ECE285/SIO209, Machine learning for physical applications, Spring 2017

Peter Gerstoft, 534-7768, gerstoft@ucsd.edu

We meet Wednesday from 5 to 6:20pm in Spies Hall 330

Text Bishop

Grading A or maybe S

Classes

First 4 weeks: Focus on theory/implementation

Middle 3 weeks: 50% Applications plus 50% theory

Last 3 weeks: 30% Final Project, 30% Applications plus 50% theory

<u>Applications</u>

Graph theory for source localization: Gerstoft

Source tracking in ocean acoustics: Grad Student Emma Reeves

Aramco Research: Weichang Li, Group leader

Seismic network using mobile phones: Berkeley

Eric Orenstein: identifying plankton

Plus more

- Dictionary learning

-

Homework: Both matlab/python will be used, Just email the code to me (I dont need anything else).

Homework is due 11am on Wednesday. That way we can discuss in class.

Hw 1:

matlab/ python/ ipython/ jupyter ?

Tritoned? https://tritoned.ucsd.edu/

Machine Learning for Geophysical Applications

Peter Gerstoft

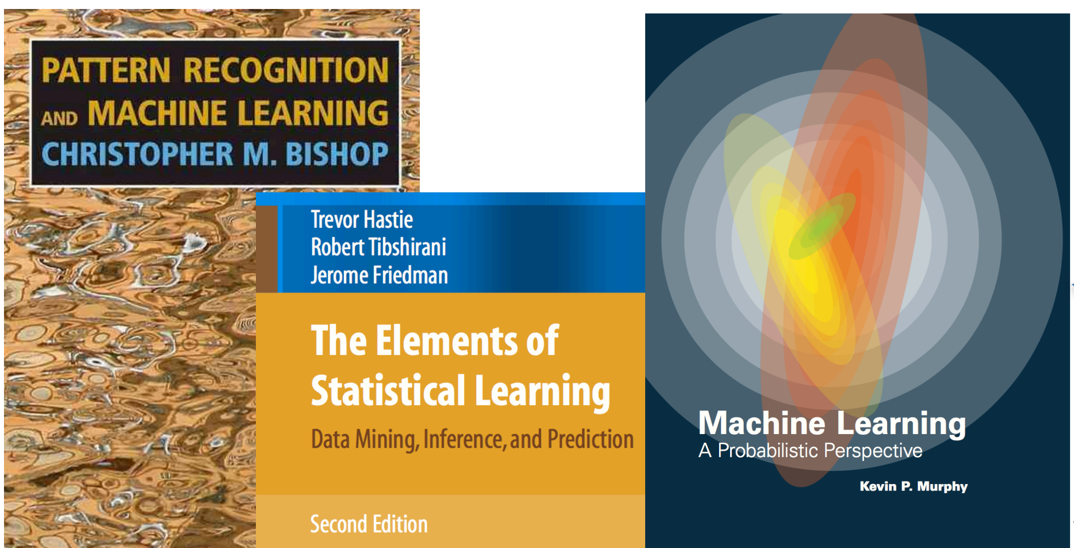
noiselab.ucsd.edu

Plan

Unsupervised source localization (graph theory)

Supervised source localization (neural network)

Unsupervised dictionary learning for sound speed



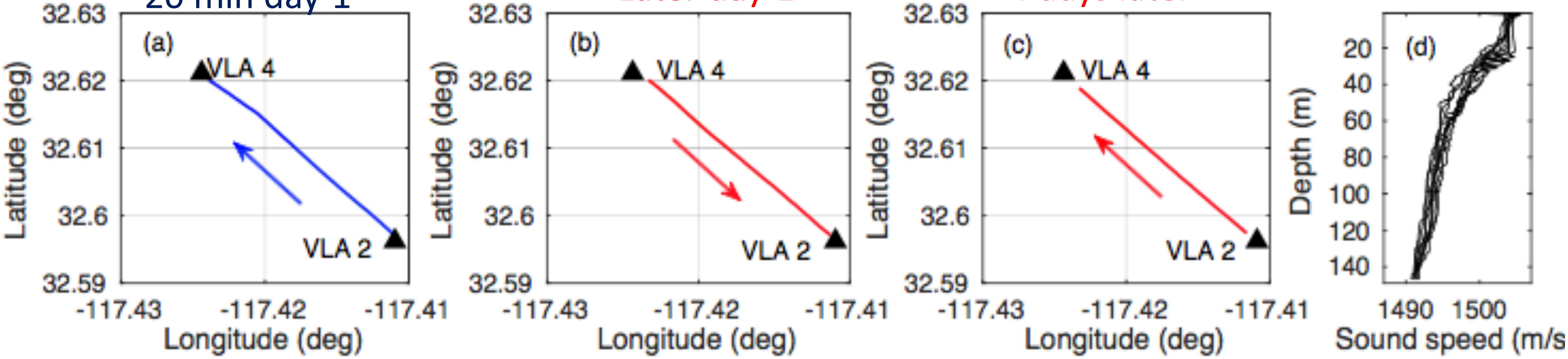
Murphy: “This books adopts the view that the best way to make machines that can learn from data is to use the *tools of probability theory*, which has been the mainstay of statistics and engineering for centuries. “

Neural Networks (TensorFlow) for tracking a ship

Training data,
20 min day-1

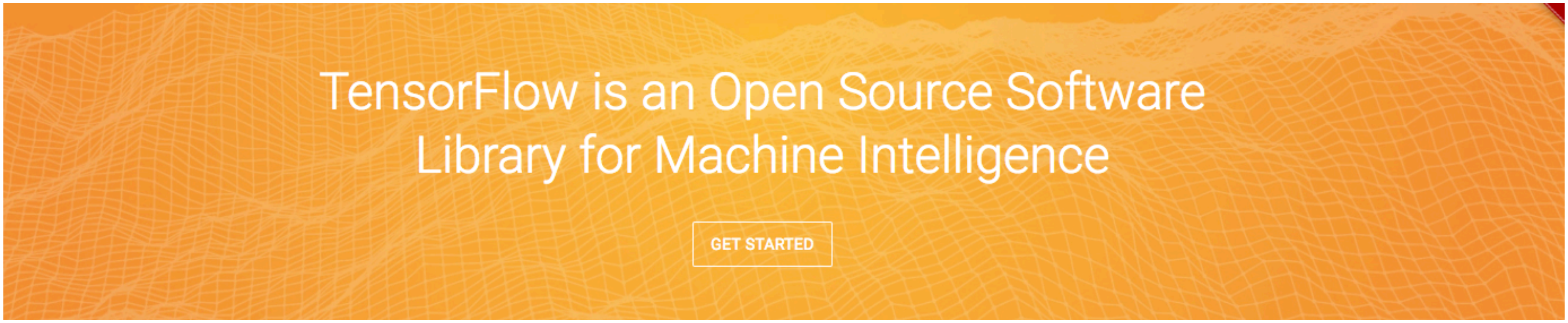
Test-data-1
Later day-1

Test-data-2
4 days later



Niu and Gerstoft JASA, 2016

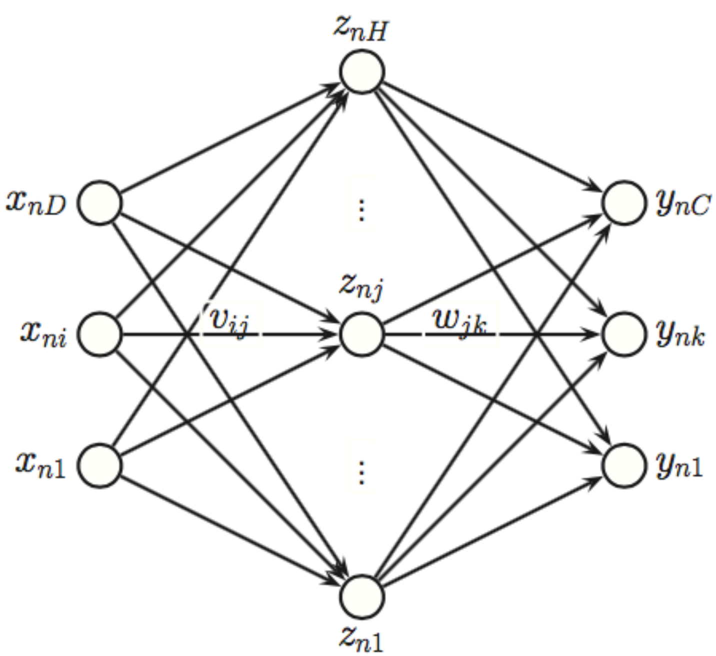
TensorFlow implementation



Input: Sample cov. matrix: 272 Neurons (16*17/2*2) per frequncy at each range

Output: binary range vector: 0.1-3km, 138 neurons

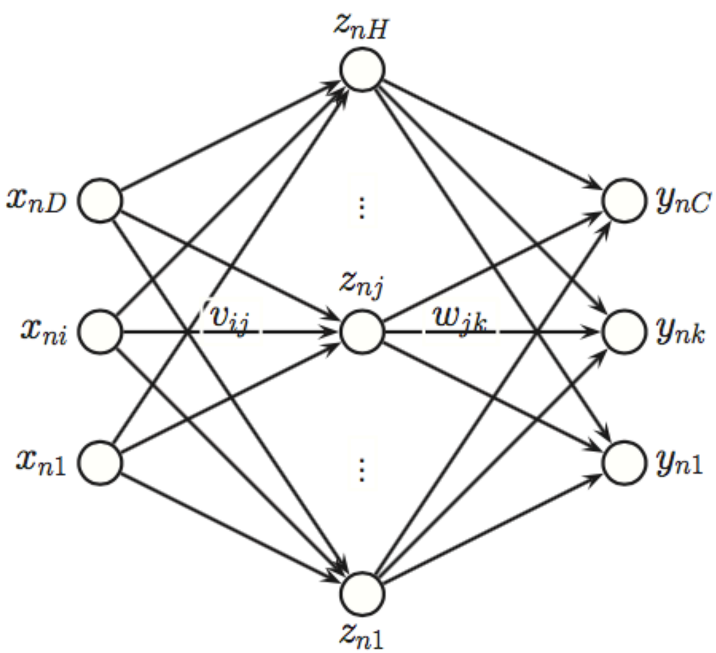
Just one middle layer 128 Neurons



*	*	*	...	*	*	*	*

$$
\begin{array}{l}
[\ 1 \quad 0 \quad 0 \quad \dots \quad 0 \quad 0 \quad 0 \quad 0\ ] \longrightarrow r_1 \\
[\ 0 \quad 1 \quad 0 \quad \dots \quad 0 \quad 0 \quad 0 \quad 0\ ] \longrightarrow r_2 \\
[\ 0 \quad 0 \quad 1 \quad \dots \quad 0 \quad 0 \quad 0 \quad 0\ ] \longrightarrow r_3 \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\
[\ 0 \quad 0 \quad 0 \quad \dots \quad 0 \quad 0 \quad 0 \quad 1\ ] \longrightarrow r_K
\end{array}
$$

TensorFlow implementation

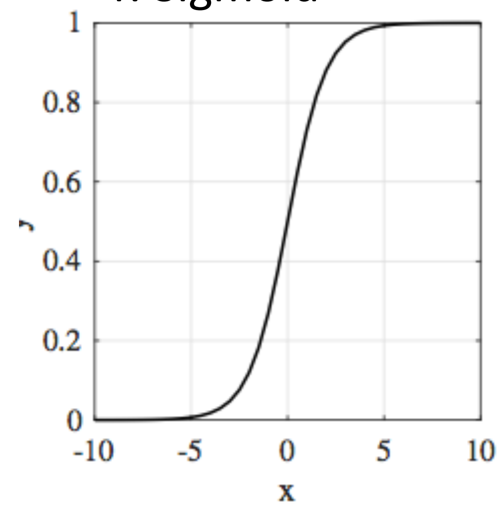


$$\boldsymbol{z} = f(\boldsymbol{V}\boldsymbol{x})$$
$$\boldsymbol{y} = h(\boldsymbol{W}\boldsymbol{z})$$

h: softmax function:

$$\boldsymbol{a} = \boldsymbol{W}\boldsymbol{z}$$
$$y_k = \frac{e^{a_k}}{\sum_j^C e^{a_j}}$$

