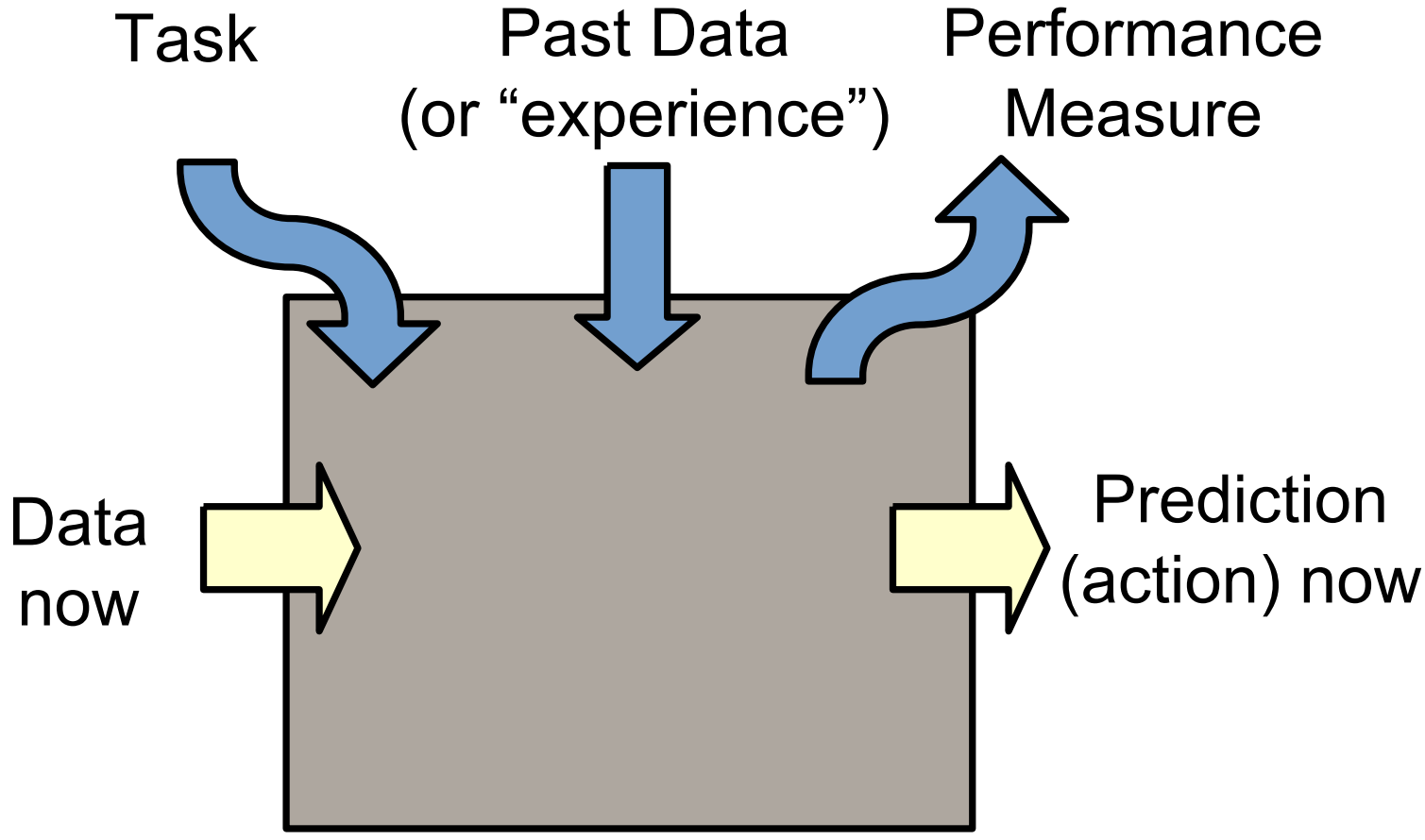
Machine Learning (CS 181):

3. Probabilistic Interpretation and Basis Functions

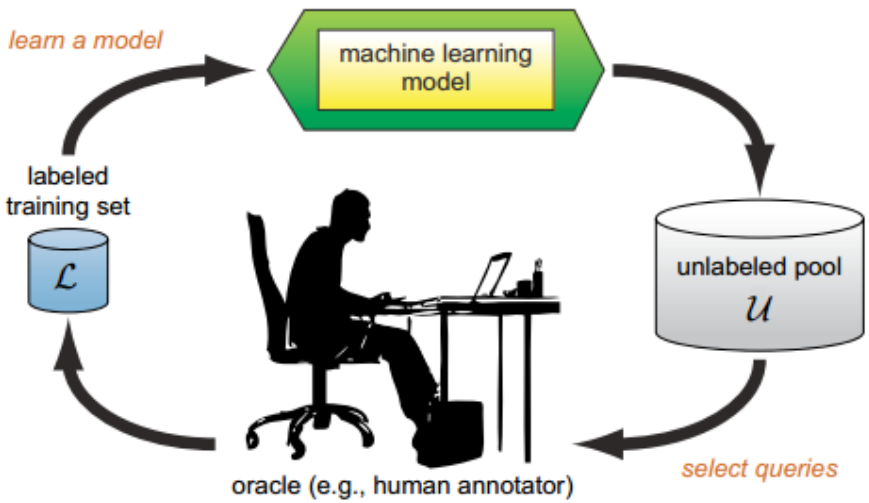
What is machine learning?



Active Learning

Q: What if we add new labeled data to make the model better? e.g.

- Ask users which movies they like.
- Allocate resources to analyze an area.
- ...



