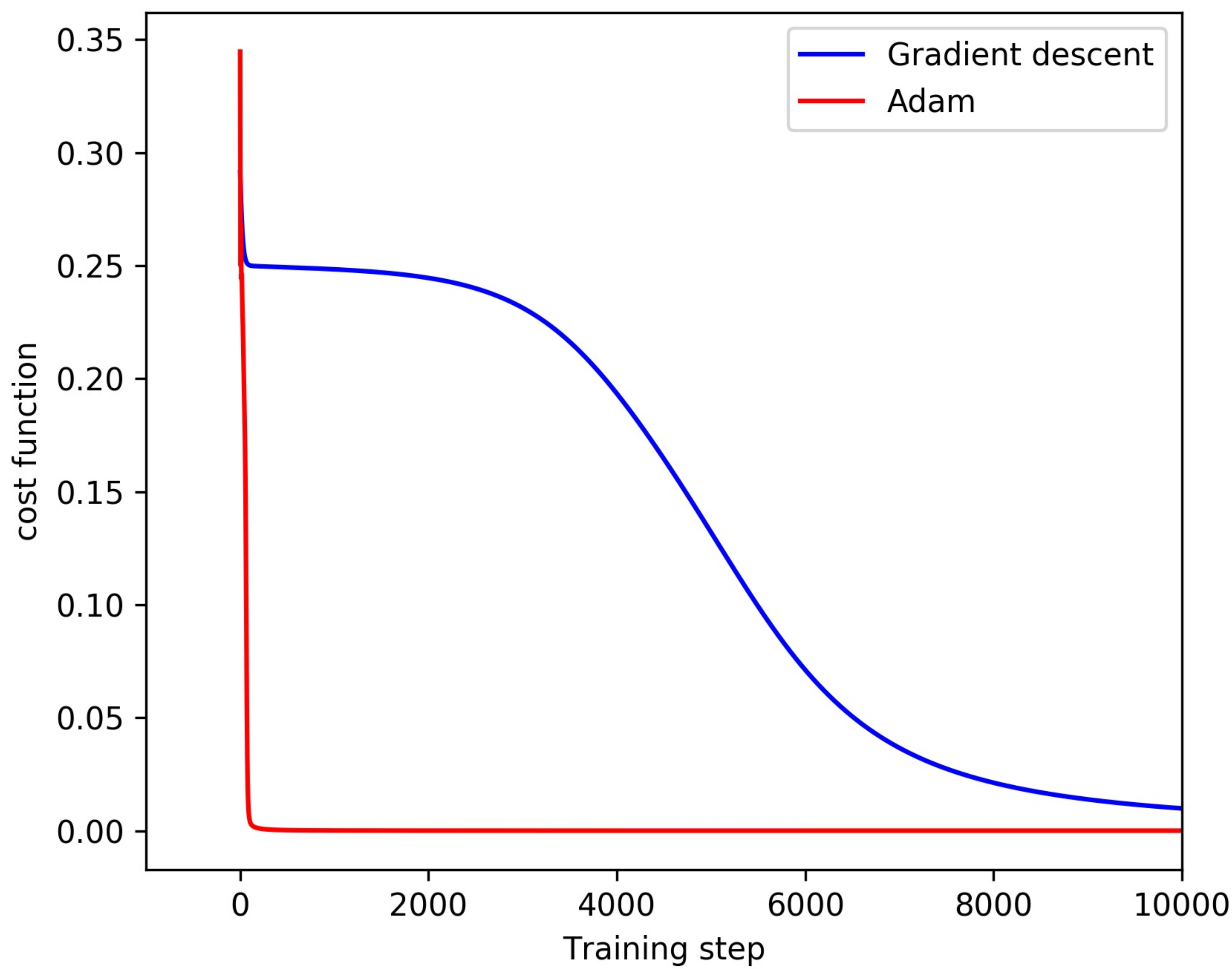

Gradient descent
Adam
cost function
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
2000
4000
6000
8000
10000
Training step

TensorFlow and Matlab training

- `tf.train.Optimizer`
- `tf.train.GradientDescentOptimizer`
- `tf.train.AdadeltaOptimizer`
- `tf.train.AdagradOptimizer`
- `tf.train.AdagradDAOptimizer`
- `tf.train.MomentumOptimizer`
- `tf.train.AdamOptimizer`
- `tf.train.FtrlOptimizer`
- `tf.train.ProximalGradientDescentOptimizer`
- `tf.train.ProximalAdagradOptimizer`
- `tf.train.RMSPropOptimizer`

> The network will be trained with Levenberg-Marquardt backpropagation algorithm (trainlm), unless there is not enough memory, in which case scaled conjugate gradient backpropagation (trainscg) will be used.

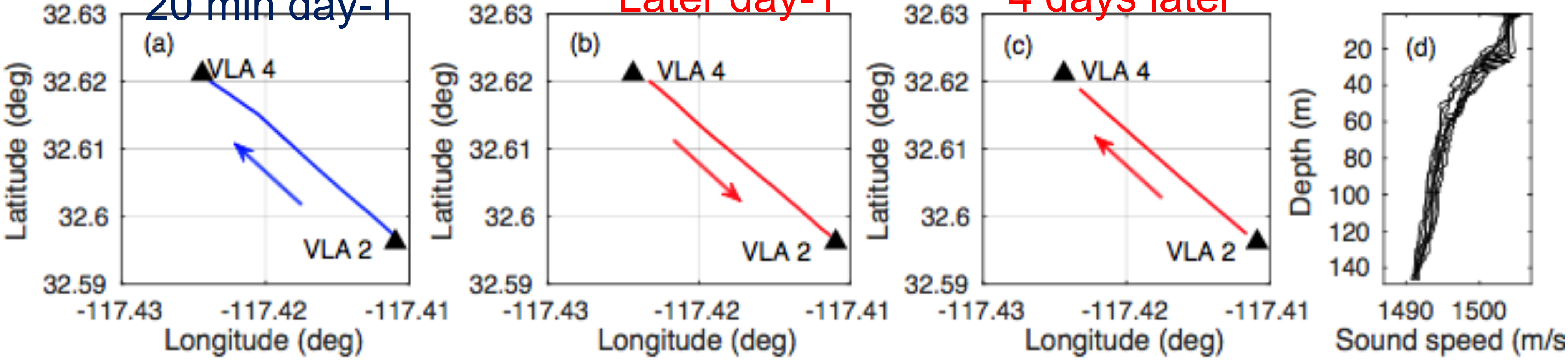
- **Grading>** Full scale of the letter grade. Grade consist of About 25 % homework, 25% seminar summary, and 50% final-project 2-4 man teams. Your and my purpose is to lean, so a good effort is sufficient. 10% reduction/day for a delayed homework.
  Seminar summary Based on one talk at the 3-day workshop Big Data and The Earth Sciences: Grand Challenges Workshop write a one or two page summary. Due at class on 7 June.
  Final project Either propose a topic before may 1. Or it will be based on my paper: Niu et al, 2017 on arXiv. We will make teams on April 24 and 26. Report due ABOUT June 16.
  Howework Homework 1 due **1 May on cody**
-