f: Sigmoid



Four-frequency localization

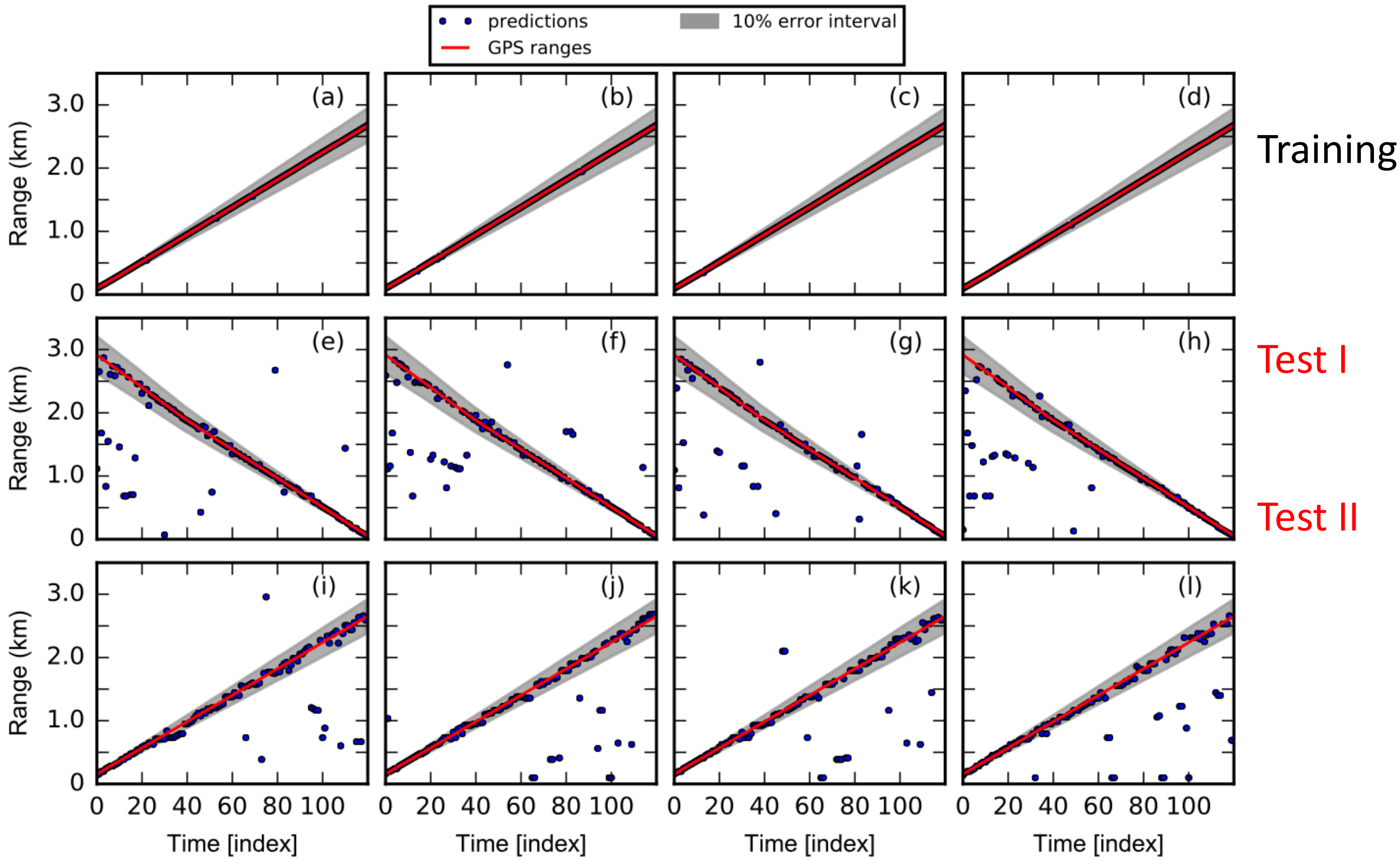
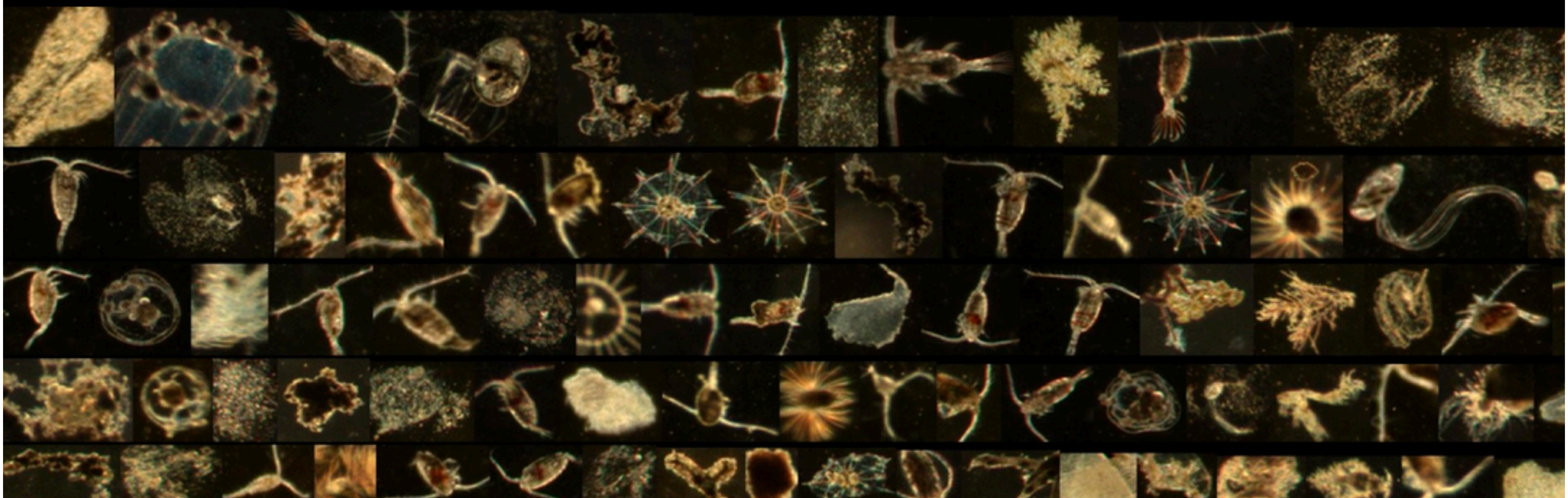


FIG. 12. (Color online) Range predictions on training data (a, b, c, d, first row), Test-Data-1 (e, f, g, h, second row) and Test-Data-2 (i, j, k, l, third row) by FNN with multi-frequency inputs. (a)(e)(i) 450, 490, 520, 550 Hz. (b)(f)(j) 560, 590, 620, 650 Hz. (c)(g)(k) 660, 690, 720, 750 Hz. (d)(h)(l) 450, 600, 750, 900 Hz. The time index increment is 10 s for training and Test-Data-1, and 5 s for Test-Data-2.

TRANSFER LEARNING AND DEEP FEATURE EXTRACTION FOR PLANKTONIC IMAGE DATA SETS



Eric C. Orenstein[1] and Oscar Beijbom[2]

[1] Scripps Institution of Oceanography – University of California San Diego

[2] Department of Electrical Engineering and Computer Science – University of California Berkeley

IEEE Winter Conference on Applications of Computer Vision 2017 – Paper ID #313

Qingkai Kong is from Berkeley, I have 3GB of data and examples of analysis by students there

RESEARCH ARTICLE

EARTH SCIENCES

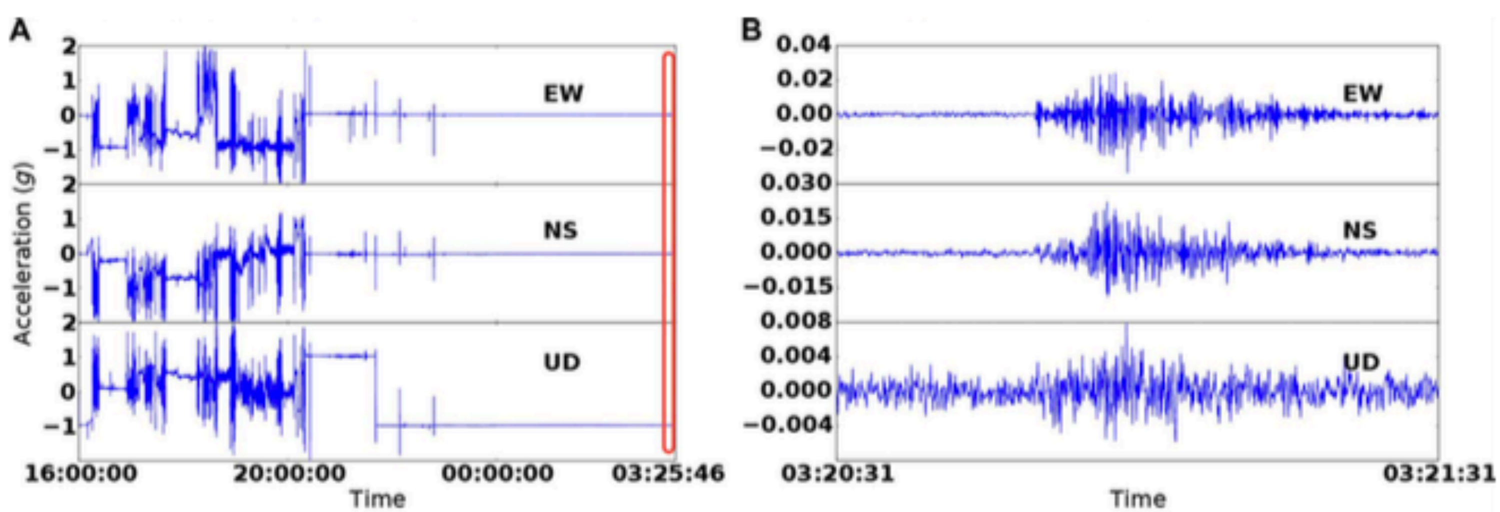
MyShake: A smartphone seismic network for earthquake early warning and beyond

10.1126/sciadv.1501055

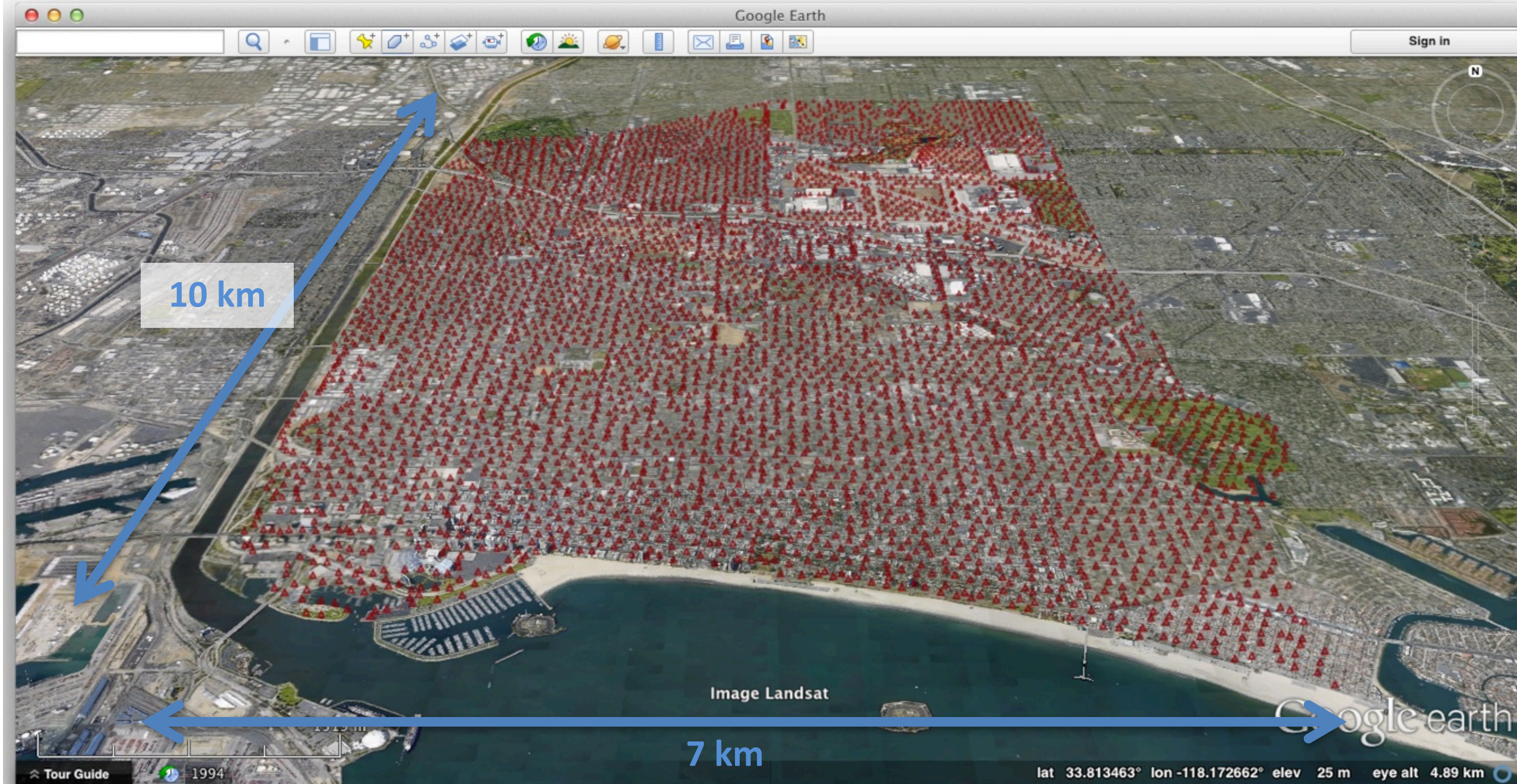
Qingkai Kong,[1]* Richard M. Allen,[1] Louis Schreier,[2] Young-Woo Kwon[3]

Large magnitude earthquakes in urban environments continue to kill and injure tens to hundreds of thousands of people, inflicting lasting societal and economic disasters. Earthquake early warning (EEW) provides seconds to minutes of warning, allowing people to move to safe zones and automated slowdown and shutdown of transit and other machinery. The handful of EEW systems operating around the world use traditional seismic and geodetic networks

