Machine Learning

Discriminative Models

Lecturer: Duc Dung Nguyen, PhD.
Contact: nddung@hcmut.edu.vn

Faculty of Computer Science and Engineering
Hochiminh city University of Technology

Contents

Discriminative Models

Discriminative vs. generative models

Generative model (BNs, HMMs)

- Joint distributions: $p(\mathbf{y}, \mathbf{x})$
- It can generate any distribution on $\mathbf{y}$ and $\mathbf{x}$.

Discriminative model:

- Conditional distributions: $p(\mathbf{y}|\mathbf{x})$
- It discriminates $\mathbf{y}$ given $\mathbf{x}$.

Feature-based Linear Classifier

- Decision is based on a linear combination of features.

Feature-based linear classifiers

- Decision is based on a linear combination of features.
- Features are essential pieces of information in an observation that decide classification.

Feature-based linear classifiers

- Decision is based on a linear combination of features.
- Features are essential pieces of information in an observation that decide classification.
- A feature can be defined as a function with a bounded real value of a class and an observation:

$$f(c, \mathbf{x})$$

Feature-based linear classifiers

- Decision is based on a linear combination of features.
- Features are essential pieces of information in an observation that decide classification.
- A feature can be defined as a function with a bounded real value of a class and an observation:

$$f(c, \mathbf{x})$$

- Linear classifier:

$$\arg\max_{c \in C} \sum_{m=1..M} \lambda_m . f_m(c, x)$$

Feature-based linear classifiers

- SVMs:

$$y(\mathbf{x}) = \mathbf{w}.\mathbf{x} + \mathbf{b} = \sum_{m=1..M} \mathbf{w}_m.\mathbf{x}_m + \mathbf{b}$$

Feature function: $f_m(c, \mathbf{x}) = x_m$

- Naive Bayes classifier:

$$c_{NB} = \arg\max_{c \in C} p(c). \prod_{m=1}^{M} p(x_m|c)$$

$$c_{NB} = \arg\max_{c \in C} \left(\log p(c) + \sum_{m=1}^{M} \log p(x_m|c) \right)$$

Feature function:

$$f_m(c, \mathbf{x}) = \log p(x_m|c)$$

- In NLP, a feature could be an indicator function whose value is boolean:

$$f(c, \mathbf{x}) = (\text{c is a certain class}) \wedge (\text{x has a certain property})$$

Feature-based linear classifiers

- In NLP, a feature could be an indicator function whose value is boolean:

$$f(c, \mathbf{x}) = (\text{c is a certain class}) \wedge (\text{x has a certain property})$$

- Example:

$$f(c, \mathbf{x}) = (\text{c} = \text{NUMBER}) \text{ and } (\text{x contains only digits}))$$

$$f(\text{NUMBER}, \text{"2018"}) = 1$$

$$f(\text{NUMBER}, \text{"may 2018"}) = 0$$

$$f(\text{IDENTIFIER}, \text{"may 2018"}) = 0$$

Feature-based linear classifiers

- Empirical count of a feature:

$$\tilde{E}(f_m) = \sum_{(c,\mathbf{x}) \in \text{observed}(c,\mathbf{x})} \tilde{p}(c,\mathbf{x}).f_m(c,\mathbf{x})$$

$$\tilde{E}(f_m) = \frac{1}{D} \sum_{(c,\mathbf{x}) \in \text{observed}(c,\mathbf{x})} f_m(c,\mathbf{x})$$

- Model expectation of a feature:

$$E(f_m) = \sum_{(c,\mathbf{x})\in(C,X)} p(c,\mathbf{x}).f_m(c,\mathbf{x})$$

$$E(f_m) = \sum_{(c,\mathbf{x})\in(C,X)} p(\mathbf{x}).p(c|\mathbf{x}).f_m(c,\mathbf{x})$$

$$E(f_m) \approx \sum_{(c,\mathbf{x})\in(C,X)} \tilde{p}(\mathbf{x}).p(c|\mathbf{x}).f_m(c,\mathbf{x})$$

$$E(f_m) \approx \frac{1}{D} \sum_{\mathbf{x}\in\text{observed}(\mathbf{x})} \sum_{c\in C} p(c|\mathbf{x}).f_m(c,\mathbf{x})$$

Feature-based linear classifiers

- Empirical count of a feature:

$$\tilde{E}(f_m) = \frac{1}{D} \sum_{(c,\mathbf{x}) \in \text{observed}(c,\mathbf{x})} f_m(c, \mathbf{x})$$

- Model expectation of a feature:

$$E(f_m) = \frac{1}{D} \sum_{\mathbf{x} \in \text{observed}(\mathbf{x})} \sum_{c \in C} p(c|\mathbf{x}).f_m(c, \mathbf{x})$$

- Consistency constraint:

$$E(f_m) = \tilde{E}(f_m)$$

Logistic Regression

Logistic regression

- A discriminative model:

$$p(y|\mathbf{x}) = \frac{\exp \sum_{m=1..M} \lambda_m . f_m(y, \mathbf{x})}{\sum_{y' \in Y} \exp \sum_{m=1..M} \lambda_m . f_m(y, \mathbf{x})}$$

Logistic regression

- A discriminative model:

$$p(y|\mathbf{x}) = \frac{\exp \sum_{m=1..M} \lambda_m . f_m(y, \mathbf{x})}{\sum_{y' \in Y} \exp \sum_{m=1..M} \lambda_m . f_m(y, \mathbf{x})}$$

- Not factorized into a product of conditional distributions, but a product of arbitrary functions.

