# CSC 4020 Fundamental of Machine Learning: Linear Regression

Baoyuan Wu
School of Data Science, CUHK-SZ

January 25/27, 2021

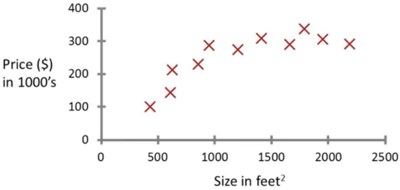
# Outline

1. Linear Regression: A Deterministic Perspective
2. Linear Regression: A Probabilistic Perspective
   - Probabilistic modeling
   - Robust linear regression
   - Ridge regression
   - Lasso regression
3. Generalized Linear Regression

# Linear regression with one variable

- Here we start from a simple example of one dimensional input variable, and the training dataset $D = \{(x_i, y_i)\}_{i=1}^m$ can be plotted on the $x - y$ plane.
- $m$ indicates the number of training samples; $x$ denotes the input variable/feature; $y$ denotes the output variable.

| Size in feet$^2$ ($x$) | Price in 1000's (y) |
|---|---|
| 2104 | 460 |
| 1416 | 232 |
| 1514 | 315 |
| 852 | 178 |
| ... | ... |

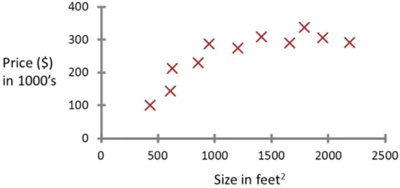


# Linear hypothesis function

- Our goal is to find a linear hypothesis function to well fit the training data $D$, *i.e.*,

$$h_\theta(x) = \theta_0 + \theta_1 \phi(x) = [\theta_0, \theta_1][1; \phi(x)] = \hat{\boldsymbol{\phi}}(x)^\top \boldsymbol{\theta} \tag{1}$$

  where $\phi(x)$ is called **basis expansion**, which is specified as different forms, such as $\phi(x) = x$ or $\phi(x) = [x^3; x^2; x]$. In the following, we will use $\phi(x) = x$ as example, while other expansions will be introduced later.
- Given $\theta_0, \theta_1$, $h_\theta(x)$ is the function of $x$.
- Given $x$, $h_\theta(x)$ is a **linear function** of $\boldsymbol{\theta} = [\theta_0; \theta_1]$. This is why it is called **linear regression**.
- Then, given $D$, how to learn $\boldsymbol{\theta}$?



# Cost function

- We design the following **cost function** to minimize the difference between the prediction $h_{\boldsymbol{\theta}}(x_i)$ and the ground-truth value $y_i$, *i.e.*,

$$J(\boldsymbol{\theta}) = \frac{1}{2}\sum_{i=1}^{m}(h_{\boldsymbol{\theta}}(x_i) - y_i)^2 \tag{2}$$

$$= \frac{1}{2}\sum_{i=1}^{m}(\theta_0 + \theta_1 x_i - y_i)^2, \tag{3}$$

$$= \frac{1}{2}\sum_{i=1}^{m}(\bar{\boldsymbol{x}}_i^\top \boldsymbol{\theta} - y_i)^2 \tag{4}$$

  which is called **residual sum of squares** (RSS) or sum of squared errors (SSE).

- $J(\boldsymbol{\theta})$ is a convex or non-convex function? What is the shape of it?

# Gradient descent

- The linear regression is formulated to the following optimization problem

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \frac{1}{2}\sum_{i=1}^{m}(\bar{\boldsymbol{x}}_i^\top \boldsymbol{\theta} - y_i)^2. \tag{5}$$

- $\boldsymbol{\theta}$ can be updated by **gradient descent algorithm**,

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}, \ \frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \sum_{i=1}^{m}(\bar{\boldsymbol{x}}_i^\top \boldsymbol{\theta} - y_i)\bar{\boldsymbol{x}}_i \tag{6}$$

  where $\alpha$ is called step-size or learning rate.
- Does gradient descent always converge to the optimal solution? (Plot the trajectory of gradient descent on curve or contours)