RESEARCH ARTICLE



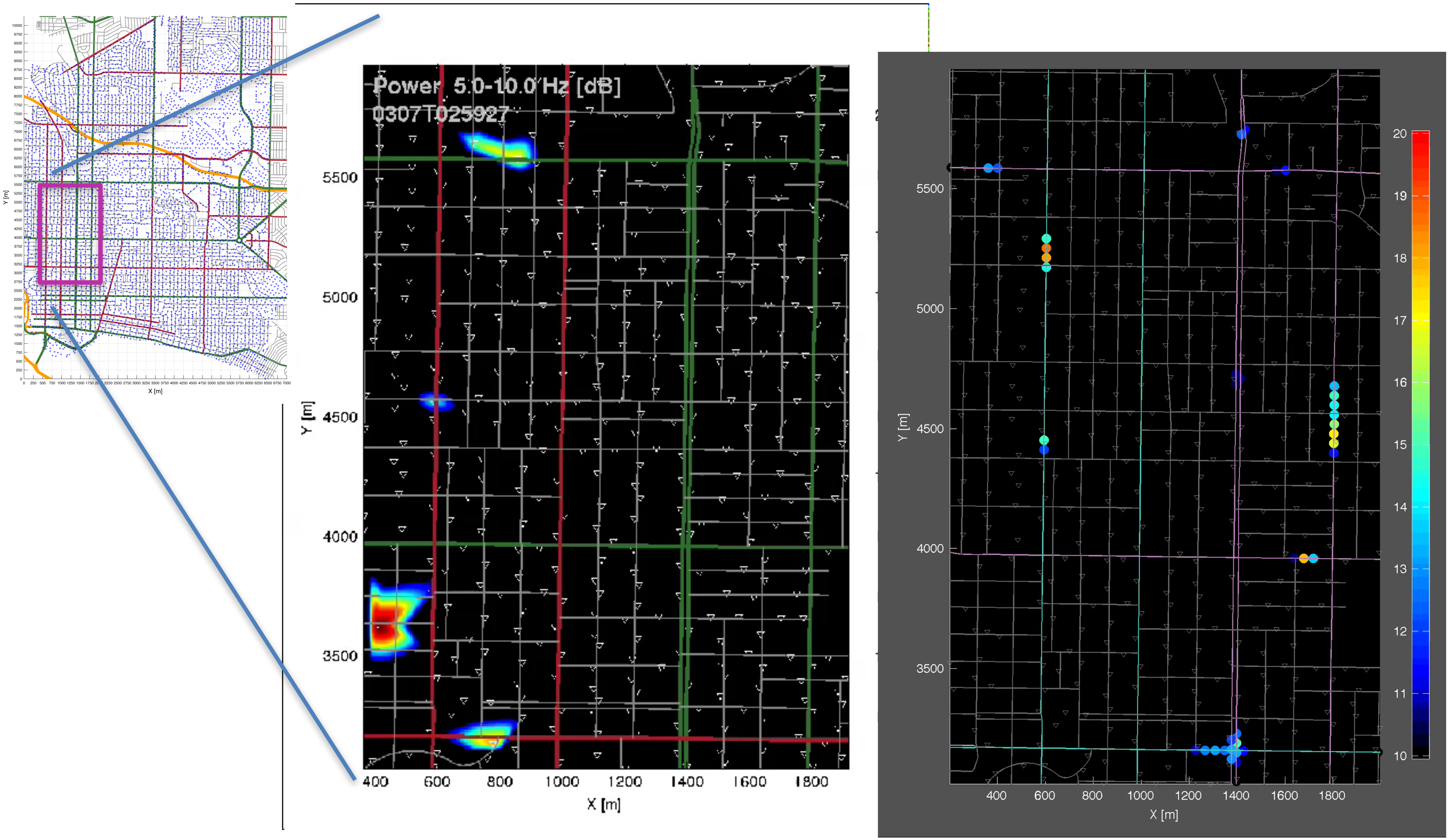
Why we got interested in traffic



March 5—12, 2011

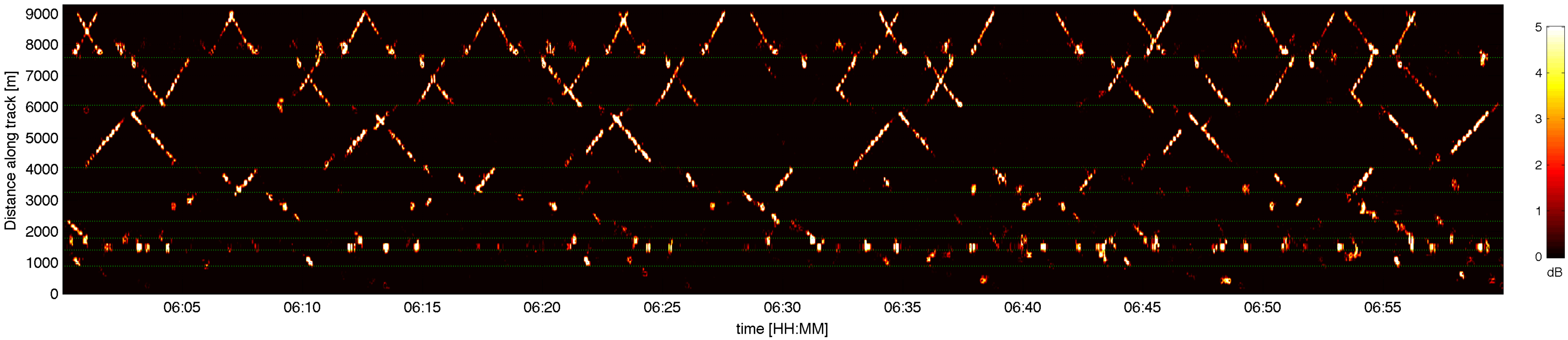
Noise Tracking of Cars/Trains/Airplanes

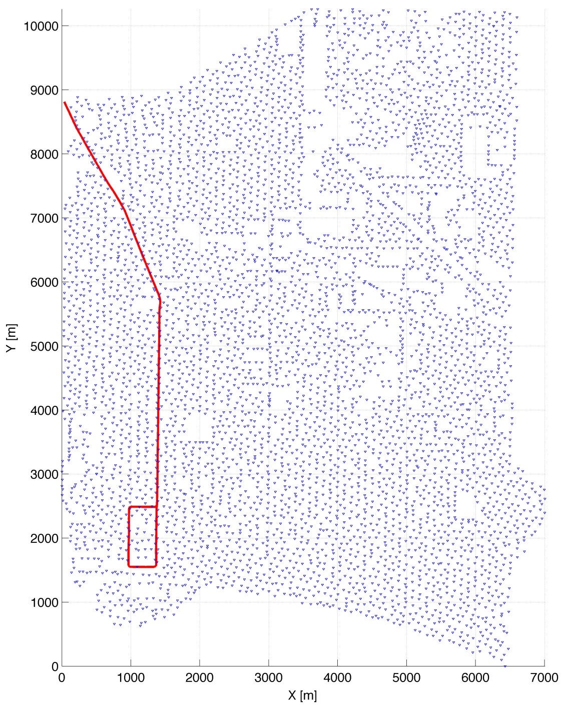
5200 element Long Beach array (Dan Hollis)



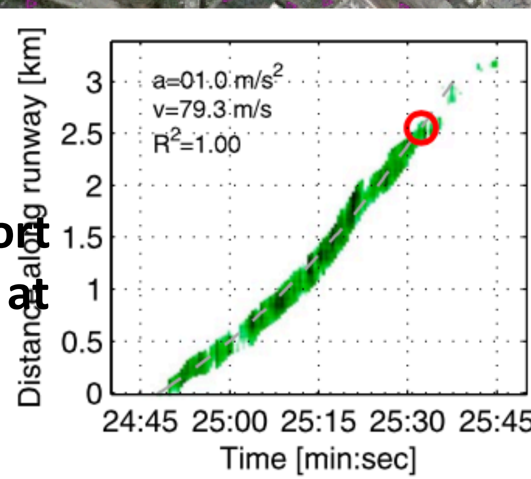
Nima Riahi 2014

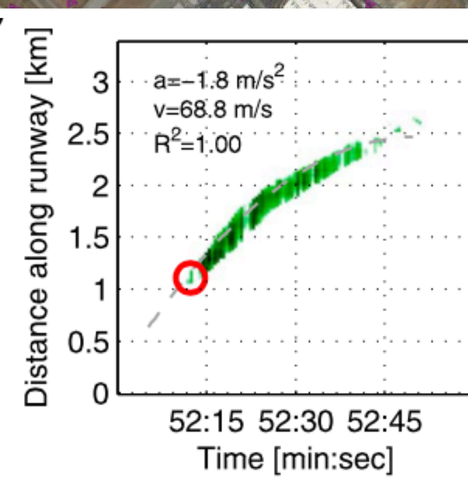
Noise Tracking of Cars/Trains/Airplanes





March 7th, 6-7am, rush hour, Blue Line



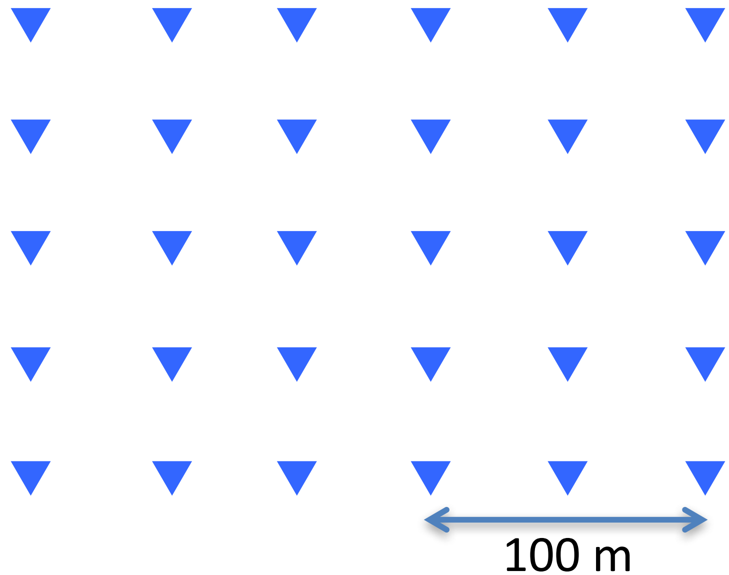


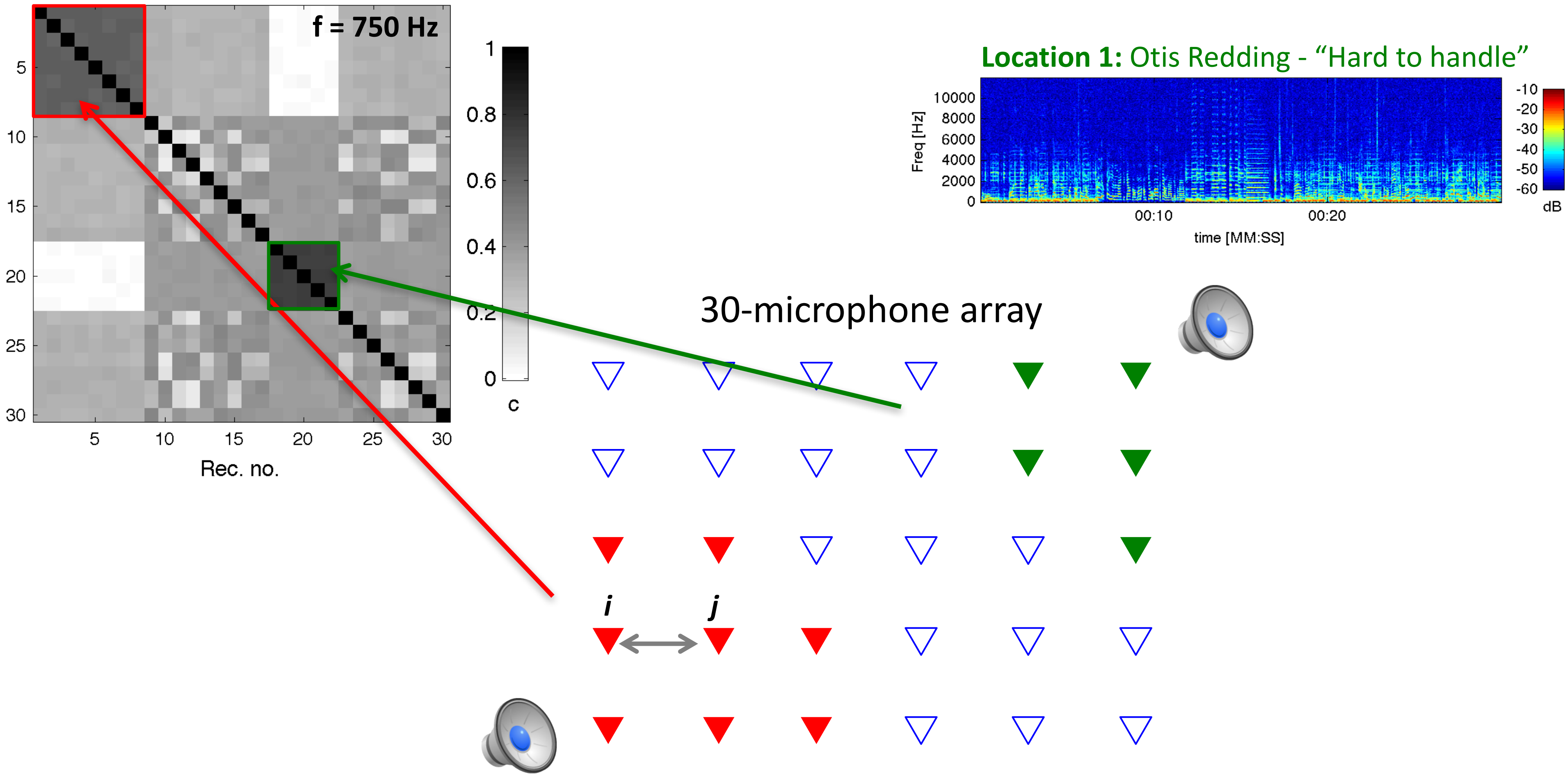
Accelerating airplane on Long Beach Airport runway, moving northwest and taking off at about 120 mi/h.

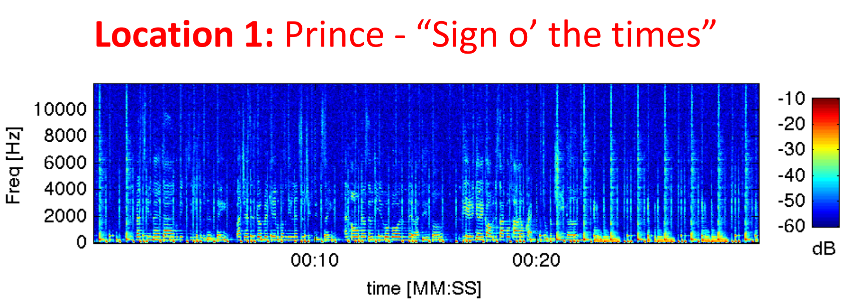
Riahi, Gerstoft, GRL 2015

Thought experiment: Party at a detection array

30-microphone array







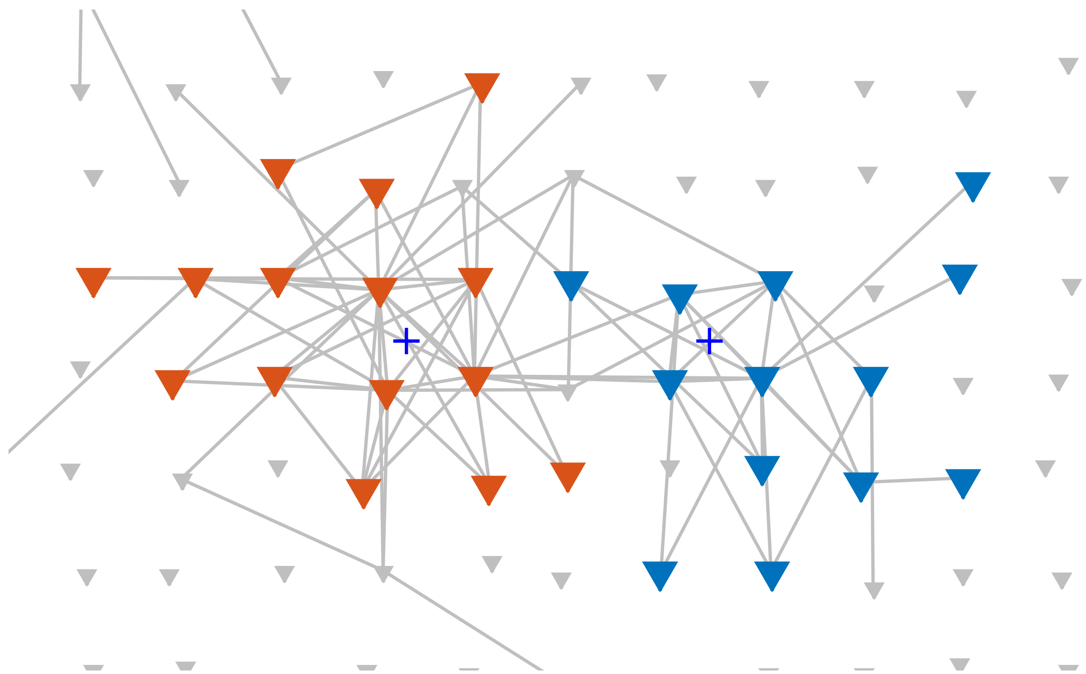
Spectral coherence between ***i*** and ***j***

$$\hat{C}_{ij}(f) = \frac{1}{N}\sum_{t=1}^{N} X_i(f,t) \cdot \bar{X}_j(f,t)$$

(Normalization: |X(f,t)|²=1)

Objective

Find coherent but very localized event in a large array.
Don't assume anything about the medium.



