



# Generalized Linear Models, Kernel Methods

Junxian He  
Feb 24, 2026

# Announcement

HW1 is out, due on March 3rd, please start early

# Exponential Family

**Rough Idea** “If  $P$  has a a special form, then inference and learning come for free”

$$P(y; \eta) = b(y) \exp \left\{ \eta^T T(y) - a(\eta) \right\}.$$

$\eta$ : natural parameter or canonical parameter

Here  $y$ ,  $a(\eta)$ , and  $b(y)$  are scalars.  $T(y)$  same dimension as  $\eta$ .

$T(y)$  is called the **sufficient statistic**. holds all information the data provides with regard to the unknown parameter values

$b(y)$  is called the **base measure** – does *not* depend on  $\eta$ .

$a(\eta)$  is called the **log partition function** – does *not* depend on  $y$ .

$$1 = \sum_y P(y; \eta) = e^{-a(\eta)} \sum_y b(y) \exp \left\{ \eta^T T(y) \right\}$$

$$\implies a(\eta) = \log \sum_y b(y) \exp \left\{ \eta^T T(y) \right\}$$

## Example: Bernoulli

Bernoulli random variable is an event (say flipping a coin) then:

$$p(y; \phi) = \phi^y (1 - \phi)^{1-y}$$

How do we put it in the required form?

$$P(y; \eta) = b(y) \exp \left\{ \eta^T T(y) - a(\eta) \right\}.$$

$$\begin{aligned} p(y; \phi) &= \phi^y (1 - \phi)^{1-y} \\ &= \exp(y \log \phi + (1 - y) \log(1 - \phi)) \\ &= \exp \left( \left( \log \left( \frac{\phi}{1 - \phi} \right) \right) y + \log(1 - \phi) \right) \end{aligned}$$

## Example: Bernoulli

$$P(y; \eta) = b(y) \exp \left\{ \eta^T T(y) - a(\eta) \right\}$$
$$\begin{aligned} p(y; \phi) &= \phi^y (1 - \phi)^{1-y} \\ &= \exp(y \log \phi + (1 - y) \log(1 - \phi)) \\ &= \exp \left( \left( \log \left( \frac{\phi}{1 - \phi} \right) \right) y + \log(1 - \phi) \right) \end{aligned}$$

So then:

$$\eta = \log \frac{\phi}{1 - \phi}, T(y) = y, a(\eta) = -\log(1 - \phi).$$

$$b(y) = 1$$

We need to show  $a(\eta)$  is a function of  $\log \frac{\phi}{1 - \phi}$

# Example: Bernoulli

We first observe that:

$$\begin{aligned}\eta = \log \frac{\phi}{1 - \phi} &\implies e^\eta(1 - \phi) = \phi \\ e^\eta = (e^\eta + 1)\phi &\implies \phi = \frac{1}{1 + e^{-\eta}}\end{aligned}$$

Now, we plug into  $\log(1 - \phi)$  and we verify:

$$a(\eta) = \log(1 - \phi) = \log \frac{e^{-\eta}}{1 + e^{-\eta}} = -\log(1 + e^\eta).$$

We have verified Bernoulli distribution is in the exponential family

# Example: Gaussian with Fixed Variance $\sigma^2 = 1$

$$P(y; \mu) = \frac{1}{\sqrt{2\pi}} \exp \left\{ -\frac{1}{2}(y - \mu)^2 \right\}.$$

Can we put it in the exponential family form?

$$P(y; \eta) = b(y) \exp \left\{ \eta^T T(y) - a(\eta) \right\}.$$

Multiply out the square and group terms:

In all the exponential family distribution we work with in the course,  $T(y) = y$

$$P(y; \mu) = \frac{1}{\sqrt{2\pi}} \exp \left\{ -y^2/2 \right\} \exp \left\{ \mu y - \frac{1}{2}\mu^2 \right\}.$$

$$\eta = \mu, T(y) = y, a(\eta) = \frac{1}{2}\eta^2.$$

# An Observation

Notice that for a Gaussian with mean  $\mu$  we had

$$\eta = \mu, T(y) = y, a(\eta) = \frac{1}{2}\eta^2.$$

$$\partial_\eta a(\eta) = \eta = \mu = \mathbb{E}[y] \text{ and } \partial_\eta^2 a(\eta) = 1 = \sigma^2 = \text{var}(y)$$

Is this true for general?

