Probabilistic Machine Learning

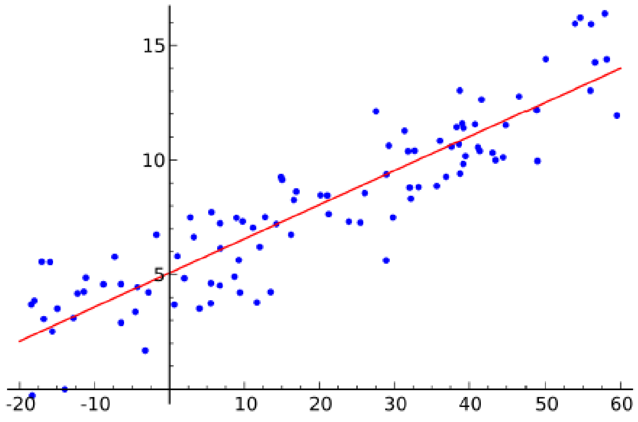
Prof. Seungchul Lee
Industrial AI Lab.

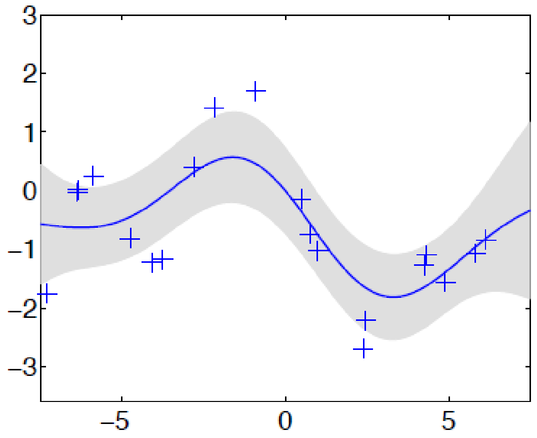
Outline

- *Probabilistic* Linear Regression
- *Probabilistic* Classification
- *Probabilistic* Clustering
- *Probabilistic* Dimension Reduction

Frequentist View of Linear Regression

Probabilistic Linear Regression





- Inference idea

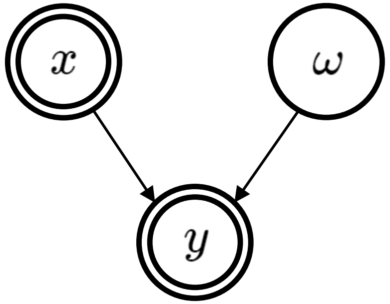
$$P(X \mid \theta) = \text{Probability}\,[\text{data} \mid \text{pattern}]$$

data = underlying pattern + independent noise

- Change your viewpoint of data
  - Generative model

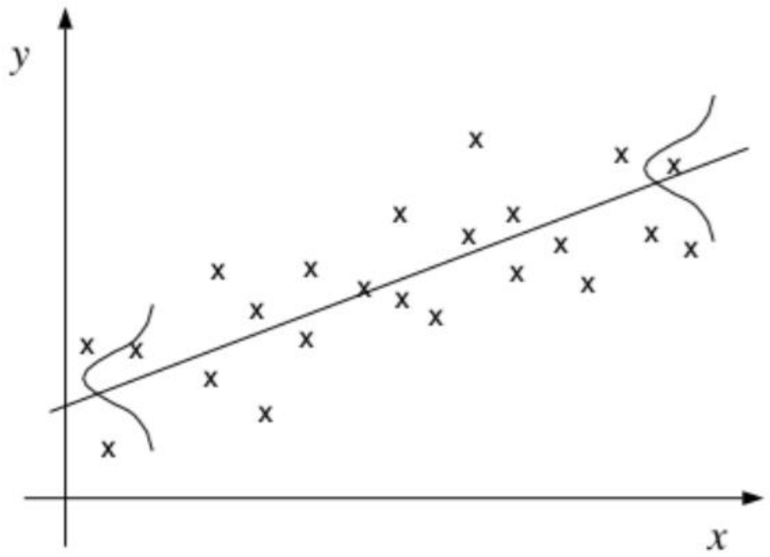
Generative Model: Regression

$$y = \hat{y} + \varepsilon = \omega^T x + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

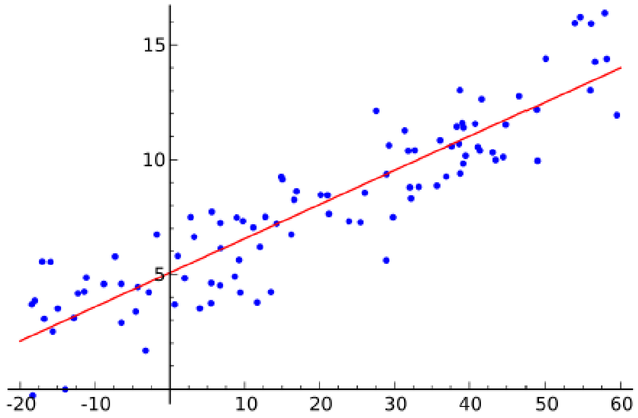


$$D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$$

$$P\left(y \mid x; \omega, \sigma^2\right) = \mathcal{N}\left(\omega^T x, \sigma^2\right)$$



Probabilistic Linear Regression



- Given observed data $D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$,
- We want to estimate the weight vector ω
- Each response generated by a linear model plus Gaussian noise

$$y = \omega^T x + \varepsilon, \quad \varepsilon \sim \mathcal{N}\left(0, \sigma^2\right)$$

- Each response y then becomes a draw from the following Gaussian:

$$y \mid x \sim \left(\omega^T x, \sigma^2\right)$$

- Probability of each response variable

$$P(y \mid x\,;\omega) = \mathcal{N}\left(\omega^T x, \sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\left(y - \omega^T x\right)^2\right)$$

Maximum Likelihood Estimation (MLE)

- Estimate parameters $\theta = (\omega, \sigma^2)$ such that maximize the likelihood given a generative model
  - Likelihood

$$\mathcal{L} = P(D \mid \theta) = P(D\,;\,\theta)$$

$$\hat{\theta}_{MLE} = \underset{\theta}{\operatorname{argmax}} \;\; P(D\,;\,\theta)$$

- Log-likelihood:

$$\begin{aligned}
\ell(\omega, \sigma) = \log \mathcal{L}(\omega, \sigma) &= \log P(D\,;\omega, \sigma^2) \\
&= \log P(Y \mid X\,;\omega, \sigma^2) \\
&= \log \prod_{i=1}^{m} P\left(y_i \mid x_i\,;\omega, \sigma^2\right) \\
&= \sum_{i=1}^{m} \log P\left(y_i \mid x_i\,;\omega, \sigma^2\right) \\
&= \sum_{i=1}^{m} \log \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(y_i - \omega^T x_i\right)^2}{2\sigma^2}\right) \\
&= \sum_{i=1}^{m} \left\{-\frac{1}{2}\log\left(2\pi\sigma^2\right) - \frac{\left(y_i - \omega^T x_i\right)^2}{2\sigma^2}\right\}
\end{aligned}$$