# Analytical solution

- If we set the gradient to 0, then we can get the following solution

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \sum_{i=1}^{m} (\bar{\boldsymbol{x}}_i^\top \boldsymbol{\theta} - y_i)\bar{\boldsymbol{x}}_i = \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{X}^\top \boldsymbol{y} = 0 \tag{7}$$

$$\Rightarrow \boldsymbol{\theta}^* = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}, \tag{8}$$

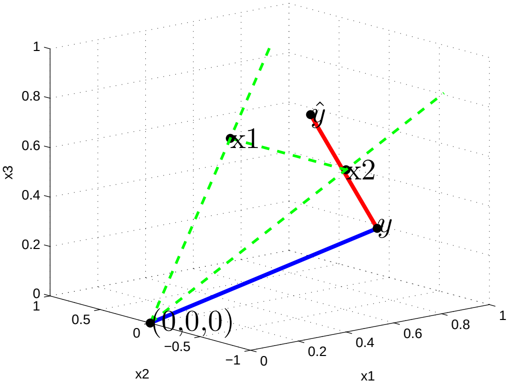
  which are called **normal equation** and **ordinary least squares** (OLS) solution, respectively. $\boldsymbol{X} = [\bar{\boldsymbol{x}}_1^\top; \bar{\boldsymbol{x}}_2^\top; \ldots; \bar{\boldsymbol{x}}_m^\top] \in \mathbb{R}^{m \times d}$.
- Since there is a closed-form solution, why do we need gradient descent algorithm?

# Geometric interpretation

- Since $\boldsymbol{\theta}^* = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{y}$, then the predictions of $\boldsymbol{X}$ can be obtained by
$$\hat{\boldsymbol{y}} = \boldsymbol{X}\boldsymbol{\theta}^* = \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{y}, \tag{9}$$
which corresponds to the **orthogonal projection** of $\boldsymbol{y}$ onto the column space of $\boldsymbol{X}$.

$$\boldsymbol{X} = \begin{pmatrix} 1 & 2 \\ 1 & -2 \\ 1 & 2 \end{pmatrix},$$

$$\mathrm{y} = \begin{pmatrix} 8.89 \\ 0.61 \\ 1.77 \end{pmatrix}$$



# Normal equation vs. gradient descent

| | Size (feet²) | Number of bedrooms | Number of floors | Age of home (years) | Price ($1000) |
|---|---|---|---|---|---|
| $x_0$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $y$ |
| 1 | 2104 | 5 | 1 | 45 | 460 |
| 1 | 1416 | 3 | 2 | 40 | 232 |
| 1 | 1534 | 3 | 2 | 30 | 315 |
| 1 | 852 | 2 | 1 | 36 | 178 |

$$X = \underbrace{\begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix}}_{m \times (n+1)} \qquad y = \underbrace{\begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}}_{m\text{-dimensional vector}}$$

$$\boxed{\theta = (X^T X)^{-1} X^T y}$$

| Gradient Descent | Normal Equation |
|---|---|
| Need to choose alpha | No need to choose alpha |
| Needs many iterations | No need to iterate |
| O $(kn^2)$ | O $(n^3)$, need to calculate inverse of $X^T X$ |
| Works well when n is large | Slow if n is very large |

# Probabilistic modeling

- We assume that the relationship between the input variable/feature $\boldsymbol{x}$ and the output variable $y$ is

$$y = \boldsymbol{\theta}^\top \boldsymbol{x} + e, \text{ where } e \sim \mathcal{N}(0, \sigma^2), \tag{10}$$

where $e$ is called **observation noise** or **residual error**, and it is independent with any specific input $\boldsymbol{x}$.