# Log Partition Function

Yes! Recall that

$$a(\eta) = \log \sum_y b(y) \exp \left\{ \eta^T T(y) \right\}$$

Then, taking derivatives

$$\nabla_\eta a(\eta) = \frac{\sum_y T(y) b(y) \exp \left\{ \eta^T T(y) \right\}}{\sum_y b(y) \exp \left\{ \eta^T T(y) \right\}} = \mathbb{E}[T(y); \eta]$$

# Many Other Exponential Models

- ▶ There are many canonical exponential family models:
  - ▶ Binary  $\mapsto$  Bernoulli
  - ▶ Multiple Classes  $\mapsto$  Multinomial
  - ▶ Real  $\mapsto$  Gaussian
  - ▶ Counts  $\mapsto$  Poisson
  - ▶  $\mathbb{R}_+$   $\mapsto$  Gamma, Exponential
  - ▶ Distributions  $\mapsto$  Dirichlet

# Recap

- Linear Regression  $h_{\theta}(x) = \theta^T x$   $\theta_j := \theta_j + \alpha \sum_{i=1}^n (y^{(i)} - h_{\theta}(x^{(i)})) x_j^{(i)}$
- Logistic Regression  $h_{\theta}(x) = g(\theta^T x)$   $\theta_j := \theta_j + \alpha \sum_{i=1}^n (y^{(i)} - h_{\theta}(x^{(i)})) x_j^{(i)}$
- Multi-class Classification Regression  $h_{\theta}(x) = softmax(\theta_1^T x, \dots, \theta_k^T x)$   
 $\theta_k := \theta_k + \alpha \sum_{i=1}^n (1\{y^{(i)} = k\} - h_{\theta}(x)_k) x^{(i)}$

Is this coincidence?

# Generalized Linear Models

We're given features  $x \in \mathbb{R}^{d+1}$  and a target  $y$ . We want a model.  
We first we pick a distribution based on  $y$ 's type.

- ▶ We assume  $y | x; \theta$  distributed as an exponential family.
  - ▶ Binary  $\mapsto$  Bernoulli
  - ▶ Multiple Classes  $\mapsto$  Multinomial
  - ▶ Real  $\mapsto$  Gaussian
  - ▶ Counts  $\mapsto$  Poisson
  - ▶  $\mathbb{R}_+$   $\mapsto$  Gamma, Exponential
  - ▶ Distributions  $\mapsto$  Dirichlet

# Generalized Linear Models

We're given features  $x \in \mathbb{R}^{d+1}$  and a target  $y$ . We want a model.  
We first we pick a distribution based on  $y$ 's type.

- ▶ We assume  $y | x; \theta$  distributed as an exponential family.
  - ▶ Binary  $\mapsto$  Bernoulli
  - ▶ Multiple Classes  $\mapsto$  Multinomial
  - ▶ Real  $\mapsto$  Gaussian
  - ▶ Counts  $\mapsto$  Poisson
  - ▶  $\mathbb{R}_+ \mapsto$  Gamma, Exponential
  - ▶ Distributions  $\mapsto$  Dirichlet
- ▶ Our model is *linear* beacuse we make the natural parameter  $\eta = \theta^T x$  in which  $\theta, x \in \mathbb{R}^{d+1}$ .

# Generalized Linear Models

**inference**

$h_\theta(x) = \mathbb{E}[y | x; \theta]$  is the **output**.

**learn**

$\max_\theta \log p(y | x; \theta)$  by maximum likelihood.

$$P(y; \eta) = b(y) \exp \left\{ \eta^T T(y) - a(\eta) \right\}.$$

$$a(\eta) = \log \sum_y b(y) \exp \left\{ \eta^T T(y) \right\}$$

Then, taking derivatives

$$\nabla_\eta a(\eta) = \frac{\sum_y T(y) b(y) \exp \left\{ \eta^T T(y) \right\}}{\sum_y b(y) \exp \left\{ \eta^T T(y) \right\}} = \mathbb{E}[T(y); \eta]$$