Maximum Likelihood Estimation (MLE)

- Maximum likelihood solution:

$$\log \mathcal{L}(\omega, \sigma) = \sum_{i=1}^{m} \left\{ -\frac{1}{2}\log\left(2\pi\sigma^2\right) - \frac{\left(y_i - \omega^T x_i\right)^2}{2\sigma^2} \right\}$$

$$\begin{aligned}
\hat{\omega}_{MLE} &= \arg\max_{\omega} \log P(D\,;\omega, \sigma^2) \\
&= \arg\max_{\omega} \; -\frac{1}{2\sigma^2}\sum_{i=1}^{m}\left(y_i - \omega^T x_i\right)^2 \\
&= \arg\min_{\omega} \frac{1}{2\sigma^2}\sum_{i=1}^{m}\left(y_i - \omega^T x_i\right)^2 \\
&= \arg\min_{\omega} \sum_{i=1}^{m}\left(y_i - \omega^T x_i\right)^2
\end{aligned}$$

- Big lesson
  - It is equivalent to the least-squares objective function for linear regression (amazing !)
  - In least squares, we implicitly assume that noise is Gaussian distributed

Compute MLE for Linear Regression

$$
\begin{aligned}
\mathcal{L}(\omega, \sigma) &= P\left(y_1, y_2, \cdots, y_m \mid x_1, x_2, \cdots, x_m;\ \underbrace{\omega, \sigma}_{\theta}\right) \\
&= \prod_{i=1}^{m} P\left(y_i \mid x_i;\ \omega, \sigma\right) \\
&= \frac{1}{(2\pi\sigma^2)^{\frac{m}{2}}} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^{m} (y_i - \omega^T x_i)^2\right)
\end{aligned}
$$

Compute MLE for Linear Regression

$$\mathcal{L}(\omega, \sigma) = \frac{1}{(2\pi\sigma^2)^{\frac{m}{2}}} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{m}(y_i - \omega^T x_i)^2\right)$$

$$\ell = -\frac{m}{2}\log 2\pi - m\log\sigma - \frac{1}{2\sigma^2}\sum_{i=1}^{m}\left(y_i - \omega^T x_i\right)^2$$

$$\frac{d\ell}{d\omega} = -2X^T Y + 2X^T X\omega = 0 \qquad \Longrightarrow \qquad \omega_{MLE} = \left(X^T X\right)^{-1} X^T Y$$

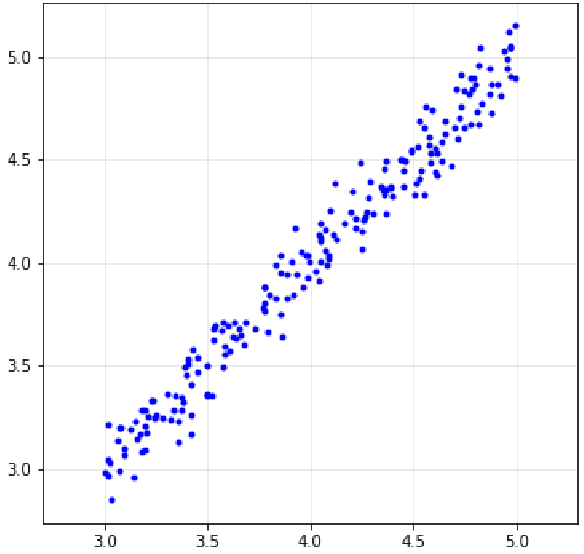
$$\frac{d\ell}{d\sigma} = -\frac{m}{\sigma} + \frac{1}{\sigma^3}\sum_{i=1}^{m}\left(y_i - \omega^T x_i\right)^2 = 0 \qquad \Longrightarrow \qquad \sigma^2_{MLE} = \frac{1}{m}\sum_{i=1}^{m}\left(y_i - \omega^T x_i\right)^2$$

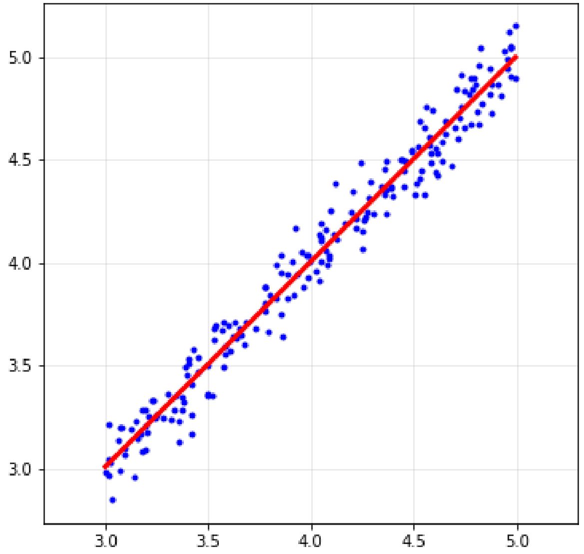
- Big lesson
  - It is equivalent to the least-squares objective function for linear regression (amazing !)
  - In least squares, we implicitly assume that noise is Gaussian distributed

Linear Regression: A Probabilistic View

```python
m = 200

a = 1
x = 3 + 2*np.random.uniform(0,1,[m,1])
noise = 0.1*np.random.randn(m,1)

y = a*x + noise;
y = np.asmatrix(y)
```

```python
A = np.hstack([np.ones([m, 1]), x])
A = np.asmatrix(A)

theta = (A.T*A).I*A.T*y

# to plot the fitted line
xp = np.linspace(np.min(x), np.max(x))
yp = theta[1,0]*xp + theta[0,0]
```