Neural Networks (Tensor flow) for tracking a ship

Training data,
20 min day-1

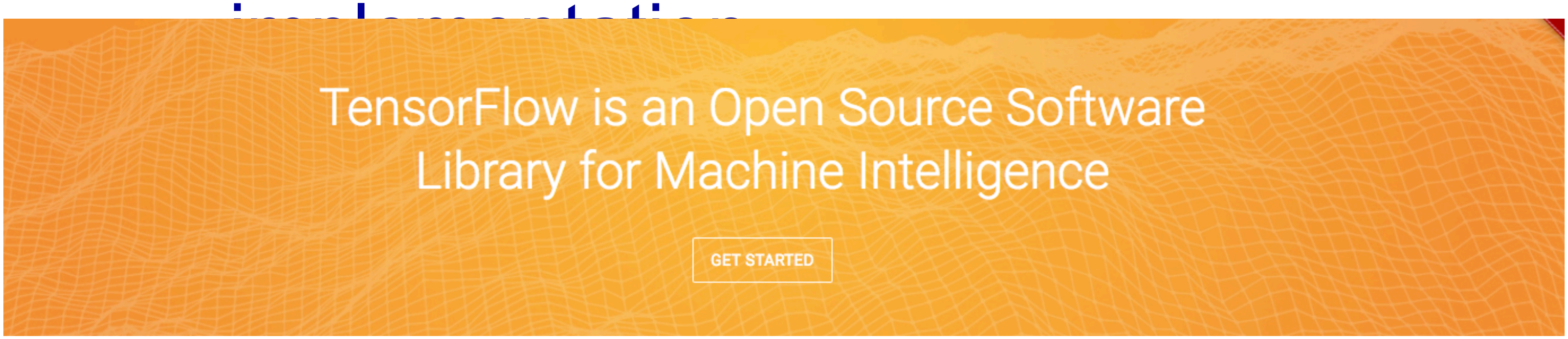
Test-data-1
Later day-1

Test-data-2
4 days later



Niu, Revees and Gerstoft JASA, 2017

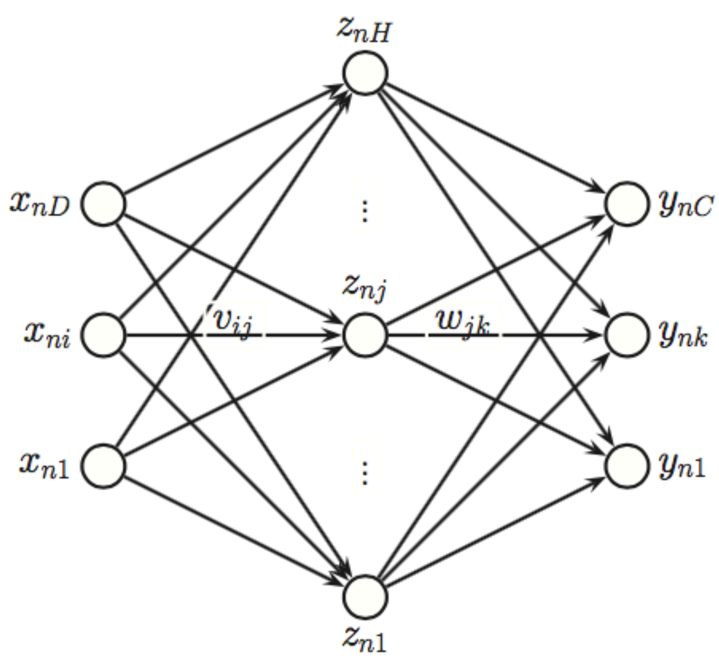
TensorFlow implementation

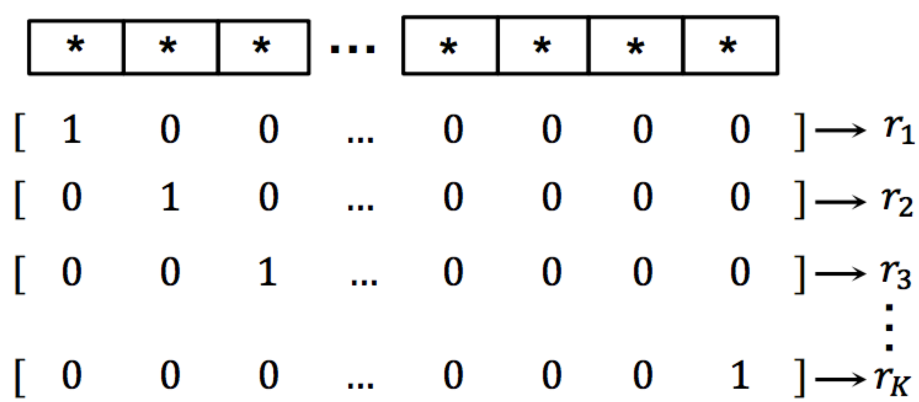


Input: Sample cov. matrix: 272 Neurons (16*17/2*2) per frequncy at each range

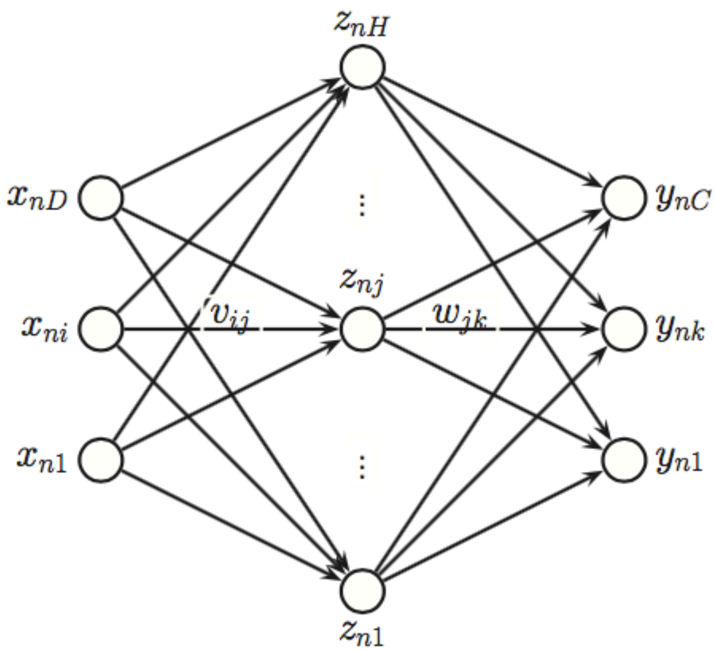
Output: binary range vector: 0.1-3km, 138 neurons