- Thus, the output $y$ can also be seen as a random variable, and its conditional probability is formulated as

$$p(y|\boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}^\top \boldsymbol{x}, \sigma^2) \tag{11}$$

# Maximum log-likelihood estimation

- The parameter $\boldsymbol{\theta}$ can be learned by maximum log-likelihood estimation (MLE), given the training dataset $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$, as follows

$$\boldsymbol{\theta}_{MLE} = \arg\max_{\boldsymbol{\theta}} \log \mathcal{L}(\boldsymbol{\theta}|D) \tag{12}$$

$$= \sum_i^m \log p(y|\boldsymbol{x}, \boldsymbol{\theta}) = \sum_i^m \log \mathcal{N}(\boldsymbol{\theta}^\top \boldsymbol{x}, \sigma^2) \tag{13}$$

$$= -\log(\sigma^m (2\pi)^{\frac{m}{2}}) - \frac{1}{2\sigma^2} \sum_i^m (y_i - \boldsymbol{\theta}^\top \boldsymbol{x}_i) \tag{14}$$

- Removing the constants w.r.t. $\boldsymbol{\theta}$,

$$\boldsymbol{\theta}_{MLE} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2} \sum_i^m (y_i - \boldsymbol{\theta}^\top \boldsymbol{x}_i)^2, \tag{15}$$

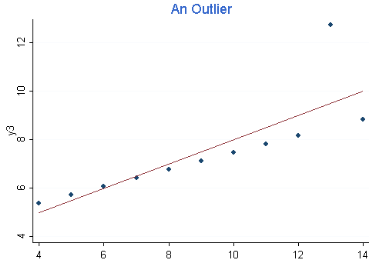
which is exactly same with the cost function from the deterministic perspective.

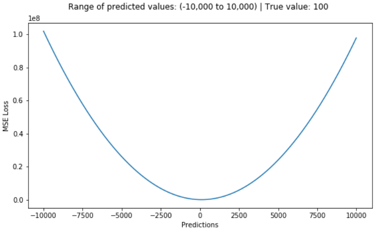
# Robust linear regression

- When there is a few outliers in the training data $D$, which are far from most other points, then learned parameters $\boldsymbol{\theta}_{MLE}$ will be significantly influenced, leading to very poor fit.
- Let's see the loss curve of the residual sum of squares (RSS),

$$J(\boldsymbol{\theta}) = \frac{1}{2}\sum_{i=1}^{m}(\bar{\boldsymbol{x}}_i^\top \boldsymbol{\theta} - y_i)^2. \tag{16}$$

- The error increases quadratically along with the residual. To minimize such a large error, the linear model will be significantly changed.
- How to alleviate the significant influence of outliers?



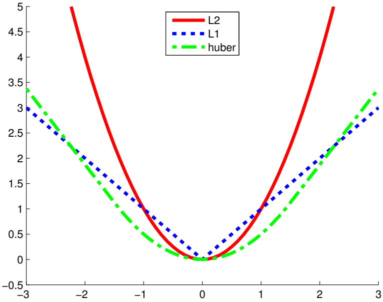



# Robust linear regression

- We adopt the $\ell_1$ loss to replace the $\ell_2$ loss, as follows

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{m} |\bar{\boldsymbol{x}}_i^\top \boldsymbol{\theta} - y_i|. \tag{17}$$

- The curves of $\ell_1$ and $\ell_2$ losses are shown ad follows.
- When the residual is large, the $\ell_1$ loss is much smaller than the $\ell_2$ loss, such that the influence of outliers could be alleviated.



# Robust linear regression

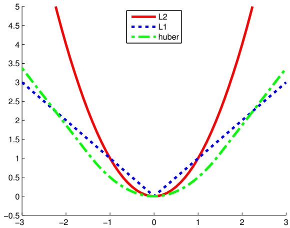
- Actually, the above $\ell_1$ loss can also be derived from the probabilistic perspective, by assuming that