(Settles, 09)

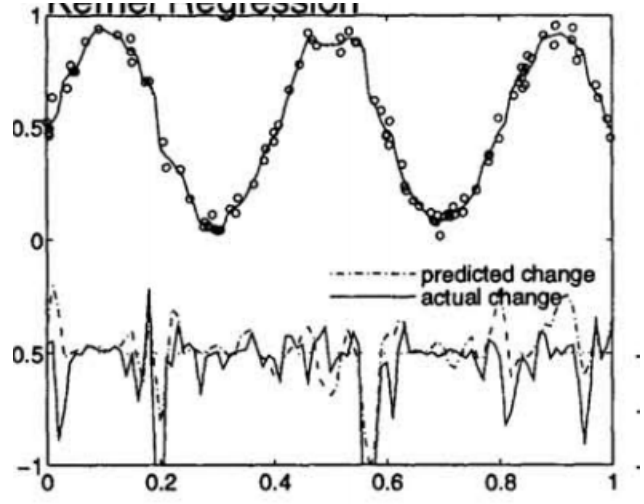
Active Learning with Statistical Models

David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan
cohn@psyche.mit.edu, zoubin@psyche.mit.edu, jordan@psyche.mit.edu

Department of Brain and Cognitive Sciences
Massachusetts Institute of Technology
Cambridge, MA 02139

- Derive active learning algorithms for neural networks and non-parametric regression.

We consider the problem of actively learning a mapping $X \rightarrow Y$ based on a set of training examples $\{(x_i, y_i)\}_{i=1}^{m}$, where $x_i \in X$ and $y_i \in Y$. The learner is allowed to iteratively select new inputs $\tilde{x}$ (possibly from a constrained set), observe the resulting output $\tilde{y}$, and incorporate the new examples $(\tilde{x}, \tilde{y})$ into its training set.



Active Learning with Statistical Models

David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan
cohn@psyche.mit.edu, zoubin@psyche.mit.edu, jordan@psyche.mit.edu

Department of Brain and Cognitive Sciences
Massachusetts Institute of Technology
Cambridge, MA 02139

Zoubin Ghahramani

From Wikipedia, the free encyclopedia

Zoubin Ghahramani FRS[9] (Persian: زوبین قهرمانی; born 8 February 1970)[1] is an Iranian researcher[3][10] and Professor of Information Engineering at the University of Cambridge. He holds joint appointments at Carnegie Mellon University[*citation needed*], University College London and the Alan Turing Institute.[*citation needed*] and has been a Fellow of St John's College, Cambridge since 2009.[1]

Zoubin Ghahramani

Contents

Contents

Model 1: Linear Regression

- Input: $\mathbf{x} \in \mathbb{R}^m$
- Output: $y \in \mathbb{R}$
- Parameters: $\mathbf{w} \in \mathbb{R}^m$
- Model:

$$h(\mathbf{x}; \mathbf{w}) = \mathbf{w}^\top \mathbf{x}$$

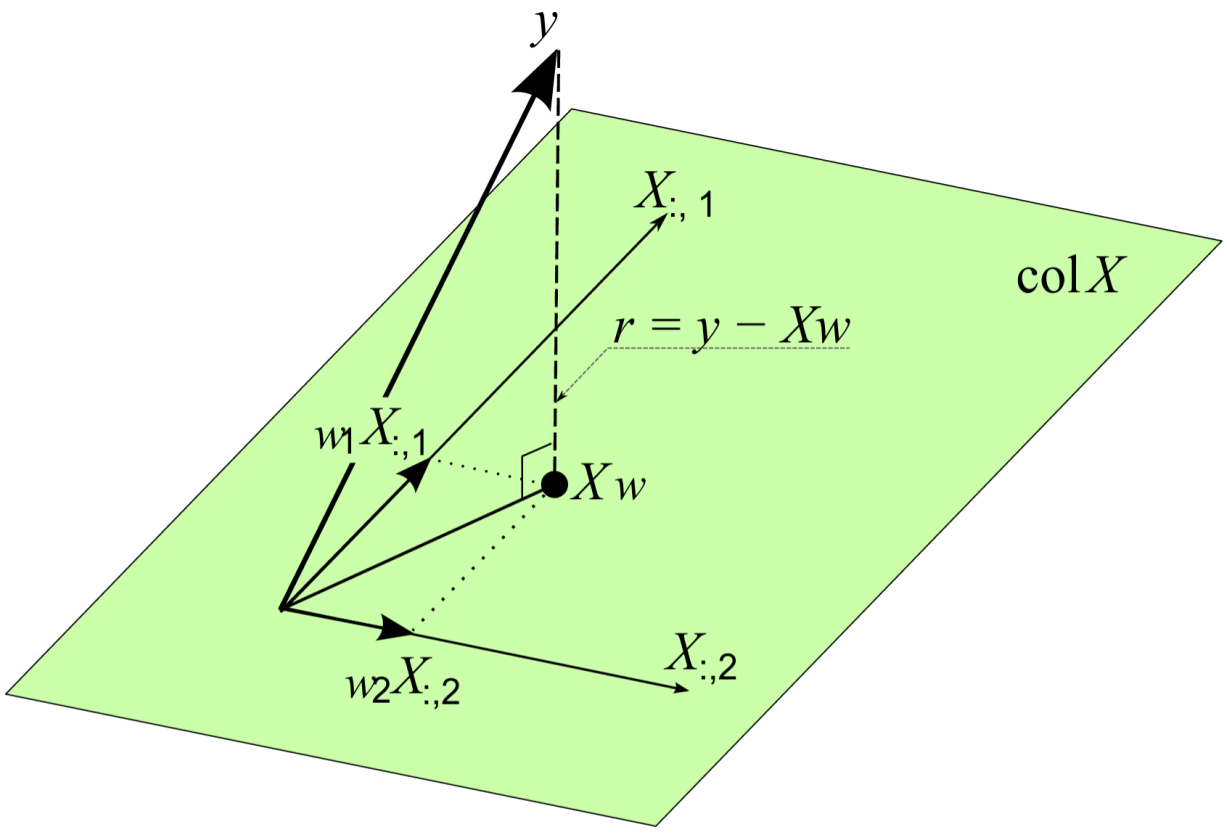
- Loss Function:

$$\mathcal{L}_D(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{n} \big(y_i - h(\mathbf{x}_i; \mathbf{w})\big)^2$$

- Optimization

$$\mathbf{w}^\star = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} = \arg\min_{\mathbf{w}} \mathcal{L}_D(\mathbf{w})$$

Linear Regression as Projection



Optimization

$$\mathbf{w}^\star = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} = \arg\min_{\mathbf{w}} \mathcal{L}_D(\mathbf{w})$$

