

ECE 5424: Introduction to Machine Learning

Topics:

- Regression

Readings: Barber 17.1, 17.2

Stefan Lee
Virginia Tech

Administrivia

- HW1
 - 39 Submissions
 - 58 Students Enrolled
 - 19 MIA (?????)
- Project Proposal
 - Due: **09/21, 11:55 pm** (LESS THAN A WEEK!)
 - <= 2pages, NIPS format
- HW2
 - Out today
 - Due on **Wednesday 09/28, 11:55pm**
 - Please please please please please start early
 - Implement Linear Regression, Naïve Bayes, Logistic Regression

Recap of last time

Learning a Gaussian

- Collect a bunch of data
 - Hopefully, i.i.d. samples
 - e.g., exam scores
- Learn parameters
 - Mean
 - Variance

$$P(x | \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

MLE for Gaussian

- Prob. of i.i.d. samples $D=\{x_1, \dots, x_N\}$:

$$P(\mathcal{D} \mid \mu, \sigma) = \left(\frac{1}{\sigma \sqrt{2\pi}} \right)^N \prod_{i=1}^N e^{\frac{-(x_i - \mu)^2}{2\sigma^2}}$$

- Log-likelihood of data:

$$\begin{aligned}\ln P(\mathcal{D} \mid \mu, \sigma) &= \ln \left[\left(\frac{1}{\sigma \sqrt{2\pi}} \right)^N \prod_{i=1}^N e^{\frac{-(x_i - \mu)^2}{2\sigma^2}} \right] \\ &= -N \ln \sigma \sqrt{2\pi} - \sum_{i=1}^N \frac{(x_i - \mu)^2}{2\sigma^2}\end{aligned}$$

Your second learning algorithm: MLE for mean of a Gaussian

- What's MLE for mean?

$$\frac{d}{d\mu} \ln P(\mathcal{D} \mid \mu, \sigma) = \frac{d}{d\mu} \left[-N \ln \sigma \sqrt{2\pi} - \sum_{i=1}^N \frac{(x_i - \mu)^2}{2\sigma^2} \right]$$

Learning Gaussian parameters

- MLE:

$$\hat{\mu}_{MLE} = \frac{1}{N} \sum_{i=1}^N x_i$$

$$\hat{\sigma}_{MLE}^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \hat{\mu})^2$$

Bayesian learning of Gaussian parameters

- Conjugate priors
 - Mean: Gaussian prior
 - Variance: Inverse Gamma or Wishart Distribution
- Prior for mean:

$$P(\mu \mid \eta, \lambda) = \frac{1}{\lambda \sqrt{2\pi}} e^{\frac{-(\mu-\eta)^2}{2\lambda^2}}$$