Approach

Construct network using pair-wise sensor coherence.
Exploit network structure to identify sources.

What is Machine Learning?

Many related terms:

- Pattern Recognition
- Neural Networks
- Data Mining
- Adaptive Control
- Statistical Modelling
- Data analytics / data science
- Artificial Intelligence
- Machine Learning

Big data

Learning:
The view from different fields

- Engineering: signal processing, system identification, adaptive and optimal control, information theory, robotics, ...
- Computer Science: Artificial Intelligence, computer vision, information retrieval, ...
- Statistics: learning theory, data mining, learning and inference from data, ...
- Cognitive Science and Psychology: perception, movement control, reinforcement learning, mathematical psychology, computational linguistics, ...
- Computational Neuroscience: neuronal networks, neural information processing, ...
- Economics: decision theory, game theory, operational research, ...

Physical science is missing!
ML cannot replace physical understanding.
It might improve or find additional trends

Machine learning is interdisciplinary focusing on both mathematical foundations and practical applications of systems that learn, reason and act.

Probabilistic Modelling

- A model describes data that one could observe from a system
- If we use the mathematics of probability theory to express all forms of uncertainty and noise associated with our model...
- ...then *inverse probability* (i.e. Bayes rule) allows us to infer unknown quantities, adapt our models, make predictions and learn from data.

Bayes Rule

$$P(\text{hypothesis}|\text{data}) = \frac{P(\text{data}|\text{hypothesis})P(\text{hypothesis})}{P(\text{data})}$$

Rev'd Thomas Bayes (1702–1761)

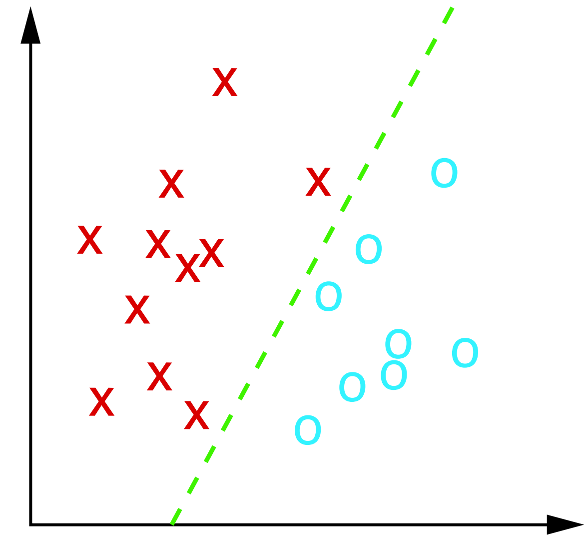
- Bayes rule tells us how to do inference about hypotheses from data.

- Learning and prediction can be seen as forms of inference.

Some Canonical Machine Learning Problems

- Linear Classification
- Polynomial Regression
- Clustering with Gaussian Mixtures (Density Estimation)

Linear Classification



Data: $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}$ for $n = 1, \ldots, N$ data points

$$\begin{aligned} \mathbf{x}^{(n)} &\in \mathbb{R}^D \\ y^{(n)} &\in \{+1, -1\} \end{aligned}$$

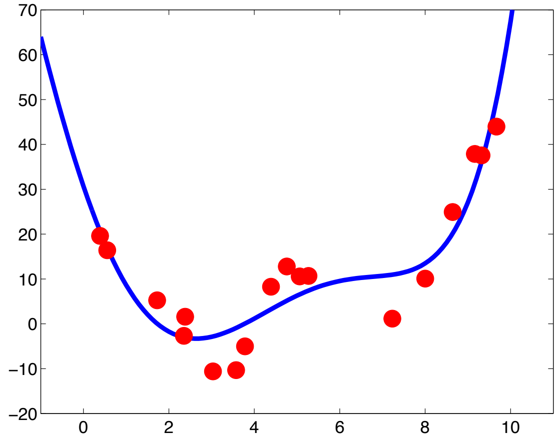
Model:

$$P(y^{(n)} = +1 | \boldsymbol{\theta}, \mathbf{x}^{(n)}) = \begin{cases} 1 & \text{if } \sum_{d=1}^{D} \theta_d \, x_d^{(n)} + \theta_0 \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Parameters: $\boldsymbol{\theta} \in \mathbb{R}^{D+1}$

Goal: To infer $\boldsymbol{\theta}$ from the data and to predict future labels $P(y|\mathcal{D}, \mathbf{x})$

Polynomial Regression



Data: $\mathcal{D} = \{(x^{(n)}, y^{(n)})\}$ for $n = 1, \ldots, N$

$$\begin{aligned} x^{(n)} &\in \mathbb{R} \\ y^{(n)} &\in \mathbb{R} \end{aligned}$$

Model:

$$y^{(n)} = a_0 + a_1 x^{(n)} + a_2 {x^{(n)}}^2 \ldots + a_m {x^{(n)}}^m + \epsilon$$

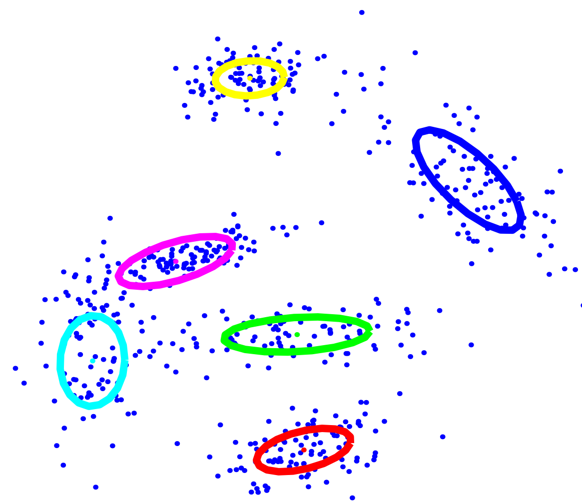
where

$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$

Parameters: $\boldsymbol{\theta} = (a_0, \ldots, a_m, \sigma)$

Goal: To infer $\boldsymbol{\theta}$ from the data and to predict future outputs $P(y|\mathcal{D}, x, m)$

Clustering with Gaussian Mixtures (Density Estimation)



Data: $\mathcal{D} = \{\mathbf{x}^{(n)}\}$ for $n = 1, \ldots, N$

$$\mathbf{x}^{(n)} \in \mathbb{R}^D$$

Model:

$$\mathbf{x}^{(n)} \sim \sum_{i=1}^{m} \pi_i \, p_i(\mathbf{x}^{(n)})$$

where

$$p_i(\mathbf{x}^{(n)}) = \mathcal{N}(\mu^{(i)}, \Sigma^{(i)})$$

Parameters: $\boldsymbol{\theta} = \big((\mu^{(1)}, \Sigma^{(1)}) \ldots, (\mu^{(m)}, \Sigma^{(m)}), \boldsymbol{\pi}\big)$

Goal: To infer $\boldsymbol{\theta}$ from the data, predict the density $p(\mathbf{x}|\mathcal{D}, m)$, and infer which points belong to the same cluster.

Bayesian Modelling

Everything follows from two simple rules:

Sum rule: $P(x) = \sum_y P(x, y)$

Product rule: $P(x, y) = P(x)P(y|x)$

$$P(\theta|\mathcal{D}, m) = \frac{P(\mathcal{D}|\theta, m)P(\theta|m)}{P(\mathcal{D}|m)}$$

$P(\mathcal{D}\|\theta, m)$	likelihood of parameters θ in model m
$P(\theta\|m)$	prior probability of θ
$P(\theta\|\mathcal{D}, m)$	posterior of θ given data $\mathcal{D}$

Prediction:

$$P(x|\mathcal{D}, m) = \int P(x|\theta, \mathcal{D}, m)P(\theta|\mathcal{D}, m)d\theta$$

Model Comparison:

$$P(m|\mathcal{D}) = \frac{P(\mathcal{D}|m)P(m)}{P(\mathcal{D})}$$

$$P(\mathcal{D}|m) = \int P(\mathcal{D}|\theta, m)P(\theta|m)\,d\theta$$

ML overview

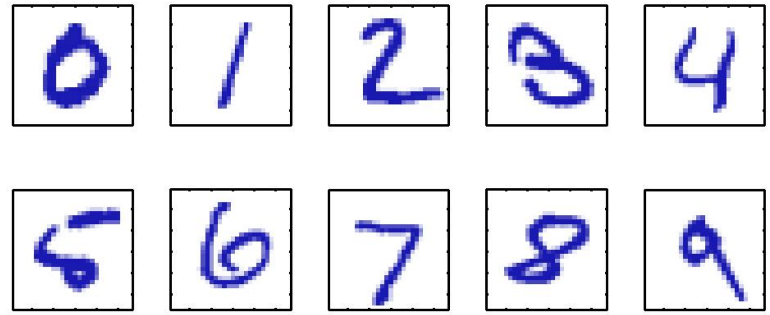
- Output: **y**(**x**)

  images x

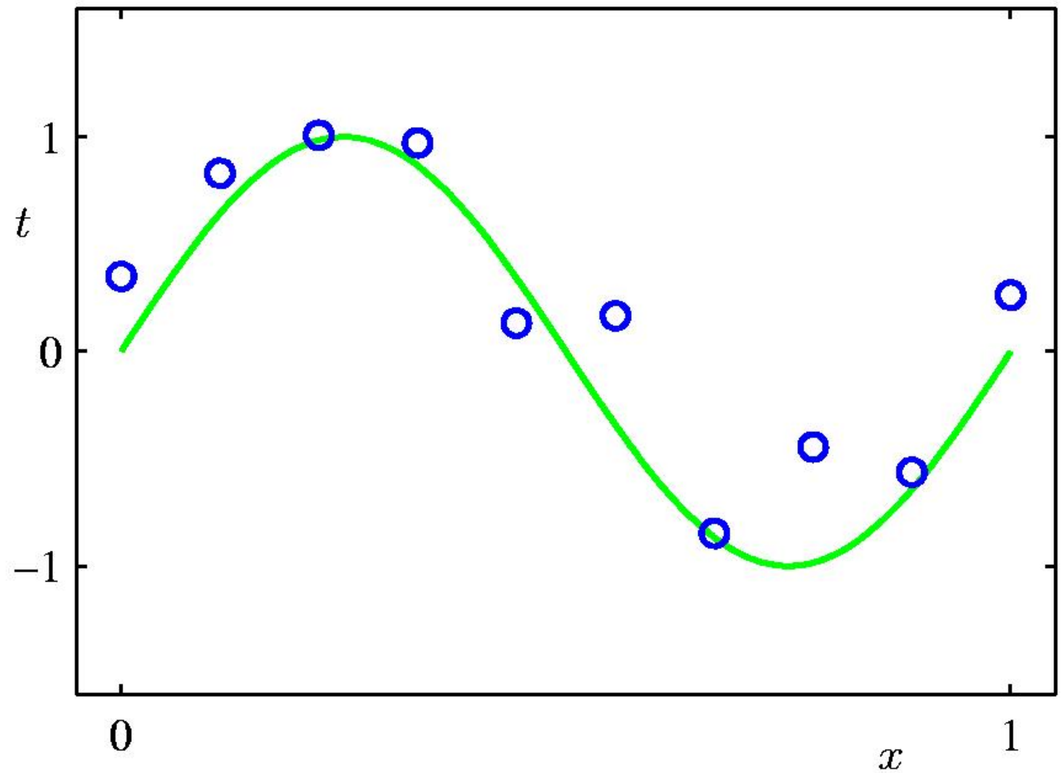
Target vector y or t

- Learning/ training [$\mathbf{x}_1$…]
- Test set
- Feature extraction
- Supervised learning--- Making predictions
  - Classification
  - Regression
- Unsupervised learning
  - Clustering
  - Density estimation
- Reinforcement learning
  - Exploration><exploitation