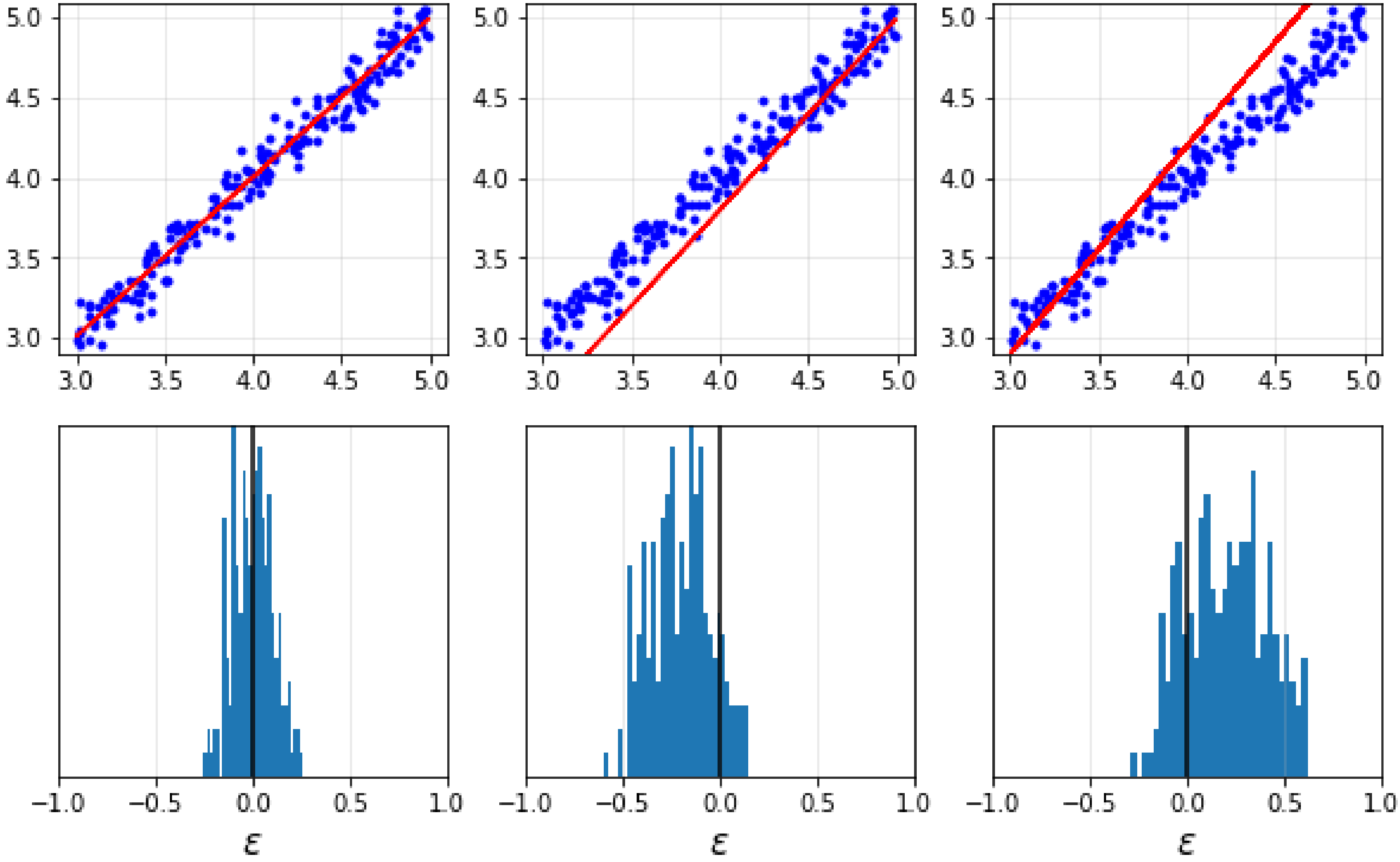


Linear Regression: A Probabilistic View

- Demonstrate

$$\varepsilon \sim \mathcal{N}\left(0, \sigma^2\right)$$

```
yhat0 = theta[1,0]*x + theta[0,0]
err0 = yhat0 - y

yhat1 = 1.2*x - 1
err1 = yhat1 - y

yhat2 = 1.3*x - 1
err2 = yhat2 - y
```

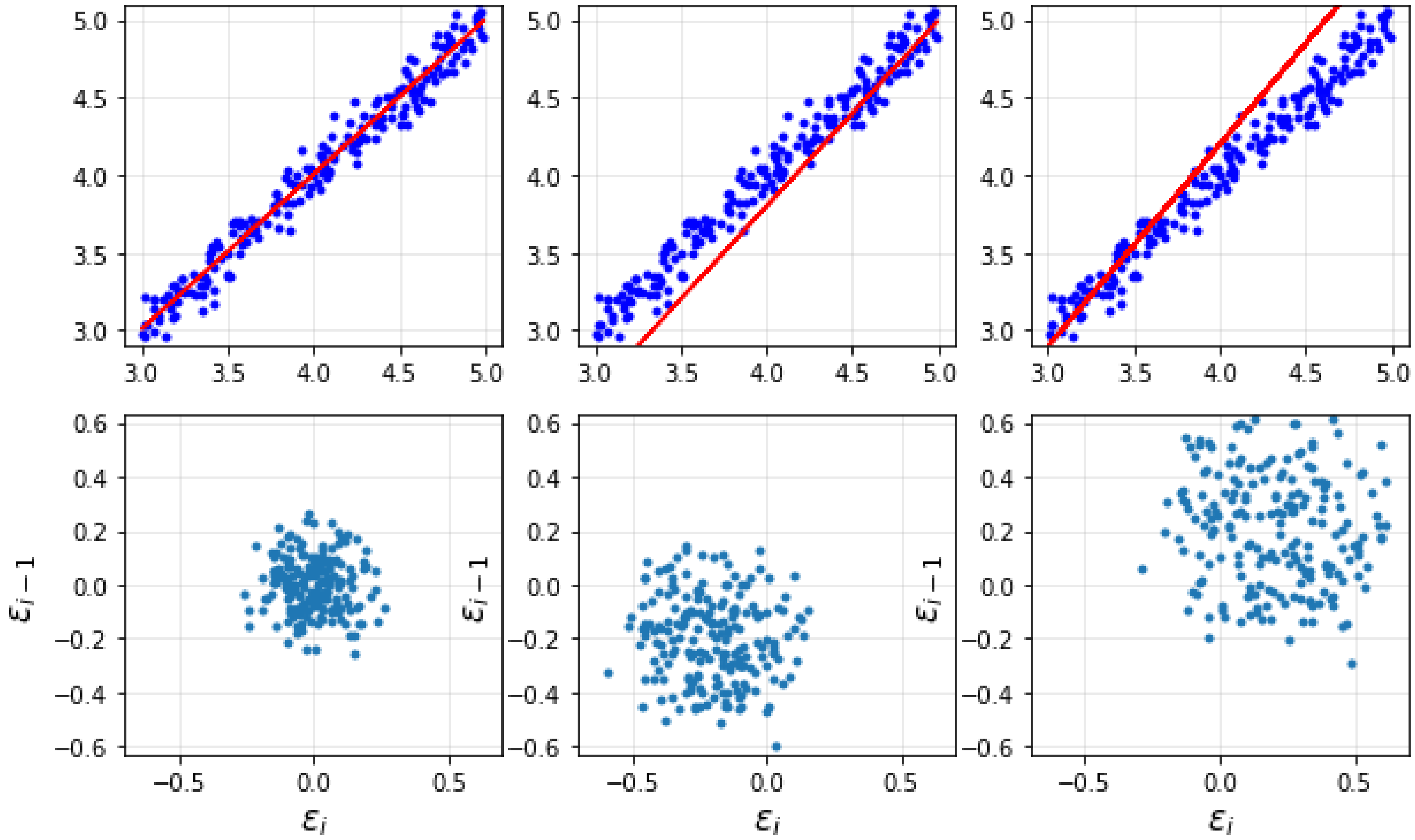


POSTECH

Linear Regression: A Probabilistic View

- Demonstrate
  - samples are independent

```
a0x = err0[1:]
a0y = err0[0:-1]

a1x = err1[1:]
a1y = err1[0:-1]

a2x = err2[1:]
a2y = err2[0:-1]
```