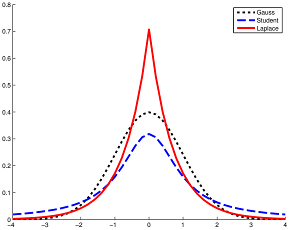
$$p(y|\boldsymbol{x}, \boldsymbol{\theta}, b) = \mathrm{Lap}(y|\boldsymbol{x}, \boldsymbol{\theta}, b) \propto \exp(-\frac{1}{b}|y - \boldsymbol{\theta}^\top \boldsymbol{x}|) \tag{18}$$

- Applying the maximum log-likelihood estimation (MLE), we will obtain

$$\boldsymbol{\theta}_{MLE} = \arg\max_{\boldsymbol{\theta}} \log \mathcal{L}(\boldsymbol{\theta}|D) = \sum_{i}^{m} \log p(y|\boldsymbol{x}, \boldsymbol{\theta}) \tag{19}$$

$$\equiv \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{m} |\bar{\boldsymbol{x}}_i^\top \boldsymbol{\theta} - y_i| \tag{20}$$



# Robust linear regression

$$\boldsymbol{\theta}_{MLE} = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{m} |\boldsymbol{x}_i^\top \boldsymbol{\theta} - y_i| \tag{21}$$

- However, the $\ell_1$ loss function is non-differentiable and non-linear. The gradient descent algorithm cannot be adopted.
- We can transform it to a linear program, as follows

$$\min_{\boldsymbol{\theta}, \boldsymbol{t}} \sum_{i}^{m} t_i \tag{22}$$

$$s.t. \ -t_i \leq \boldsymbol{x}_i^\top \boldsymbol{\theta} - y_i \leq t_i, 1 \leq i \leq m. \tag{23}$$

# Robust linear regression

$$\boldsymbol{\theta}_{MLE} = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{m} |\boldsymbol{x}_i^\top \boldsymbol{\theta} - y_i| \tag{24}$$

- We can also utilize the following equation:

$$|a| = \min_{\mu} \frac{1}{2}(\frac{a^2}{\mu} + \mu) \tag{25}$$

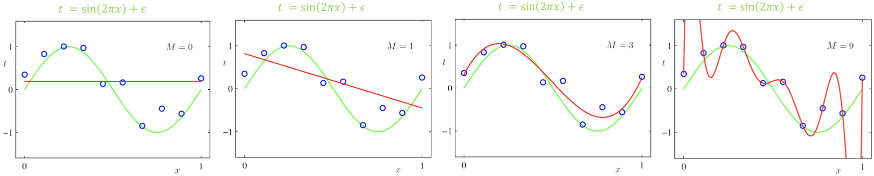
- Then, the $\ell_1$ minimization problem can be reformulated as follows

$$\min_{\boldsymbol{\theta}} \min_{\mu} \frac{1}{2}(\frac{(\boldsymbol{x}^\top \boldsymbol{\theta} - y_i)^2}{\mu} + \mu). \tag{26}$$

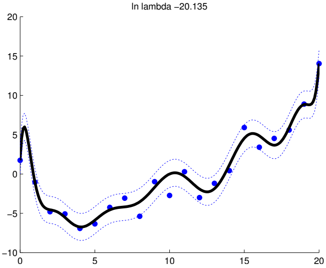
- It can be iteratively and alternatively optimized as follows:
  - Given $\boldsymbol{\theta}$, $\mu = |\boldsymbol{x}^\top \boldsymbol{\theta} - y_i|$
  - Given $\mu$, $\boldsymbol{\theta} = \min_{\boldsymbol{\theta}} (\boldsymbol{x}^\top \boldsymbol{\theta} - y_i)^2$
- It is called **iteratively reweighted least squares** method.

# Ridge regression

- As demonstrated in the first week, overfitting is an important challenge for linear regression.
- What approaches we have introduced to alleviate ovefitting? Ocam's razor or cross-validation
- Is there other more theoretical approaches? SURE!