Least Squares Loss

Given data, minimized least-squares loss,

$$D = (\mathbf{x}_1, y_1) \dots (\mathbf{x}_n, y_n) = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$$

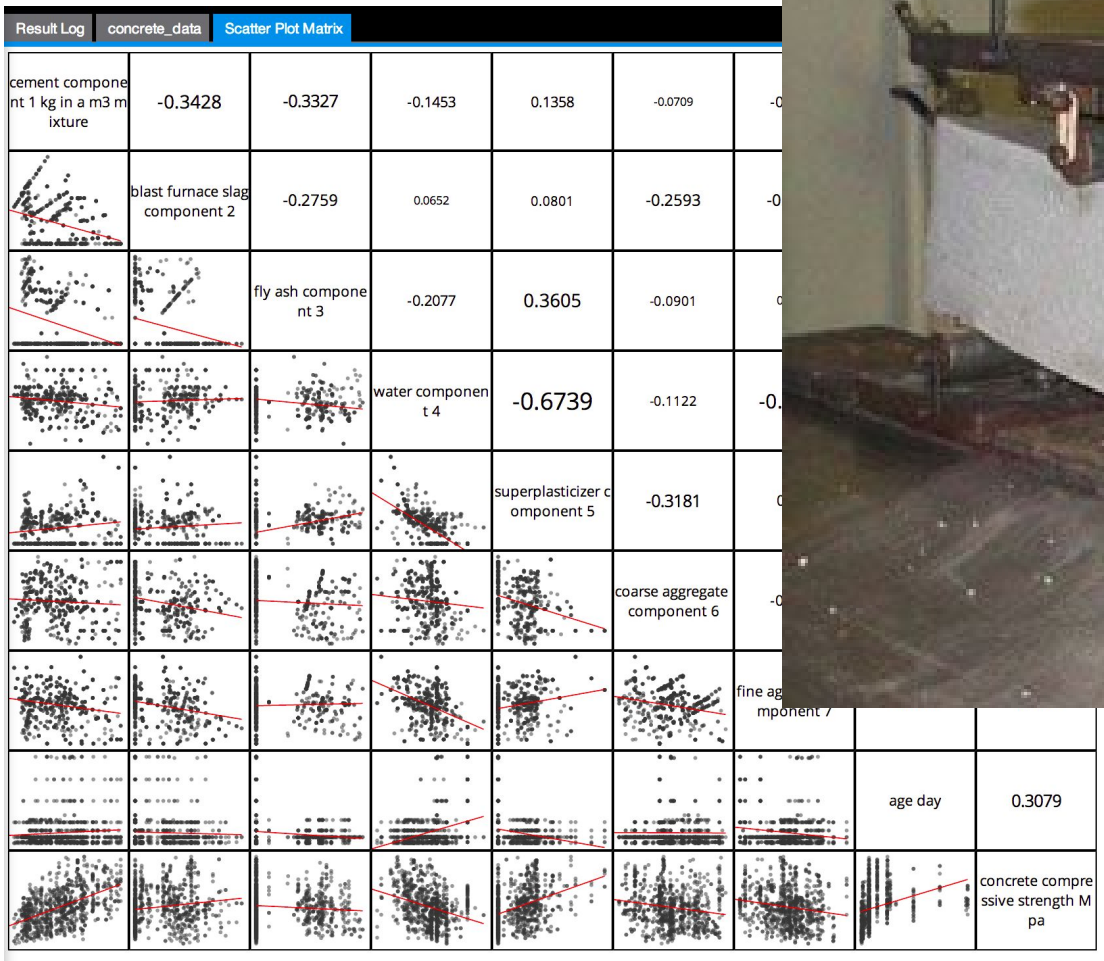
$$\mathcal{L}(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{n} (y_i - h(\mathbf{x}_i; \mathbf{w}))^2$$

Residual:

$$r_i = y_i - h(\mathbf{x}_i; \mathbf{w})$$

- Issue: least squares seems arbitrary (penalty increases quadratically)
- This class: Probabilistic view of this optimization

Regression



www.geocheminc.com alpine.atlassian.net

http://www.hindawi.com/journals/tswj/2014/381549/

Table 1: UCI dataset for estimating concrete compressive strength.

Name of the component	Data type	Measurement	Description	Symbol
Cement	Quantitative	kg in a m3 mixture	Input variable	**C**
Blast Furnace Slag	Quantitative	kg in a m3 mixture	Input variable	**BFS**
Fly Ash	Quantitative	kg in a m3 mixture	Input variable	**F**
Water	Quantitative	kg in a m3 mixture	Input variable	**W**
Superplasticizer	Quantitative	kg in a m3 mixture	Input variable	**S**
Coarse Aggregate	Quantitative	kg in a m3 mixture	Input variable	**CA**
Fine Aggregate	Quantitative	kg in a m3 mixture	Input variable	**FA**
Age	Quantitative	Day (1~365)	Input variable	**A**
Concrete compressive strength	Quantitative	MPa	Output variable	**CCS**

Exploring Data: Residuals

2009

OkCupid QuickMatch Scores

men rating women	ASIAN women	BLACK women	LATINA women	WHITE women
ASIAN men rating...	11%	-16%	-1%	7%
BLACK men rating...	3%	-3%	3%	-3%
LATINO men rating...	7%	-22%	6%	9%
WHITE men rating...	6%	-18%	2%	10%

women rating men	ASIAN men	BLACK men	LATINO men	WHITE men
ASIAN women rating...	10%	-14%	-12%	16%
BLACK women rating...	-11%	16%	-4%	0%
LATINA women rating...	-16%	-4%	11%	10%
WHITE women rating...	-12%	-6%	1%	17%