MAP for mean of Gaussian

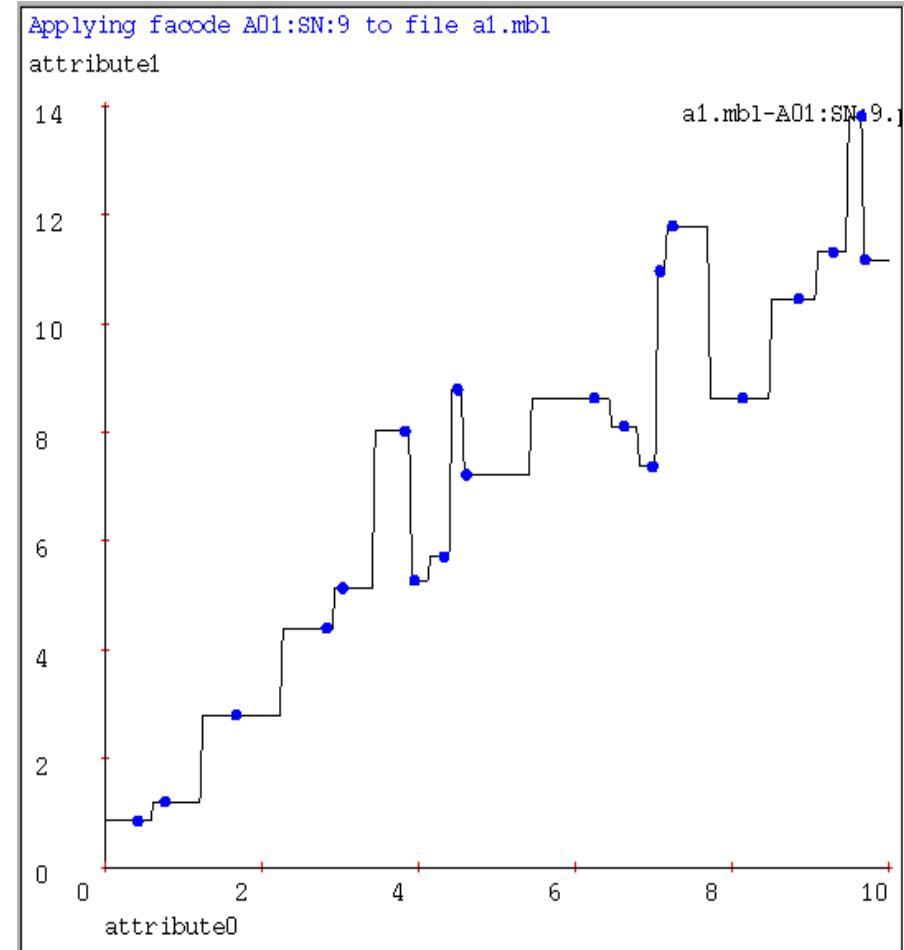
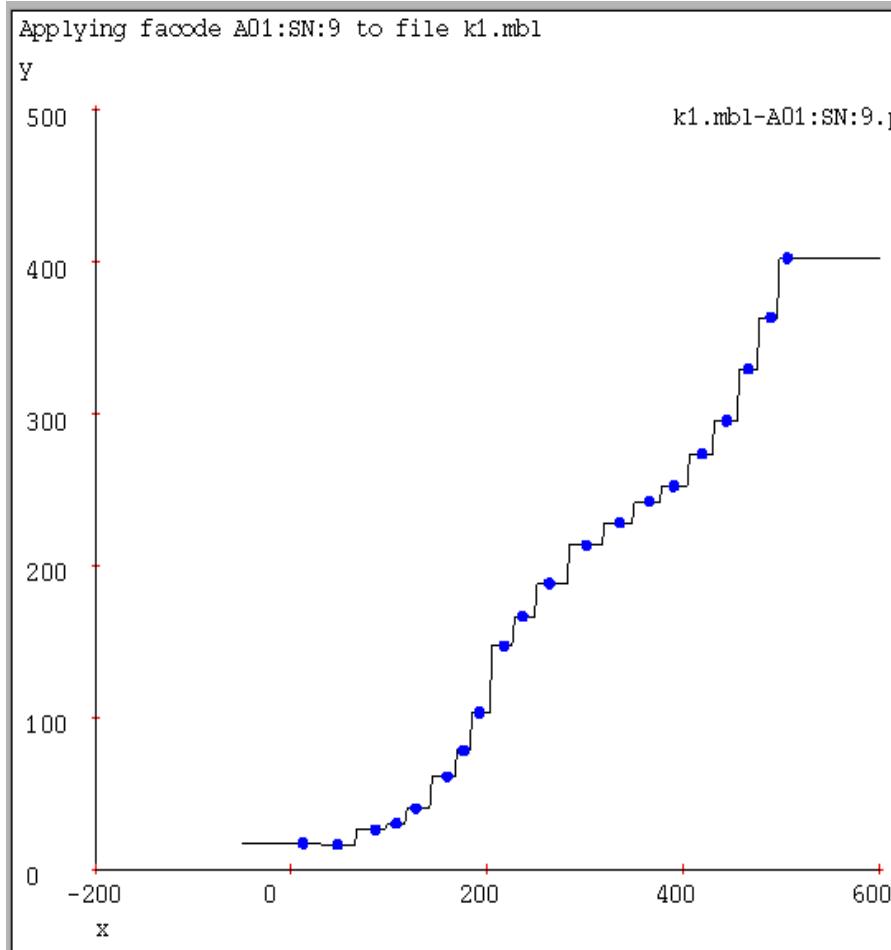
$$P(\mu \mid \eta, \lambda) = \frac{1}{\lambda \sqrt{2\pi}} e^{\frac{-(\mu-\eta)^2}{2\lambda^2}}$$
$$P(\mathcal{D} \mid \mu, \sigma) = \left(\frac{1}{\sigma \sqrt{2\pi}} \right)^N \prod_{i=1}^N e^{\frac{-(x_i - \mu)^2}{2\sigma^2}}$$

$$\frac{d}{d\mu} [\ln P(\mathcal{D} \mid \mu) P(\mu)] = \frac{d}{d\mu} [\ln P(\mathcal{D} \mid \mu) + \ln P(\mu)]$$

