Recitation

Lyle Ungar

Computer and information Science

Learning Objectives

Representation, loss function, search
Selection of loss functions in practice
Generalized linear models and RBF

K-NN

- When doing k-nn with *y* a real number, what is the loss function L(y, ŷ) being minimized?

Decision Trees

- When doing decision trees with y a Boolean, what is the loss function being minimized?

Which model to use?

$y = \mathbf{x}^T\mathbf{w}$

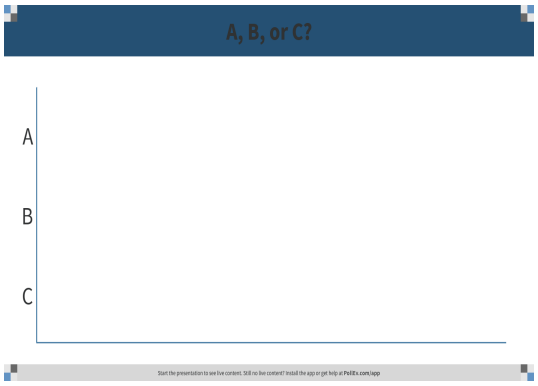
Predict income based on age, sex, and state or country you were born in

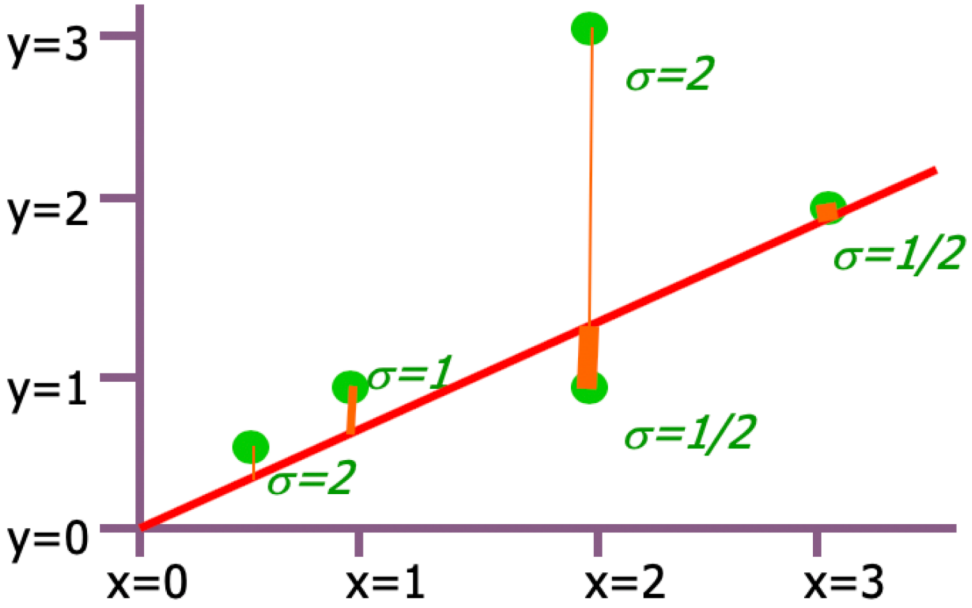
What exactly are **x** and y?

Which loss function to use?

$$||y - Xw||_p$$

a) p=0
b) p=1
c) p=2





Which loss function to use?

You are building a model to estimate the cost, *y*, of a software project that you are bidding on as a contractor (as a function of lots of features of the project, including estimates of lines of code, hours of meetings, complexity of specifications).

Which loss function to use?

You are writing a search algorithm that returns web pages as a function of the search query, the words on the web page the person is searching from, and the search history of that user.

Which regression penalty to use?

$$\text{Error} + \lambda_2 ||\mathbf{w}||_2^2 + \lambda_1 ||\mathbf{w}||_1 + \lambda_0 ||\mathbf{w}||_0$$

