Introduction to Machine Learning

Acting under uncertainty: Bayesian decision theory

Nils M. Kriege

WS 2023

Data Mining and Machine Learning
Faculty of Computer Science
University of Vienna

Credit: Slides based on the IML Lectures by Sebastian Tschiatschek and Andreas Krause

Acting under uncertainty

- So far, have seen how we can interpret supervised learning as fitting probabilistic models of the data
- Next, we will see how we can use the estimated models to make decisions

Acting under uncertainty

- Suppose we have estimated a logistic regression model (say, for spam filtering), and obtain $P(Y = \text{spam}|\mathbf{x})$
- Further suppose we have three actions: `Spam`, `NotSpam` and `AskUser`
- ? Which action should we pick?

Cost

Action	Spam	Not spam
`Spam`	0	10
`NotSpam`	1	0
`AskUser`	0.5	0.5

Expected cost

Action	$p = 0.2$	$p = 0.8$
`Spam`		
`NotSpam`		
`AskUser`		

Acting under uncertainty

- Suppose we have estimated a logistic regression model (say, for spam filtering), and obtain $P(Y = \text{spam}|\mathbf{x})$
- Further suppose we have three actions: `Spam`, `NotSpam` and `AskUser`
- ? Which action should we pick?

Cost		
Action	Spam	Not spam
`Spam`	0	10
`NotSpam`	1	0
`AskUser`	0.5	0.5

Expected cost		
Action	$p = 0.2$	$p = 0.8$
`Spam`	8.0	2.0
`NotSpam`	**0.2**	0.8
`AskUser`	0.5	**0.5**

Bayesian decision theory

- Given
  - Conditional distribution over labels $P(y|\mathbf{x})$
  - Set of actions $\mathcal{A}$
  - Cost function $C\colon \mathcal{Y} \times \mathcal{A} \to \mathbb{R}$
- Bayesian Decision Theory recommends to pick the action that minimizes the expected cost:

$$a^* = \operatorname*{argmin}_{a \in \mathcal{A}} \mathbb{E}_y[C(y, a) \mid \mathbf{x}]$$

Bayesian decision theory

- Given
  - Conditional distribution over labels $P(y|\mathbf{x})$
  - Set of actions $\mathcal{A}$
  - Cost function $C\colon \mathcal{Y} \times \mathcal{A} \to \mathbb{R}$
- Bayesian Decision Theory recommends to pick the action that minimizes the expected cost:

$$a^* = \operatorname*{argmin}_{a \in \mathcal{A}} \mathbb{E}_y[C(y, a) \mid \mathbf{x}]$$

- If we had access to the true distribution $P(Y|\mathbf{x})$ this decision implements the Bayesian optimal decision
- In practice, can only estimate it, e.g., (logistic) regression

Recall: Logistic regression

- Learning:
  - Find optimal weights by minimizing logistic loss + regularizer:

$$\hat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w}} \sum_{i=1}^{n} \log\left(1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)\right) + \lambda \|\mathbf{w}\|_2^2$$
$$= \operatorname*{argmax}_{\mathbf{w}} P(\mathbf{w} | \mathbf{x}_1, \ldots, \mathbf{x}_n, y_1, \ldots, y_n)$$

- Classification:
  - Use conditional distribution:

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

  - E.g., predict more likely class label

$$\operatorname*{argmax}_{y} P(y | \mathbf{x}, \hat{\mathbf{w}}) = \operatorname{sign}(\hat{\mathbf{w}}^T \mathbf{x})$$

Optimal decisions for logistic regression

- Est. cond. dist: $\hat{P}(Y \mid \mathbf{x}) = \mathrm{Ber}(y; \sigma(\hat{\mathbf{w}}^T \mathbf{x}))$
- Action set: $\mathcal{A} = \{+1, -1\}$
- Cost function: $C(y, a) = [y \neq a] = \begin{cases} 1 & \text{if } a \neq y, \\ 0 & \text{otherwise} \end{cases}$