Just one middle layer 128 Neurons



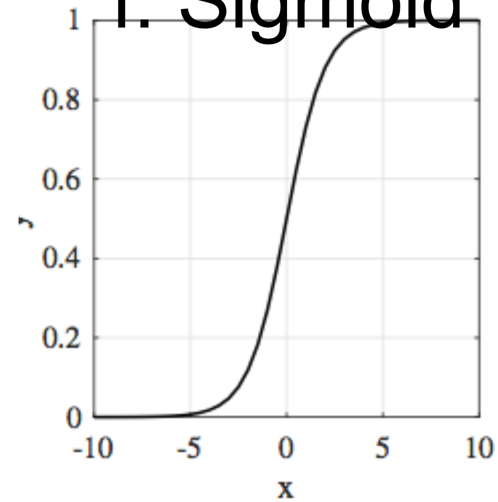


TensorFlow implementation



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

f: Sigmoid



h: softmax function:

$$\boldsymbol{a} = \boldsymbol{W}\boldsymbol{z}$$

Four-frequency localization

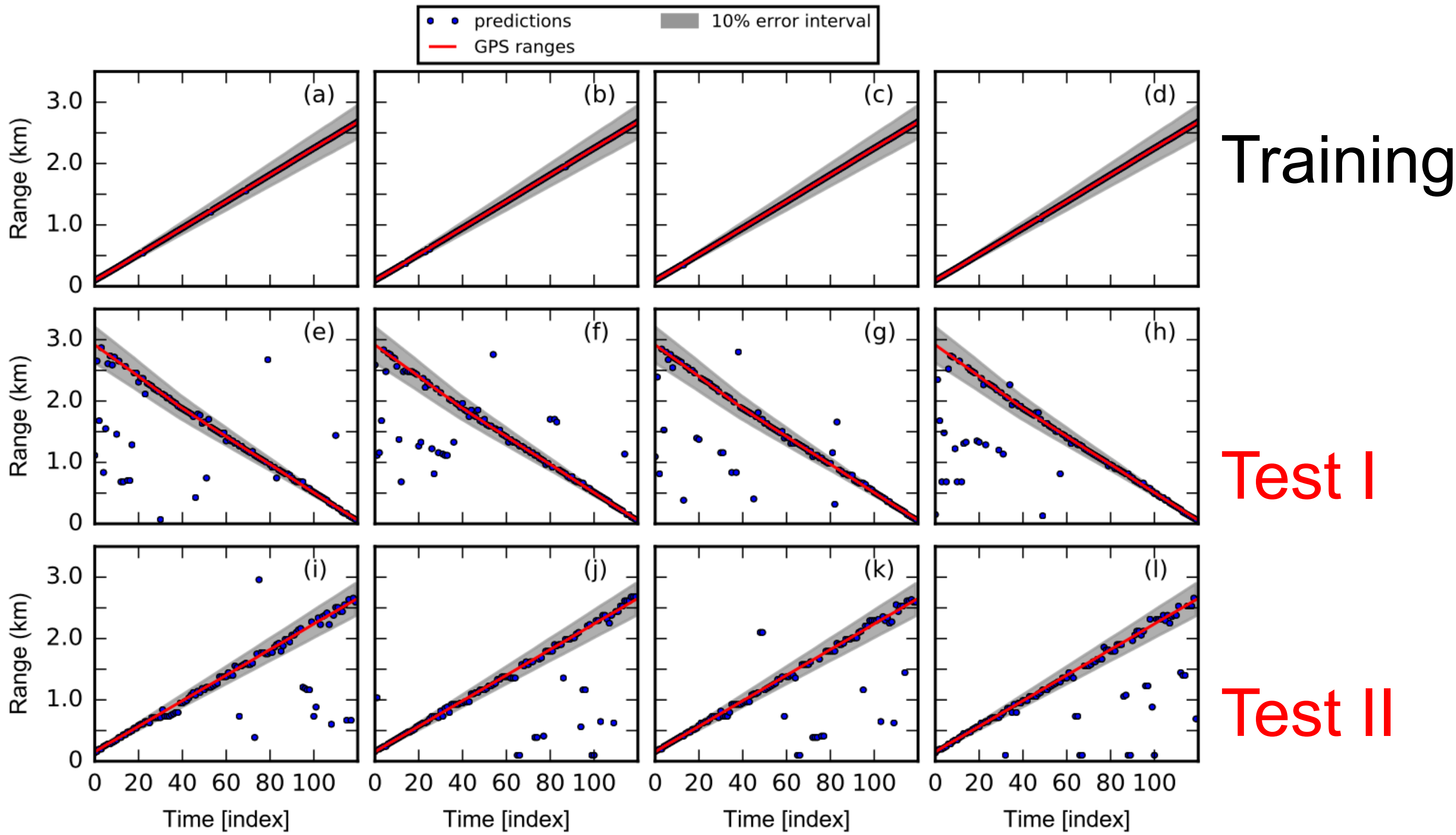
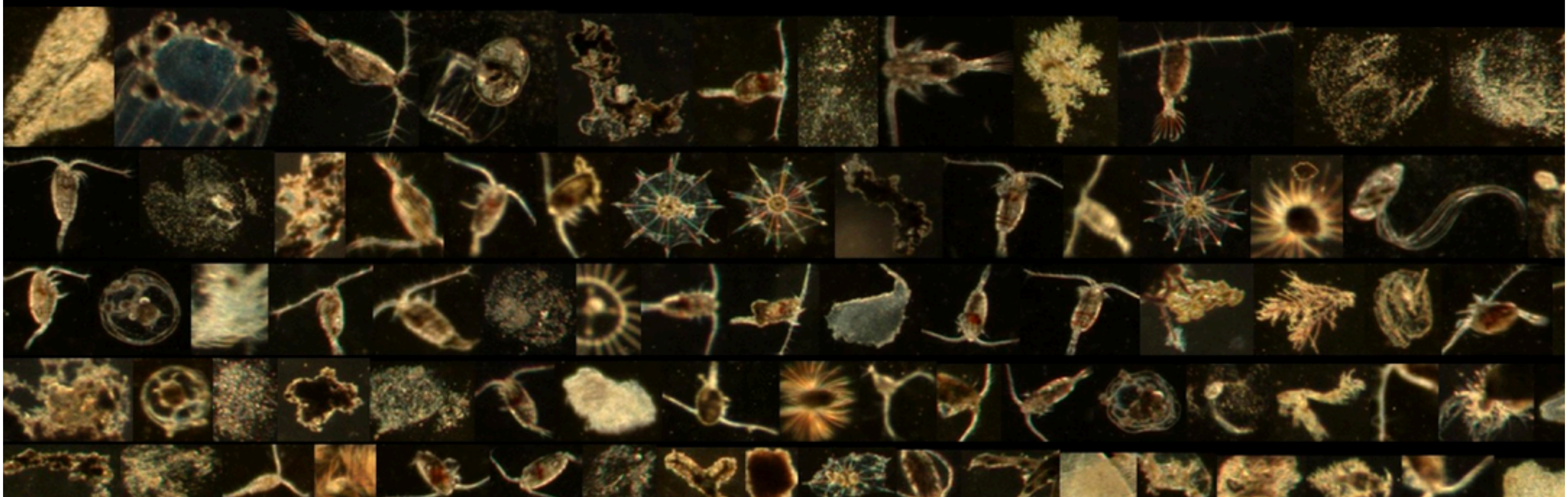