$T(y) = y$  for most of the examples you will see in this course

# Generalized Linear Models

**inference**  $h_{\theta}(x) = \mathbb{E}[y \mid x; \theta]$  is the **output**.

**learn**  $\max_{\theta} \log p(y \mid x; \theta)$  by maximum likelihood.

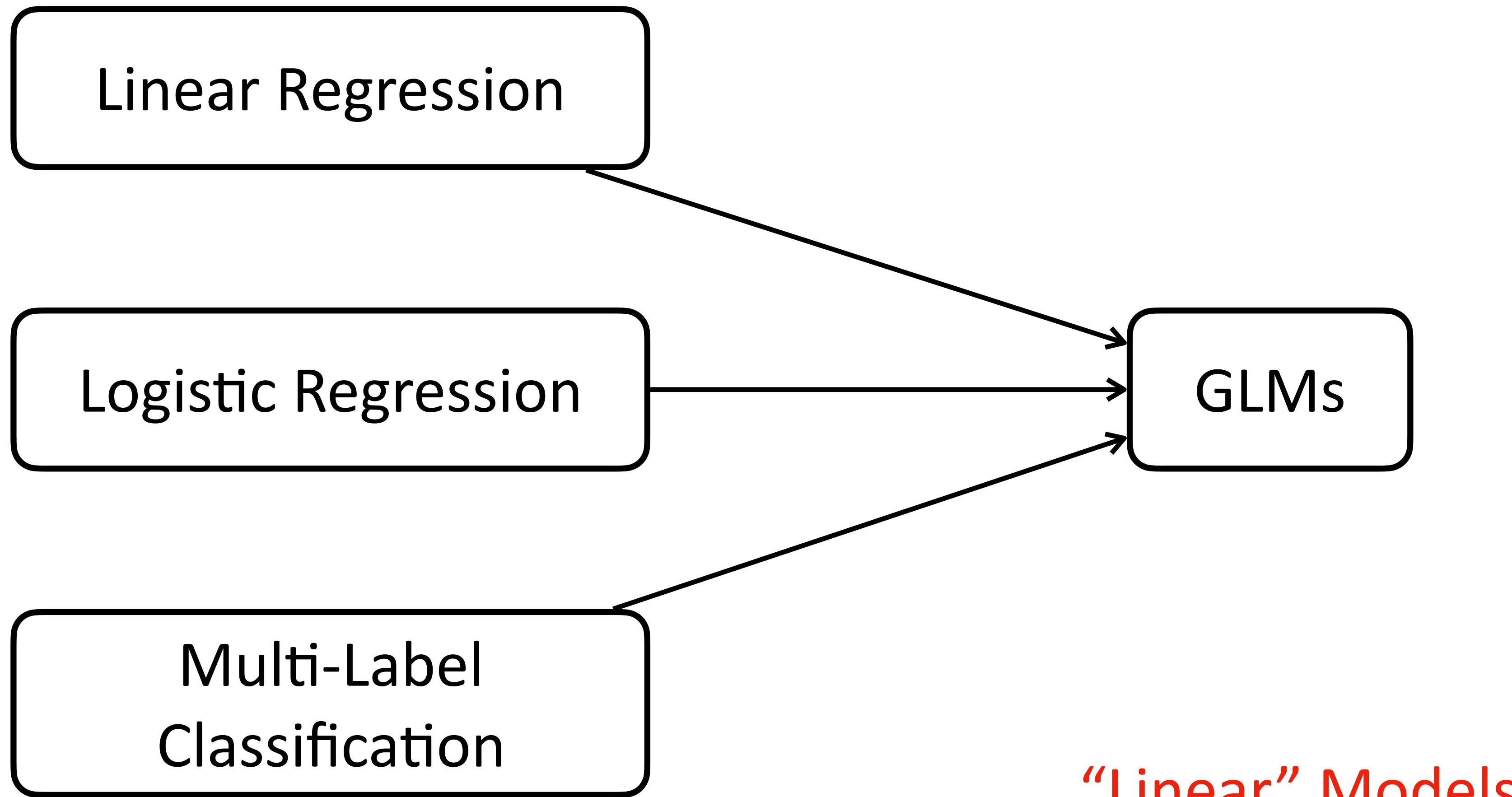
**algorithm: SGD**  $\theta^{(t+1)} = \theta^{(t)} + \alpha \left( y^{(i)} - h_{\theta^{(t)}}(x^{(i)}) \right) x^{(i)}$

# Constructing GLMs

- Pick an exponential family distribution given the type of  $y$  (Poisson, Multinomial, Gaussian...)
- $\eta = \theta^T x$ , or  $\eta_i = \theta_i^T x$
- Training with maximum likelihood estimation
- Inference:  $h(x) = E[y | x]$