# Ridge regression




- Let's see one simple example, we use a polynomial function with 14 degree to fit $m = 21$ data points. The learned curve is very "wiggly" (see above).
- The parameter values of this curve are as follows

$$6.56, -36.934, -109.25, 543.452, 1022.561, -3046.224, -3768.013, 8524.54,$$
$$6607.897, -12640.058, -5530.188, 9479.73, 1774, 639, -2821.526$$

- There are many large positive/negative values, such that a small change of features could lead to significant change of output.

# Ridge regression

- How to get smaller parameter values?
- We can assume that the parameter follow a zero-mean Gaussian prior

$$p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\mathbf{0}, \tau^2\mathbf{I}) \tag{27}$$

- Utilizing this prior, we obtain the maximum a posteriori (MAP) estimation

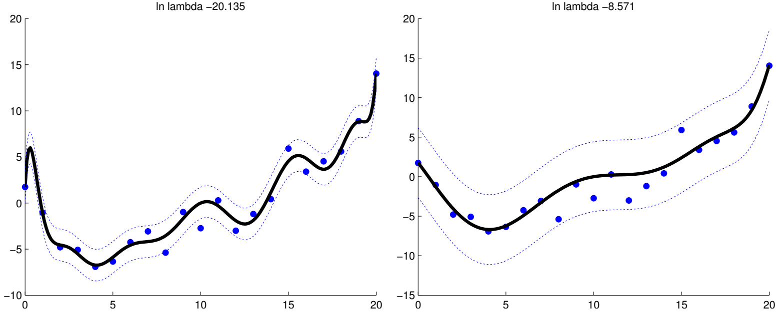
$$\begin{aligned}
\boldsymbol{\theta}_{MAP} &= \arg\max_{\boldsymbol{\theta}} \sum_i^m \log p(y|\boldsymbol{x}, \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) && (28)\\
&= \sum_i^m \log \mathcal{N}(\boldsymbol{\theta}^\top \boldsymbol{x}, \sigma^2) + \mathcal{N}(\boldsymbol{\theta}|\mathbf{0}, \tau^2\mathbf{I}) && (29)\\
&\equiv \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^m (\bar{\boldsymbol{x}}_i^\top \boldsymbol{\theta} - y_i)^2 + \lambda\|\boldsymbol{\theta}\|_2^2. && (30)
\end{aligned}$$

- The corresponding closed-form solution is given by

$$\boldsymbol{\theta}_{MAP} = (\lambda \boldsymbol{I}) + \boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top y. \tag{31}$$

# Ridge regression

- The above method is also known as **ridge regression**, or **penalized least squares**.
- In general, adding a Gaussian prior to the parameters of a model to encourage them to be small is called $\ell_2$ **regularization** or **weight decay**.
- As shown below, when we set a larger $\lambda$, *i.e.*, more weight on the prior, the resulting curve will be smoother.



# Lasso regression

- We can replace the Gaussian prior by a Laplacian prior, *i.e.*,

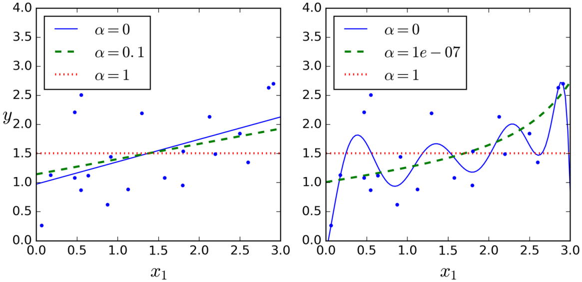
$$p(\boldsymbol{\theta}) = \text{Lap}(\boldsymbol{\theta}|\mathbf{0}, b) = \frac{1}{2b} \exp\left( - \frac{|\boldsymbol{\theta}|}{b} \right), \tag{32}$$