FIG. 12. (Color online) Range predictions on training data (a, b, c, d, first row), Test-Data-1 (e, f, g, h, second row) and Test-Data-2 (i, j, k, l, third row) by FNN with multi-frequency inputs. (a)(e)(i) 450, 490, 520, 550 Hz. (b)(f)(j) 560, 590, 620, 650 Hz. (c)(g)(k) 660, 690, 720, 750 Hz. (d)(h)(l)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

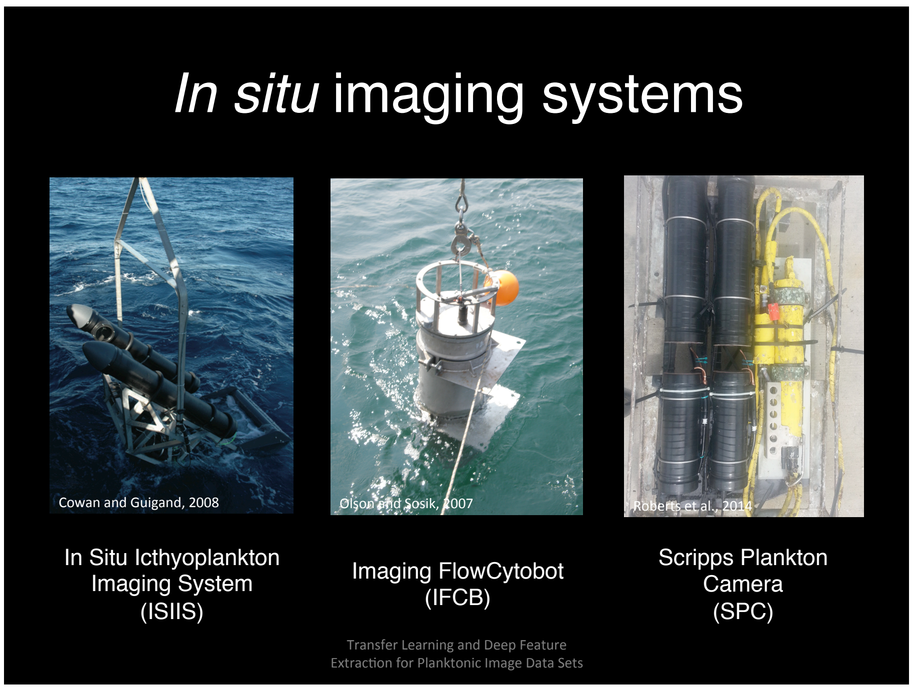


Eric C. Orenstein[1] and Oscar Beijbom[2]

[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



- Feature extraction example in python
- ~100 features
  Using SVM and RF
  better with CNN

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

RESEARCH ARTICLE

EARTH SCIENCES

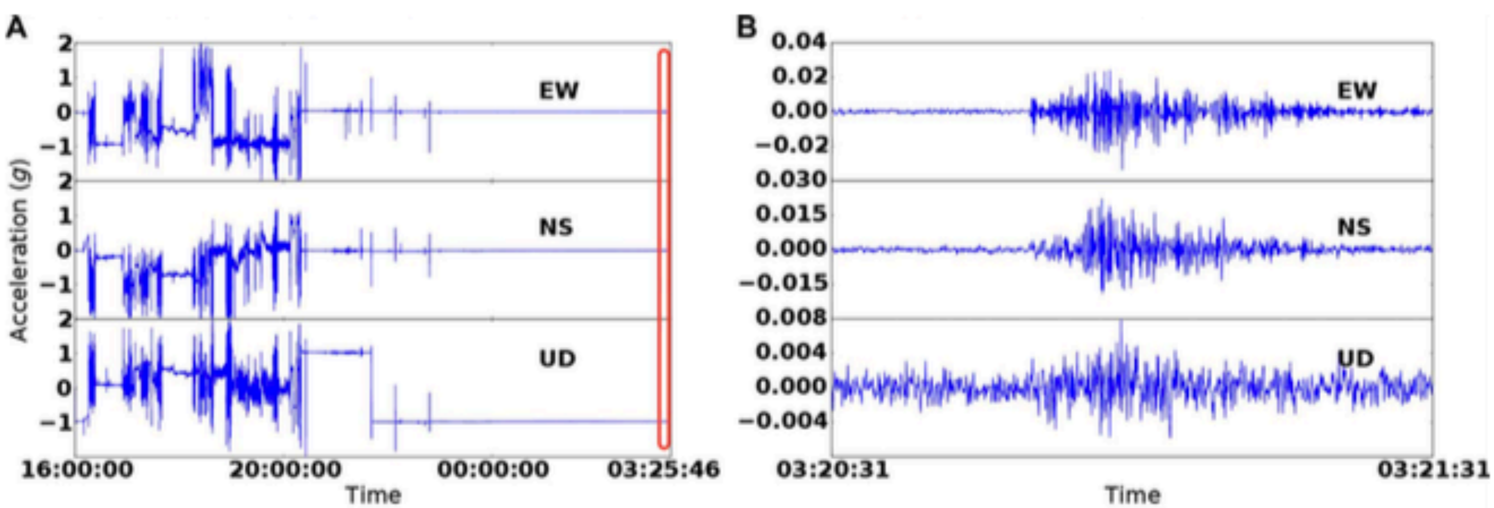
MyShake: A smartphone seismic network for earthquake early warning and beyond