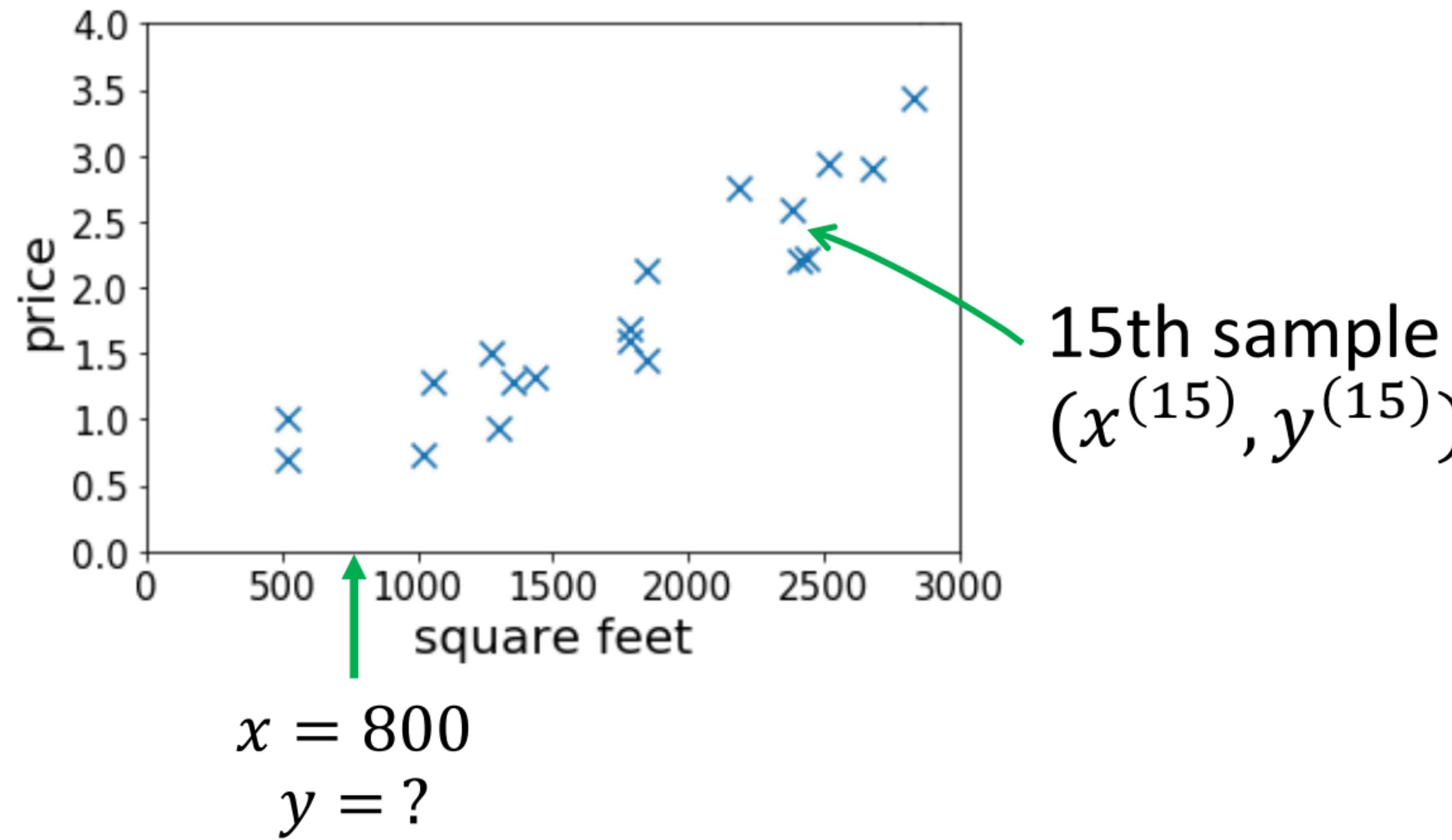
Enjoy closed-form solution for various statistics

# Generalized Linear Models



# Kernel Methods

# Feature Map



Feature map  
 $\phi : R^d \rightarrow R^p$

$$y = \theta x$$
$$y = \theta_3 x^3 + \theta_2 x^2 + \theta_1 x + \theta_0$$

$$\phi(x) = \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \end{bmatrix} \in \mathbb{R}^4.$$

$$y = \theta^T \phi(x)$$

# LMS Update Rule with Features

Linear Regression:

$$\begin{aligned}\theta &:= \theta + \alpha \sum_{i=1}^n (y^{(i)} - h_\theta(x^{(i)})) x^{(i)} \\ &:= \theta + \alpha \sum_{i=1}^n (y^{(i)} - \theta^T x^{(i)}) x^{(i)}.\end{aligned}$$

With Features:

$$\theta := \theta + \alpha \sum_{i=1}^n (y^{(i)} - \theta^T \phi(x^{(i)})) \phi(x^{(i)})$$

How about Generalized Linear Models with Features?

# New Feature Vector Can Be Very High-Dimensional

$$\phi(x) = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_1^2 \\ x_1 x_2 \\ x_1 x_3 \\ \vdots \\ x_2 x_1 \\ \vdots \\ x_1^3 \\ x_1^2 x_2 \\ \vdots \end{bmatrix}$$

Computationally expensive

Is the computation evitable given  $\theta \in R^p$ ?