Nmist data set

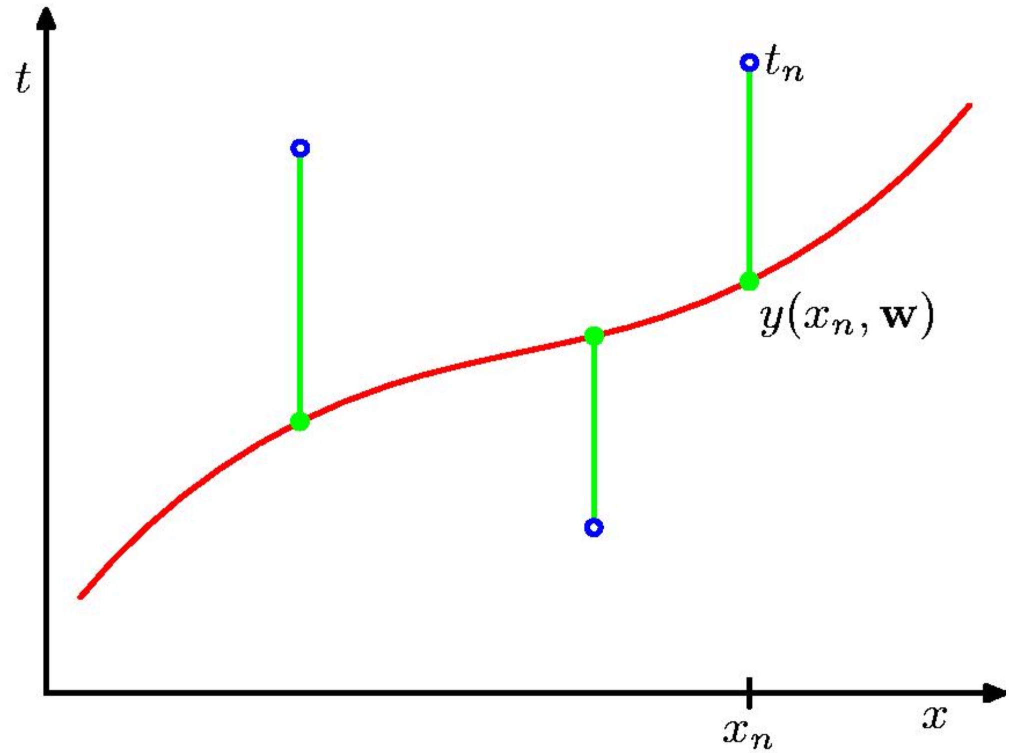


Polynomial Curve Fitting



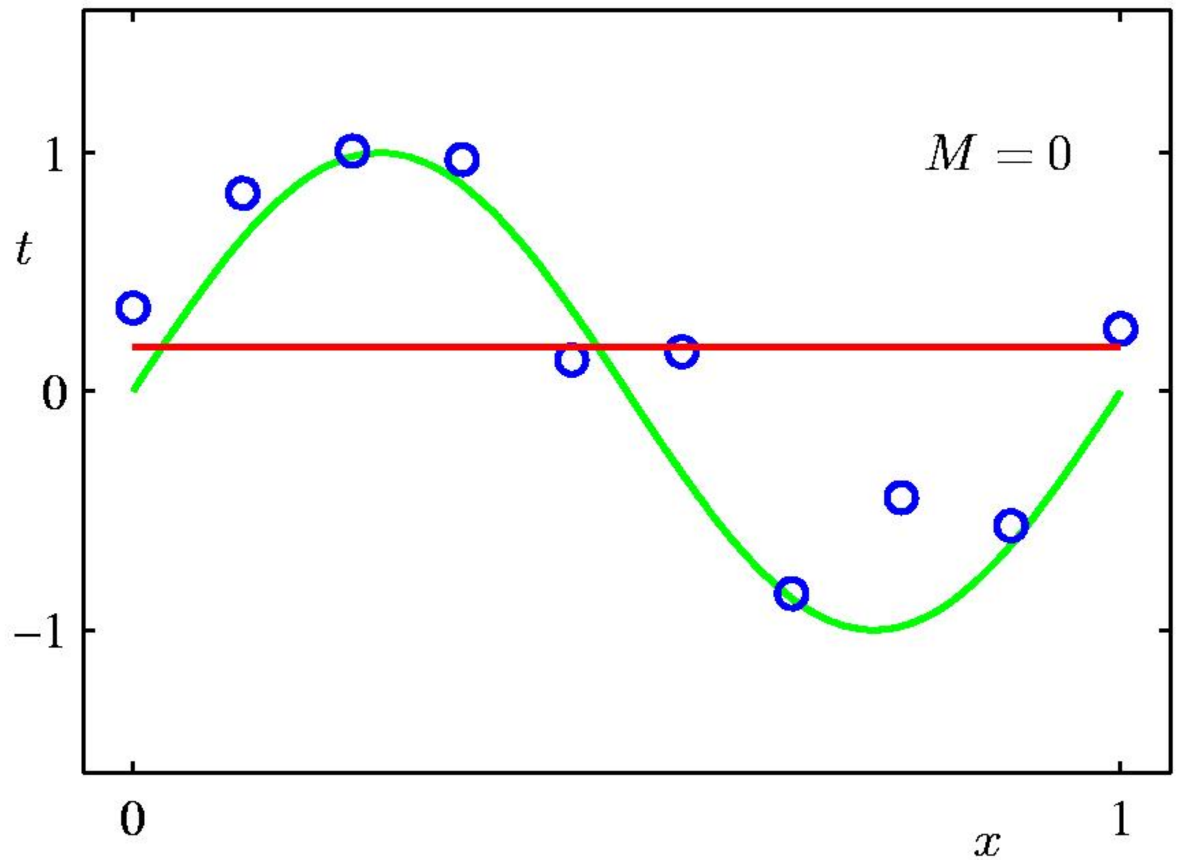
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Sum-of-Squares Error Function

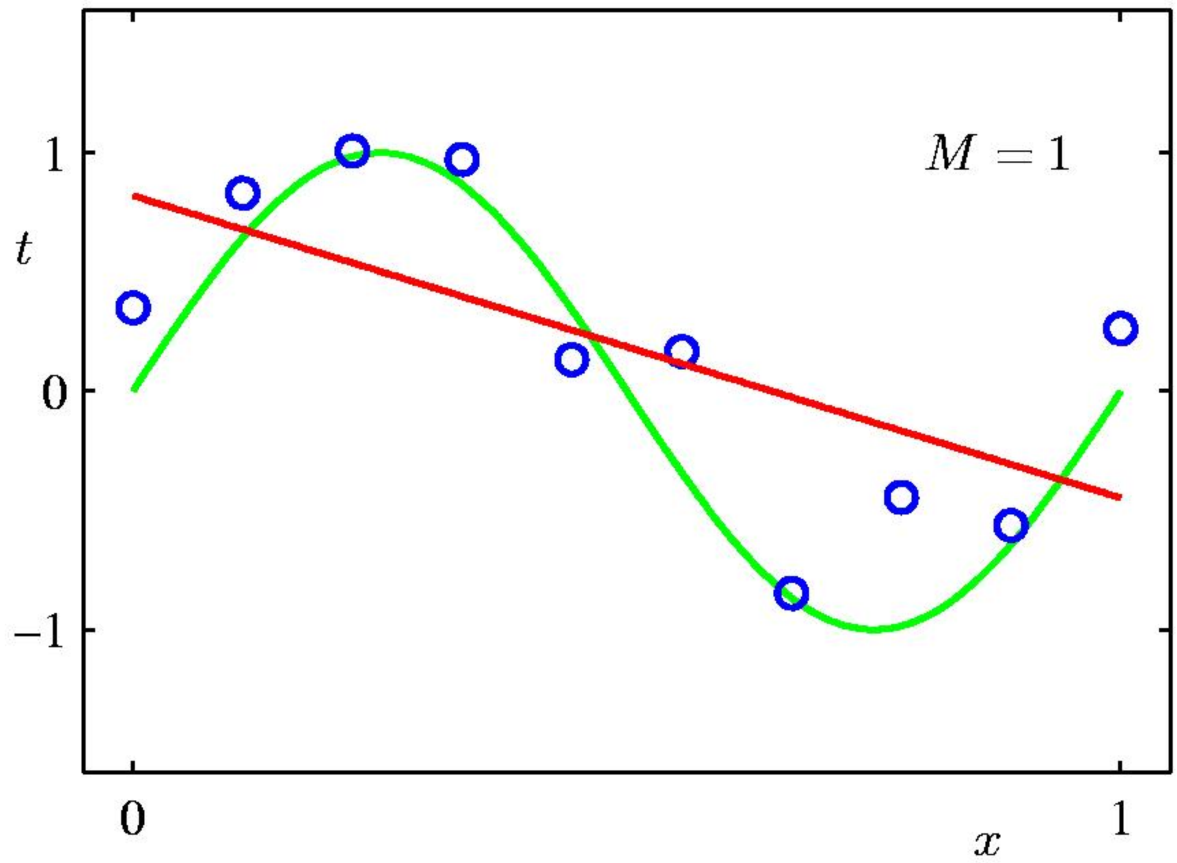


$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$$

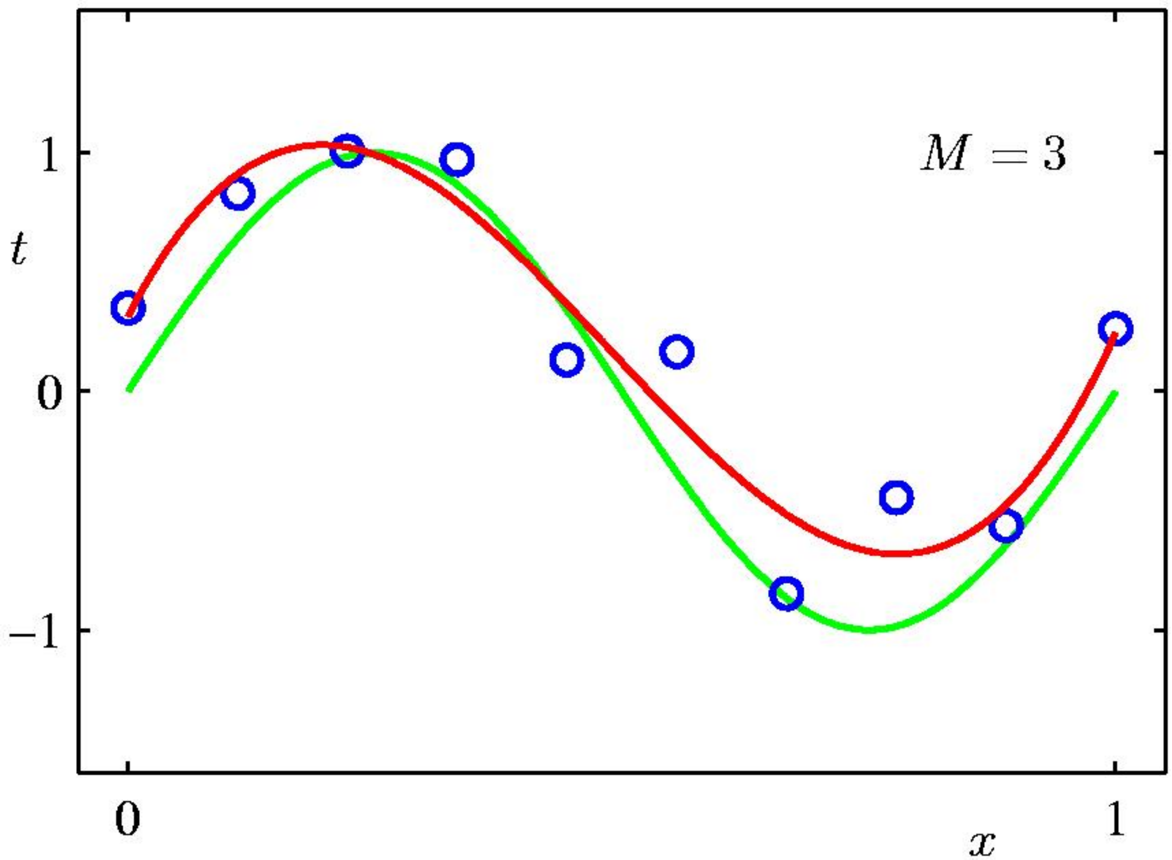
0^{th} Order Polynomial



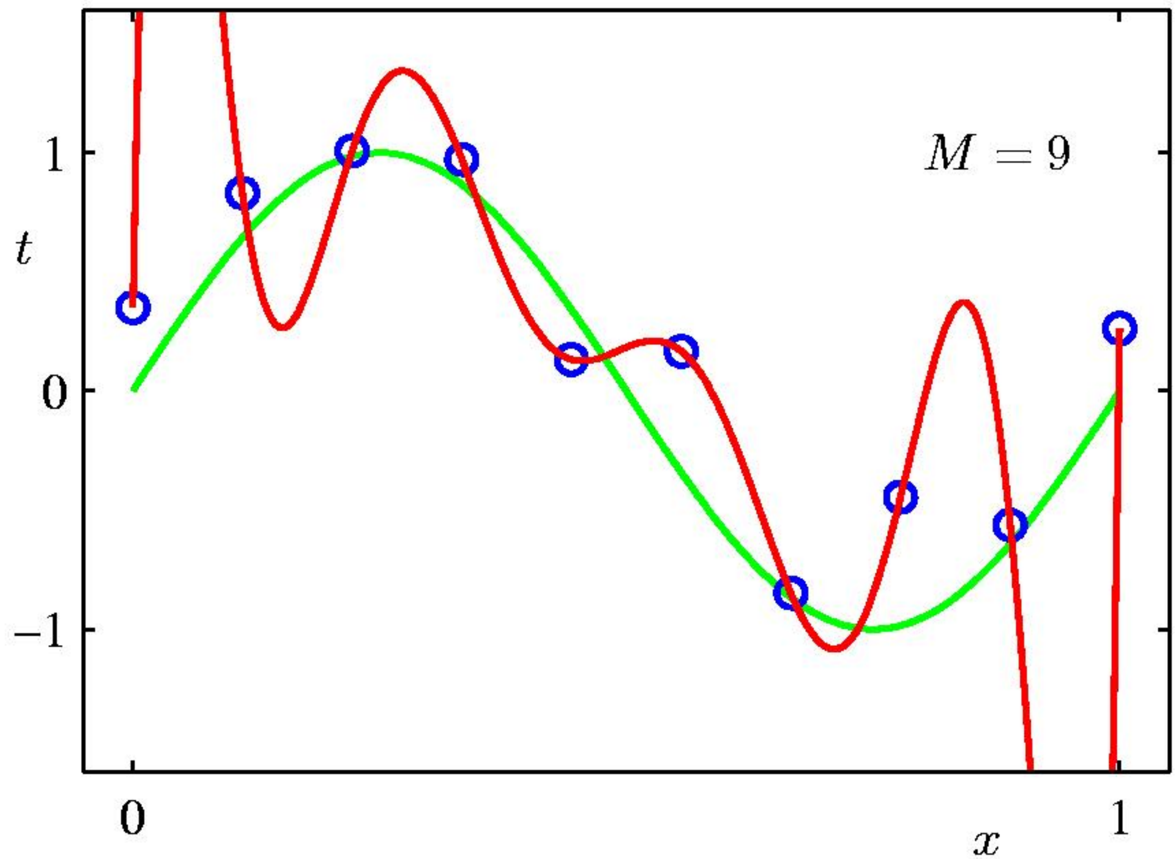
1^{st} Order Polynomial



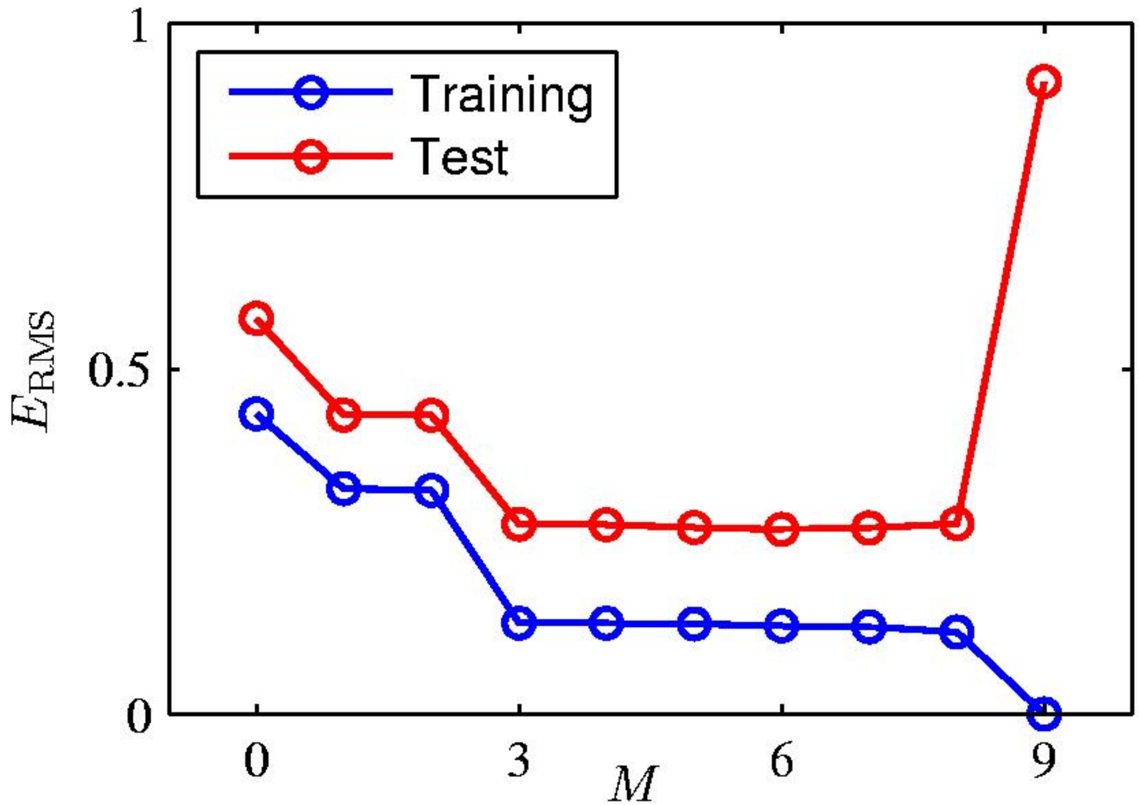
3^{rd} Order Polynomial



9th Order Polynomial



Over-fitting



Root-Mean-Square (RMS) Error: $E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^\star)/N}$