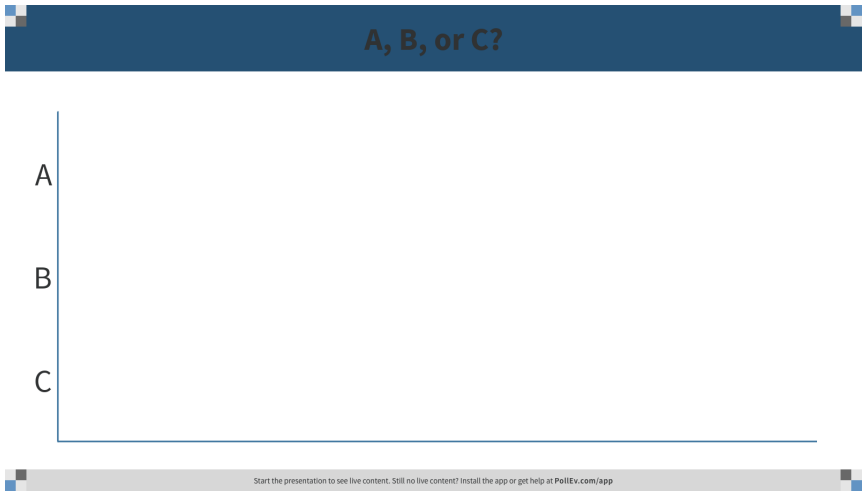
- If you want the model to be scale invariant?
- If you want to have a small model?
- If you want a convex optimization problem?

- Your training error for ridge regression is substantially lower than your testing error.
- You shouid

  a) increase λ

  b) decrease λ

  c) no change in λ

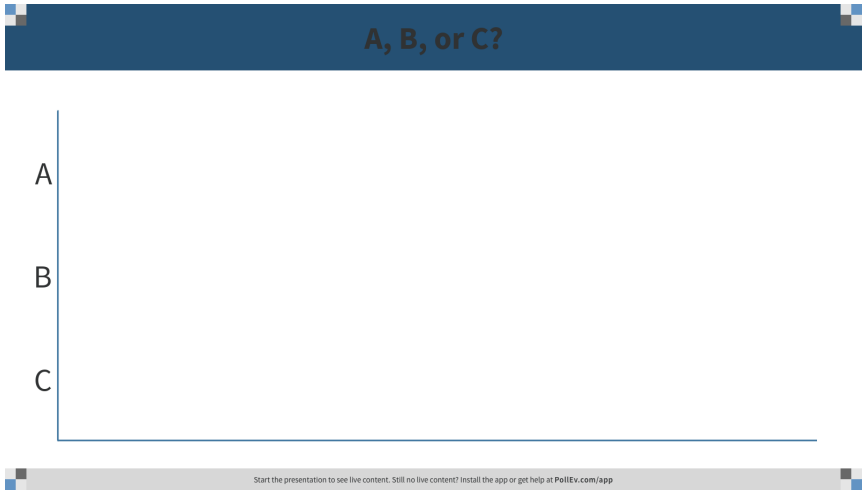


- Your training error for ridge regression is the same as your testing error.
- You shouid

  a) increase λ

  b) decrease λ

  c) no change in λ



Generalized linear models

- **Basic Linear Model:** $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=0}^{d} \theta_j x_j$
- **Generalized Linear Model:** $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=0}^{d} \theta_j \phi_j(\boldsymbol{x})$
- **Or add a link function:** $h_{\theta}(\boldsymbol{x}) = f(\boldsymbol{w}^T \boldsymbol{x})$

Based on slide by Geoff Hinton

Linear Basis Function Models

- Generally,

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=0}^{d} \theta_j \phi_j(\boldsymbol{x})$$

basis function

- Typically, $\phi_0(\boldsymbol{x}) = 1$ so that θ_0 acts as a bias
- In the simplest case, we use linear basis functions

$$\phi_j(\boldsymbol{x}) = x_j$$

- Could use polynomials or Gaussians

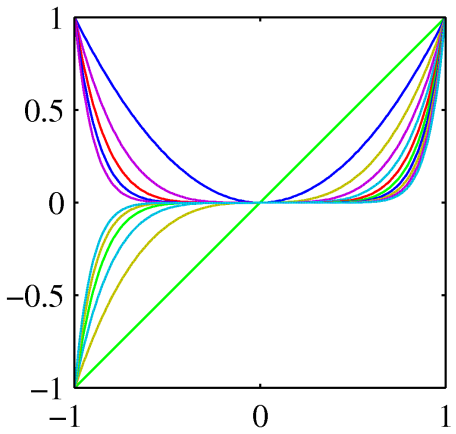
Based on slide by Christopher Bishop (PRML)