- A discriminative model:

$$p(y|\mathbf{x}) = \frac{\exp \sum_{m=1..M} \lambda_m . f_m(y, \mathbf{x})}{\sum_{y' \in Y} \exp \sum_{m=1..M} \lambda_m . f_m(y, \mathbf{x})}$$

- Not factorized into a product of conditional distributions, but a product of arbitrary functions.
- Linear classifier:

$$\log p(y|\mathbf{x}) = \sum_{m=1..M} \lambda_m . f_m(y, \mathbf{x})$$

Logistic regression

- Comparision to Naive Bayes classifier:

$$
\begin{aligned}
y_{NB} &= \arg\max_{y \in Y} \Pi_{m=1..M} p(x_m|y).p(y) \\
&= \arg\max_{y \in Y} \mathsf{exp}(\log p(y) + \sum_{m=1..M} \log p(x_m|y)) \\
&= \arg\max_{y \in Y} \mathsf{exp} \sum_{m=1..M} \lambda_m f_m(y, x_m)
\end{aligned}
$$

- Comparision to Naive Bayes classifier:

$$
\begin{aligned}
y_{NB} &= \arg\max_{y \in Y} \Pi_{m=1..M} p(x_m|y).p(y) \\
&= \arg\max_{y \in Y} \mathsf{exp}(\log p(y) + \sum_{m=1..M} \log p(x_m|y)) \\
&= \arg\max_{y \in Y} \mathsf{exp} \sum_{m=1..M} \lambda_m f_m(y, x_m)
\end{aligned}
$$

- Naive Bayes classifier is just an exponential model.

- A discriminative model:

$$p(y|\mathbf{x}) = \frac{\exp \sum_{m=1..M} \lambda_m . f_m(y, \mathbf{x})}{\sum_{y^* \in Y} \exp \sum_{m=1..M} \lambda_m . f_m(y^*, \mathbf{x})}$$

- A two-class case:

$$p(\oplus|\mathbf{x}) = \frac{\exp \sum_{m=1..M} \lambda_m . f_m(y, \mathbf{x})}{1 + \exp \sum_{m=1..M} \lambda_m . f_m(y, \mathbf{x})} p(\ominus|\mathbf{x}) = \frac{1}{1 + \exp \sum_{m=1..M} \lambda_m . f_m(y, \mathbf{x})}$$

- Learning the parameters λ's by an iterative algorithm until the convergence of the consistency constraint.

Maximum Entropy Model

Maximum entropy model

- **Principle of maximum entropy**: the only unbiased assumption is a distribution that is as uniform as possible given the available information.

Maximum entropy model

- **Principle of maximum entropy**: the only unbiased assumption is a distribution that is as uniform as possible given the available information.
- **Maximum entropy model**: the proper probability distribution is the one that maximizes the entropy given the constraints from the training data.

- Conditional entropy:

$$H(y|\mathbf{x}) = -\sum_{(y,\mathbf{x})\in(Y,X)} p(y,\mathbf{x}).\log p(y|\mathbf{x})$$

$$H(y|\mathbf{x}) \approx -\sum_{(y,\mathbf{x})\in(Y,X)} \tilde{p}(\mathbf{x}).p(y|\mathbf{x}).\log p(y|\mathbf{x})$$

Maximum entropy model

- Maximum entropy model:

$$p^*(y|\mathbf{x}) = \arg\max_{p(y|\mathbf{x})\in P} H(y|\mathbf{x})$$

Constraints:

$$E(f_m) = \tilde{E}(f_m)$$

$$\sum_{y\in Y} p(y|\mathbf{x}) = 1$$

Maximum entropy model

- Lagrange function $L(p(y|\mathbf{x}))$:

$$H(y|\mathbf{x}) + \sum_{m=1..M} \lambda_m(E(f_m) - \tilde{E}(f_m)) + \lambda_{M+1}(\sum_{y \in Y} p(y|\mathbf{x}) - 1)$$

- Optimization:

$$\partial L(p(y|\mathbf{x}))/\partial p(y|\mathbf{x}) = 0$$

$$H(y|\mathbf{x}) = - \sum_{(y,\mathbf{x}) \in (Y,X)} \tilde{p}(\mathbf{x}).p(y|\mathbf{x}). \log p(y|\mathbf{x})$$

$$\partial H(y|\mathbf{x})/\partial p(y|\mathbf{x}) = -\tilde{p}(\mathbf{x}).(\log p(y|\mathbf{x}) + 1)$$