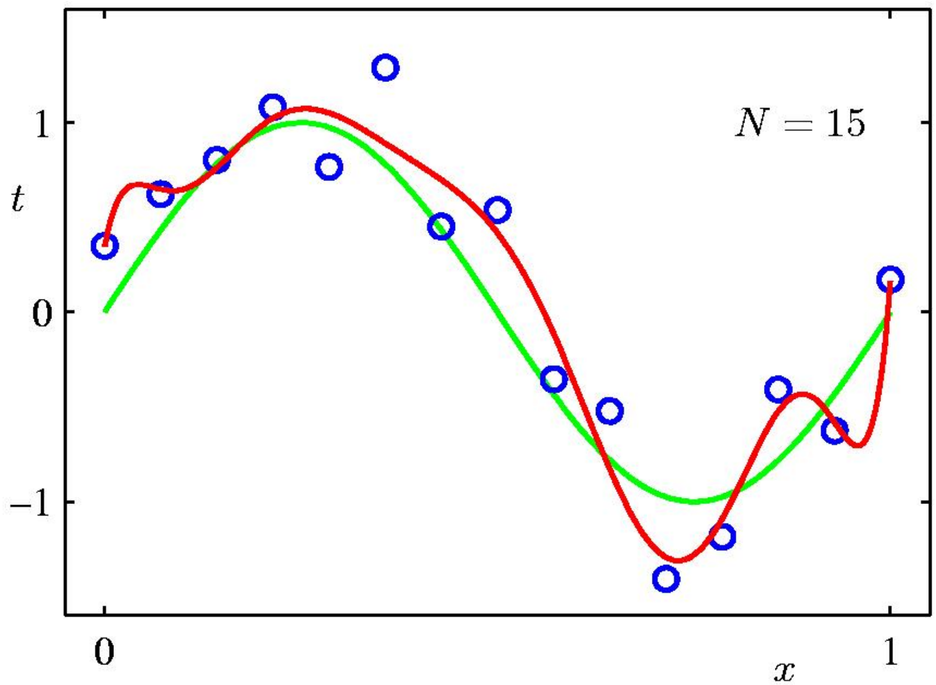
Polynomial Coefficients

	$M = 0$	$M = 1$	$M = 3$	$M = 9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

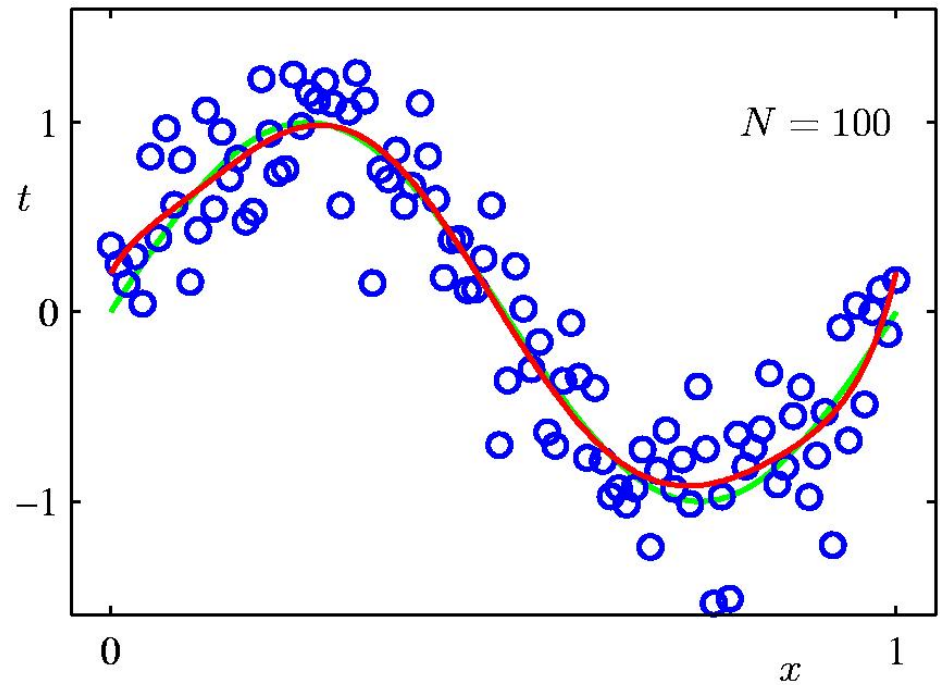
Data Set Size:

9th Order Polynomial

$N = 15$



$N = 100$

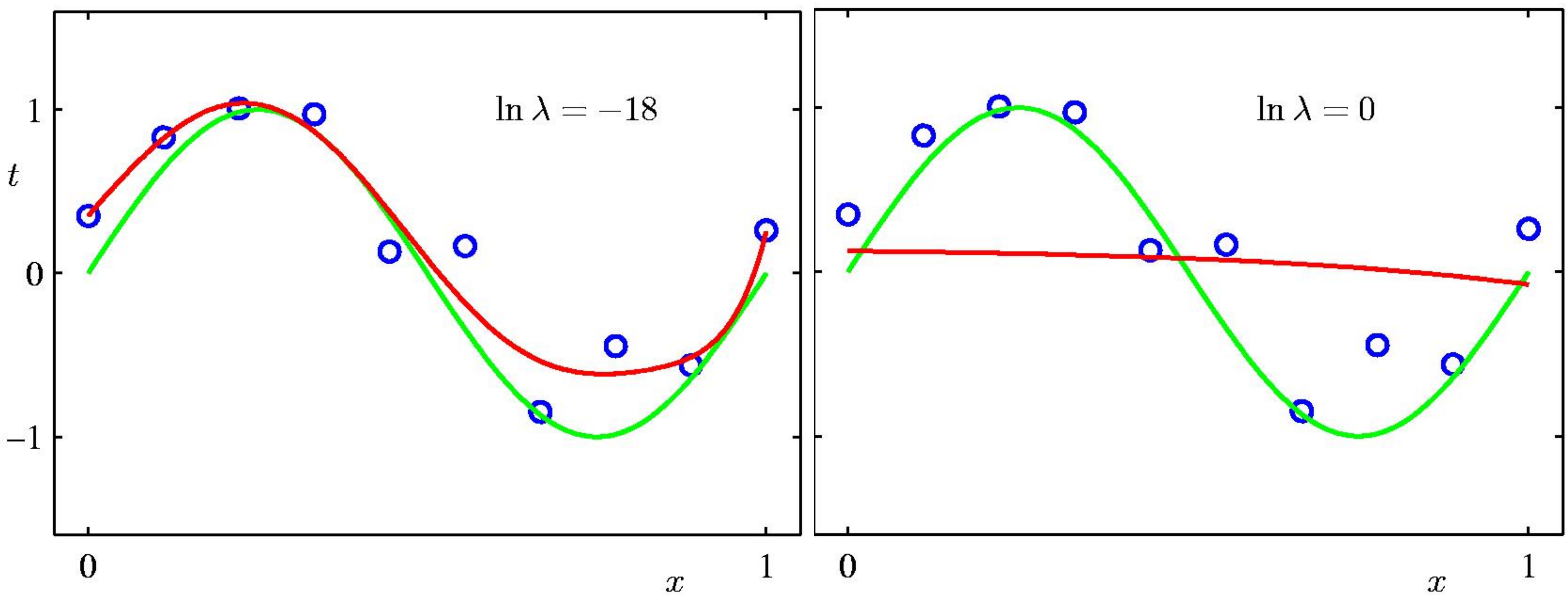


Regularization

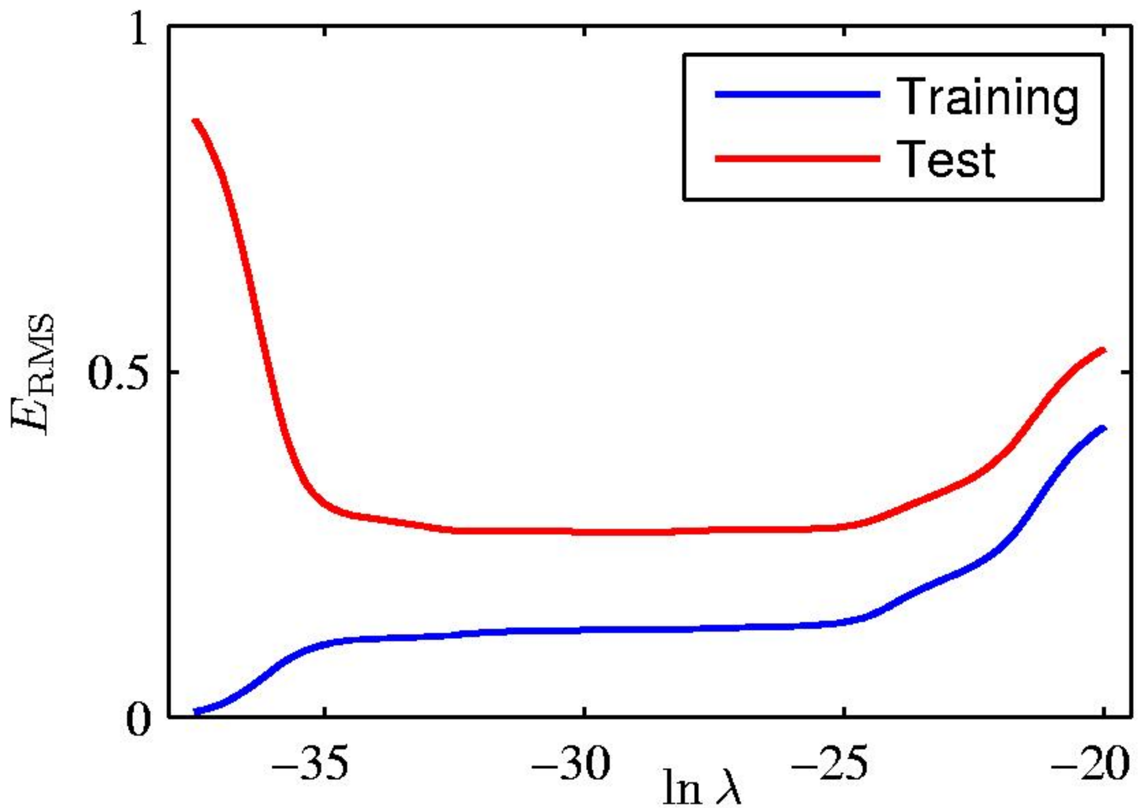
- Penalize large coefficient values

$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

Regularization:



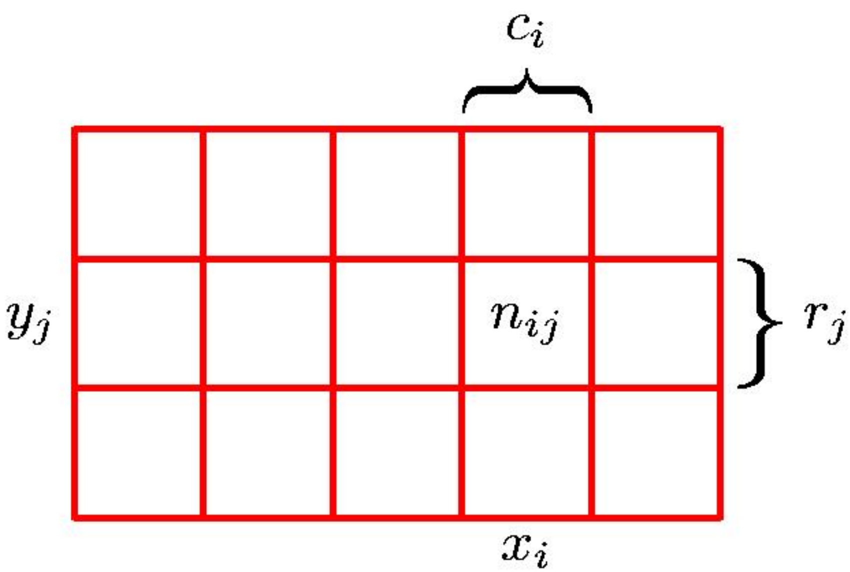
Regularization: E_{RMS} vs. $\ln \lambda$



Polynomial Coefficients

	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01

Probability Theory



- Marginal Probability

$$p(X = x_i) = \frac{c_i}{N}.$$

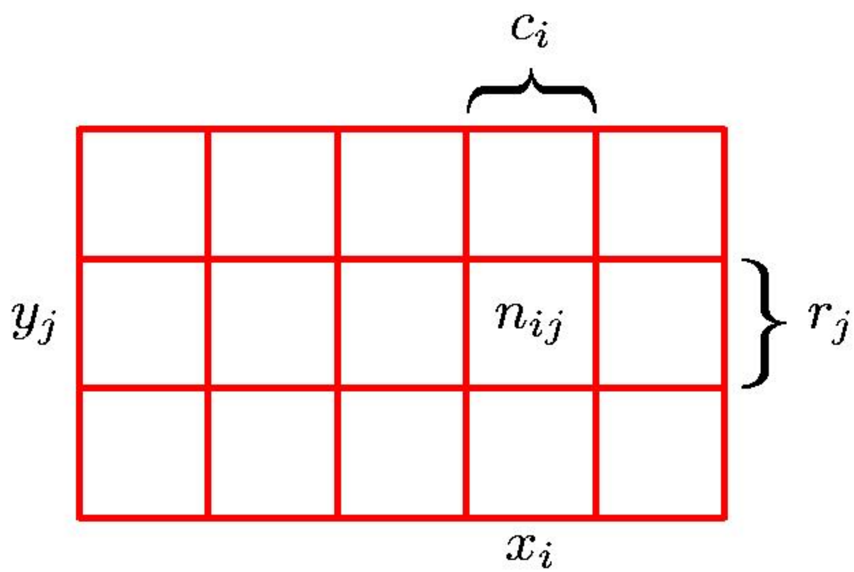
Joint Probability

$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N}$$

- Conditional Probability

$$p(Y = y_j | X = x_i) = \frac{n_{ij}}{c_i}$$

Probability Theory



- Sum Rule

$$p(X = x_i) = \frac{c_i}{N} = \frac{1}{N}\sum_{j=1}^{L} n_{ij}$$

$$= \sum_{j=1}^{L} p(X = x_i, Y = y_j)$$

Product Rule

$$\begin{aligned} p(X = x_i, Y = y_j) &= \frac{n_{ij}}{N} = \frac{n_{ij}}{c_i} \cdot \frac{c_i}{N} \\ &= p(Y = y_j | X = x_i) p(X = x_i) \end{aligned}$$