- The combination of the Gaussian distribution of $p(y|\boldsymbol{x}, \boldsymbol{\theta})$ and the Laplacian prior, leading to

$$\begin{aligned} \boldsymbol{\theta}_{MAP} &= \arg\max_{\boldsymbol{\theta}} \sum_{i}^{m} \log p(y|\boldsymbol{x}, \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) & (33) \\ &= \sum_{i}^{m} \log \mathcal{N}(\boldsymbol{\theta}^\top \boldsymbol{x}, \sigma^2) + \text{Lap}(\boldsymbol{\theta}|\mathbf{0}, b) & (34) \\ &\equiv \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{m} (\bar{\boldsymbol{x}}_i^\top \boldsymbol{\theta} - y_i)^2 + \lambda|\boldsymbol{\theta}|. & (35) \end{aligned}$$

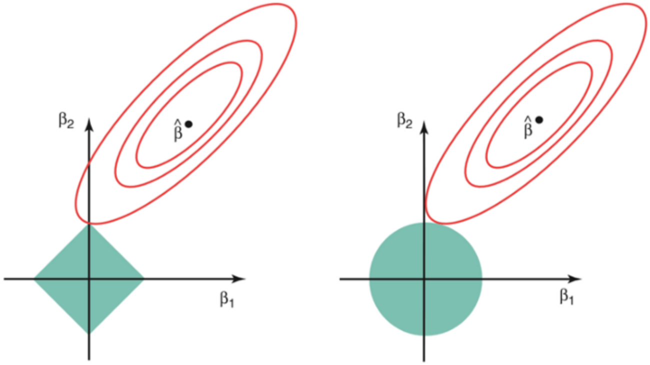
# Lasso regression

- It is **Lasso regression**, and the regularization is called $\ell_1$ **regularization**. It will encourage the sparse parameters.
- As shown below, when we set a larger $\lambda$, *i.e.*, more weight on the prior, the resulting curve will be smoother.



# Geometry of Ridge and Lasso regression

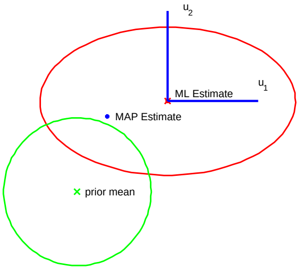
- Geometry of Ridge and Lasso regression. Which one is Ridge?



# Summary of different linear regressions

Note that the uniform distribution will not change the mode of the likelihood. Thus, MAP estimation with a uniform prior corresponds to MLE.

| $p(y\|\boldsymbol{x}, \boldsymbol{\theta})$ | $p(\boldsymbol{\theta})$ | regression method |
|---|---|---|
| Gaussian | Uniform | Least squares |
| Gaussian | Gaussian | Ridge regression |
| Gaussian | Laplace | Lasso regression |
| Laplace | Uniform | Robust regression |
| Student | Uniform | Robust regression |



# Generalized linear regression

- **Linear model**:

$$\mu(\boldsymbol{x}|\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \phi(\boldsymbol{x}), \tag{36}$$
$$y(x|\boldsymbol{\theta}) \sim f(\mu(\boldsymbol{x}|\boldsymbol{\theta})), \tag{37}$$

where $f$ denotes a distribution function.

- **Generalized linear model (GLM)**:

$$\mu(\boldsymbol{x}|\boldsymbol{\theta}) = g^{-1}(\boldsymbol{\theta}^\top \phi(\boldsymbol{x})), \tag{38}$$
$$y(x|\boldsymbol{\theta}) \sim f(\mu(\boldsymbol{x}|\boldsymbol{\theta})), \tag{39}$$

where $g$ is called **link function**, which is required to be monotonically increasing differentiable.

- The standard linear model is a special case of GLM with $g(a) = a$.

# Why we need generalized linear regression

- **Why we need generalized linear model?** Let's see one example.

  In the early stages of a disease epidemic, the rate at which new cases occur can often increase exponentially through time. Hence, if $\mu_i$ is the expected number of new cases on day $t_i$, a model of the form

  $$\mu_i = \gamma \exp(\delta t_i)$$

  seems appropriate.

