Logistic Regression

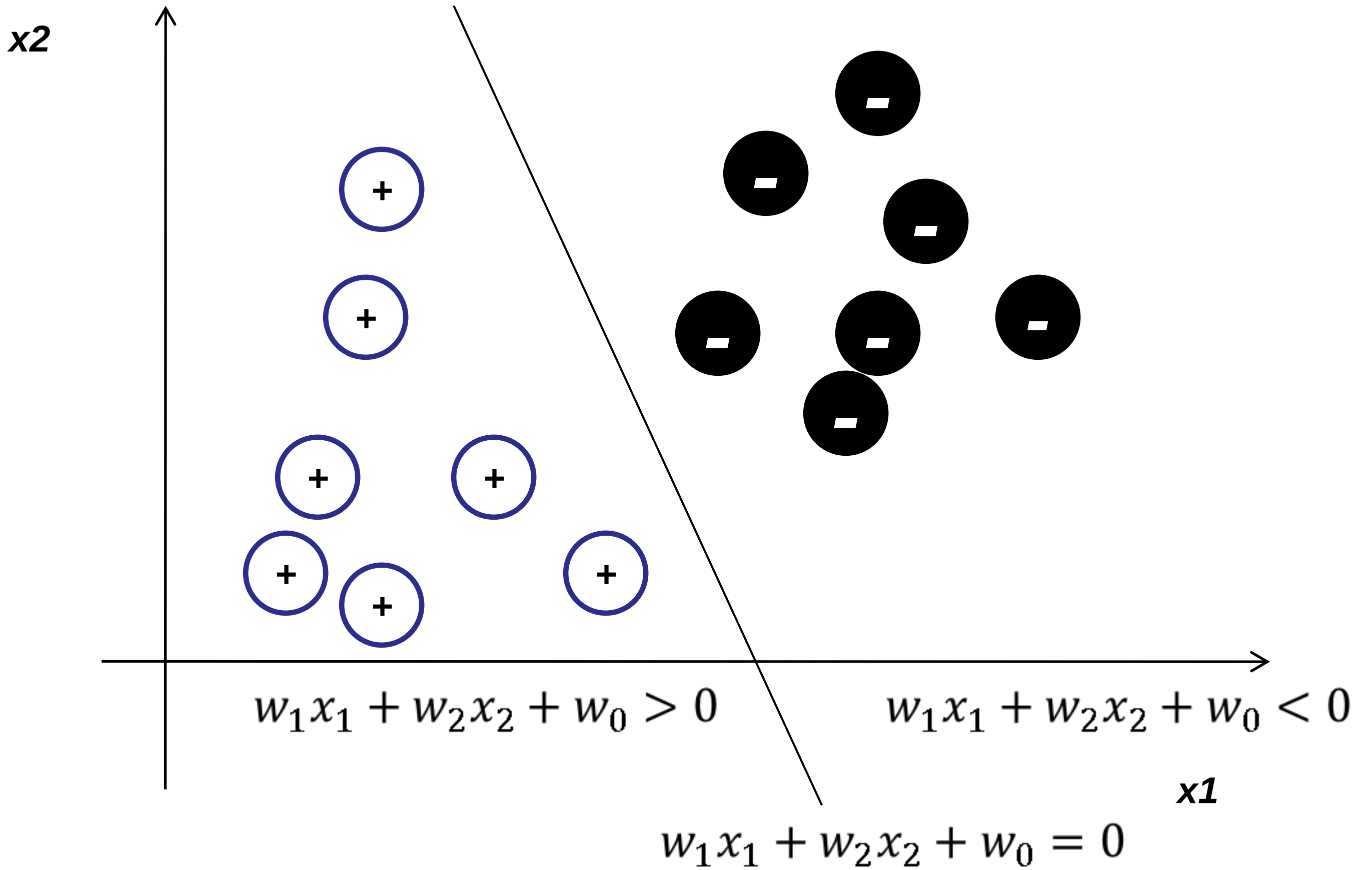
Concepts:
Probabilistic predictions and decision theory
Maximum likelihood estimation application to LR
Maximum a posterior (MAP) estimation and connection to regularization