http://blog.okcupid.com/index.php/page/8/

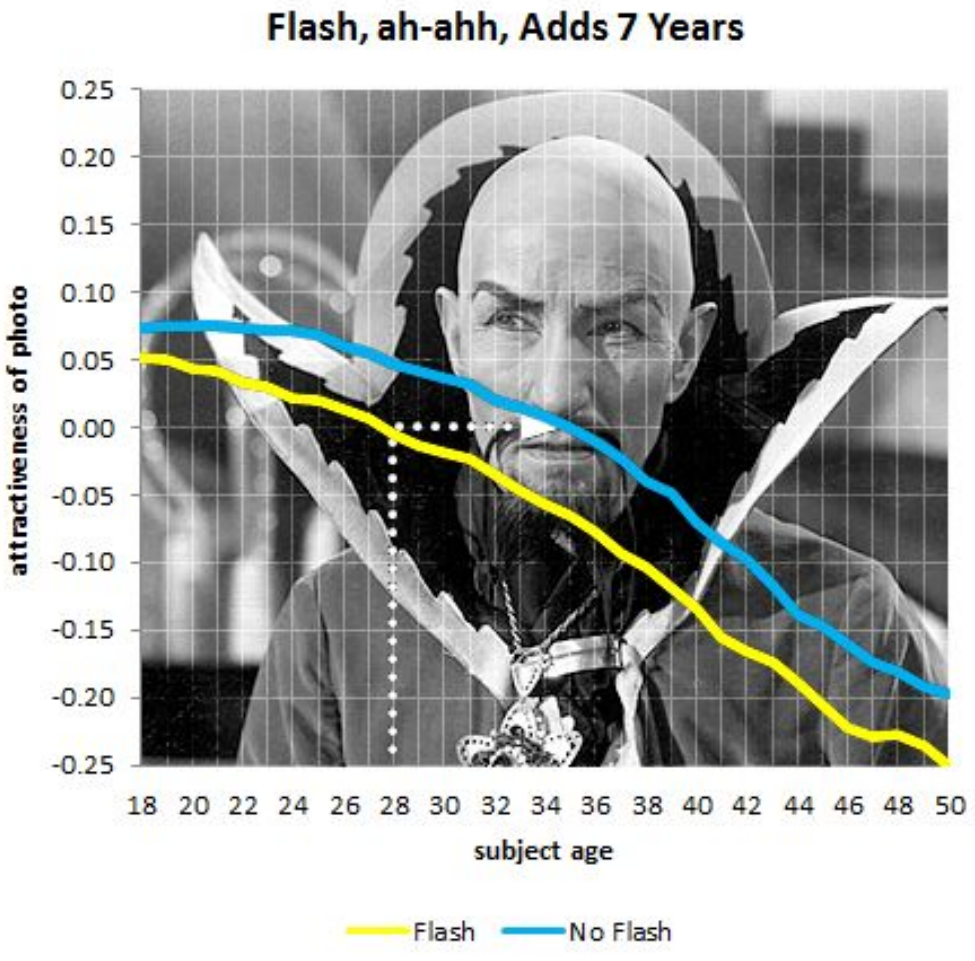
Exploring Data: Residuals

2009

OkCupid QuickMatch Scores

men rating women	ASIAN women	BLACK women	LATINA women	WHITE women
ASIAN men rating...	11%	-16%		
BLACK men rating...	3%	-3%		
LATINO men rating...	7%	-22%		
WHITE men rating...	6%	-18%		

women rating men	ASIAN men	BLACK men	LA
ASIAN women rating...	10%	-14%	
BLACK women rating...	-11%	16%	
LATINA women rating...	-16%	-4%	
WHITE women rating...	-12%	-6%	



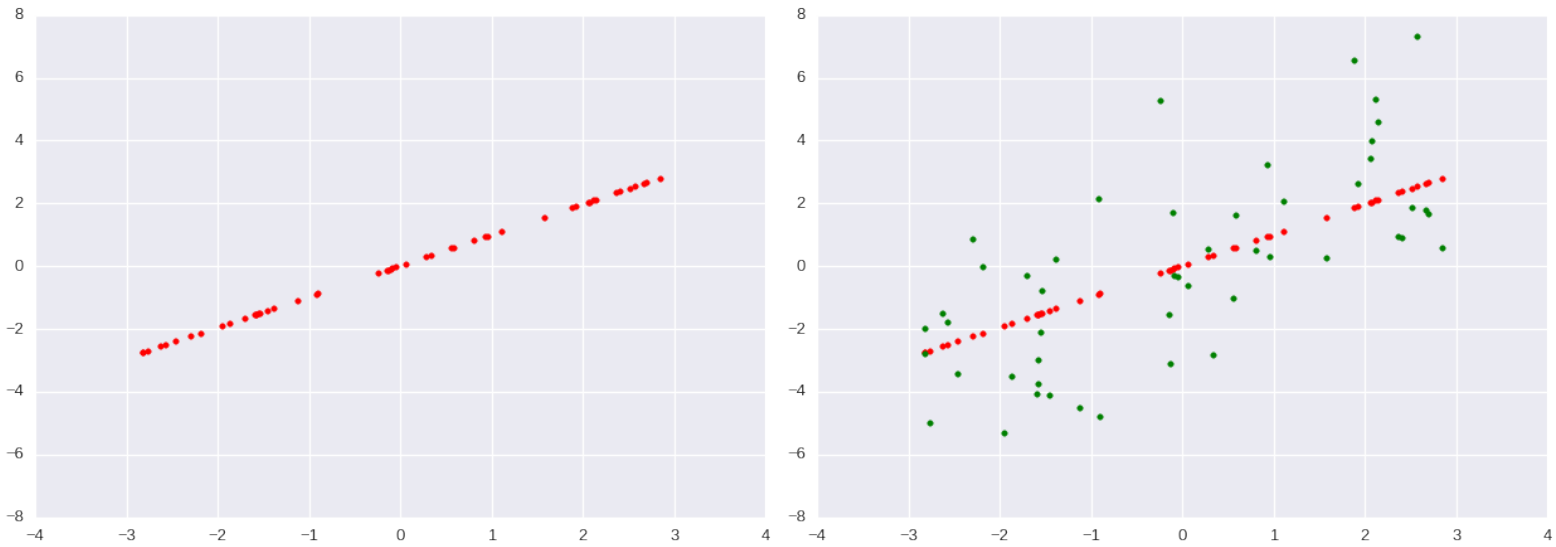
http://blog.okcupid.com/index.php/page/8/

Contents

Generative Model of Linear Regression

Generative Process:

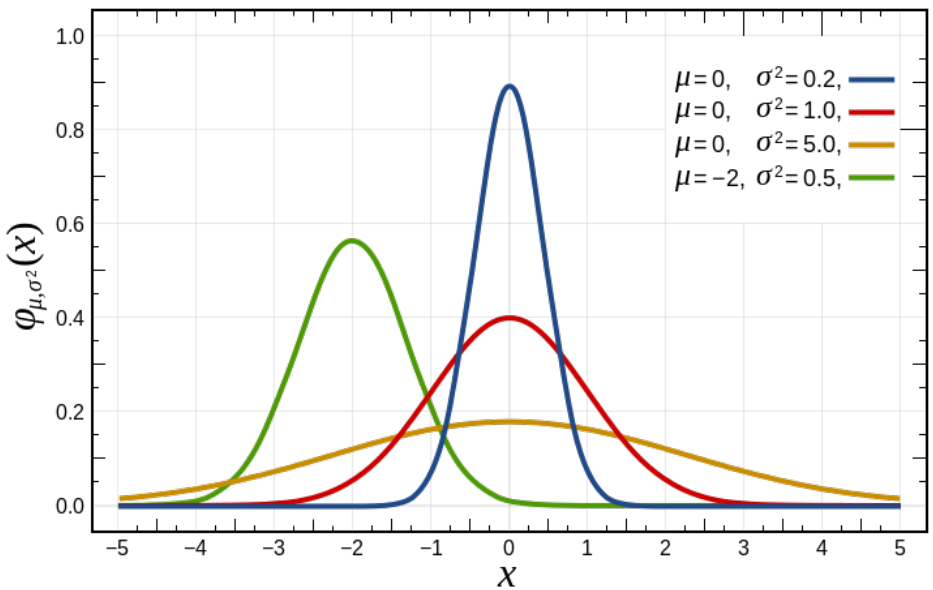
- Data is generated from a linear model with unknown parameters but corrupted with Gaussian noise.



Assume data comes from linear model, but we see noisy version.

Review: Gaussian Distribution

$$\begin{aligned}\mathcal{N}(z|\mu,\sigma^2) &= \frac{1}{\sqrt{2\pi\sigma^2}}\exp(-\frac{(z-\mu)^2}{2\sigma^2}) \\ &= \frac{1}{\sqrt{2\pi\sigma^2}}\exp(-\frac{1}{2}(z-\mu)\sigma^{-2}(z-\mu))\end{aligned}$$

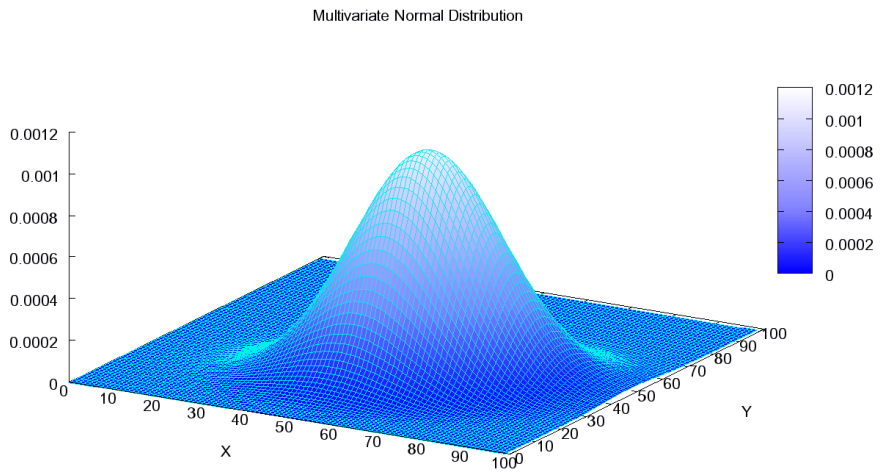


Density of example Gaussian distributions.

Multivariate Gaussian Distribution

$$\mathcal{N}(\mathbf{z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}|}} \exp(-\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{z}-\boldsymbol{\mu}))$$

- $\boldsymbol{\mu} \in \mathbb{R}^m$; mean vector
- $\boldsymbol{\Sigma} \in \mathbb{R}^{m \times m}$; covariance matrix



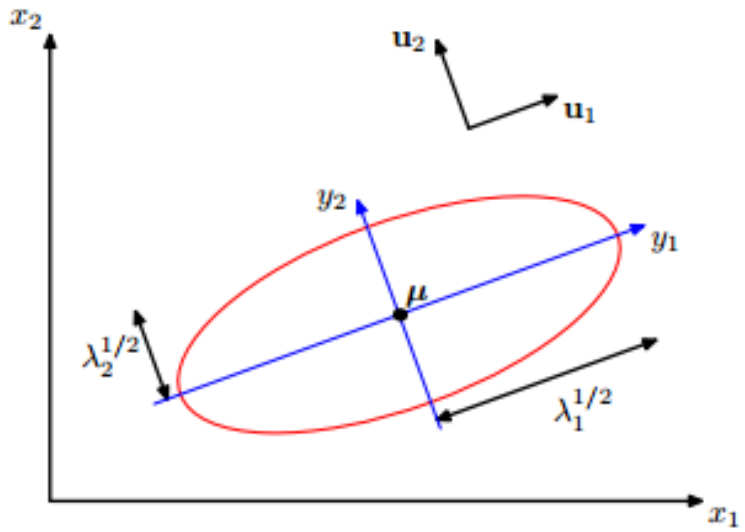
Density of 2D multivariate Gaussian

Covariance Matrix Properties

$$\mathcal{N}(\mathbf{z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}|}} \exp(-\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{z}-\boldsymbol{\mu}))$$

- $\boldsymbol{\Sigma} \in \mathbb{R}^{m \times m}$; symmetric and semi-definite (non-negative eigenvalues)
- Mahalanobis Distance (generalizes Euclidean distance $\boldsymbol{\Sigma} = \mathbf{I}$)

$$\Delta^2 = (\mathbf{z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{z}-\boldsymbol{\mu})$$



Bishop: Ellipse with density $\exp(-\frac{1}{2})$.