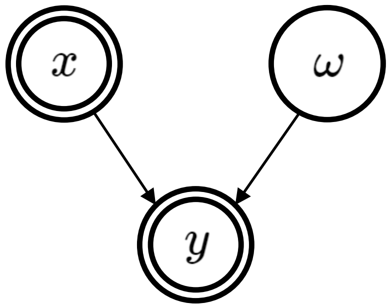
Bayesian View of Linear Regression

Generative Model: Regression

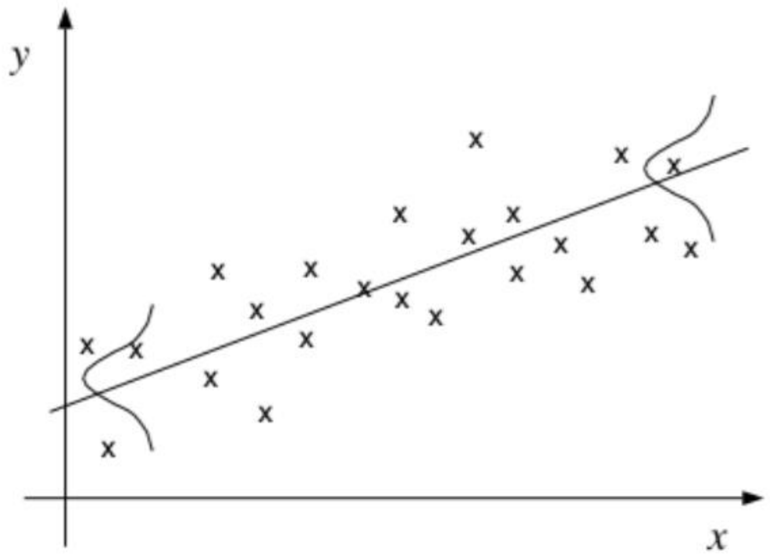
$$y = \hat{y} + \varepsilon = \omega^T x + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

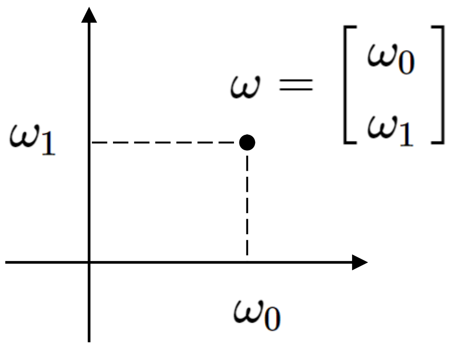
$$D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$$

$$P\left(y \mid x; \omega, \sigma^2\right) = \mathcal{N}\left(\omega^T x, \sigma^2\right)$$

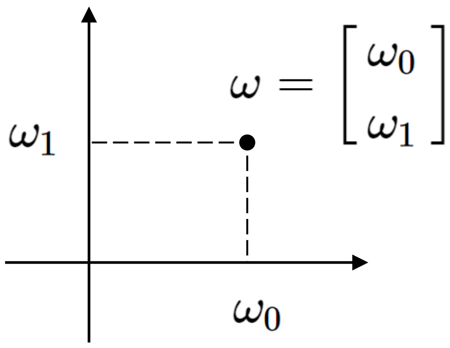


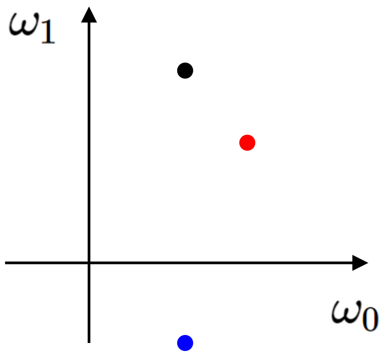
$$y = \omega_1 x + \omega_0 + \varepsilon$$