Linear Basis Function Models

- **Polynomial basis functions**

$$\phi_j(x) = x^j$$

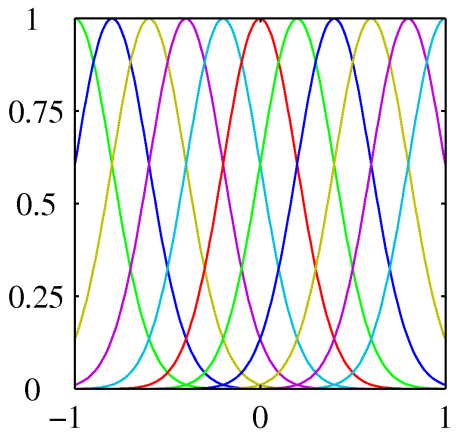
  - **Global – mostly crappy**



- **Gaussian basis functions:**

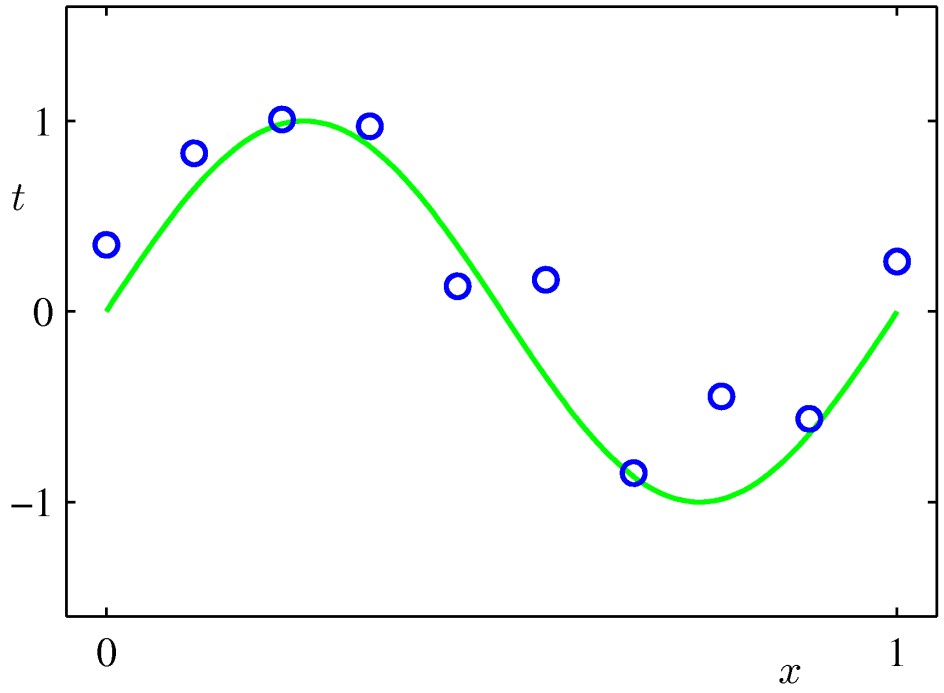
$$\phi_j(x) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\}$$

  - **Local – good!**



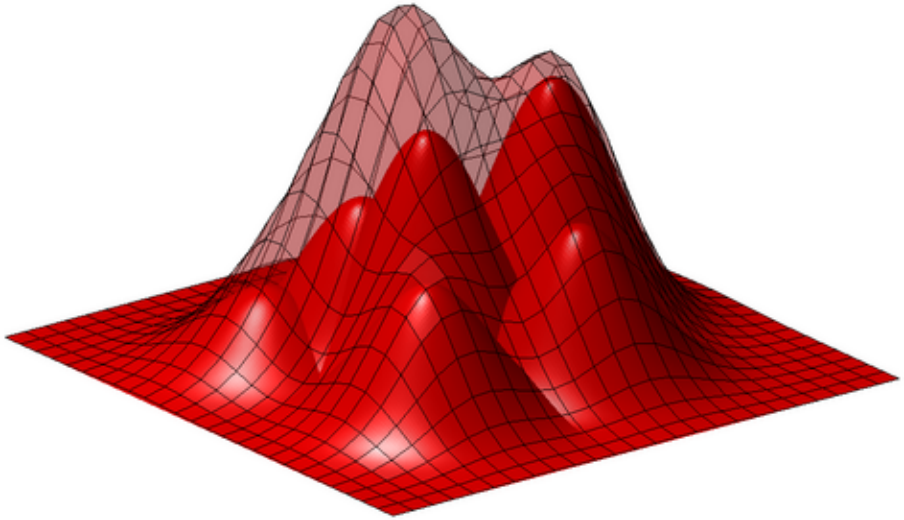
Based on slide by Christopher Bishop (PRML)