Maximum entropy model

- Lagrange function $L(p(y|x))$:

$$H(y|\mathbf{x}) + \sum_{m=1..M} \lambda_m(E(f_m) - \tilde{E}(f_m)) + \lambda_{M+1}(\sum_{y \in Y} p(y|\mathbf{x}) - 1)$$

- Optimization:

$$E(f_m) - \tilde{E}(f_m) = \sum_{(y,x) \in (Y,X)} \tilde{p}(\mathbf{x}).p(y|\mathbf{x}).f_m(y,\mathbf{x}) - \sum_{(y,\mathbf{x}) \in (Y,X)} \tilde{p}(y,\mathbf{x}).f_m(y,\mathbf{x})$$

$$\partial \sum_{m=1..M} \lambda_m(E(f_m) - \tilde{E}(f_m))/\partial p(y|\mathbf{x}) = \sum_{m=1..M} \lambda_m \tilde{p}(\mathbf{x}).f_m(y,\mathbf{x})$$

- Lagrange function $L(p(y|\mathbf{x}))$:

$$H(y|\mathbf{x}) + \sum_{m=1..M} \lambda_m(E(f_m) - \tilde{E}(f_m)) + \lambda_{M+1}(\sum_{y \in Y} p(y|\mathbf{x}) - 1)$$

- Optimization:

$$\partial \lambda_{M+1}(\sum_{y \in Y} p(y|\mathbf{x}) - 1)/\partial p(y|\mathbf{x}) = \lambda_{M+1}$$

Maximum entropy model

- Lagrange function $L(p(y|\mathbf{x}))$:

$$H(y|x) + \sum_{m=1..M} \lambda_m (E(f_m) - \tilde{E}(f_m)) + \lambda_{M+1}(\sum_{y \in Y} p(y|\mathbf{x}) - 1)$$

- Optimization:

$$-\tilde{p}(\mathbf{x}).(\log p(y|\mathbf{x}) + 1) + \sum_{m=1..M} \lambda_m \tilde{p}(\mathbf{x}).f_m(y, \mathbf{x}) + \lambda_{M+1} = 0$$

$$p(y|\mathbf{x}) = \exp \sum_{m=1..M} \lambda_m f_m(y, \mathbf{x}) \exp(\lambda_{M+1}/\tilde{p}(\mathbf{x}) - 1)$$

$$\sum_{y \in Y} p(y|\mathbf{x}) = 1$$

Maximum entropy model

- Lagrange function $L(p(y|\mathbf{x}))$:

$$H(y|\mathbf{x}) + \sum_{m=1..M} \lambda_m (E(f_m) - \tilde{E}(f_m)) + \lambda_{M+1} (\sum_{y \in Y} p(y|\mathbf{x}) - 1)$$

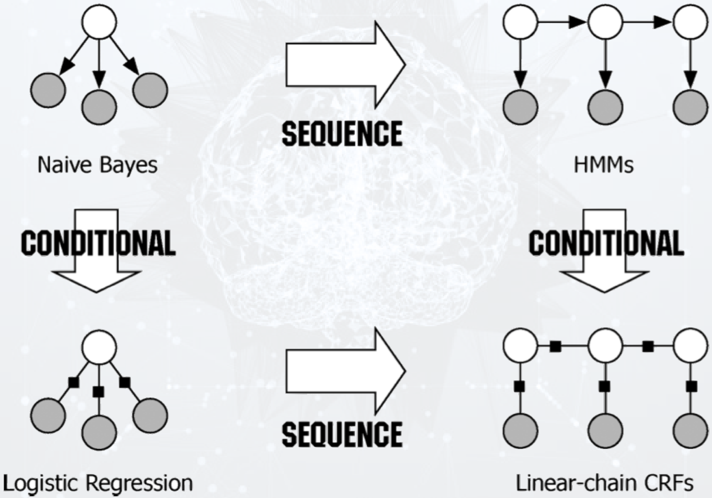
- Optimization:

$$\frac{\partial L(p(y|\mathbf{x}))}{\partial p(y|\mathbf{x})} = 0$$

- Solution:

$$p(y|\mathbf{x}) = \frac{\exp \sum_{m=1..M} \lambda_m . f_m(y, \mathbf{x})}{\sum_{y^* \in Y} \exp \sum_{m=1..M} \lambda_m . f_m(y^*, \mathbf{x})}$$

Conditional Random Field


SEQUENCE
Naive Bayes
HMMs
CONDITIONAL
CONDITIONAL
SEQUENCE
Logistic Regression
Linear-chain CRFs

Conditional Random Fields

- Linear chain CRF:

$$p(\mathbf{y}|\mathbf{x}) = \frac{\Pi_{t=1..T}(\exp \sum_{m=1..M} \lambda_m . f_m(y_t, y_{t-1}, \mathbf{x}_t))}{\sum_{y^* \in Y} \Pi_{t=1..T}(\exp \sum_{m=1..M} \lambda_m . f_m(y^*_t, y^*_{t-1}, \mathbf{x}_t))}$$