Classification problem

- Given input $\mathbf{x}$, the goal is to predict y**,** which is a categorical variable
  - $\mathbf{x}$**:** the feature vector
  - y: the class label
- Example:
  - $\mathbf{x}$: monthly income and bank saving amount;
    y: risky or not risky
  - $\mathbf{x}$: review text for a product
    y: sentiment positive, negative or neutral

Binary Linear Classifier

- We will be begin with the simplest choice: linear classifiers for binary classification problems ($y \in \{0,1\}$)

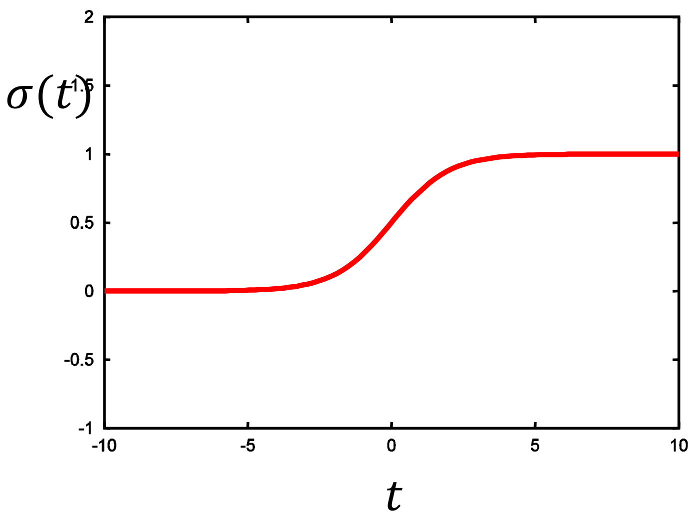


Logistic Regression

- Input $\mathbf{x} \in R^d$, target output $y \in \{0,1\}$
- Logistic regression is a probabilistic classifier:

$$\mathrm{P}(\mathrm{y}=1|\mathbf{x};\mathbf{w}) = \sigma\left(\mathbf{w}^{\mathbf{T}}\mathbf{x}\right) = \frac{1}{1+\exp(-\mathbf{w}^{\mathbf{T}}\mathbf{x})}$$

$$\mathrm{P}(\mathrm{y}=0|\mathbf{x};\mathbf{w}) = 1 - \sigma(\mathbf{w}^{\mathbf{T}}\mathbf{x})$$



- linear function $\mathbf{w}^T\mathbf{x}$ has range $(-\infty, \infty)$
- Sigmoid function σ warps the value of $\mathbf{w}^{\mathbf{T}}\mathbf{x}$ to a value between 0 and 1

Logistic regression: linear classifier

- Maximum A Posteriori (MAP) estimation of y:

$$y_{map} = \arg\max_{v \in \{0,1\}} P(y = v | \mathbf{x}; \mathbf{w})$$

- We will predict $y = 1$ if

$$P(y = 1 | \mathbf{x}; \mathbf{w}) \geq P(y = 0 | \mathbf{x}; \mathbf{w}) \Rightarrow$$

$$\frac{1}{1 + \exp(-\mathbf{w}^{\mathbf{T}}\mathbf{x})} \geq \frac{\exp(-\mathbf{w}^{\mathbf{T}}\mathbf{x})}{1 + \exp(-\mathbf{w}^{\mathbf{T}}\mathbf{x})} \Rightarrow$$

$$1 \geq \exp\left(-\mathbf{w}^{\mathbf{T}}\mathbf{x}\right) \Rightarrow$$

$$0 \geq -\mathbf{w}^{\mathbf{T}}\mathbf{x} \Rightarrow \mathbf{w}^{\mathbf{T}}\mathbf{x} \geq \mathbf{0}$$

- We refer to $\mathbf{w}^{\mathbf{T}}\mathbf{x} = 0$ as our decision boundary

A more general decision rule

If we have some knowledge about the cost of different types of mistakes, given $P(y|\mathbf{x})$, we can choosing the prediction that ***minimizes <u>the expected cost:</u>***

$$y^* = \arg\min_y \sum_{y'} L(y, y') P(y'|\mathbf{x})$$

True label→ Predicted ↓	**Spam**	**Non-spam**
Spam	0	10
Non-spam	1	0

For example: $P(y = spam|\mathbf{x}) = 0.6$

- The expected cost if predict <u>spam</u>?
- What if we predict <u>non-spam</u>?
- Which prediction minimizes the expected cost?