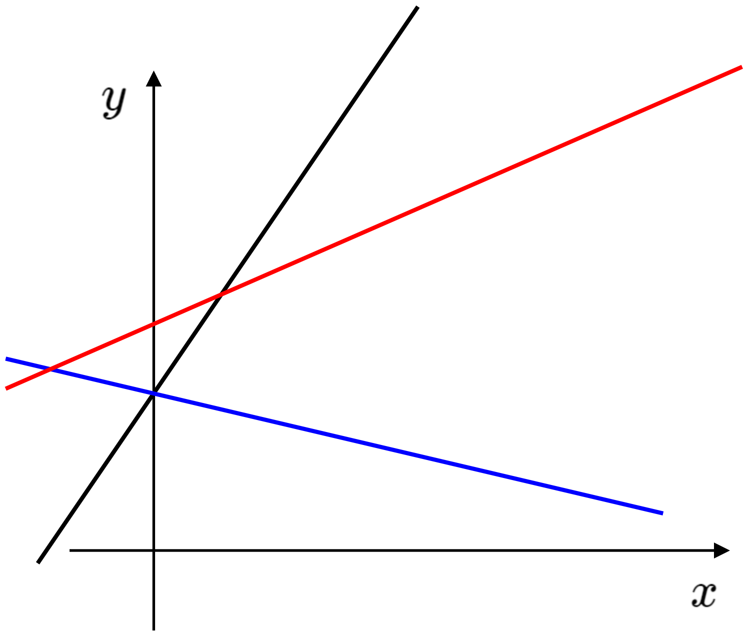




Meaning of ω

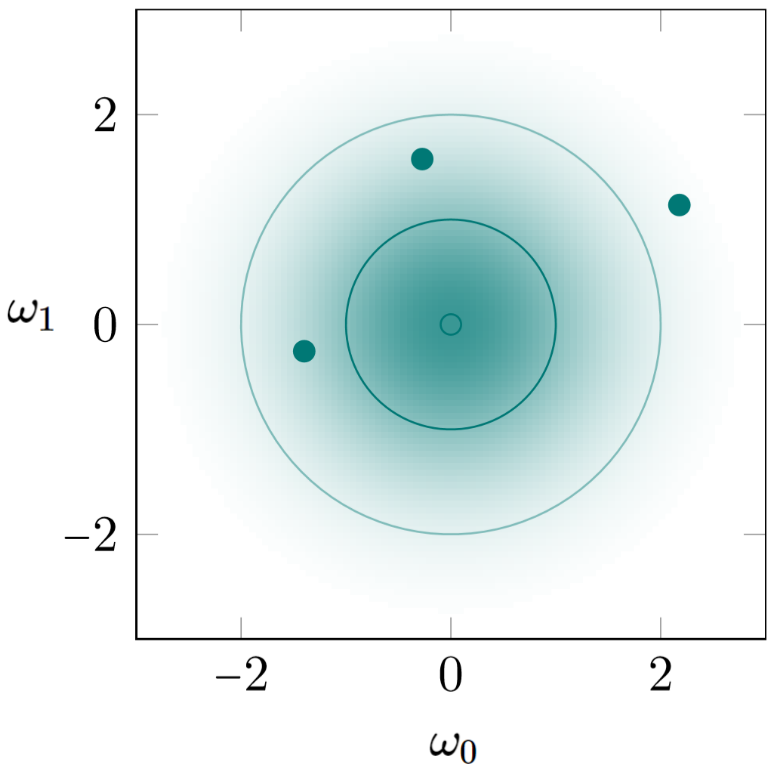


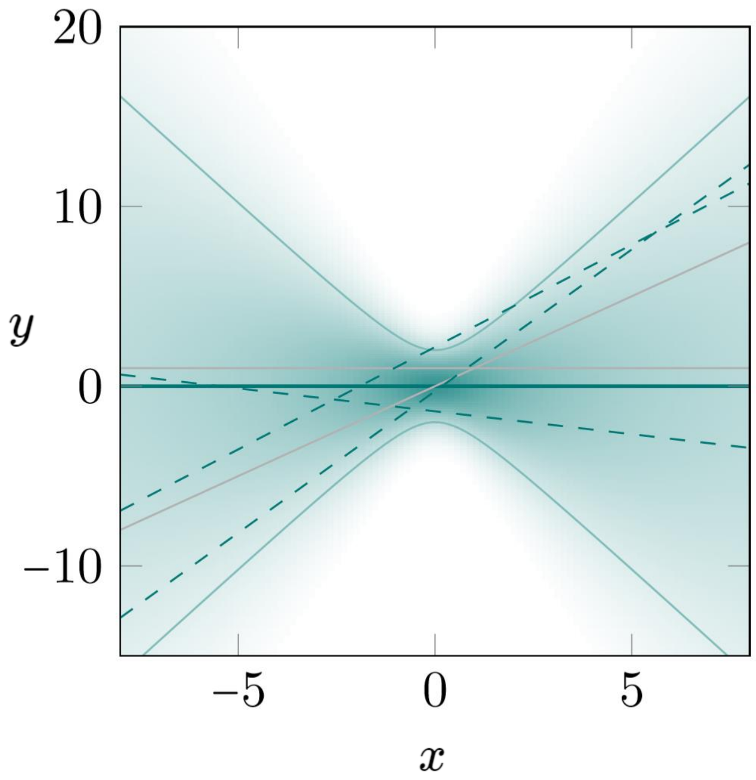




Prior on ω

- Suppose to assume a Gaussian prior distribution over the weight vector ω

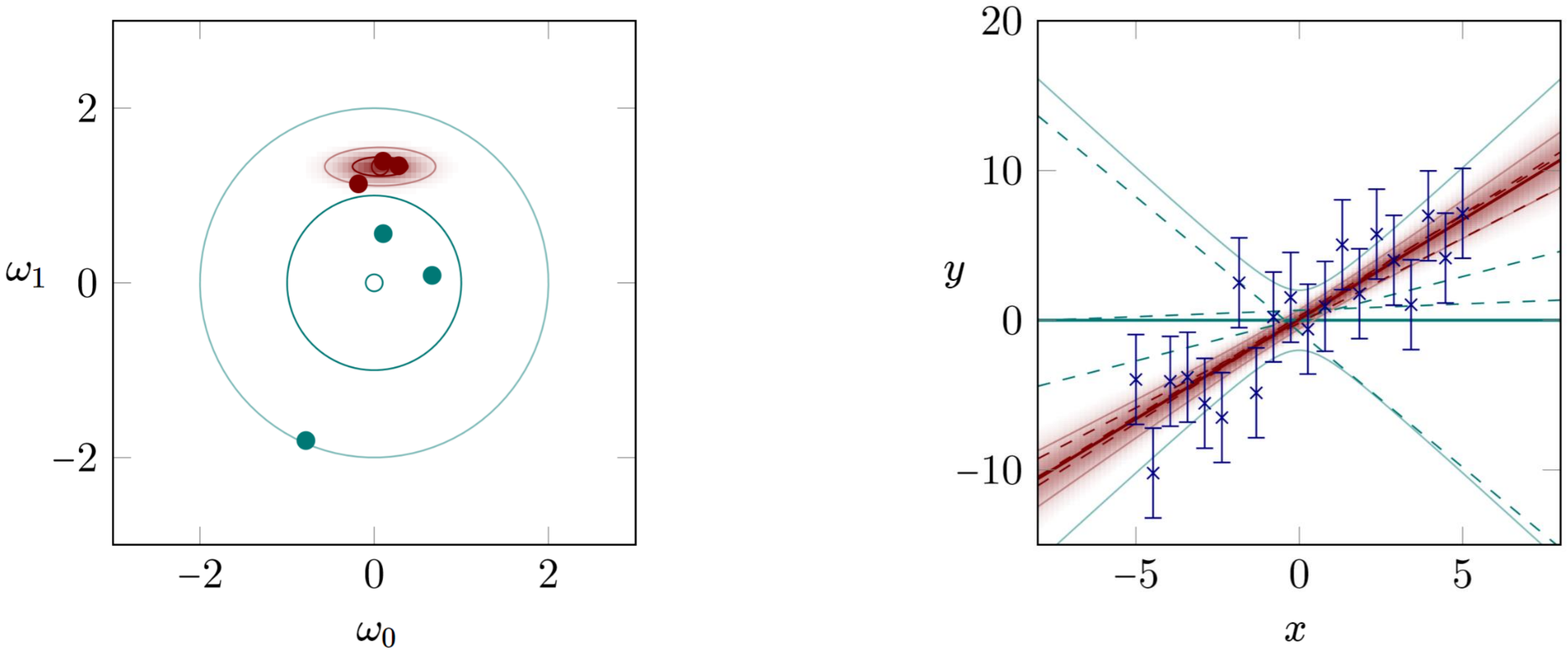




Images from Prof. Philipp Hennig at University of Tubingen

Prior on ω

- Suppose to assume a Gaussian prior distribution over the weight vector ω



Maximum-a-Posteriori (MAP)

