large p), we make the bias small but the variance is increased. If we make* $\hat{f}$ *simple (small p), we make the bias large but the variance is decreased.*
- *Most of Modern Statistical Learning is rich with model selection techniques that seek to achieve a trade-off between bias and variance to get the optimal model. Principle of parsimony (sparsity), Ockham's razor principle.*

Effect of Bias-Variance Dilemma of Prediction



- *Optimal Prediction achieved at the point of bias-variance trade-off.*

Theoretical Aspects of Statistical Regression Learning

- *Just like we have a VC bound for classification, there is one for regression, ie when* $\mathcal{Y} = \mathbb{R}$ *and*

$$\hat{R}_n(f) = \frac{1}{n}\sum_{i=1}^{n} |\mathrm{y}_i - f(\mathbf{x}_i)|^2 = \textit{Squared error loss}$$

Indeed, for every $f \in \mathcal{F}$*, with probability at least* $1 - \eta$*, we have*

$$R(f) \leq \frac{\hat{R}_n(f)}{(1 - c\sqrt{\delta})_+}$$

where

$$\delta = \frac{a}{n}\left[v + v\log\left(\frac{bn}{v}\right) - \log\left(\frac{\eta}{4}\right)\right]$$

- *Note once again as before that these bounds are not asymptotic*
- *Unfortunately these bounds are known to be very loose in practice.*

The pitfalls of memorization and overfitting

The trouble - limitation - with naively using a criterion on the whole sample lies in the fact, given a sample $(\boldsymbol{x}_1, \mathrm{y}_1), (\boldsymbol{x}_2, \mathrm{y}_2), \cdots, (\boldsymbol{x}_n, \mathrm{y}_n)$*, the function* $\hat{f}_{\texttt{memory}}$ *defined by*

$$\hat{f}_{\texttt{memory}}(\boldsymbol{x}_i) = \mathrm{y}_i, \qquad i = 1, \cdots, n$$

always achieves the best performance, since $\widehat{\texttt{MSE}}(\hat{f}_{\texttt{memory}}) = 0$*, which is the minimum achievable.*

`Where does the limitation of` $\hat{f}_{\texttt{memory}}$ `come from?` *Well,* $\hat{f}_{\texttt{memory}}$ *does not really learn the dependency between* X *and* Y*. While it may have some of it, it also grabs a lot of the noise in the data, and ends overfitting the data. As a result of not really learning the structure of the relationship between* X *and* Y *and only merely memorizing the present sample values,* $\hat{f}_{\texttt{memory}}$ `will predict very poorly when presented with observations that were not in the sample.`

Training Set Test Set Split

Splitting the data into training set and test set: *It makes sense to judge models (functions), not on how they perform with in sample observations, but instead how they perform on* out of sample cases. *Given a collection* $\mathcal{D} = (\boldsymbol{x}_1, \mathrm{y}_1), (\boldsymbol{x}_2, \mathrm{y}_2), \cdots, (\boldsymbol{x}_n, \mathrm{y}_n)$ *of pairs,*

- *Randomly split* $\mathcal{D}$ *into training set of size* $\mathtt{ntr}$ *and test set of size* $\mathtt{nte}$, *such that* $\mathtt{ntr} + \mathtt{nte} = n$
  - *Training set*

$$\mathtt{Tr} = \left\{ (\boldsymbol{x}_1^{(\mathtt{tr})}, \mathrm{y}_1^{(\mathtt{tr})}), (\boldsymbol{x}_2^{(\mathtt{tr})}, \mathrm{y}_2^{(\mathtt{tr})}), \cdots, (\boldsymbol{x}_{\mathtt{ntr}}^{(\mathtt{tr})}, \mathrm{y}_{\mathtt{ntr}}^{(\mathtt{tr})}) \right\}$$

  - *Training set*

$$\mathtt{Te} = \left\{ (\boldsymbol{x}_1^{(\mathtt{te})}, \mathrm{y}_1^{(\mathtt{te})}), (\boldsymbol{x}_2^{(\mathtt{te})}, \mathrm{y}_2^{(\mathtt{te})}), \cdots, (\boldsymbol{x}_{\mathtt{nte}}^{(\mathtt{te})}, \mathrm{y}_{\mathtt{nte}}^{(\mathtt{te})}) \right\}$$