Fitting a Polynomial Curve with a Linear Model



$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \ldots + \theta_p x^p = \sum_{j=0}^{p} \theta_j x^j$$

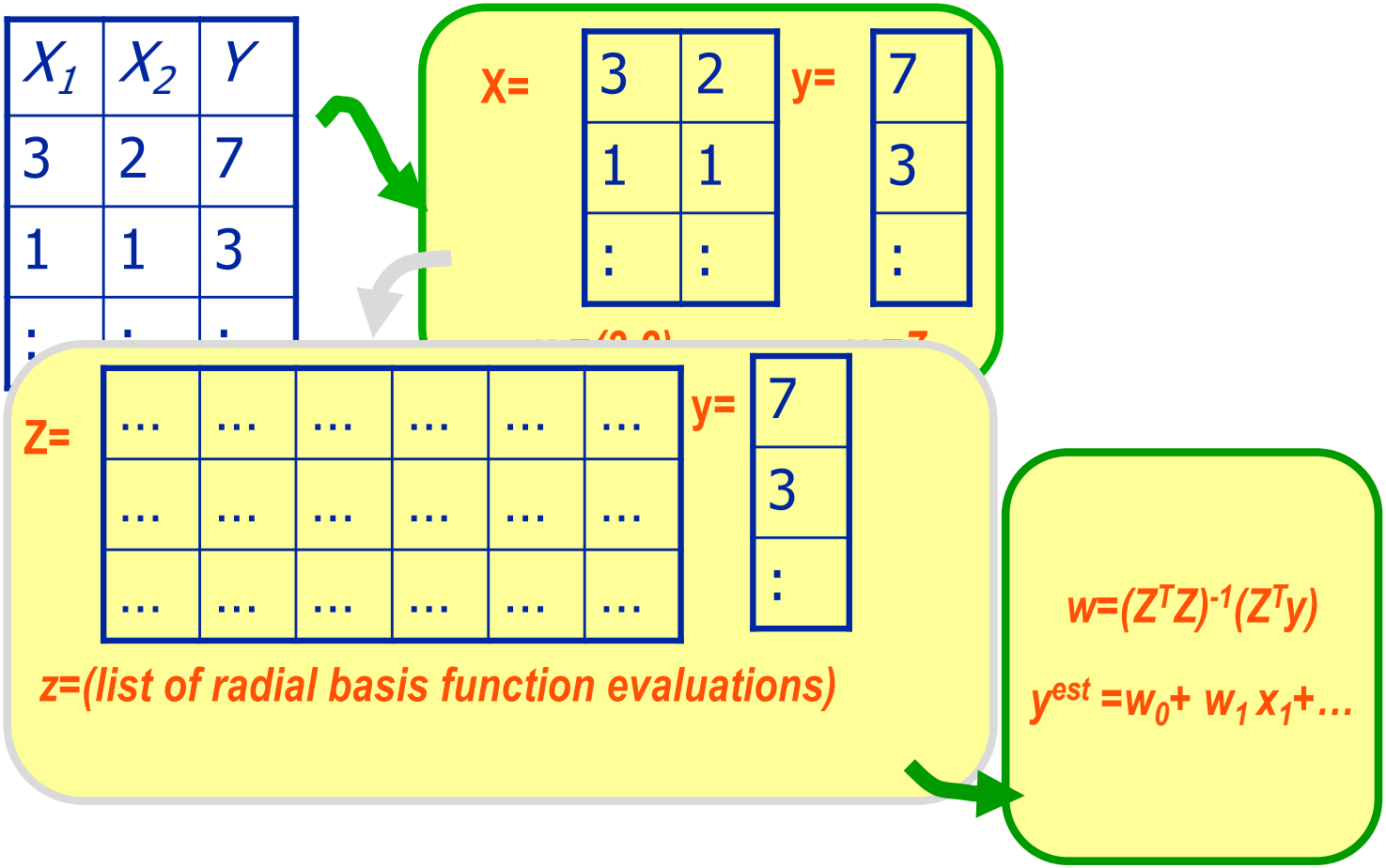
Radial Basis Functions



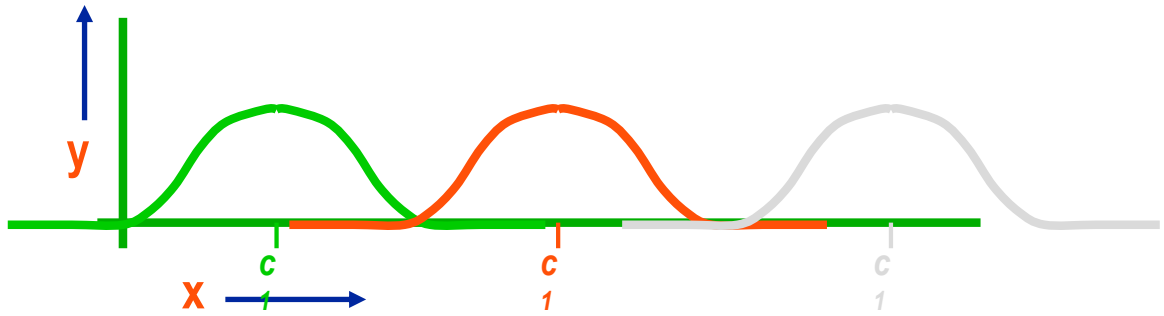
http://www.it.uu.se/research/project/rbf/rbf.png

Originally by Andrew Moore;
now heavily edited by Lyle Ungar

Radial Basis Functions (RBFs)



1-d RBFs



$y^{est} = w_1 \phi_1(x) + w_2 \phi_2(x) + w_3 \phi_3(x)$

where

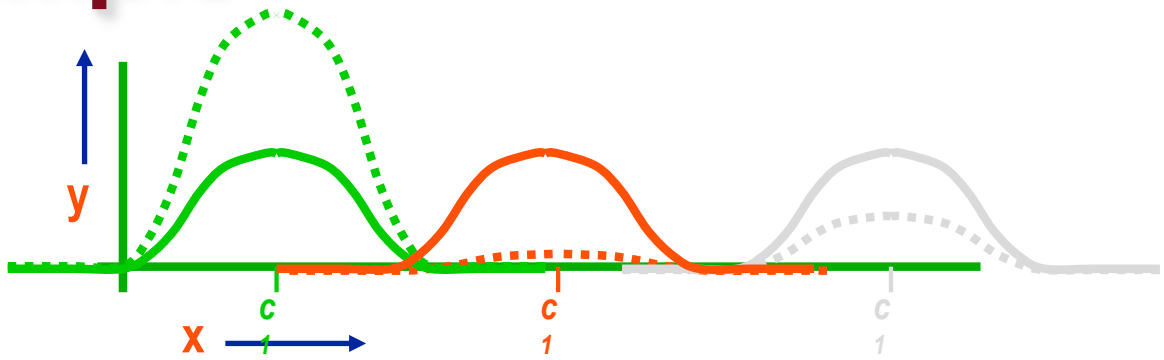
$\phi_i(x) = KernelFunction(||x - \mu_j|| / C)$

For RBF:

$KernelFunction(|| x - \mu_j || / C) = exp\{-||x - \mu_j||_2^2 / C\}$

C = "Kernel Width"

Example



$y^{est} = 2\phi_1(x) + 0.05\phi_2(x) + 0.5\phi_3(x)$

where

$\phi_j(x) = KernelFunction(|x - \mu_j| / C)$

RBFs can do …

- **Use k < p basis vectors**
  - Dimensionality reduction
  - Good for high dimensional feature spaces
- **Use k > p basis vectors**
  - Increases the dimensionality
  - Can make a formerly nonlinear problem linear
- **Use k=n basis vectors**
  - We will use this to switch to a *dual* representation

How to find the kernel centers?

- **Pick random points**
  - Generally a bad idea
- **Standard RBF: do k-means clustering and use the centers of the clusters**
  - Works great!
- **Use all n of the training data points as kernel centers**
  - Requires regularization
- **Estimate them: nonlinear regression**
  - A good initialization helps

Link functions

- **Link function** $f(x)$: $h_\theta(\boldsymbol{x}) = f(\boldsymbol{w}^T\boldsymbol{x})$
  - $f(x) = e^x$
  - $f(x) = log(x)$
- **Equivalent to** $f^{-1}(h_\theta(\boldsymbol{x})) = \boldsymbol{w}^T\boldsymbol{x}$

What you should know

- Loss functions depend on the problem
- Basis functions allow one to fit a nonlinear function using linear regression
- Link functions give a nonlinear regression