- No prior information or uniform distribution on ω leads to MLE

- Suppose to assume a Gaussian prior distribution over the weight vector ω
  - (Make sure you understand what it means)
  - Assume $E[\omega] = 0$ for simplicity

$$P(\omega) \sim \mathcal{N}\left(0, \Sigma\right) = \mathcal{N}\left(0, \lambda^{-1} I\right) = \frac{1}{(2\pi)^{D/2}} \exp\left(-\frac{\lambda}{2} \omega^T \omega\right)$$

- Excellent explanation by Philipp Henning
  - https://www.youtube.com/watch?v=50Vgw11qn0o

Posterior

- Posterior probability
  - Bayes rule

$$P(\omega \mid D) = \frac{P(D \mid \omega)P(\omega)}{P(D)}$$

- Log posterior probability

$$\log P(\omega \mid D) = \log \frac{P(D \mid \omega)P(\omega)}{P(D)} = \log P(D \mid \omega) + \log P(\omega) - \underbrace{\log P(D)}_{\text{constant}}$$

- Maximize log posterior probability

Maximum-a-Posteriori (MAP)

$$\begin{aligned}
\hat{\omega}_{MAP} &= \arg\max_{\omega} \log P(\omega \mid D) \\
&= \arg\max_{\omega} \{\log P(D \mid \omega) + \log P(\omega)\} \\
&= \arg\max_{\omega} \left\{ \sum_{i=1}^{m} \left\{ -\frac{1}{2}\log\left(2\pi\sigma^2\right) - \frac{\left(y_i - \omega^T x_i\right)^2}{2\sigma^2} \right\} - \frac{D}{2}\log(2\pi) - \frac{\lambda}{2}\omega^T\omega \right\} \\
&= \arg\min_{\omega} \frac{1}{2\sigma^2}\sum_{i=1}^{m}\left(y_i - \omega^T x_i\right)^2 + \frac{\lambda}{2}\omega^T\omega
\end{aligned}$$

(ignoring constants and changing max to min)

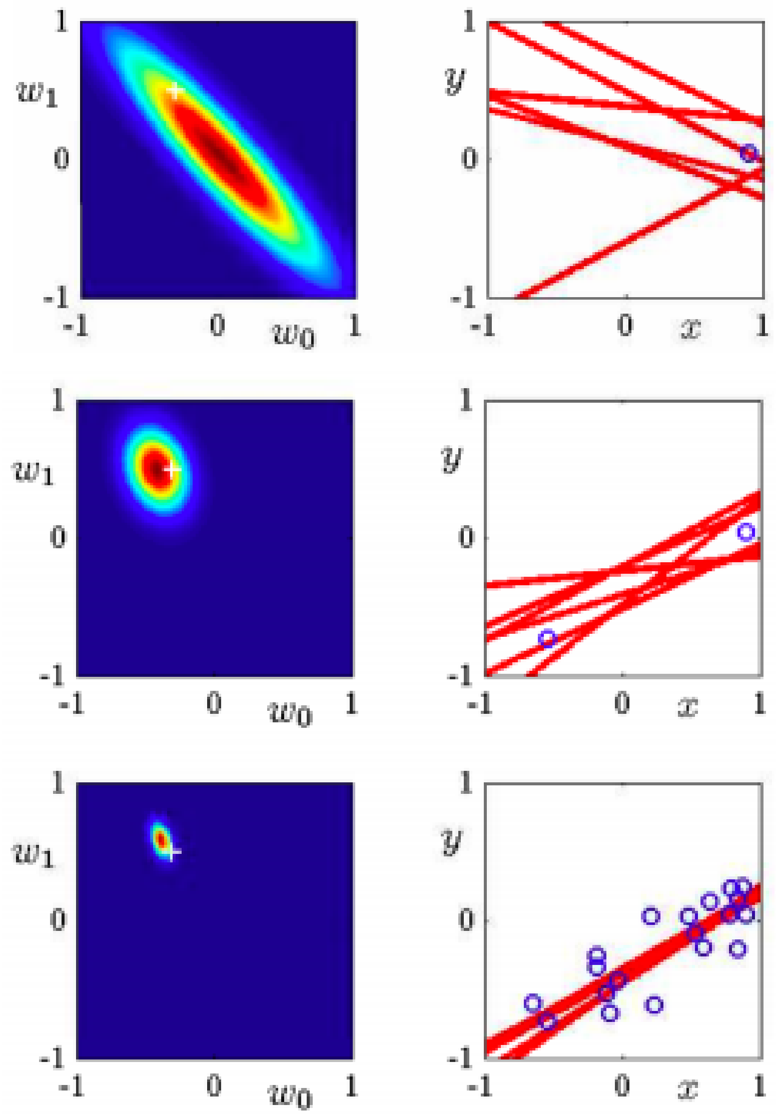
- For $\sigma = 1$ (or some constant) for each input, it's equivalent to the regularized least-squares objective (amazing !)

$$\hat{\omega}_{MAP} = \arg\min_{\omega} \left\{ \sum_{i=1}^{m} \left(y_i - \omega^T x_i\right)^2 + \lambda\omega^T\omega \right\}$$

- Big lesson: MAP = l_2 norm regularization

MAP Illustration

- One observation
- Two observations
- 20 observations



Images from David S. Rosenberg at Bloomberg ML EDU

Summary: MLE vs MAP

- MLE solution:

$$\hat{\omega}_{MLE} = \arg\min_{\omega} \frac{1}{2\sigma^2} \sum_{i=1}^{m} \left(y_i - \omega^T x_i\right)^2$$

- MAP solution:

$$\hat{\omega}_{MAP} = \arg\min_{\omega} \frac{1}{2\sigma^2} \sum_{i=1}^{m} \left(y_i - \omega^T x_i\right)^2 + \frac{\lambda}{2} \omega^T \omega$$

- Take-home messages:
  - MLE estimation of a parameter leads to unregularized solutions
  - MAP estimation of a parameter leads to regularized solutions
  - The prior distribution acts as a regularizer in MAP estimation

- Note : for MAP, different prior distributions lead to different regularizers
  - Gaussian prior on ω regularizes the l_2 norm of ω
  - Laplace prior $exp(-C\|\omega\|_1)$ on ω regularizes the l_1 norm of ω