Training Set Test Set Split

- *For each function class $\mathcal{F}$ (linear models, nonparametrics, etc ...)*
  - *Find the best in its class based on the training set* $\texttt{Tr}$
  - *For all the estimated functions* $\hat{f}_1, \hat{f}_2, \cdots, \hat{f}_m$*, compute the training error*

$$\widehat{\texttt{MSE}}_{\texttt{Tr}}(\hat{f}_j) = \frac{1}{\texttt{ntr}} \sum_{i=1}^{\texttt{ntr}} (\texttt{y}_i^{(\texttt{tr})} - \hat{f}_j(\boldsymbol{x}_i^{(\texttt{tr})}))^2$$

  - *For all the estimated functions* $\hat{f}_1, \hat{f}_2, \cdots, \hat{f}_m$*, compute the test error*

$$\widehat{\texttt{MSE}}_{\texttt{Te}}(\hat{f}_j) = \frac{1}{\texttt{nte}} \sum_{i=1}^{\texttt{nte}} (\texttt{y}_i^{(\texttt{te})} - \hat{f}_j(\boldsymbol{x}_i^{(\texttt{te})}))^2$$

- *Compute the averages of both* $\widehat{\texttt{MSE}}_{\texttt{Tr}}$ *and* $\widehat{\texttt{MSE}}_{\texttt{Te}}$ *over many random splits of the data, and tabulate (if necessary) those averages.*
- *Select* $\hat{f}_{j^*}$ *such that*

$$\texttt{mean}[\widehat{\texttt{MSE}}_{\texttt{Te}}(\hat{f}_{j^*})] < \texttt{mean}[\widehat{\texttt{MSE}}_{\texttt{Te}}(\hat{f}_j)], \qquad j = 1, 2, \cdots, m, \quad j \neq j^*$$

Computational Comparisons

- *Ideally, we would like to compare the true theoretical performances measured by the risk functional*

$$\mathcal{R}(f) = \mathbb{E}[\ell(Y, f(X))] = \int_{\mathcal{X}\times\mathcal{Y}} \ell(\mathbf{x}, \mathrm{y}) dP(\mathbf{x}, \mathrm{y}), \tag{6}$$

- *Instead, we build the estimators using other optimality criteria, and then compare their predictive performances using the average test error* $\texttt{AVTE}(\cdot)$*, namely*

$$\texttt{AVTE}(\widehat{f}) = \frac{1}{R}\sum_{r=1}^{R}\left\{\frac{1}{m}\sum_{t=1}^{m}\ell(\mathrm{y}_{i_t}^{(r)}, \widehat{f}_r(\mathbf{x}_{i_t}^{(r)}))\right\}, \tag{7}$$