Optimal decisions for logistic regression

- Est. cond. dist: $\hat{P}(Y \mid \mathbf{x}) = \text{Ber}(y; \sigma(\hat{\mathbf{w}}^T \mathbf{x}))$
- Action set: $\mathcal{A} = \{+1, -1\}$
- Cost function: $C(y, a) = [y \neq a] = \begin{cases} 1 & \text{if } a \neq y, \\ 0 & \text{otherwise} \end{cases}$
- Then the action that minimizes the expected cost

$$a^* = \operatorname*{argmin}_{a \in \mathcal{A}} \mathbb{E}_y[C(y, a) \mid \mathbf{x}]$$

is the most likely class

$$a^* = \operatorname*{argmax}_{y} \hat{P}(y \mid \mathbf{x}, \hat{\mathbf{w}}) = \text{sign}(\hat{\mathbf{w}}^T \mathbf{x})$$

Asymmetric costs

- Est. cond. dist: $\hat{P}(Y = y \mid \mathbf{x}) = \mathrm{Ber}(y; \sigma(\hat{\mathbf{w}}^T \mathbf{x}))$
- Action set: $\mathcal{A} = \{+1, -1\}$
- Cost function:

$$C(y, a) = \begin{cases} c_{\mathrm{FP}} & \text{if } y = -1 \text{ and } a = +1 \\ c_{\mathrm{FN}} & \text{if } y = +1 \text{ and } a = -1 \\ 0 & \text{otherwise} \end{cases}$$

Asymmetric costs

- Est. cond. dist: $\hat{P}(Y = y \mid \mathbf{x}) = \text{Ber}(y; \sigma(\hat{\mathbf{w}}^T \mathbf{x}))$
- Action set: $\mathcal{A} = \{+1, -1\}$
- Cost function:

$$C(y, a) = \begin{cases} c_{\text{FP}} & \text{if } y = -1 \text{ and } a = +1 \\ c_{\text{FN}} & \text{if } y = +1 \text{ and } a = -1 \\ 0 & \text{otherwise} \end{cases}$$

Let $p = P(Y = +1|\mathbf{x})$. The expected cost of the actions is:

$$C_+ = \mathbb{E}_y[C(y, +1) \mid \mathbf{x}] = (1 - p) \cdot c_{\text{FP}} + p \cdot 0 = (1 - p)c_{\text{FP}}$$
$$C_- = \mathbb{E}_y[C(y, -1) \mid \mathbf{x}] = (1 - p) \cdot 0 + p \cdot c_{\text{FN}} = pc_{\text{FN}}$$

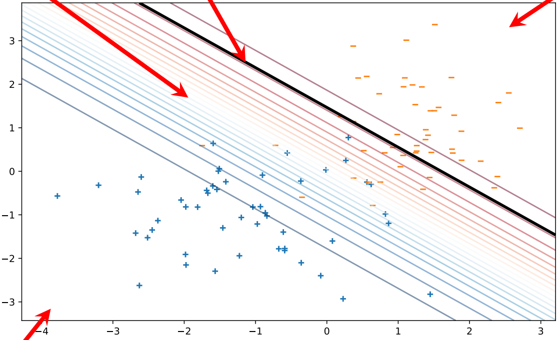
Take action $+1$ if $C_+ < C_- \Leftrightarrow (1 - p)c_{\text{FP}} < pc_{\text{FN}} \Leftrightarrow p > \frac{c_{\text{FP}}}{c_{\text{FP}} + c_{\text{FN}}}$

Demo: Asymmetric costs

decision boundary $\frac{c_{FP}}{c_{FP}+c_{FN}}$

$P(Y = +1|x) = 0.5$

$P(Y = +1|x) \approx 0$



$P(Y = +1|x) \approx 1$

[probabilistic-modeling/lr-demo2.ipynb]

"Doubtful" logistic regression

- Est. cond. dist: $\hat{P}(Y \mid \mathbf{x}) = \mathrm{Ber}(y; \sigma(\hat{\mathbf{w}}^T \mathbf{x}))$
- Action set: $\mathcal{A} = \{+1, -1, D\}$, where D represents doubt
- Cost function:

$$C(y, a) = \begin{cases} [y \neq a] & \text{if } a \in \{+1, -1\} \\ c & \text{if } a = D \end{cases}$$

"Doubtful" logistic regression

- Est. cond. dist: $\hat{P}(Y \mid \mathbf{x}) = \text{Ber}(y; \sigma(\hat{\mathbf{w}}^T \mathbf{x}))$
- Action set: $\mathcal{A} = \{+1, -1, D\}$, where D represents doubt
- Cost function:

$$C(y, a) = \begin{cases} [y \neq a] & \text{if } a \in \{+1, -1\} \\ c & \text{if } a = D \end{cases}$$