Jointly Gaussian Properties

- If Z_1 and Z_2 are Gaussian and independent with σ_1^2 and σ_2^2, they jointly form a bivariate Gaussian :

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$$

  (Generalizes to multivariate case)

- However a pair of jointly Gaussian variables need not be independent (correlation terms in Σ)

- Finally two non-independent Gaussians do not need to form a joint Gaussian distribution.

Jointly Gaussian Properties

- If Z_1 and Z_2 are Gaussian and independent with σ_1^2 and σ_2^2, they jointly form a bivariate Gaussian :

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$$

  (Generalizes to multivariate case)

- However a pair of jointly Gaussian variables need not be independent (correlation terms in $\mathbf{\Sigma}$)

- Finally two non-independent Gaussians do not need to form a joint Gaussian distribution.

Jointly Gaussian Properties

- If Z_1 and Z_2 are Gaussian and independent with σ_1^2 and σ_2^2, they jointly form a bivariate Gaussian :

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$$

  (Generalizes to multivariate case)

- However a pair of jointly Gaussian variables need not be independent (correlation terms in $\mathbf{\Sigma}$)

- Finally two non-independent Gaussians do not need to form a joint Gaussian distribution.

Multivariate Gaussian Distribution: Intuition

[IPython Demo]

Generative Model of Linear Regression

Generative Process:

- Data D is generated from a linear model $h(\mathbf{x}; \mathbf{w})$ with unknown parameters $\mathbf{w}$, but with Gaussian noise with variance β^{-1}.

$$y_i = h(\mathbf{x}; \mathbf{w}) + \epsilon_n$$
$$\epsilon_n \sim \mathcal{N}(0, \beta^{-1}).$$

- Therefore,

$$p(y|\mathbf{x}, \mathbf{w}) = \mathcal{N}(y|h(\mathbf{x}; \mathbf{w}), \beta^{-1})$$

- Likelihood of data set (regression) : $p(y|\mathbf{X}, \mathbf{w})$

Generative Model of Linear Regression

Generative Process:

- Data D is generated from a linear model $h(\mathbf{x}; \mathbf{w})$ with unknown parameters $\mathbf{w}$, but with Gaussian noise with variance β^{-1}.

$$y_i = h(\mathbf{x}; \mathbf{w}) + \epsilon_n$$
$$\epsilon_n \sim \mathcal{N}(0, \beta^{-1}).$$

- Therefore,

$$p(y|\mathbf{x}, \mathbf{w}) = \mathcal{N}(y|h(\mathbf{x}; \mathbf{w}), \beta^{-1})$$

- Likelihood of data set (regression) : $p(y|\mathbf{X}, \mathbf{w})$

Maximum Likelihood Estimation

1. Describe a generative process behind the data.
2. Write down the likelihood of the observed data D.
3. Select parameters that make the data most likely.

- General strategy for parameter estimation:

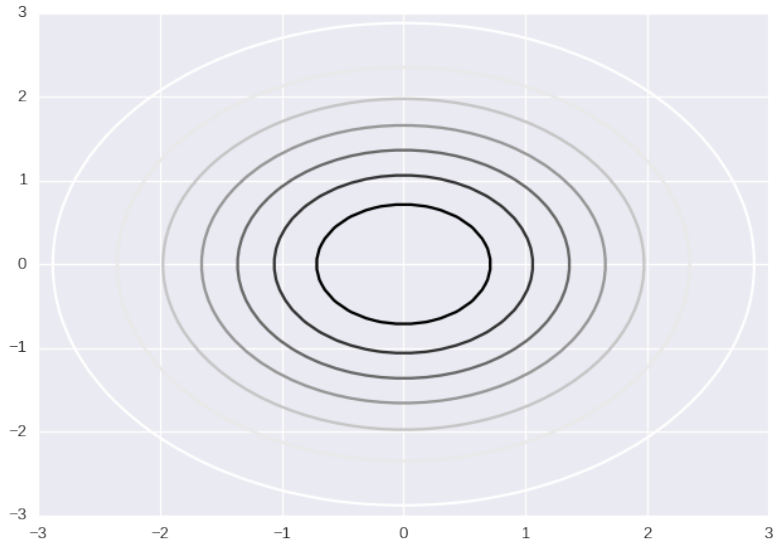
$$\mathbf{w}^* = \arg\max_{\mathbf{w}} p(\mathbf{y}|\mathbf{X}, \mathbf{w}) = \arg\max_{\mathbf{w}} \prod_{i=1}^{n} p(y_i|\mathbf{x}_i, \mathbf{w})$$

MLE For Linear Regression

Linear regression data likelihood under independent noise assumption:

$$
\begin{aligned}
p(\mathbf{y}|\mathbf{X}, \mathbf{w}) &= \prod_{i=1}^{n} \mathcal{N}(y_i|h(\mathbf{x}_i; \mathbf{w}), \beta^{-1}) \\
&= \mathcal{N}(\mathbf{y}|\mathbf{X}\mathbf{w}, \mathbf{I}\beta^{-1})
\end{aligned}
$$

- Note: Diagonal covariance indicate noise independence.



Maximizing Likelihood

- Maximum likelihood estimates is done by *minimizing* negative log-likelihood.
- MLE loss function:

$$\mathcal{L}(\mathbf{w}) = -\ln p(\mathbf{y}|\mathbf{X}, \mathbf{w})$$

Maximizing Likelihood 2

$$\mathcal{N}(\mathbf{z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}|}} \exp(-\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{z}-\boldsymbol{\mu}))$$

$$\begin{aligned}
\mathcal{L}(\mathbf{w}) &= -\ln \mathcal{N}(\mathbf{y}|\mathbf{X}\mathbf{w}, \mathbf{I}\beta^{-1}) \\
&= c + \frac{1}{2}(\mathbf{y}-\mathbf{X}\mathbf{w})^\top (\mathbf{I}\beta)(\mathbf{y}-\mathbf{X}\mathbf{w}) \\
&= c + \frac{\beta}{2}(\mathbf{y}-\mathbf{X}\mathbf{w})^\top (\mathbf{y}-\mathbf{X}\mathbf{w})
\end{aligned}$$

where $c = \ln \sqrt{|2\pi\mathbf{I}\beta^{-1}|} = (n/2)\ln 2\pi\beta$ is a term not dependent on $\mathbf{w}$