Probabilistic Classification

Probabilistic Classification

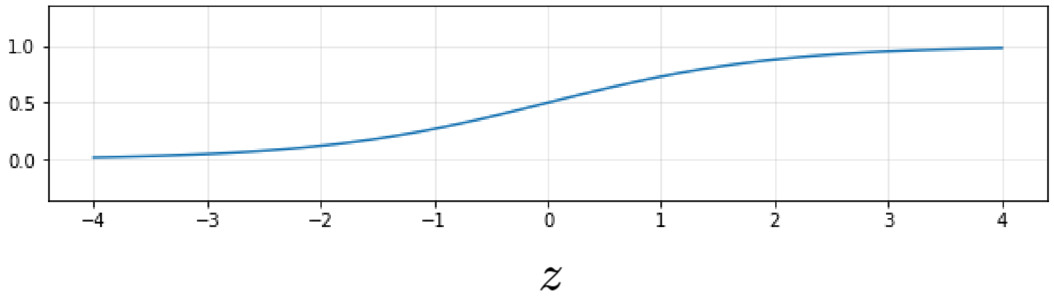
- We want to predict the label probabilities
  - E.g., $P(y = +1|\ x, \omega)$: the probability that the label is $P(y|x, \omega)$
  - In a sense, it is our confidence in the predicted label $+1$

Probabilistic Linear Classification

- Probabilistic classification models allow us do that $(y = -1/+1)$
- Consider the following function in a compact expression

$$P(y \mid x, \omega) = \sigma\left(y\omega^T x\right) = \frac{1}{1 + \exp(-y\omega^T x)}$$

- σ is the logistic function which maps all real number into $(0, 1)$



Logistic Regression

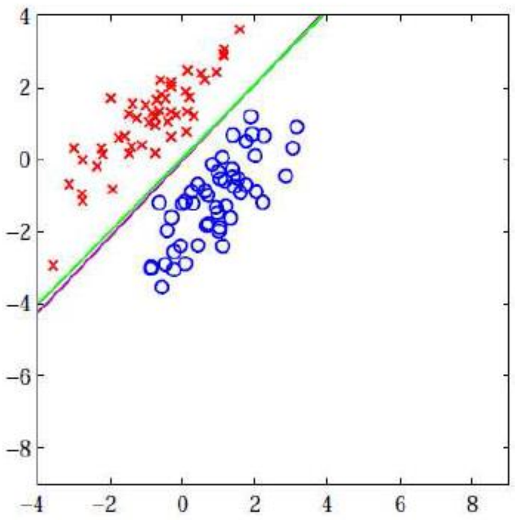
- What does the decision boundary look like for logistic regression?
- At the decision boundary labels $-1/+1$ becomes equiprobable

$$P(y = +1 \mid x, \omega) = P(y = -1 \mid x, \omega)$$

$$\frac{1}{1 + \exp(-\omega^T x)} = \frac{1}{1 + \exp(\omega^T x)}$$

$$\exp\left(-\omega^T x\right) = \exp\left(\omega^T x\right)$$

$$\omega^T x = 0$$



- The decision boundary is therefore linear $\Longrightarrow$ logistic regression is a linear classifier

- Note: it is possible to kernelize and make it nonlinear

Maximum Likelihood Solution

- Goal: want to estimate ω from the data $D = \{(x_1, y_1), \cdots, (x_m, y_m)\}$
- Log-likelihood:

$$
\begin{aligned}
\ell(\omega) = \log \mathcal{L}(\omega) &= \log P(D \mid \omega) \\
&= \log P(Y \mid X, \omega) \\
&= \log \prod_{i=1}^{m} P(y_i \mid x_i, \omega) \\
&= \sum_{i=1}^{m} \log P(y_i \mid x_i, \omega) \\
&= \sum_{i=1}^{m} \log \frac{1}{1 + \exp(-y_i \omega^T x_i)} \\
&= \sum_{i=1}^{m} -\log\left[1 + \exp\left(-y_i \omega^T x_i\right)\right]
\end{aligned}
$$

Maximum Likelihood Solution

- Maximum Likelihood Solution:

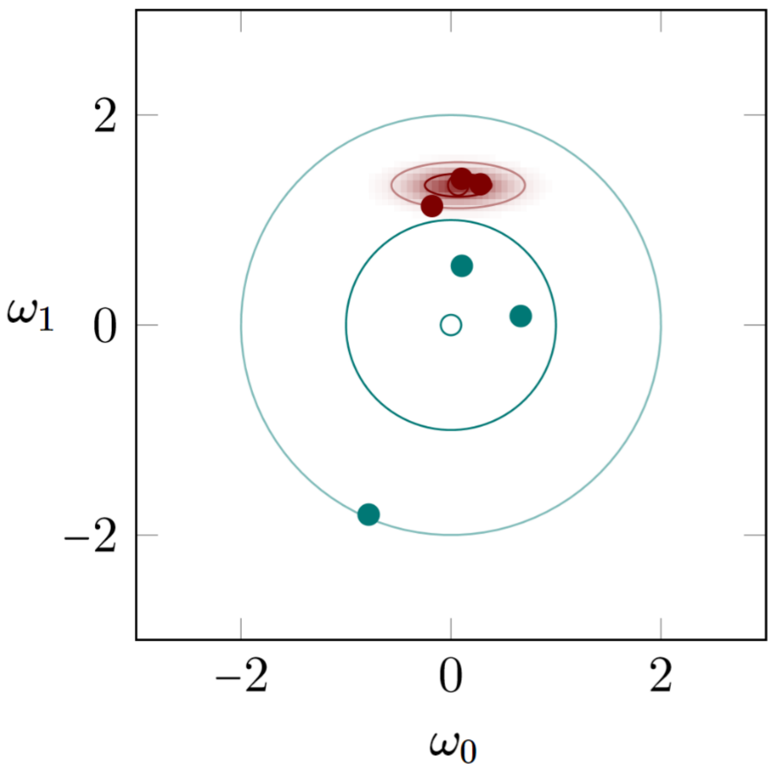
$$\hat{\omega}_{MLE} = \arg\max_{\omega} \log \mathcal{L}(\omega) = \arg\min_{\omega} \sum_{i=1}^{m} \log\left[1 + \exp\left(-y_i \omega^T x_i\right)\right]$$