- Then the action that minimizes the expected cost

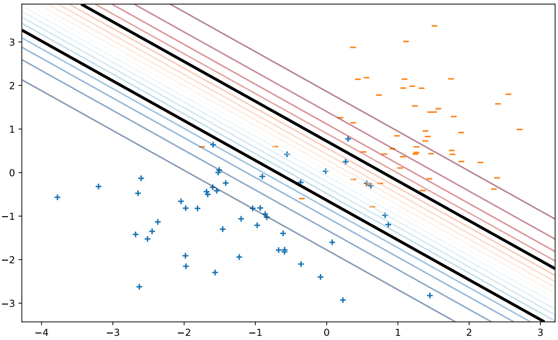
$$a^* = \operatorname*{argmin}_{a \in \mathcal{A}} \mathbb{E}_y[C(y, a) \mid \mathbf{x}]$$

is given by

$$a^* = \begin{cases} y & \text{if } \hat{P}(y \mid \mathbf{x}) \geq 1 - c \\ D & \text{otherwise} \end{cases}$$

- I.e., pick most likely class only if confident enough!

Demo: “Doubtful” logistic regression



[probabilistic-modeling/lr-demo2.ipynb]

Optimal decisions for LS regression

- Est. cond. dist: $\hat{P}(Y \mid \mathbf{x}) = \mathcal{N}(y; \hat{\mathbf{w}}^T\mathbf{x}, \sigma^2)$
- Action set: $\mathcal{A} = \mathbb{R}$
- Cost function: $C(y, a) = (y - a)^2$
- Then the action that minimizes the expected cost

$$a^* = \operatorname*{argmin}_{a \in \mathcal{A}} \mathbb{E}_y[C(y, a) \mid \mathbf{x}] = \operatorname*{argmin}_{a \in \mathcal{A}} \int \hat{P}(Y \mid \mathbf{x}) C(y, a) \, \mathrm{d}y$$

is the conditional mean

$$a^* = \mathbb{E}_y[y \mid \mathbf{x}] = \int \hat{P}(Y \mid \mathbf{x}) y \, \mathrm{d}y = \hat{\mathbf{w}}^T\mathbf{x}$$

Example: Asymmetric cost for regression

- Est. cond. dist: $\hat{P}(Y \mid \mathbf{x}) = \mathcal{N}(y; \hat{\mathbf{w}}^T \mathbf{x}, \sigma^2)$
- Action set: $\mathcal{A} = \mathbb{R}$
- Cost function:

$$C(y, a) = c_1 \underbrace{\max(y - a, 0)}_{\text{Underestimation}} + c_2 \underbrace{\max(a - y, 0)}_{\text{Overestimation}}$$

- Then the action that minimizes the expected cost

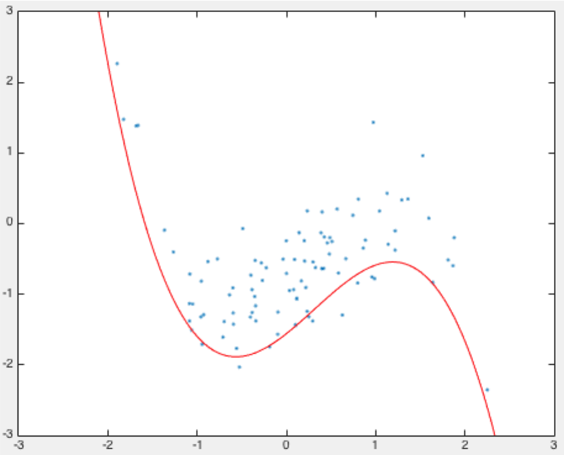
$$a^* = \operatorname*{argmin}_{a \in \mathcal{A}} \mathbb{E}_y[C(y, a) \mid \mathbf{x}]$$