Maximizing Likelihood 2

$$\mathcal{N}(\mathbf{z}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}|}}\exp(-\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{z}-\boldsymbol{\mu}))$$

$$\begin{aligned}\mathcal{L}(\mathbf{w}) &= -\ln\mathcal{N}(\mathbf{y}|\mathbf{X}\mathbf{w},\mathbf{I}\beta^{-1}) \\ &= c+\frac{1}{2}(\mathbf{y}-\mathbf{X}\mathbf{w})^\top(\mathbf{I}\beta)(\mathbf{y}-\mathbf{X}\mathbf{w}) \\ &= c+\frac{\beta}{2}(\mathbf{y}-\mathbf{X}\mathbf{w})^\top(\mathbf{y}-\mathbf{X}\mathbf{w})\end{aligned}$$

where $c = \ln\sqrt{|2\pi\mathbf{I}\beta^{-1}|} = (n/2)\ln 2\pi\beta$ is a term not dependent on $\mathbf{w}$

Maximizing Likelihood 3

$$\mathcal{L}(\mathbf{w}) = c + \frac{\beta}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w})$$

Now taking gradients, (using chain rule)

$$\begin{aligned}\frac{\partial}{\partial \mathbf{w}}\mathcal{L}(\mathbf{w}) &= -\beta\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) \\ &= -\beta(\mathbf{X}^\top\mathbf{y} - \mathbf{X}^\top\mathbf{X}\mathbf{w})\end{aligned}$$

Maximizing Likelihood 4

$$
\begin{aligned}
\frac{\partial}{\partial \mathbf{w}} \mathcal{L}(\mathbf{w}) &= 0 \\
\mathbf{X}^\top \mathbf{y} - \mathbf{X}^\top \mathbf{X} \mathbf{w} &= 0 \\
\mathbf{X}^\top \mathbf{X} \mathbf{w} &= \mathbf{X}^\top \mathbf{y} \\
\mathbf{w}^* &= (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}
\end{aligned}
$$

Same least squares result.

Maximizing Likelihood 4

$$
\begin{aligned}
\frac{\partial}{\partial \mathbf{w}} \mathcal{L}(\mathbf{w}) &= 0 \\
\mathbf{X}^\top \mathbf{y} - \mathbf{X}^\top \mathbf{X} \mathbf{w} &= 0 \\
\mathbf{X}^\top \mathbf{X} \mathbf{w} &= \mathbf{X}^\top \mathbf{y} \\
\mathbf{w}^* &= (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}
\end{aligned}
$$

Same least squares result.

Why take a probabilistic approach?

- Allows us to get calibrated probabilty estimates $p(y|\mathbf{x})$
- Separates predictions from modeling
- A general framework for parameter estimation.
  - Can use to fit other parameters of the model.

MLE of Variance

Find MLE of β^{-1}. Estimate the variance/noise.

Let $\mathbf{w} = \mathbf{w}^*$ and including c we have,

$$\mathcal{L}(\mathbf{w}, \beta) = (n/2) \ln 2\pi\beta + \frac{\beta}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w})$$

$$\frac{\partial}{\partial \beta} \ln \mathcal{N}(\mathbf{y} \mid \mathbf{X}\mathbf{w}, \beta^{-1}\mathbf{I}) = \frac{n}{2\beta} - \frac{1}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) = 0$$

$$\frac{n}{\beta} = (\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w})$$

$$\beta^{-1} = \frac{1}{n}(\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w})$$

$$= \frac{1}{n}\sum_{i=1}^{n}(y_i - h(\mathbf{x}_i; \mathbf{w}))^2$$

MLE of Variance

Find MLE of β^{-1}. Estimate the variance/noise.

Let $\mathbf{w} = \mathbf{w}^*$ and including c we have,

$$\mathcal{L}(\mathbf{w}, \beta) = (n/2) \ln 2\pi\beta + \frac{\beta}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w})$$

$$\begin{aligned}
\frac{\partial}{\partial\beta} \ln \mathcal{N}(\mathbf{y} \,|\, \mathbf{X}\mathbf{w}, \beta^{-1}\mathbf{I}) &= \frac{n}{2\beta} - \frac{1}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) = 0 \\
\frac{n}{\beta} &= (\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) \\
\beta^{-1} &= \frac{1}{n}(\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w}) \\
&= \frac{1}{n}\sum_{i=1}^{n}(y_i - h(\mathbf{x}_i; \mathbf{w}))^2
\end{aligned}$$

Two Last Notes (For Homework)

Next classes will cover these topics:

- Regularization and *ridge regression*.

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}\left(y_i - h(\mathbf{x}_i;\mathbf{w})\right)^2 + \frac{\lambda}{2}\mathbf{w}^\top\mathbf{w}$$

- Bayes rule and $p(\mathbf{w})$

$$p(\mathbf{w}|\mathbf{y},\mathbf{X}) = \frac{p(\mathbf{y}|\mathbf{w},\mathbf{X})p(\mathbf{w})}{p(\mathbf{y})}$$

Contents

Issues with Linearity

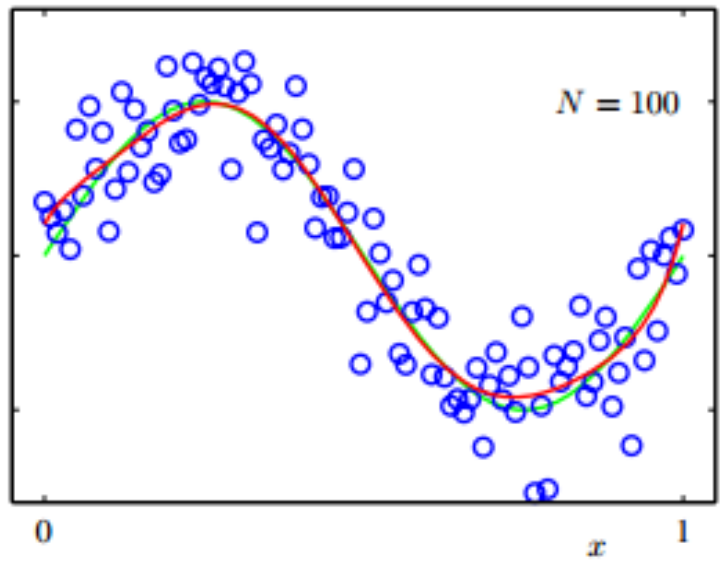
- Assume data is linear in the input $\mathbf{x}$

$$h(\mathbf{x}; \mathbf{w}) = x_1 w_1 + \cdots + x_m w_m$$

- For small m, probably unlikely (projection of data)
- Need non-linear feature interactions to model many problems.