The Rules of Probability

- Sum Rule $$p(X) = \sum_Y p(X, Y)$$
- Product Rule $$p(X, Y) = p(Y|X)p(X)$$

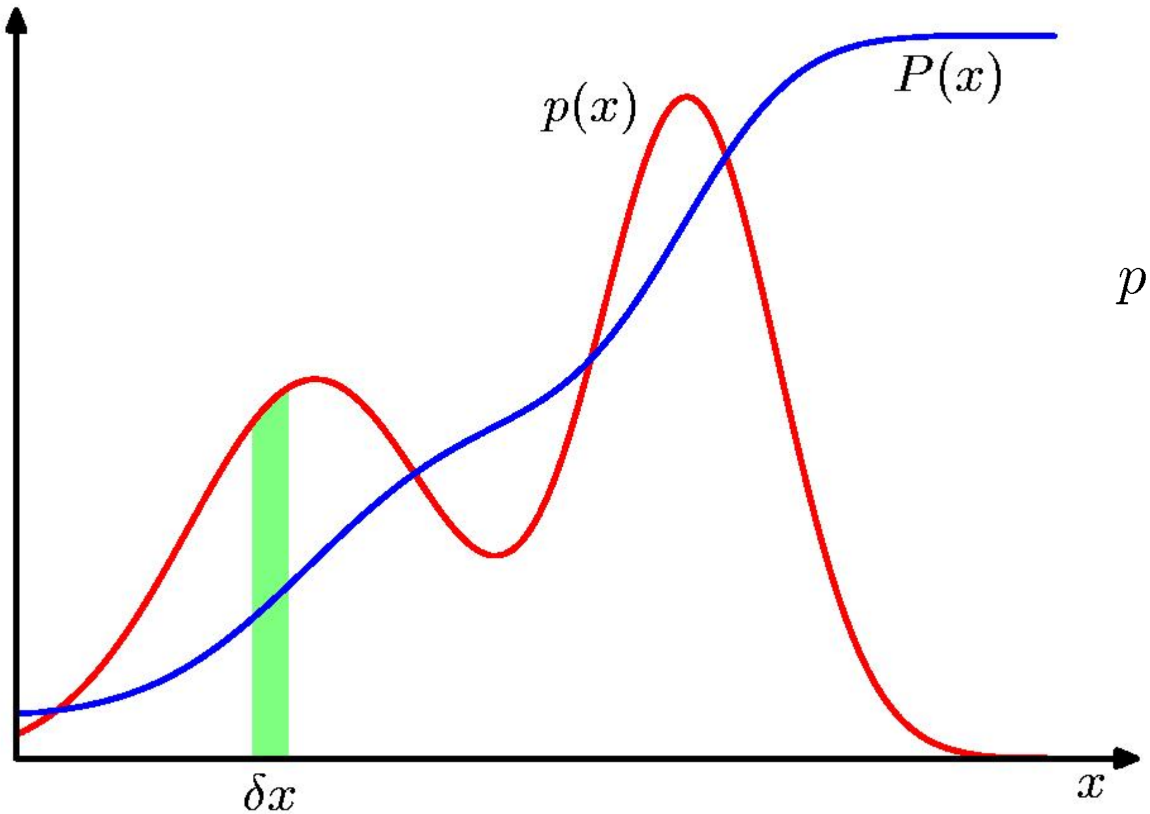
Bayes' Theorem

$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$$

$$p(X) = \sum_{Y} p(X|Y)p(Y)$$

posterior $\propto$ likelihood × prior

Probability Densities

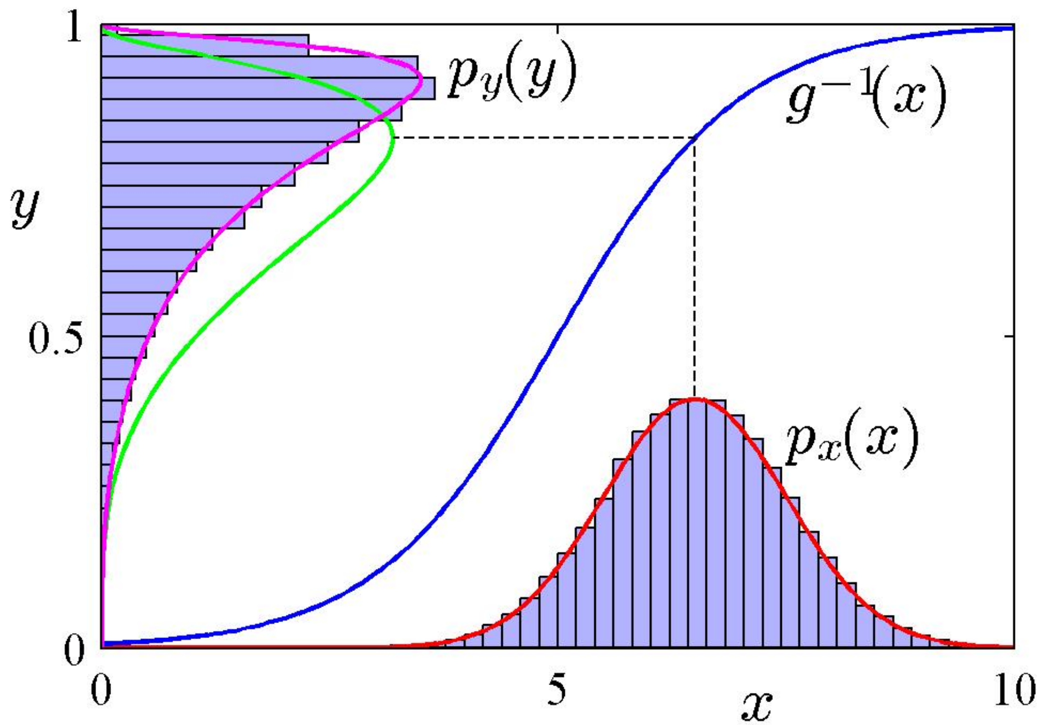


$$p(x \in (a, b)) = \int_a^b p(x)\,\mathrm{d}x$$

$$P(z) = \int_{-\infty}^{z} p(x)\,\mathrm{d}x$$

$$p(x) \geqslant 0 \qquad \int_{-\infty}^{\infty} p(x)\,\mathrm{d}x = 1$$

Transformed Densities



$$
\begin{aligned}
p_y(y) &= p_x(x)\left|\frac{\mathrm{d}x}{\mathrm{d}y}\right| \\
&= p_x(g(y))\,|g'(y)|
\end{aligned}
$$

Expectations

$$\mathbb{E}[f] = \sum_x p(x) f(x)$$

$$\mathbb{E}[f] = \int p(x) f(x)\,\mathrm{d}x$$

$$\mathbb{E}_x[f|y] = \sum_x p(x|y) f(x)$$

Conditional Expectation(discrete)

$$\mathbb{E}[f] \simeq \frac{1}{N}\sum_{n=1}^{N} f(x_n)$$

Approximate Expectation
(discrete and continuous)

Variances and Covariances

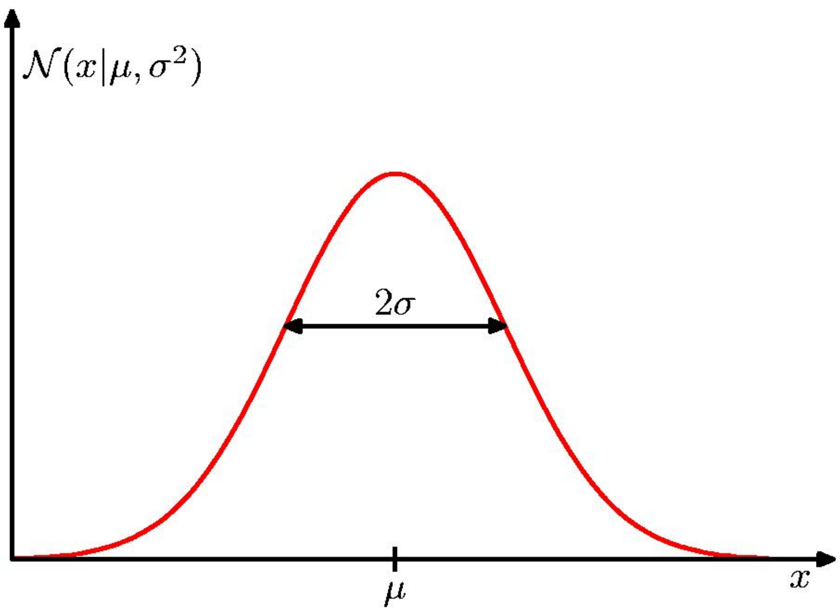
$$\operatorname{var}[f] = \mathbb{E}\left[\left(f(x) - \mathbb{E}[f(x)]\right)^2\right] = \mathbb{E}[f(x)^2] - \mathbb{E}[f(x)]^2$$

$$\begin{aligned} \operatorname{cov}[x, y] &= \mathbb{E}_{x,y}\left[\{x - \mathbb{E}[x]\}\{y - \mathbb{E}[y]\}\right] \\ &= \mathbb{E}_{x,y}[xy] - \mathbb{E}[x]\mathbb{E}[y] \end{aligned}$$

$$\begin{aligned} \operatorname{cov}[\mathbf{x}, \mathbf{y}] &= \mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\{\mathbf{x} - \mathbb{E}[\mathbf{x}]\}\{\mathbf{y}^{\mathrm{T}} - \mathbb{E}[\mathbf{y}^{\mathrm{T}}]\}\right] \\ &= \mathbb{E}_{\mathbf{x},\mathbf{y}}[\mathbf{x}\mathbf{y}^{\mathrm{T}}] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{y}^{\mathrm{T}}] \end{aligned}$$

The Gaussian Distribution

$$\mathcal{N}\left(x|\mu,\sigma^2\right) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$



$$\mathcal{N}(x|\mu,\sigma^2) > 0$$

$$\int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu,\sigma^2\right)\,\mathrm{d}x = 1$$

Gaussian Mean and Variance

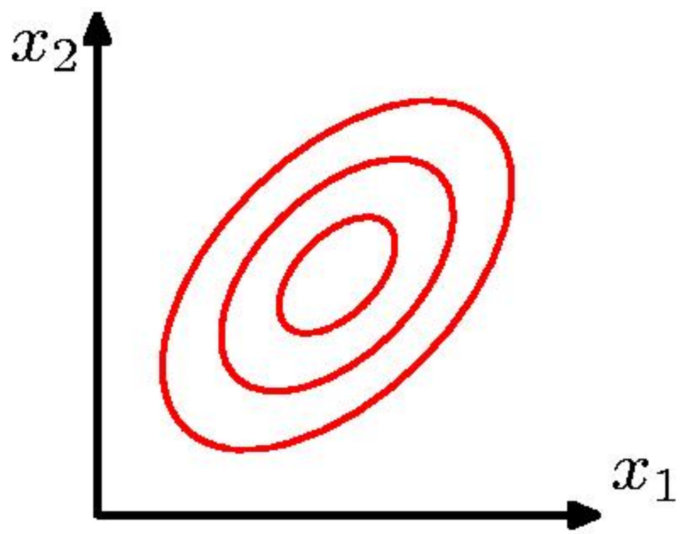
$$\mathbb{E}[x] = \int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu,\sigma^2\right) x\,\mathrm{d}x = \mu$$

$$\mathbb{E}[x^2] = \int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu,\sigma^2\right) x^2\,\mathrm{d}x = \mu^2 + \sigma^2$$

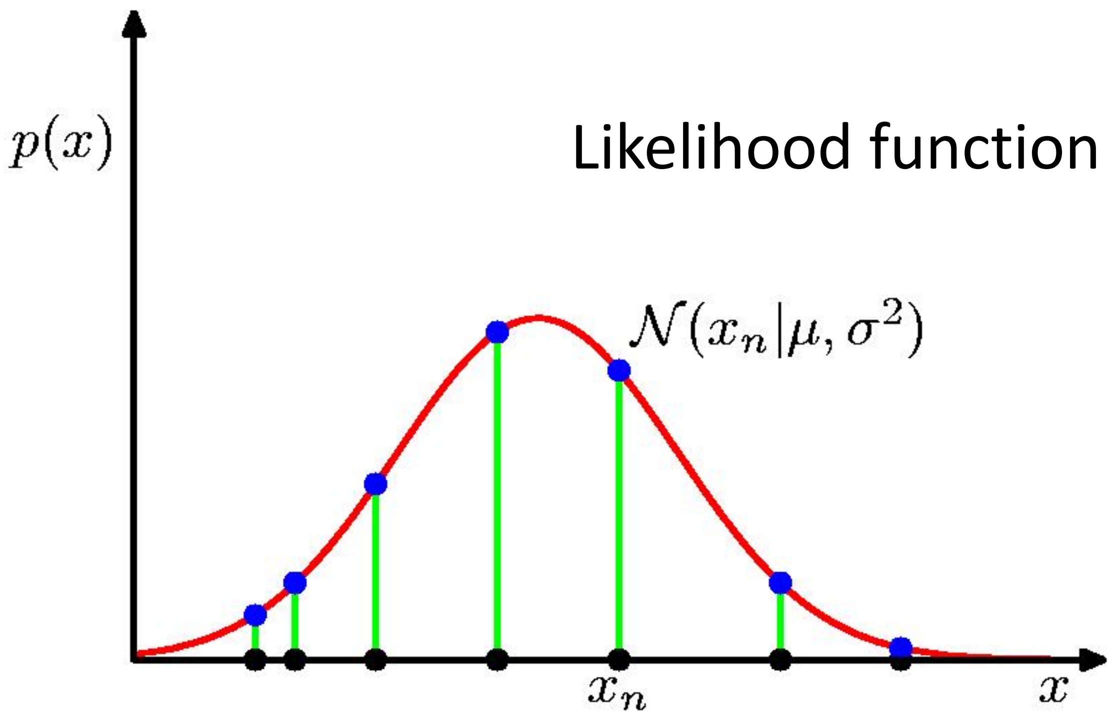
$$\mathrm{var}[x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2 = \sigma^2$$