With this more general decision rule, it can be shown that Logistic regression still leads to a linear classifier, just with a different threshold

Learning for Logistic Regression

- Given a set of training examples:

- We assume examples are identically, independently distributed (I.I.D.) following:

$$P(y = 1|\mathbf{x}; \mathbf{w}) = \frac{1}{1 + \exp(-\mathbf{w}^T\mathbf{x})}$$

- Learn $\mathbf{w}$ from the training data using <u>Maximum Likelihood Estimation</u>

Maximum (conditional) Likelihood Estimation

- Data log-likelihood:

$$\log \prod_i P(\mathbf{x}^i, y^i; \mathbf{w}) = \sum_i \log P(\mathbf{x}^i, y^i; \mathbf{w})$$

$$= \sum_i \log P\left(y^i \middle| \mathbf{x}^i; \mathbf{w}\right) P\left(\mathbf{x}^i; \mathbf{w}\right)$$

$$= \sum_i \log P\left(y^i \middle| \mathbf{x}^i; \mathbf{w}\right) + C$$

- Maximum (conditional) likelihood estimation of $\mathbf{w}$:

$$\mathbf{w}_{\boldsymbol{MLE}} = \mathrm{argmax}_{\mathrm{w}} \sum_i \log P\left(y^i \middle| \mathbf{x}^i; \mathbf{w}\right)$$

Computing Log-likelihood

$$l(\mathbf{w}) = \sum_i \log P\left(y^i \middle| \mathbf{x}^i; \mathbf{w}\right)$$

$$= \sum_{y^i=1} \log P\left(y = 1 \middle| \mathbf{x}^{\boldsymbol{i}}; \mathbf{w}\right) + \sum_{y^i=0} \log P\left(y = 0 \middle| \mathbf{x}^{\boldsymbol{i}}; \mathbf{w}\right)$$

$$= \sum_i y^i \log P\left(y = 1 \middle| \mathbf{x}^{\boldsymbol{i}}; \mathbf{w}\right) + \left(1 - y^i\right) \log P\left(y = 0 \middle| \mathbf{x}^{\boldsymbol{i}}; \mathbf{w}\right)$$

$$= \sum_i y^i \log \frac{P\left(y = 1 \middle| \mathbf{x}^{\boldsymbol{i}}; \mathbf{w}\right)}{P\left(y = 0 \middle| \mathbf{x}^{\boldsymbol{i}}; \mathbf{w}\right)} + \log P(y = 0 | \mathbf{x}^{\boldsymbol{i}}; \mathbf{w})$$

$$= \sum_i y^i \mathbf{w}^{\mathbf{T}} \mathbf{x}^i + \log\left(\frac{\exp\left(-\mathbf{w}^T \mathbf{x}^i\right)}{1 + \exp(-\mathbf{w}^T \mathbf{x}^i)}\right)$$

$$= \sum_i y^i \mathbf{w}^{\mathbf{T}} \mathbf{x}^i - \log\left(1 + \exp\left(\mathbf{w}^T \mathbf{x}^i\right)\right)$$

Concave!

Gradient

$$l(\mathbf{w}) = \sum_i \left[y^i \mathbf{w}^{\mathbf{T}} \mathbf{x}^i - \log\left(1 + \exp\left(\mathbf{w}^T \mathbf{x}^i\right)\right)\right]$$

$$\nabla l = \sum_i \left[y^i \mathbf{x}^i - \frac{\exp\left(\mathbf{w}^T \mathbf{x}^i\right) \mathbf{x}^i}{1 + \exp(\mathbf{w}^T \mathbf{x}^i)} \right]$$

$$= \sum_i \left[y^i \mathbf{x}^i - \frac{\mathbf{x}^i}{1 + \exp(-\mathbf{w}^T \mathbf{x}^i)} \right]$$

$$= \sum_i \left[y^i - P(y = 1 | \mathbf{x}^i; \mathbf{w})\right] \mathbf{x}^i$$

Batch Gradient Ascent for LR

Given : training examples $(\mathbf{x}^i, y^i)$, $i = 1,\ldots,N$

Let $\mathbf{w} \leftarrow \mathbf{w}_0$ // e.g., (0,0,0,...,0)