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

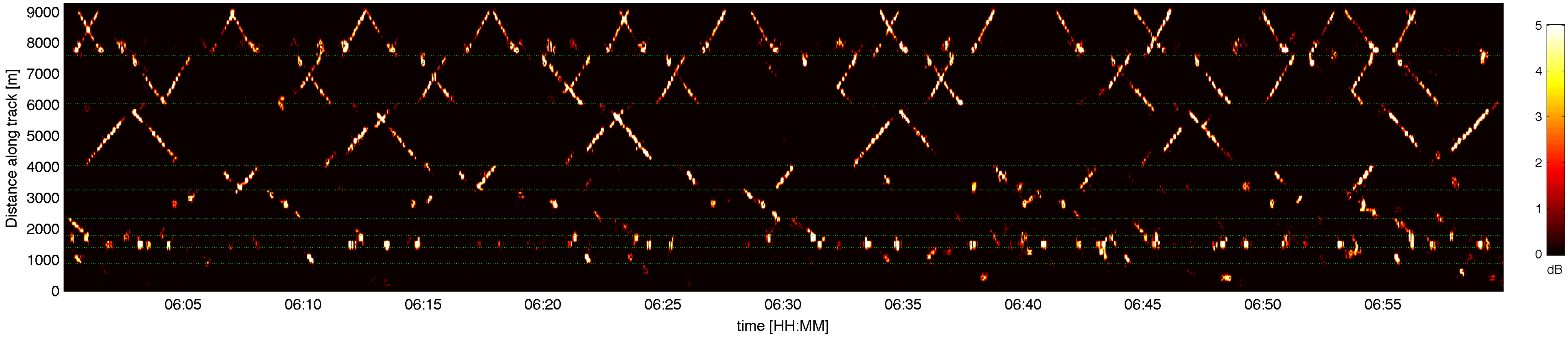
10.1126/sciadv.1501055

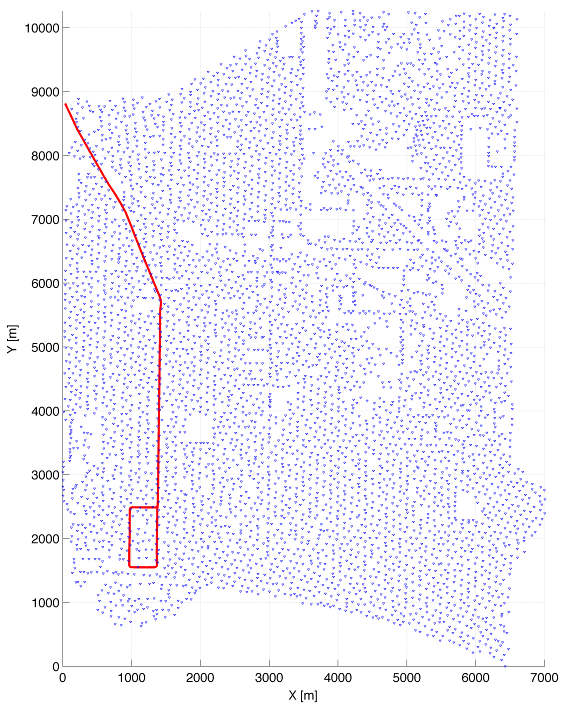
Large magnitude earthquakes in urban environments continue to kill and injure tens to hundreds of thousands of people, inflicting lasting societal and economic disasters. Earthquake early warning (EEW) provides seconds to minutes of warning, allowing people to move to safe zones and automated slowdown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

RESEARCH ARTICLE

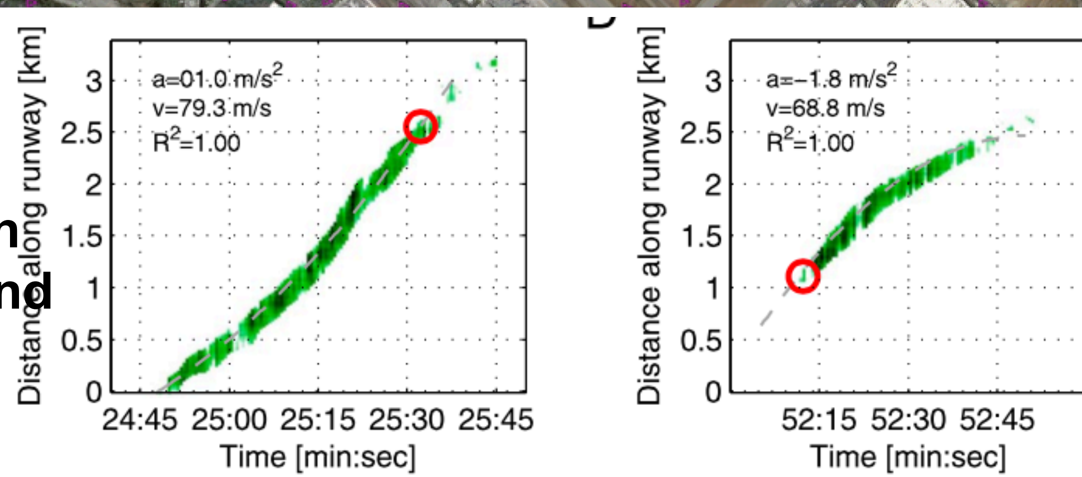


Noise Tracking of Cars/Trains/Airplanes





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

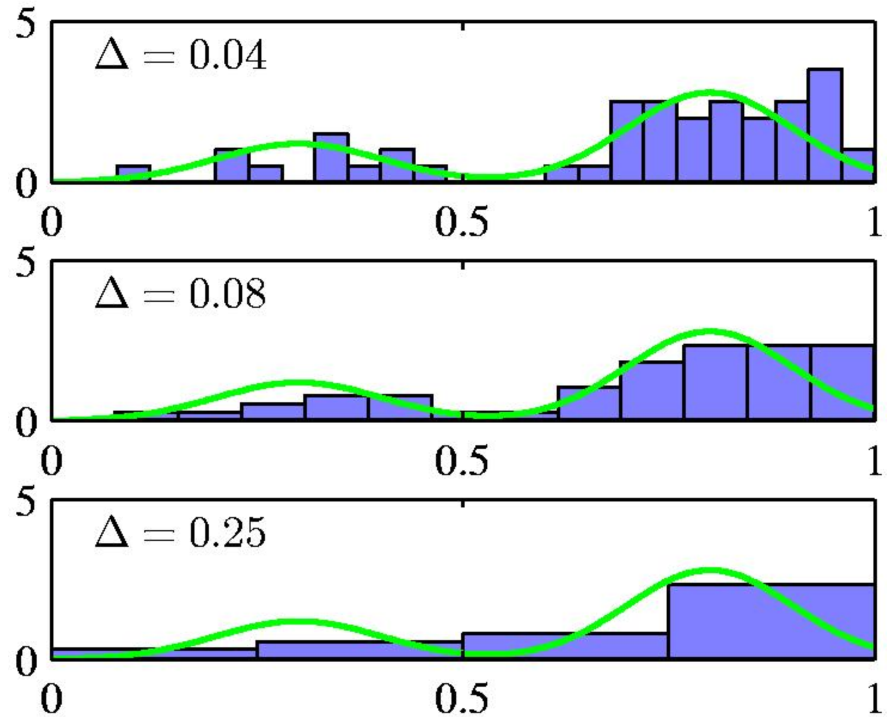
Nonparametric Methods (1)