Representation Questions

- In practice though, we select the features $\mathbf{x}$
- Intuitively this allows us to explicitly define interactions/relationships between inputs.
- Different representation can allow for nonlinear interactions within linear regression.



Basis Functions

Define basis functions $\{\phi_j\}_{j=1}^{d}$,

$$\mathbf{w} \in \mathbb{R}^d$$

$$\phi_j : \mathbb{R}^m \to \mathbb{R}$$

$$h(\mathbf{x}; \mathbf{w}) = \sum_{j=1}^{d} w_j \phi_j(\mathbf{x}).$$

- Weights for each dimension of new basis input d

Basis Vector Representation

More compact notation:

$$\boldsymbol{\phi}(\mathbf{x}) = \begin{bmatrix} \phi_1(\mathbf{x}) = 1 \\ \phi_2(\mathbf{x}) \\ \vdots \\ \phi_d(\mathbf{x}) \end{bmatrix}$$

$$h(\mathbf{x}; \mathbf{w}) = \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}).$$

Basis Functions

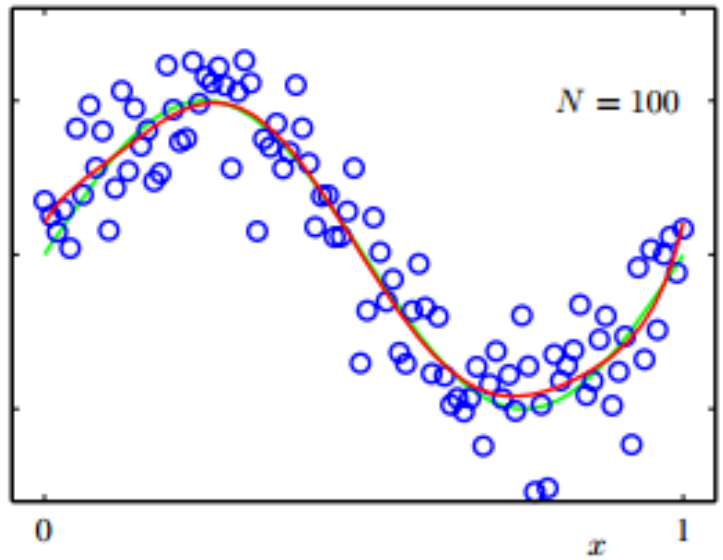
Many different ways to construct basis functions.

- Mathematical transforms of input data.
- Domain specific "higher-order features"
- Learned parameterized basis functions

Examples: Polynomial Basis

Any linear regression can be made nonlinear by using a polynomial basis.

$$\boldsymbol{\phi}(x)^\top = [\phi_1(x) = 1, \phi_2(x) = x, \phi_3(x) = x^2]$$
$$h(x; \mathbf{w}) = \mathbf{w}^\top \boldsymbol{\phi}(x).$$



Polynomial Basis Regression

Examples: Radial Basis

Given data $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, let $\phi_j(\mathbf{x}) = \exp(-(\mathbf{x}_j - \mathbf{x})^\top(\mathbf{x}_j - \mathbf{x}))$ for all $j \in \{1, \ldots n\}$ i.e. a distance with each training point.

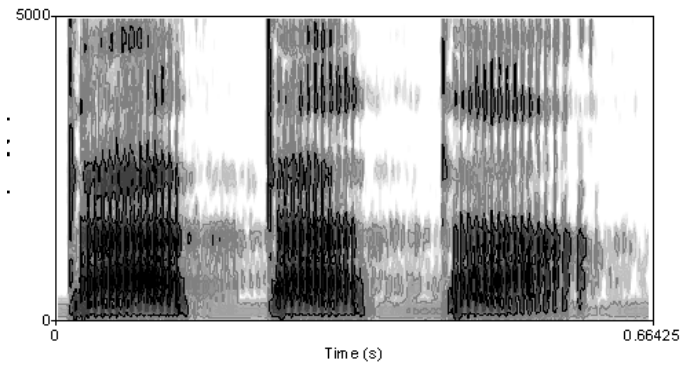
$$\boldsymbol{\phi}(\mathbf{x})^\top = [1, \{\exp(-(\mathbf{x}_j - \mathbf{x})^\top(\mathbf{x}_j - \mathbf{x}))\}_{j=1}^d]$$
$$h(x; \mathbf{w}) = \mathbf{w}^\top \boldsymbol{\phi}(x).$$

Connects non-parametric and parametric regression.

Examples: Speech Recognition

Speech recognition uses a frequency basis, i.e.

- $\mathbf{x}$; is a time-domain signal (short duration)
- ϕ; extracts the frequency-domain coefficients by applying a Fourier (among other processing)
- Resulting "spectrogram" has cleaner linear signal.



Speech Spectrogram ("ta ta ta")

Examples: Bag-Of-Words

Consider a text regression problem,

> *Given a movie review represented as a document, predict a continuous value representing the rating of the move (e.g. Metacritic score).*

- $\mathbf{x}$; the complete text of the review
- $\delta_{\mathbf{x},\text{word}}$; 1 iff `word` appears in the review
- $\mathcal{V}$; a list of the most common words

$$\boldsymbol{\phi}(\mathbf{x})^\top = [\{\delta_{\mathbf{x},v}\}_{v \in \mathcal{V}}]$$
$$h(\mathbf{x}; \mathbf{w}) = \mathbf{w}^\top \boldsymbol{\phi}(x).$$

Notebook: Basis Regression

[IPython Demo]