Repeat until convergence

$\quad \mathbf{d} \leftarrow (0,0,0,\ldots,0)$

$\quad$ For $i = 1$ to N do

$$\hat{y}^i \leftarrow \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}^i}}$$

$$error = y^i - \hat{y}^i$$

$$\mathbf{d} = \mathbf{d} + error \cdot \mathbf{x}^i$$

$\quad \mathbf{w} \leftarrow \mathbf{w} + \eta\,\mathbf{d}$

- **Online** gradient ascent algorithm can be easily constructed

Soft-max Logistic Regression

- For $k > 2$ classes, we can define the posterior probability using the <u>soft-max function</u>

$$p(y = k|\mathbf{x}) = \hat{y}_k = \frac{\exp\left(\mathbf{w}_k^T \mathbf{x}\right)}{\sum_{j=1}^{K} \exp(\mathbf{w}_j^T \mathbf{x})}$$

- Going through the same MLE derivations, we arrive at the following gradient:

$$\nabla_{\mathbf{w}_k} L = \sum_{i=1}^{N} (y_k^i - \hat{y}_k^i)\mathbf{x}^i$$

where $y_k^i = 1$ if $y^i = k$, and 0 otherwise for $k = 1, \ldots, K$

- So far we have introduced MLE for logistic regression
- We will now introduce another paradigm for estimating model parameters
  - The Bayesian paradigm

Bayesian vs. Frequentist

- Two different views for parameter estimation
- Frequentist: a parameter is a deterministic unknown value
- Bayesian: a parameter is a random variable with a distribution
  - Use priors to express our belief/preference about the parameter before observing any data
  - After observing the data, update our belief by computing the posterior distribution of the parameter

Prior distribution of θ

$$p(\theta|D) = \frac{p(\theta)p(D|\theta)}{p(D)} = \frac{p(\theta)p(D|\theta)}{\int p(D|\theta)p(\theta)d\theta}$$

Posterior distribution of θ

Maximum A Posteriori (MAP) estimation as a penalty method

$$\hat{\theta}_{MAP} = \underset{\theta}{\operatorname{argmax}}\, p(\theta|D)$$

$$= \underset{\theta}{\operatorname{argmax}}\, p(D|\theta)p(\theta)$$

$$= \underset{\theta}{\operatorname{argmax}}\; \log p(D|\theta) + \log p(\theta)$$

Penalty term

MAP for Logistic Regression

$$\underset{\mathbf{w}}{\operatorname{argmax}} P(\mathbf{w}|\boldsymbol{D}) = \underset{\mathbf{w}}{\operatorname{argmax}} P(\boldsymbol{D}|\mathbf{w})\, P(\mathbf{w})$$
$$= \underset{\mathbf{w}}{\operatorname{argmax}} \log P(\boldsymbol{D}|\mathbf{w}) + \log P(\mathbf{w})$$

- $\log P(D|\mathbf{w})$: the log-likelihood of $\mathbf{w}$

$$\sum_i \log P\left(y^i \middle| \mathbf{x}^i, \mathbf{w}\right)$$

- $P(\mathbf{w})$: a prior distribution

$$\mathbf{w} \sim N(0, \sigma^2 \mathbf{I})$$

Large weights correspond to more complex models, this prior prefer simpler hypothesis (zero mean)

Logistic Regression: MAP

$$\underset{\mathbf{w}}{\operatorname{argmax}} \log P(\boldsymbol{D}|\mathbf{w}) + \log P(\mathbf{w})$$

$$= \underset{\mathrm{w}}{\operatorname{argmax}}\, l(\mathbf{w}) + \log N(\mathbf{w}; 0, \sigma^2\mathbf{I})$$

$$= \underset{\mathrm{w}}{\operatorname{argmax}}\, l(\mathbf{w}) + \sum_j \log\left(\frac{1}{\sqrt{2\pi}\sigma} \exp \frac{-w_j^2}{2\sigma^2}\right)$$

$$= \underset{\mathrm{w}}{\operatorname{argmax}}\, l(\mathbf{w}) + \sum_j \frac{-w_j^2}{2\sigma^2}$$

$$\boxed{\lambda = \frac{1}{\sigma^2}}$$

$$= \underset{\mathrm{w}}{\operatorname{argmax}}\, l(\mathbf{w}) - \frac{\lambda}{2}\sum_j w_j^2$$