- Parametric distribution models are restricted to specific forms, which may not always be suitable; for example, consider modelling a multimodal distribution with a single, unimodal model.
- Nonparametric approaches make few assumptions about the overall shape of the distribution being modelled.
- 1000 parameter versus 10 parameter

Histograms as density models

$$p_i = \frac{n_i}{N\Delta_i}$$

- For low dimensional data we can use a histogram as a density model.
  - How wide should the bins be? (width=regulariser)
  - Do we want the same bin-width everywhere?
  - Do we believe the density is zero for empty bins?



green curve is true density

Some good and bad properties of histograms as density estimators

- There is no need to fit a model to the data.
  - We just compute some very simple statistics (the number of datapoints in each bin) and store them.
- The number of bins is exponential in the dimensionality of the dataspace. So high-dimensional data is tricky:
  - We must either use big bins or get lots of zero counts (or adapt the local bin-width to the density)
- The density has silly discontinuities at the bin boundaries.
  - We must be able to do better by some kind of smoothing.

Local density estimators

- Estimate the density in a small region to be

$$p(x) = \frac{K}{N\,V}$$

K ← points in region

V ← volume of region

N ↑ total points

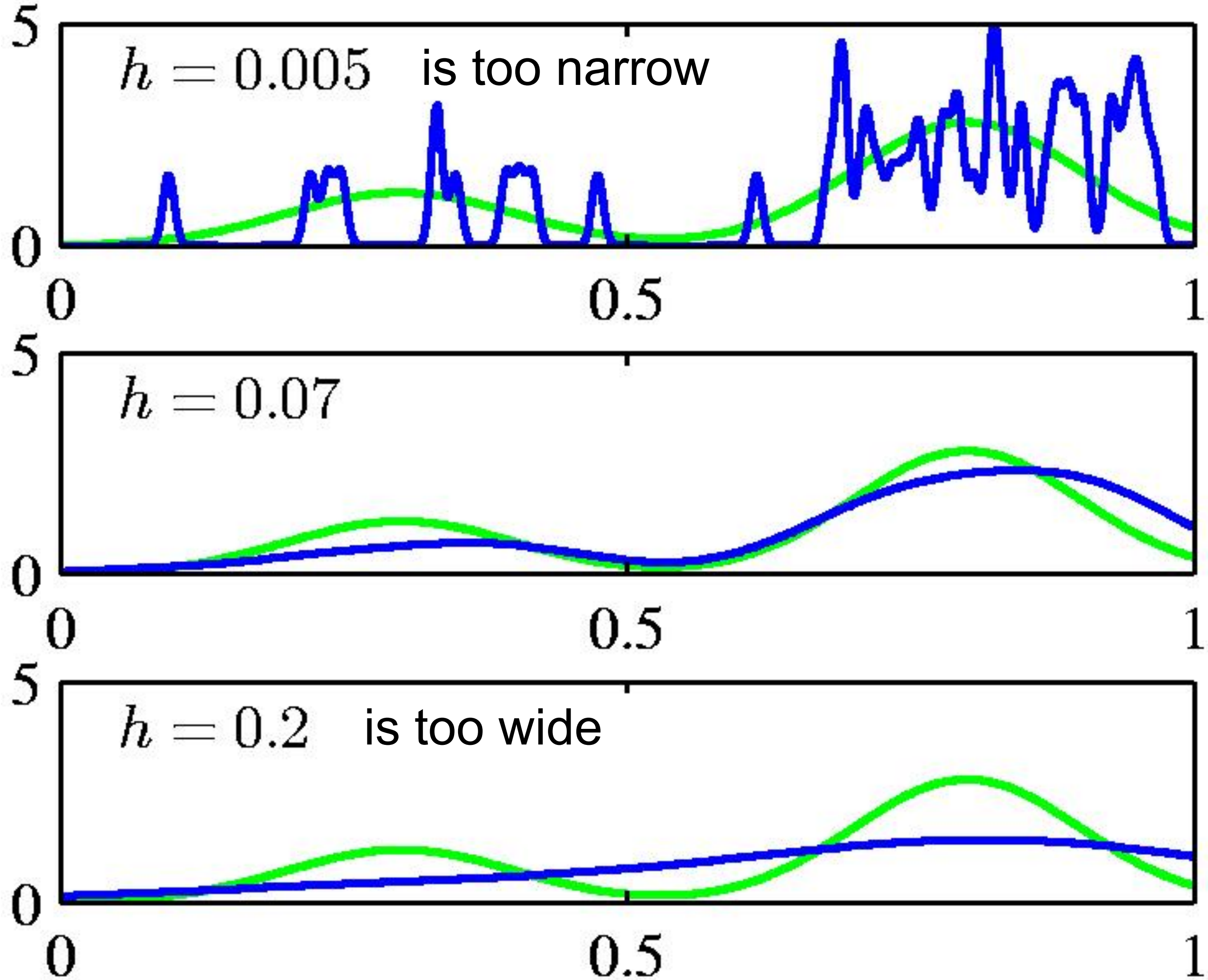
- Problem 1: Variance in estimate if K is small.
- Problem 2: Unmodelled variation across the region if V is big compared with the smoothness of the true density

Kernel density estimators

- Use regions centered on the datapoints
  - Allow the regions to overlap.
  - Let each individual region contribute a total density of 1/N
  - Use regions with soft edges to avoid discontinuities (e.g. isotropic Gaussians)

$$p(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N} \frac{1}{(2\pi\sigma^2)^{D/2}} \exp\left(\frac{\| \mathbf{x} - \mathbf{x}_n \|^2}{2\sigma^2}\right)$$