is given by

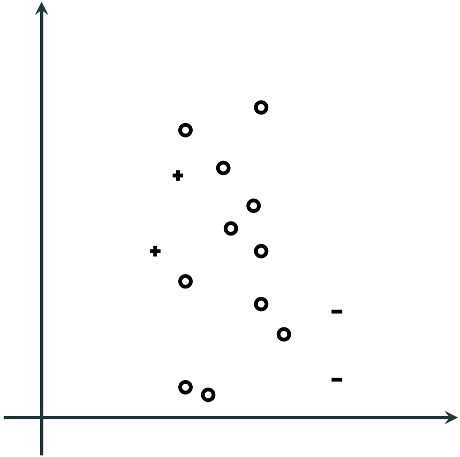
$$a^* = \hat{\mathbf{w}}^T \mathbf{x} + \sigma \Phi^{-1} \left(\frac{c_1}{c_1 + c_2} \right),$$

where Φ^{-1} is the inverse Gaussian CDF.

Demo: Asymmetric cost for regression



Outlook: Active learning



- Labels are expensive (need to ask expert)
- 💡 Want to minimize the number of labels

Uncertainty sampling

Simple strategy: Always pick the example that we are most uncertain about

- Given $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, estimate $\hat{P}(y|x)$
- For every unlabeled $\mathbf{x}_j$: $\hat{P}(y_j = +1|\mathbf{x}_j) = p_j$
- Uncertainty score $u_j = f(p_j)$, where f is a function with $f(0.5)$ maximum
- Pick point x_{j*} with $j^* = \text{argmax}_j\, u_j$

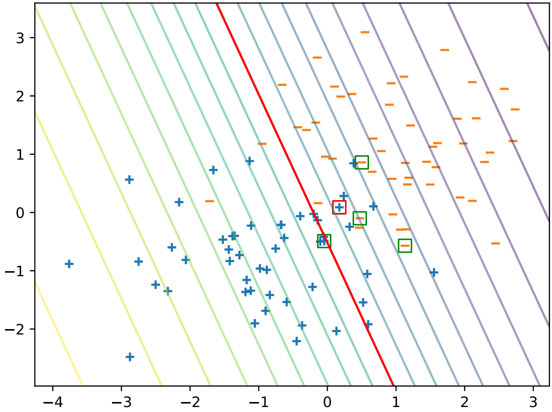
Uncertainty sampling

- **Given:** Pool of unlabeled examples $\mathcal{D}_X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$
- Also maintain labeled data set $\mathcal{D}$, initially empty
- For $t = 1, 2, 3, \ldots$
  - Estimate $\hat{P}(Y_i \mid \mathbf{x}_i)$ given current data $\mathcal{D}$
  - Pick unlabeled example that we are most uncertain about

$$i_t \in \operatorname*{argmin}_i |0.5 - \hat{P}(Y_i \mid \mathbf{x}_i)|$$

  - Query label y_i and set $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{x}_i, y_i)\}$

Demo: Uncertainty sampling



[probabilistic-modeling/lr-demo2.ipynb]

Further comments

- Active learning violates i.i.d. assumption!
- Can get stuck with bad models
- More advanced selection criteria available
  - E.g.: query point that reduces uncertainty of other points as much as possible

Deriving decision rules

- Bayesian decision theory provides a principled way to derive decision rules from conditional distributions $P(Y \mid \mathbf{x})$
- Standard rules arise as special cases:
  - Linear regression: $\hat{\mathbf{w}}^T\mathbf{x}$
  - Logistic regression: $\text{sign}(\hat{\mathbf{w}}^T\mathbf{x})$
- Can accommodate more complex settings
  - "Doubt" (i.e., requiring sufficient confidence)
  - Asymmetric losses
  - Active learning
  - ...

Summary: Learning through MAP inference

- Start with statistical assumptions on data:
  Data points modeled as iid (can be relaxed)
- Choose likelihood function
  - Examples: Gaussian, student-t, logistic, exponential, ...
    $\Rightarrow$ loss function
- Choose prior
  - Examples: Gaussian, Laplace, exponential, ...
    $\Rightarrow$ regularizer
- Optimize for MAP parameters
- Choose hyperparameters (i.e., variance, etc.) through cross-validation
- Make predictions via Bayesian Decision Theory

What you should be able to do

- Understand and apply logistic regression and its variants
- Relate logistic regression and Perceptron/SVM
- Derive MAP estimation problems for different priors and likelihood functions
- Solve resulting optimization problems by applying gradient descent
- Derive decision rules from cost functions via Bayesian decision theory
- Apply uncertainty sampling for binary classification