New Topic: Regression

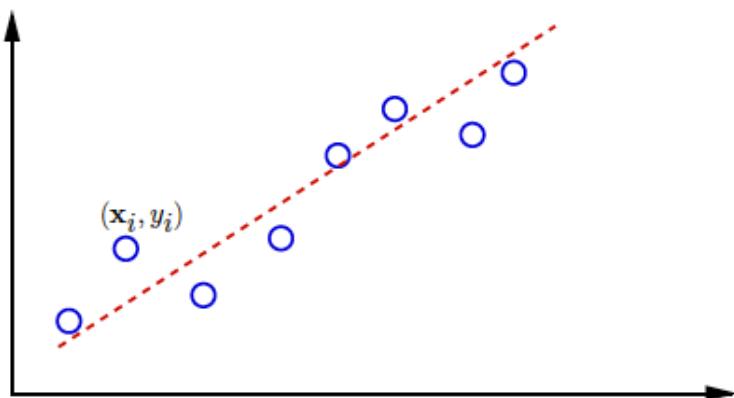
1-NN for Regression

- Often bumpy (overfits)



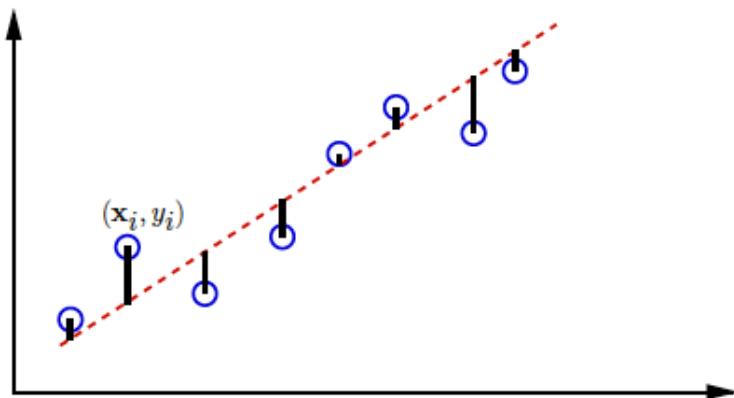
Linear fitting to data

- We want to fit a linear function to an observed set of points $X = [\mathbf{x}_1, \dots, \mathbf{x}_N]$ with associated labels $Y = [y_1, \dots, y_N]$.
 - Once we fit the function, we want to use it to *predict* the y for new \mathbf{x} .



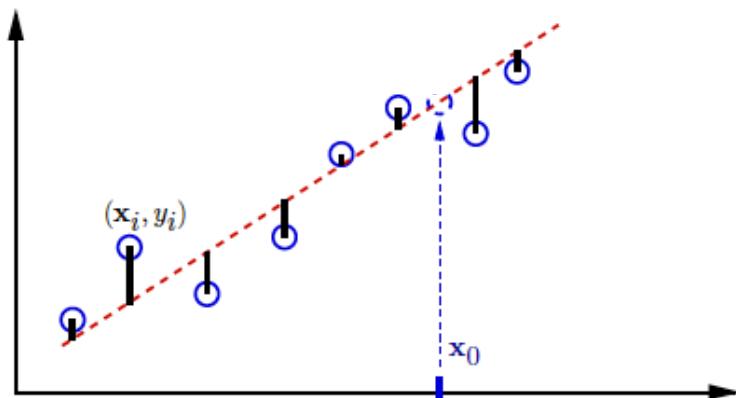
Linear fitting to data

- We want to fit a linear function to an observed set of points $X = [\mathbf{x}_1, \dots, \mathbf{x}_N]$ with associated labels $Y = [y_1, \dots, y_N]$.
 - Once we fit the function, we want to use it to *predict* the y for new \mathbf{x} .
- Least squares (LSQ) fitting criterion: find the function that minimizes sum (or average) of square distances between actual ys in the training set and predicted ones.