The density modeled by a kernel density estimator



Nearest neighbor methods for density estimation

$$p(x) = \frac{K}{N\,V}$$

points in region

volume of region

total points

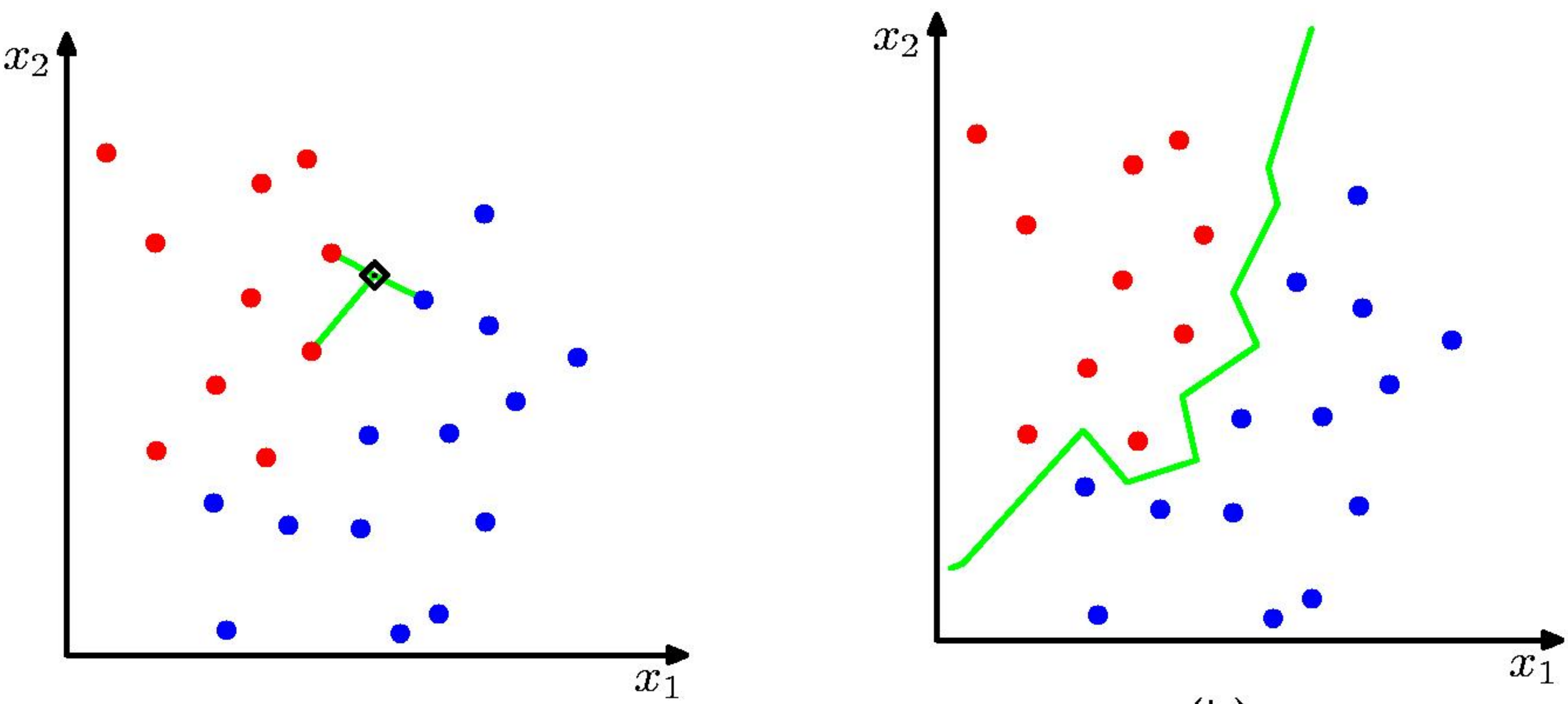
- Vary the size of a hyper-sphere around each test point so that exactly K training datapoints fall inside the hyper-sphere.
  - Does this give a fair estimate of the density?
- Nearest neighbors is usually used for classification or regression:
  - For regression, average the predictions of the K nearest neighbors.
  - For classification, pick the class with the most votes.
    - How should we break ties?

Nearest neighbor methods for classification and regression

- Nearest neighbors is usually used for classification or regression:
- For regression, average the predictions of the K nearest neighbors.
  - How should we pick K?
- For classification, pick the class with the most votes.
    - How should we break ties?
    - Let the k'th nearest neighbor contribute a count that falls off with k. For example, $1+\frac{1}{2^k}$

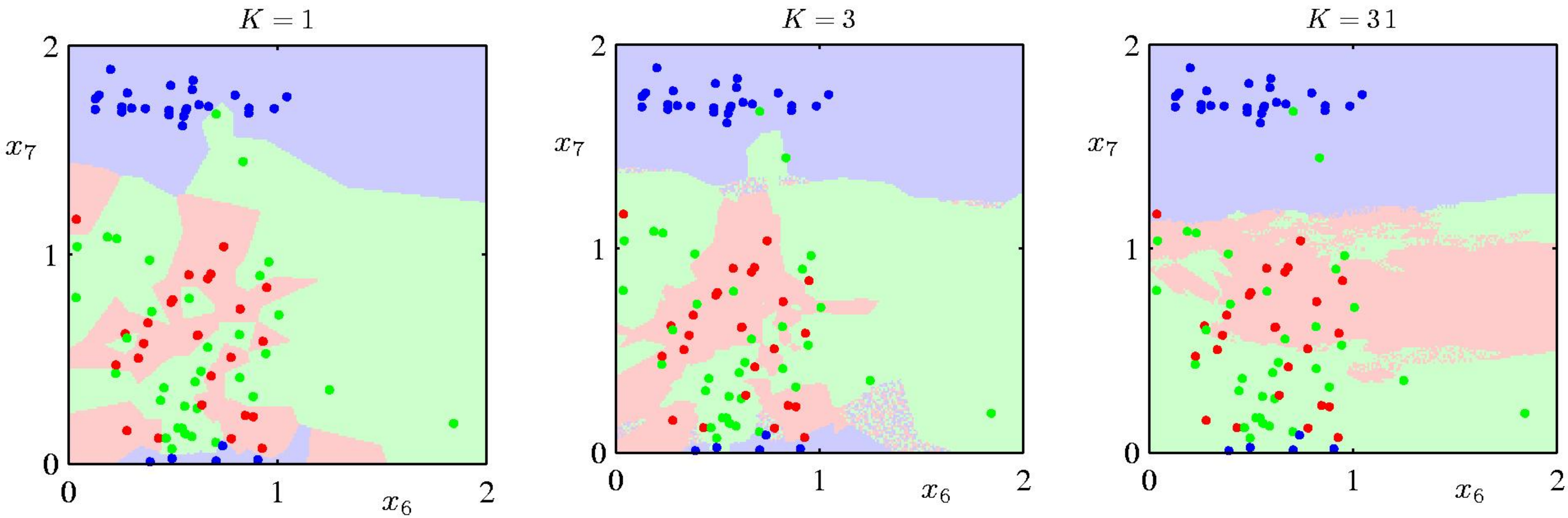
The decision boundary implemented by 3NN

The boundary is always the perpendicular bisector of the line between two points (Vornoi tesselation)



(a) (b)

Regions defined by using various numbers of neighbors



Gaussian Kernels

- Gaussian Kernel

$$k(x, x') = \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}')^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{x}')\right)$$