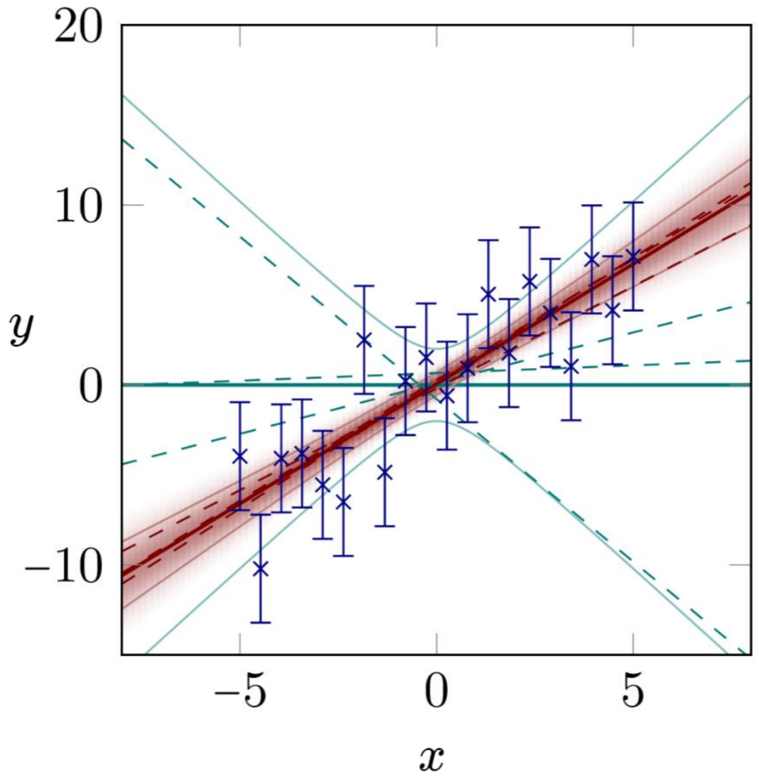
- No closed-form solution exists, but we can do
  - CVXPY (we did it)
  - Gradient descent on ω

$$\begin{aligned}\nabla_{\omega} \log \mathcal{L}(\omega) &= \sum_{i=1}^{m} -\frac{1}{1+\exp(-y_i\omega^T x_i)} \exp\left(-y_i \omega^T x_i\right)\left(-y_i x_i\right) \\ &= \sum_{i=1}^{m} \frac{1}{1+\exp(y_i \omega^T x_i)} y_i x_i\end{aligned}$$

Prior on ω

- Suppose to assume a Gaussian prior distribution over the weight vector ω



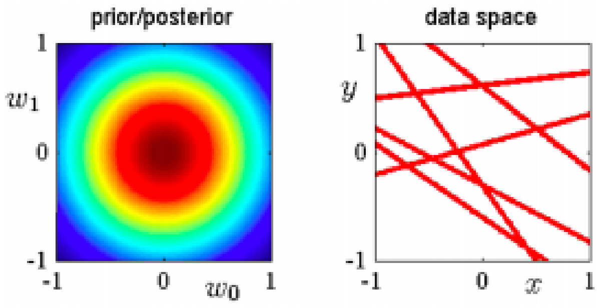


Images from Prof. Philipp Hennig at University of Tubingen

Maximum-a-Posteriori Solution

- Let's assume a Gaussian prior distribution over the weight vector ω

$$P(\omega) = \mathcal{N}\left(0, \lambda^{-1} I\right) = \frac{1}{(2\pi)^{D/2}} \exp\left(-\frac{\lambda}{2} \omega^T \omega\right)$$



- Maximum-a-Posteriori Solution:

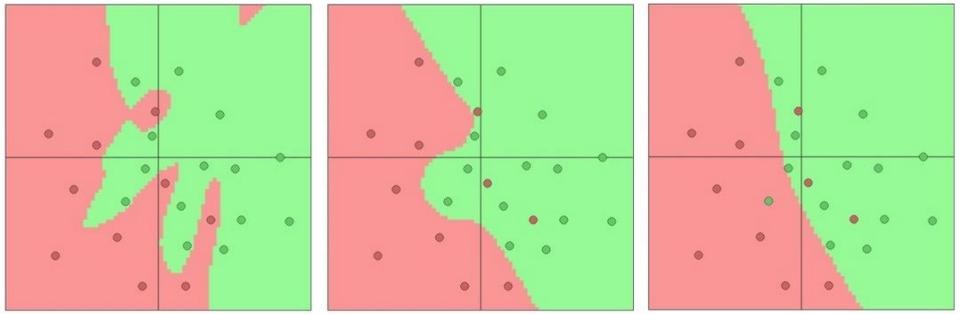
$$\begin{aligned}
\hat{\omega}_{MAP} &= \arg\max_{\omega} \log P(\omega \mid D) \\
&= \arg\max_{\omega} \{\log P(D \mid \omega) + \log P(\omega) - \underbrace{\log P(D)}_{\text{constant}}\} \\
&= \arg\max_{\omega} \{\log P(D \mid \omega) + \log P(\omega)\} \\
&= \arg\max_{\omega} \left\{ \sum_{i=1}^{m} - \log\left[1 + \exp\left(-y_i \omega^T x_i\right)\right] - \frac{D}{2}\log(2\pi) - \frac{\lambda}{2}\omega^T \omega \right\} \\
&= \arg\min_{\omega} \sum_{i=1}^{m} \log\left[1 + \exp\left(-y_i \omega^T x_i\right)\right] + \frac{\lambda}{2}\omega^T \omega
\end{aligned}$$

(ignoring constants and changing max to min)

- Big lesson: MAP = l_2 norm regularization

Maximum-a-Posteriori Solution

- Q: What does regularizer do in a classifier?
- A: Nonlinear classifier gives more intuitive explanation



- No closed-form solution exists but we can do gradient descent on ω
  - See "A comparison of numerical optimizers for logistic regression" by Tom Minka on optimization techniques (gradient descent and others) for logistic regression
  - (both MLE and MAP)

Summary: MLE vs MAP

- MLE solution:

$$\hat{\omega}_{MLE} = \arg\min_{\omega} \sum_{i=1}^{m} \log\left[1 + \exp\left(-y_i \omega^T x_i\right)\right]$$

- MAP solution:

$$\hat{\omega}_{MAP} = \arg\min_{\omega} \sum_{i=1}^{m} \log\left[1 + \exp\left(-y_i \omega^T x_i\right)\right] + \frac{\lambda}{2}\omega^T \omega$$

- Take-home messages (we already saw these before)
  - MLE estimation of a parameter leads to unregularized solutions
  - MAP estimation of a parameter leads to regularized solutions
  - The prior distribution acts as a regularizer in MAP estimation

- Note: For MAP, different prior distributions lead to different regularizers
  - Gaussian prior on ω regularizer the l_2 norm of ω
  - Laplace prior $exp(-C\|\omega\|_1)$ on ω regularizes the l_1 norm of ω

Probabilistic Clustering

- will not cover in this course

Probabilistic Dimension Reduction

- will not cover in this course

Summary

- *Probabilistic* Linear Regression
- *Probabilistic* Classification
- *Probabilistic* Clustering
- *Probabilistic* Dimension Reduction