Regularization

Old delta:

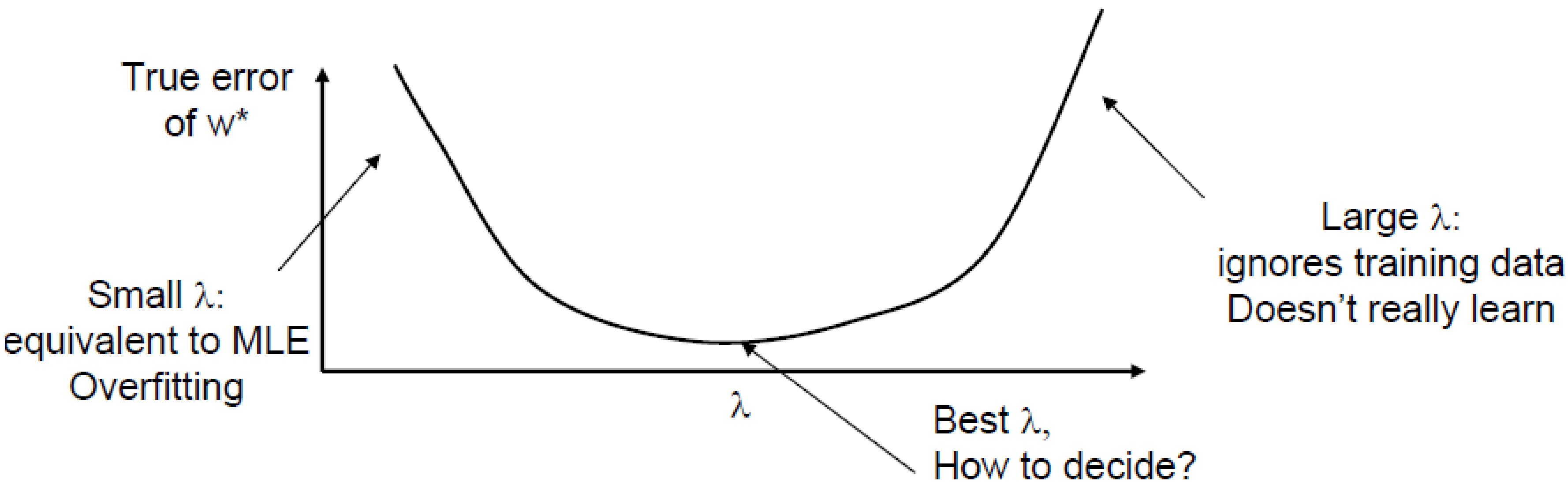
$$\boxed{\nabla L(\mathbf{w}) = \sum_{i=1}^{N} (y^i - \hat{y}^i)\mathbf{x}^i}$$

$$\Rightarrow$$

$$\boxed{\nabla L(\mathbf{w}) = \sum_{i=1}^{N} (y^i - \hat{y}^i)\mathbf{x}^i - \lambda\mathbf{w}}$$

Impact of λ

- λ is inversely proportional to the variance of our prior belief $\boxed{\lambda = \frac{1}{\sigma^2}}$



- Use cross-validation to choose

Summary of Logistic Regression

- A popular discriminative classifier
- Learns conditional probability distribution $P(y \mid \mathbf{x})$
  - Defined by a logistic function
  - Produces a **linear** decision boundary
  - Nonlinear classifier by using basis functions
- Maximum likelihood estimation (MLE)
  - Gradient ascent bears interesting similarity with perceptron
  - Unstable for linearly separable case, regularization can avoid this issue
  - Multi-class logistic regression: use the soft-max function
- Maximum posterior estimation (MAP)
  - Gaussian prior on the weights $= L_2$ regularization
  - Overfitting controlled by the variance on the prior