Diagonal Σ: (this gives ARD)

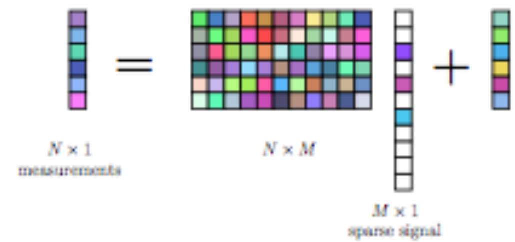
$$k(x, x') = \exp\left(-\frac{1}{2}\sum_{i}^{N}\frac{\left(x_i - x_i'\right)^2}{\sigma_i^2}\right)$$

Isotropic σ_i^2 gives an RBF

$$k(x, x') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2}{2\sigma^2}\right)$$

Sparse Bayesian Learning (SBL)

$$\text{Model}: \ \mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$$
$$\text{Prior}: \ \mathbf{x} \sim \mathcal{N}(\mathbf{x}; 0, \boldsymbol{\Gamma})$$
$$\boldsymbol{\Gamma} = \text{diag}(\gamma_1, \ldots, \gamma_M)$$
$$\text{Likelihood}: \ p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}; \mathbf{A}\mathbf{x}, \sigma^2 \mathbf{I}_N)$$



$$\text{Evidence}: \ p(\mathbf{y}) = \int_{\mathbf{x}} p(\mathbf{y}|\mathbf{x}) p(\mathbf{x}) d\mathbf{x} = \mathcal{N}(\mathbf{y}; 0, \boldsymbol{\Sigma}_{\mathbf{y}})$$
$$\boldsymbol{\Sigma}_{\mathbf{y}} = \sigma^2 \mathbf{I}_N + \mathbf{A}\boldsymbol{\Gamma}\mathbf{A}^H$$

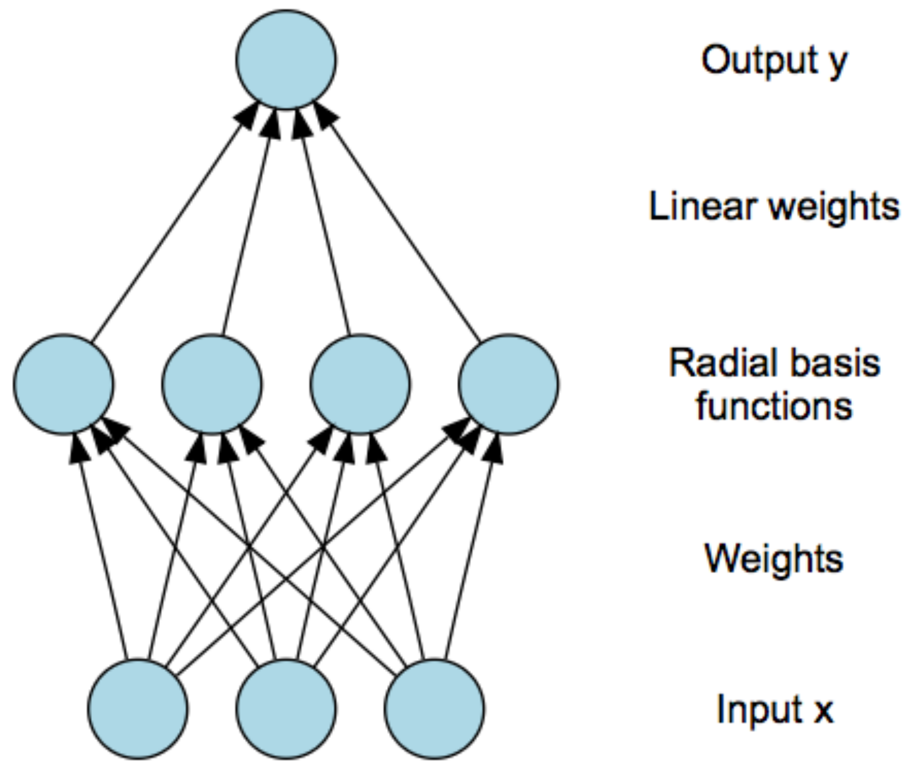
$$\text{SBL solution}: \ \hat{\boldsymbol{\Gamma}} = \arg\max_{\boldsymbol{\Gamma}} p(\mathbf{y})$$
$$= \arg\min_{\boldsymbol{\Gamma}} \left\{ \log |\boldsymbol{\Sigma}_{\mathbf{y}}| + \mathbf{y}^H \boldsymbol{\Sigma}_{\mathbf{y}}^{-1} \mathbf{y} \right\}$$

M.E.Tipping, "Sparse Bayesian learning and the relevance vector machine," Journal of Machine Learning Research, June 2001.

Kernel machine

- GLM with feature vector
- $\phi(x) = [k(x, \mu_1), \ldots, k(x, \mu_K)]$
- If k is RBF then this is RBF network

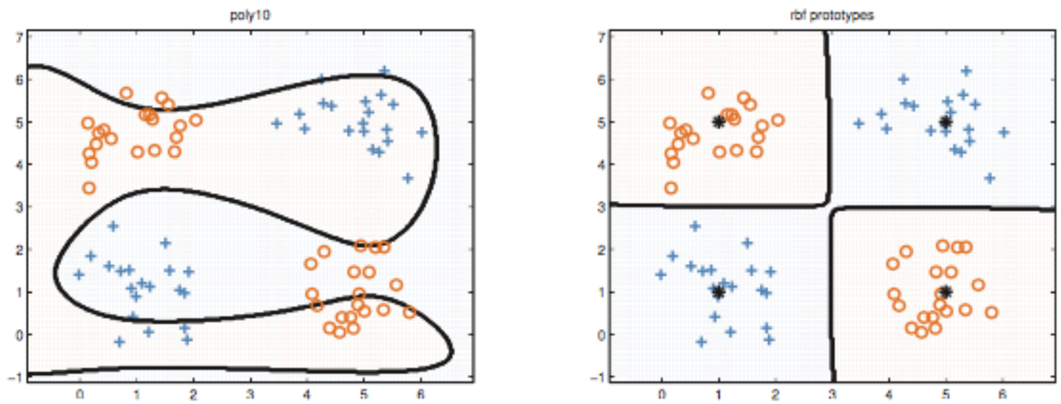
How to chose μ?



Kernels

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

(a)



(b) (c)

Figure 14.2 (a) xor truth table. (b) Fitting a linear logistic regression classifier using degree 10 polynomial expansion. (c) Same model, but using an RBF kernel with centroids specified by the 4 black crosses. Figure generated by `logregXorDemo`.

$$k(x, x') = k(x', x) \geq 0$$

NOT

USED

YET