  - Such a model can be turned into GLM form, by using a log link so that

    $$\log(\mu_i) = \log(\gamma) + \delta t_i = \beta_0 + \beta_1 t_i.$$

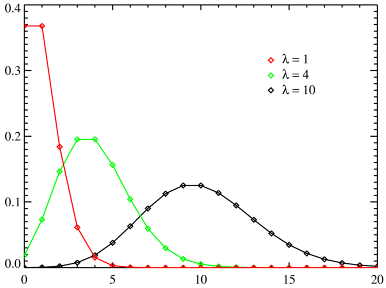
  - Since this is a count, the Poisson distribution (with expected value $\mu_i$) is probably a reasonable distribution to try.

# Log linear regression

- **Poisson distribution** The Poisson distribution is popular for modeling the number of times an event occurs in an interval of time or space.
- A discrete random variable $X$ is said to have a Poisson distribution with parameter $\lambda > 0$ if for $k = 0, 1, 2, \ldots$, the probability mass function of $X$ is given by

$$f(k;\lambda) = P(X = k|\lambda) = \frac{\lambda^k e^{-\lambda}}{k!}, \tag{40}$$

where $e$ is Euler's number ($e = 2.71828...$), we $k$ is the number of occurrences, $k!$ is the factorial of k.



# Log linear regression

- We assume that the conditional probability follows

$$P(y_i|\boldsymbol{x}_i, \boldsymbol{\theta}) = Poisson(\lambda_i) = \frac{\lambda^{y_i} e^{-\lambda}}{y_i!}, \quad \ln \lambda_i = \boldsymbol{\theta}^\top \boldsymbol{x}_i \tag{41}$$

- The log-likelihood function is formulated as follows

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{m} \log P(y_i|\boldsymbol{x}_i, \boldsymbol{\theta}) = \sum_{i=1}^{m} y_i \log \lambda_i - \lambda_i - \log y_i! \tag{42}$$

- We have

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \sum_{i=1}^{m} (y_i \boldsymbol{x}_i - e^{\boldsymbol{\theta}^\top \boldsymbol{x}_i}) = 0 \quad \Rightarrow \quad \ln y_i = (\boldsymbol{\theta}^*)^\top \boldsymbol{x}_i \tag{43}$$

- Plot the log-linear regression as below.

# Logistic regression

- We assume that the conditional probability follows

$$P(y_i|\boldsymbol{x}_i, \boldsymbol{\theta}, N) = \text{Bin}(y_i|N, \mu_i) = \binom{N}{y_i} \mu_i^{y_i} (1-\mu_i)^{N-y_i}, \quad \mu_i = \frac{1}{1+e^{-\boldsymbol{\theta}^\top \boldsymbol{x}_i}}. \tag{44}$$

- The log-likelihood function is formulated as follows

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{m} \log P(y_i|\boldsymbol{x}_i, \boldsymbol{\theta}) = y_i \log \mu_i + (N-y_i)\log(1-\mu_i) \tag{45}$$

- We have

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \sum_{i=1}^{m} (y_i - N\mu_i)\boldsymbol{x}_i = 0 \quad \Rightarrow \quad \frac{y_i}{N} = \mu_i = \frac{1}{1+e^{-\boldsymbol{\theta}^\top \boldsymbol{x}_i}}. \tag{46}$$

- Since the $\sigma(a) = \frac{1}{1+e^{-a}}$ is called **sigmoid function** or **logit function**, the above model is called **logit regression** or **logistic regression**.
- Since $\frac{y_i}{N} \in [0,1]$, it can be seen as the posterior probability. Thus, logistic regression is a classification model, rather than regression.

# Summary

- Linear model is the linear function of the parameter $\boldsymbol{\theta}$, rather than the input feature
- Linear model is a special case of generalized linear model, while generalized linear model is not always linear
- Choosing different linear models is equivalent to choosing different distributions of $p(y|\boldsymbol{x}, \boldsymbol{\theta})$ and $p(\boldsymbol{\theta})$, according to the task and the data

# Reading material

- https://www.stat.cmu.edu/~ryantibs/advmethods/notes/glm.pdf