The Multivariate Gaussian

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$



Gaussian Parameter Estimation



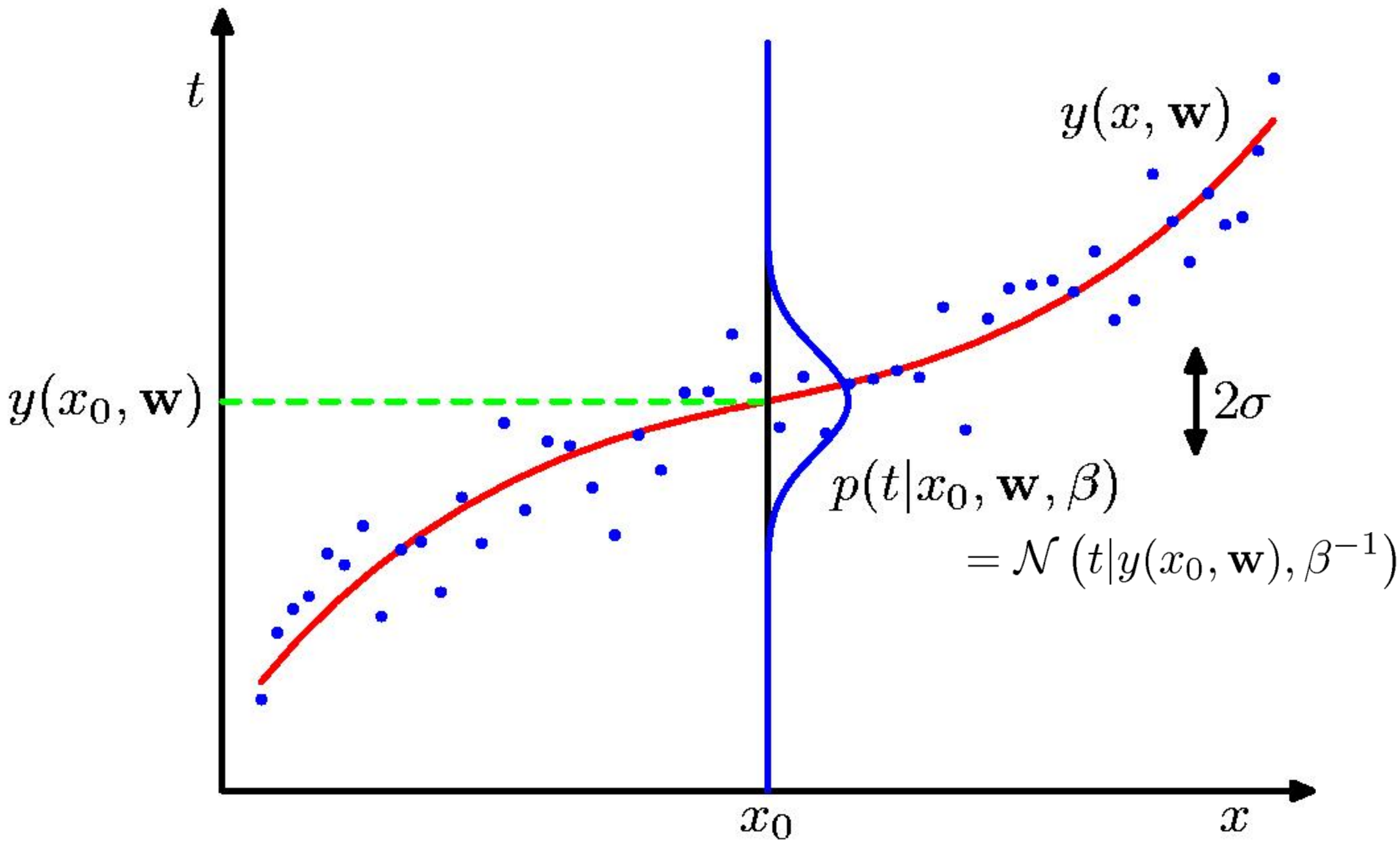
$$p(\mathbf{x}|\mu, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}\left(x_n|\mu, \sigma^2\right)$$

Maximum (Log) Likelihood

$$\ln p\left(\mathbf{x}|\mu, \sigma^2\right) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n - \mu)^2 - \frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi)$$

$$\mu_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n \qquad \sigma^2_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu_{\mathrm{ML}})^2$$

Curve Fitting Re-visited



Maximum Likelihood

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}\left(t_n | y(x_n, \mathbf{w}), \beta^{-1}\right)$$

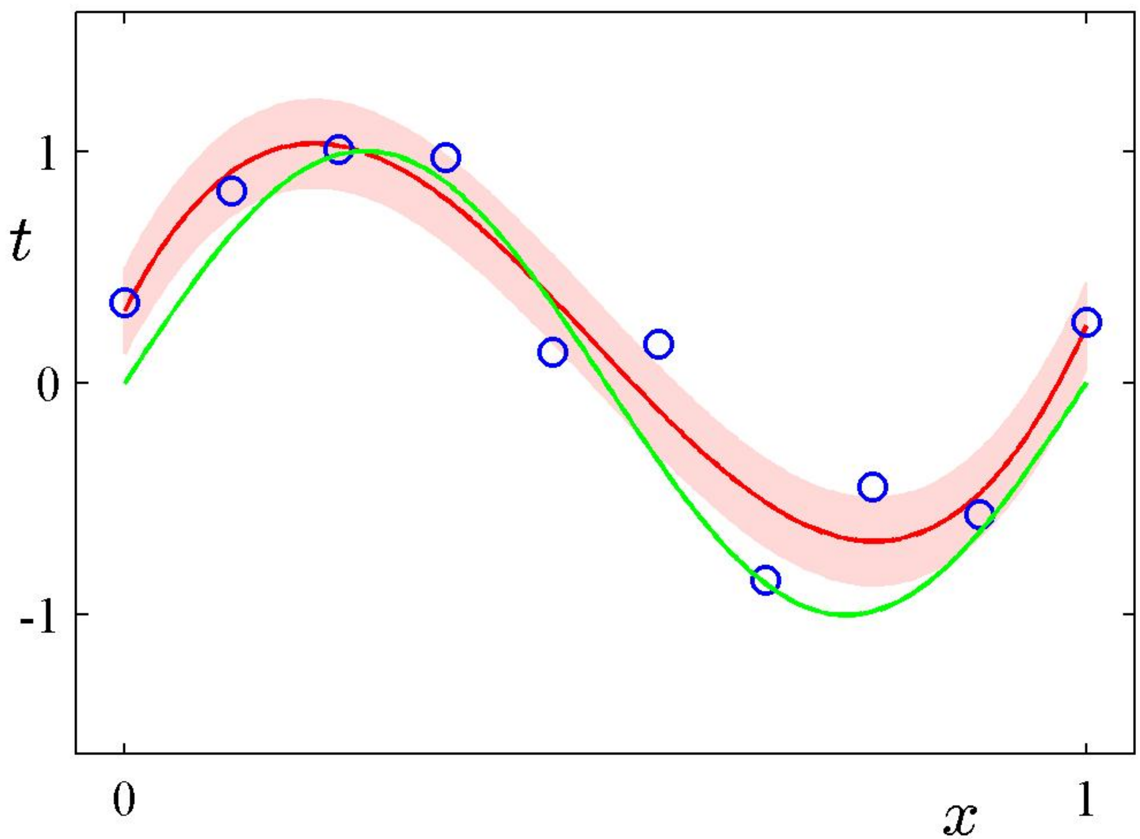
$$\ln p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \underbrace{-\frac{\beta}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2}_{\beta E(\mathbf{w})} + \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi)$$

Determine $\mathbf{w}_{\mathrm{ML}}$ by minimizing sum-of-squares error, $E(\mathbf{w})$.

$$\frac{1}{\beta_{\mathrm{ML}}} = \frac{1}{N} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}_{\mathrm{ML}}) - t_n\}^2$$

Predictive Distribution

$$p(t|x, \mathbf{w}_{\mathrm{ML}}, \beta_{\mathrm{ML}}) = \mathcal{N}\left(t|y(x, \mathbf{w}_{\mathrm{ML}}), \beta_{\mathrm{ML}}^{-1}\right)$$



MAP: A Step towards Bayes

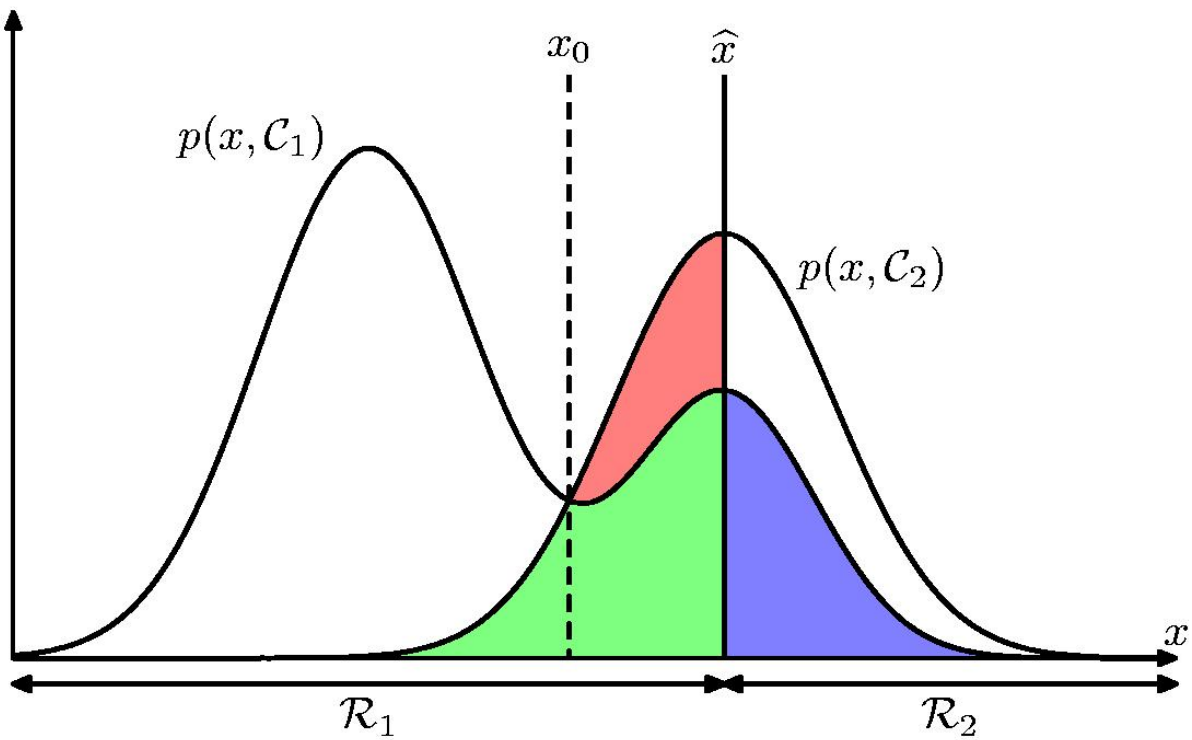
$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I}) = \left(\frac{\alpha}{2\pi}\right)^{(M+1)/2} \exp\left\{-\frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}\right\}$$

$$p(\mathbf{w}|\mathbf{x}, \mathbf{t}, \alpha, \beta) \propto p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta)p(\mathbf{w}|\alpha)$$

$$\beta\widetilde{E}(\mathbf{w}) = \frac{\beta}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

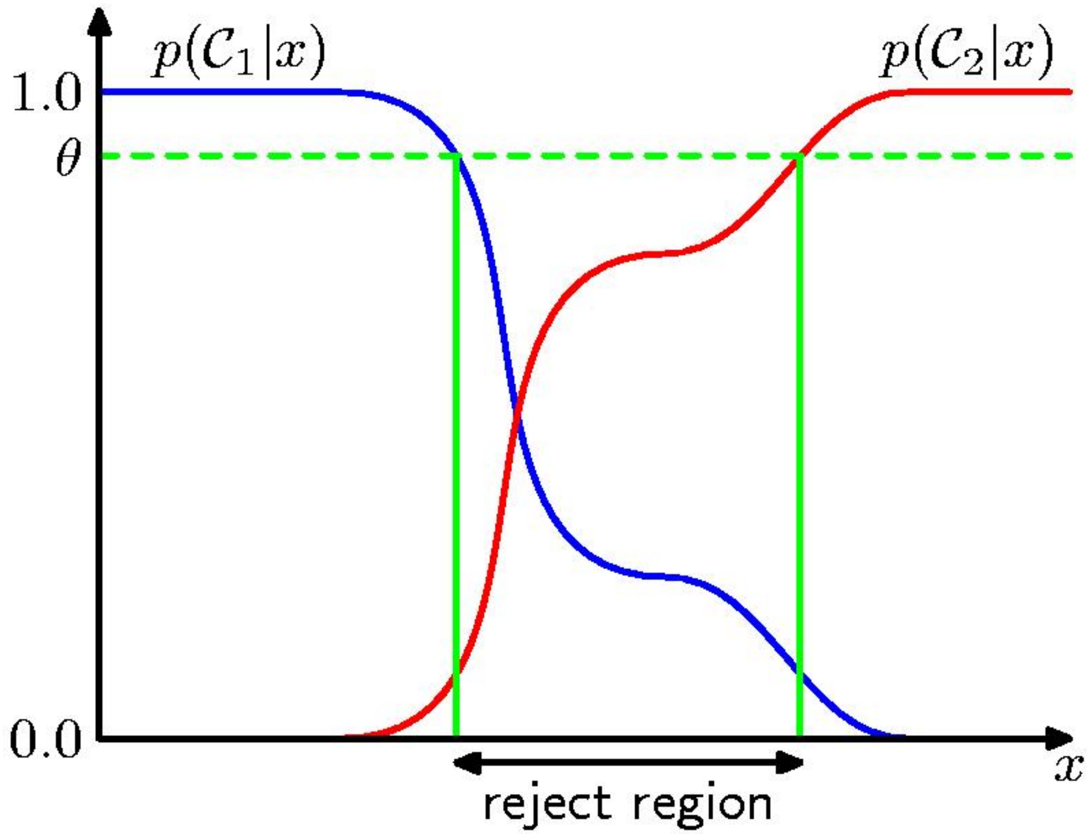
Determine $\mathbf{w}_{\mathrm{MAP}}$ by minimizing regularized sum-of-squares error, $\widetilde{E}(\mathbf{w})$.

Minimum Misclassification Rate



$$
\begin{aligned}
p(\text{mistake}) &= p(\mathbf{x} \in \mathcal{R}_1, \mathcal{C}_2) + p(\mathbf{x} \in \mathcal{R}_2, \mathcal{C}_1) \\
&= \int_{\mathcal{R}_1} p(\mathbf{x}, \mathcal{C}_2)\,\mathrm{d}\mathbf{x} + \int_{\mathcal{R}_2} p(\mathbf{x}, \mathcal{C}_1)\,\mathrm{d}\mathbf{x}.
\end{aligned}
$$

Reject Option



OLD