Linear fitting to data

- We want to fit a linear function to an observed set of points $X = [\mathbf{x}_1, \dots, \mathbf{x}_N]$ with associated labels $Y = [y_1, \dots, y_N]$.
 - Once we fit the function, we want to use it to *predict* the y for new \mathbf{x} .
- Least squares (LSQ) fitting criterion: find the function that minimizes sum (or average) of square distances between actual ys in the training set and predicted ones.



The fitted line is used as a predictor

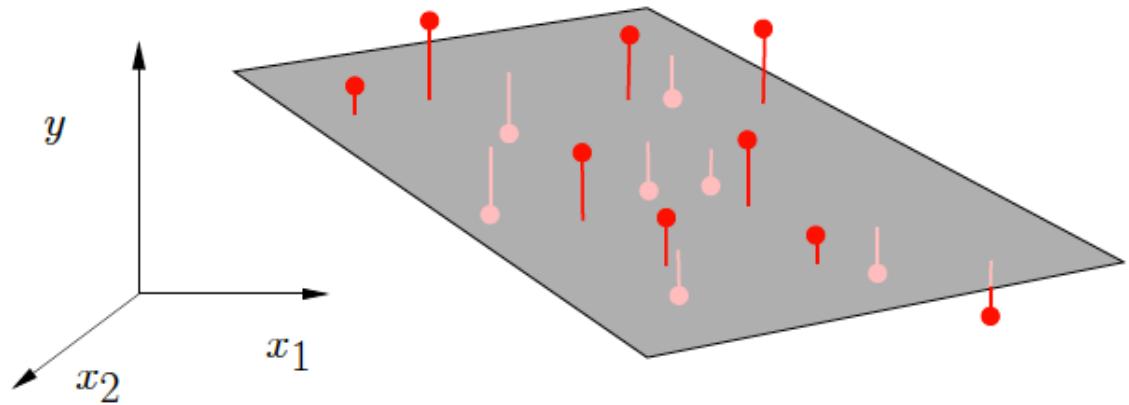
Linear Regression

- Demo
 - <http://hspm.sph.sc.edu/courses/J716/demos/LeastSquares/LeastSquaresDemo.html>

Linear functions

- General form: $f(\mathbf{x}; \mathbf{w}) = w_0 + w_1x_1 + \dots + w_dx_d$
- 1D case ($\mathcal{X} = \mathbb{R}$): a line

- $\mathcal{X} = \mathbb{R}^2$: a plane



- *Hyperplane* in general, d -D case.

Plan for Today

- Regression
 - Linear Regression Recap
 - Some matrix calculus review
 - Connections with Gaussians
 - The outlier problem ☹
 - Robust Least Squares

Least squares: estimation

- We need to minimize w.r.t. \mathbf{w}

$$\begin{aligned} L(\mathbf{w}, \mathbf{X}) &= L(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^N (y_i - \mathbf{w}^T \mathbf{x}_i)^2 \\ &= \frac{1}{N} \sum_{i=1}^N (y_i - w_0 - w_1 x_{i1} - \dots - w_d x_{id})^2 \end{aligned}$$

- Necessary condition to minimize L : derivatives w.r.t. w_0, w_1, \dots, w_d must be zero.

Least squares in matrix form

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1d} \\ \vdots & \vdots & & \vdots \\ 1 & x_{N1} & \cdots & x_{Nd} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} w_0 \\ \vdots \\ w_d \end{bmatrix}.$$

- Predictions: $\hat{\mathbf{y}} = \mathbf{X}\mathbf{w}$, errors: $\mathbf{y} - \mathbf{X}\mathbf{w}$, empirical loss:

$$L(\mathbf{w}, \mathbf{X}) = \frac{1}{N} (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w})$$