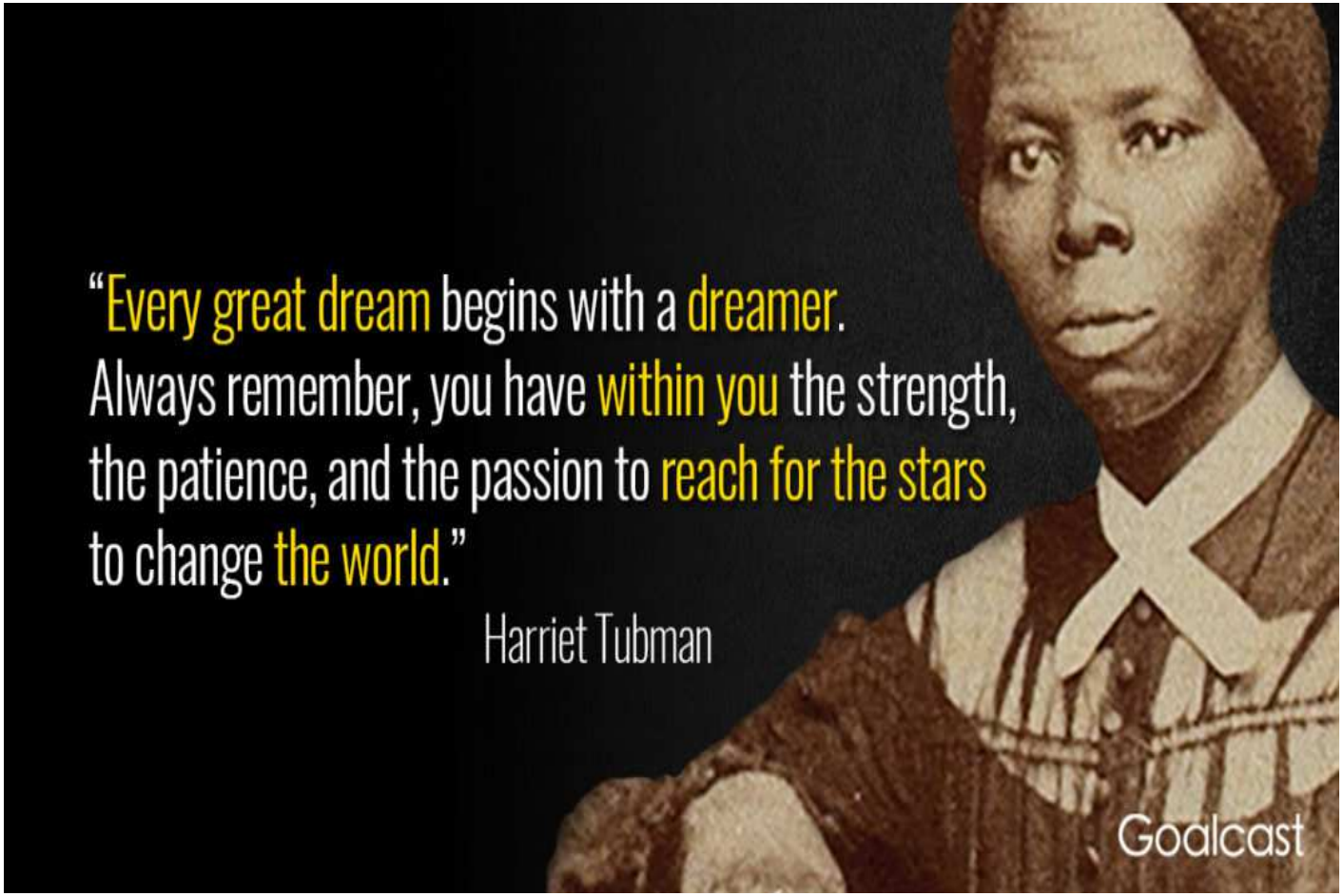
where $\widehat{f}_r(\cdot)$ *is the* r*-th realization of the estimator* $\widehat{f}(\cdot)$ *built using the training portion of the split of* $\mathcal{D}$ *into training set and test set, and* $\left(\mathbf{x}_{i_t}^{(r)}, \mathrm{y}_{i_t}^{(r)}\right)$ *is the* t*-th observation from the test set at the* r*-th random replication of the split of* $\mathcal{D}$*.*

Roadmap for Statistical Machine Learning

- *Applications:* Sharpen your intuition and your commonsense by questioning things, reading about interesting open applied problems, and attempt to solve as many problems as possible
- *Methodology:* Read and learn about the fundamental of statistical estimation and inference, get acquainted with the most commonly used methods and techniques, and consistently ask yourself and others what the natural extensions of the techniques could be.
- *Computation:* Learn and master at least two programming languages. I strongly recommend getting acquainted with *R*

  http://www.r-project.org
- *Theory:* "Nothing is more practical than a good theory" (Vladimir N. Vapnik). When it comes to data mining and machine learning and predictive analytics, those who truly understand the inner workings of algorithms and methods always solve problems better.

How will you uniquely change the world?




THANK
YOU

References

[1] *Clarke, B., Fokoué, E. and Zhang, H. H. (2009).* Principles and Theory for Data Mining and Machine Learning. *Springer Verlag, New York, (ISBN: 978-0-387-98134-5), (2009)*

[2] *James, G, Witten, D, Hastie, T and Tibshirani, R (2013).* An Introduction to Statistical Learning with Applications in R. *Springer, New York, (e-ISBN: 978-1-4614-7138-7),(2013)*

[3] *Vapnik, N. V.(1998).* Statistical Learning Theory. *Wiley, ISBN: 978-0-471-03003-4, (1998)*

[4] *Vapnik, N. V.(2000).* The Nature of Statistical Learning Theory. *Springer, ISBN 978-1-4757-3264-1, (2000)*

[5] *Hastie, T. and Tibshirani, R. and Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd Edition. Springer, ISBN 978-0-387-84858-7*