# Kernel Trick

- If  $\theta$  is initialized as 0, then at any step of the gradient descent:

$$\theta = \sum_{i=1}^n \beta_i \phi(x^{(i)}) \quad \beta_i \in R$$

$$\theta := \theta + \alpha \sum_{i=1}^n (y^{(i)} - \theta^T \phi(x^{(i)})) \phi(x^{(i)})$$

$$= \sum_{i=1}^n \beta_i \phi(x^{(i)}) + \alpha \sum_{i=1}^n (y^{(i)} - \theta^T \phi(x^{(i)})) \phi(x^{(i)})$$

$$= \sum_{i=1}^n \underbrace{(\beta_i + \alpha (y^{(i)} - \theta^T \phi(x^{(i)})))}_{\text{new } \beta_i} \phi(x^{(i)})$$

$$\beta_i := \beta_i + \alpha (y^{(i)} - \theta^T \phi(x^{(i)}))$$

$$\beta_i := \beta_i + \alpha \left( y^{(i)} - \sum_{j=1}^n \beta_j \phi(x^{(j)})^T \phi(x^{(i)}) \right)$$

# Kernel Trick

$$\beta_i := \beta_i + \alpha \left( y^{(i)} - \sum_{j=1}^n \beta_j \phi(x^{(j)})^T \phi(x^{(i)}) \right)$$

Rewrite  $\phi(x^{(j)})^T \phi(x^{(i)}) = \langle \phi(x^{(j)}), \phi(x^{(i)}) \rangle$

We can precompute all pairwise  $\langle \phi(x^{(j)}), \phi(x^{(i)}) \rangle$  beforehand, and reuse it for every gradient descent update

# Kernel Trick

$$\beta_i := \beta_i + \alpha \left( y^{(i)} - \sum_{j=1}^n \beta_j \phi(x^{(j)})^T \phi(x^{(i)}) \right)$$

Kernel  $K(x, z)$   $\mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$   $\mathcal{X}$  is the space of the input

$$K(x, z) \triangleq \langle \phi(x), \phi(z) \rangle$$

# The Algorithm

- Compute  $K(\phi(x^{(i)}), \phi(x^{(j)})) = \langle \phi(x^{(i)}), \phi(x^{(j)}) \rangle$  for all  $i, j$
- Loop  $\beta_i := \beta_i + \alpha \left( y^{(i)} - \sum_{j=1}^n \beta_j K(x^{(i)}, x^{(j)}) \right) \quad \forall i \in \{1, \dots, n\}$   
Recall that  $n$  is the number of data samples

Or in vector notation, letting  $K$  be the  $n \times n$  matrix with  $K_{ij} = K(x^{(i)}, x^{(j)})$ , we have

$$\beta := \beta + \alpha(\vec{y} - K\beta)$$

# Inference

We do not need to explicitly compute  $\theta$  !

$$\theta^T \phi(x) = \sum_{i=1}^n \beta_i \phi(x^{(i)})^T \phi(x) = \sum_{i=1}^n \beta_i K(x^{(i)}, x)$$

The Kernel function is all we need for training and inference!

# Implicit Feature Map

Do we still need to define feature maps?

$$K(x, z) \triangleq \langle \phi(x), \phi(z) \rangle$$

What kinds of kernel functions  $K()$  can correspond to some feature map  $\phi$

# Example

$$K(x, z) = (x^T z)^2 \quad x, z \in \mathbb{R}^d$$

What is the feature map to make K a valid kernel function?

$$\begin{aligned} K(x, z) &= \left( \sum_{i=1}^d x_i z_i \right) \left( \sum_{j=1}^d x_j z_j \right) \\ &= \sum_{i=1}^d \sum_{j=1}^d x_i x_j z_i z_j \\ &= \sum_{i,j=1}^d (x_i x_j)(z_i z_j) \end{aligned}$$

$$\phi(x) = \begin{bmatrix} x_1 x_1 \\ x_1 x_2 \\ x_1 x_3 \\ x_2 x_1 \\ x_2 x_2 \\ x_2 x_3 \\ x_3 x_1 \\ x_3 x_2 \\ x_3 x_3 \end{bmatrix}$$

Requires  $O(d^2)$  compute  
for feature mapping

Requires  $O(d)$  compute for  
Kernel function

# Next Lecture

- What kinds of functions would make a kernel function?
- Infinite dimensions of feature mapping?
- Support Vector Machines

**Thank You!**  
**Q & A**