Derivative of loss

$$L(\mathbf{w}) = \frac{1}{N} (\mathbf{y}^T - \mathbf{w}^T \mathbf{X}^T) (\mathbf{y} - \mathbf{X}\mathbf{w}).$$

$$\frac{\partial \mathbf{a}^T \mathbf{b}}{\partial \mathbf{a}} = \frac{\partial \mathbf{b}^T \mathbf{a}}{\partial \mathbf{a}} = \mathbf{b}, \quad \frac{\partial \mathbf{a}^T \mathbf{B} \mathbf{a}}{\partial \mathbf{a}} = 2\mathbf{B}\mathbf{a}$$

Derivative of loss

$$L(\mathbf{w}) = \frac{1}{N} (\mathbf{y}^T - \mathbf{w}^T \mathbf{X}^T) (\mathbf{y} - \mathbf{X}\mathbf{w}).$$

$$\frac{\partial \mathbf{a}^T \mathbf{b}}{\partial \mathbf{a}} = \frac{\partial \mathbf{b}^T \mathbf{a}}{\partial \mathbf{a}} = \mathbf{b}, \quad \frac{\partial \mathbf{a}^T \mathbf{B} \mathbf{a}}{\partial \mathbf{a}} = 2\mathbf{B}\mathbf{a}$$

$$\begin{aligned} \frac{\partial L(\mathbf{w})}{\partial \mathbf{w}} &= \frac{1}{N} \frac{\partial}{\partial \mathbf{w}} [\mathbf{y}^T \mathbf{y} - \mathbf{w}^T \mathbf{X}^T \mathbf{y} - \mathbf{y}^T \mathbf{X} \mathbf{w} + \mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w}] \\ &= \frac{1}{N} [\mathbf{0} - \mathbf{X}^T \mathbf{y} - (\mathbf{y}^T \mathbf{X})^T + 2\mathbf{X}^T \mathbf{X} \mathbf{w}] \\ &= -\frac{2}{N} (\mathbf{X}^T \mathbf{y} - \mathbf{X}^T \mathbf{X} \mathbf{w}) \end{aligned}$$

Least squares solution

$$\frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}) = -\frac{2}{N} (\mathbf{X}^T \mathbf{y} - \mathbf{X}^T \mathbf{X} \mathbf{w}) = 0$$

Least squares solution

$$\frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}) = -\frac{2}{N} (\mathbf{X}^T \mathbf{y} - \mathbf{X}^T \mathbf{X} \mathbf{w}) = 0$$

$$\mathbf{X}^T \mathbf{y} = \mathbf{X}^T \mathbf{X} \mathbf{w} \Rightarrow \mathbf{w}^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

- $\mathbf{X}^\dagger \triangleq (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$ is called the *Moore-Penrose pseudoinverse* of \mathbf{X} .

- Linear regression in Matlab:

```
% X(i,:) is i-th example, y(i) is i-th label  
wLSQ = pinv([ones(size(X,1),1) X])*y;
```

- Prediction:

$$\hat{y} = \mathbf{w}^{*T} \begin{bmatrix} 1 \\ \mathbf{x}_0 \end{bmatrix} = \mathbf{y}^T \mathbf{X}^{\dagger T} \begin{bmatrix} 1 \\ \mathbf{x}_0 \end{bmatrix}$$

But, why?

- Why sum squared error???
- Gaussians, Watson, Gaussians...

Gaussian noise model

$$y = f(\mathbf{x}; \mathbf{w}) + \nu, \quad \nu \sim \mathcal{N}(\nu; 0, \sigma^2)$$

- Given the input \mathbf{x} , the label y is a random variable

$$p(y|\mathbf{x}; \mathbf{w}, \sigma) = \mathcal{N}(y; f(\mathbf{x}; \mathbf{w}), \sigma^2)$$

that is,

$$p(y|\mathbf{x}; \mathbf{w}, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y - f(\mathbf{x}; \mathbf{w}))^2}{2\sigma^2}\right)$$

- This is an explicit model of y that allows us, for instance, to *sample* y for a given \mathbf{x} .

MLE Under Gaussian Model

- On board

Is OLS Robust?

- Demo
 - <http://www.calpoly.edu/~srein/StatDemo/All.html>
- Bad things happen when the data does not come from your model!
- How do we fix this?

Robust Linear Regression

- $y \sim \text{Lap}(w'x, b)$

- On board

