Chapter (6-7)-2: Advanced Classification Methods: Extra

Richard Liu

May 3, 2020

Contents

Source

- Zhihua Zhou, Machine Learning
- CS61B, University of California, Berkeley

Section 1

K-D tree: Basic Examples and Illustration

KNN Graph

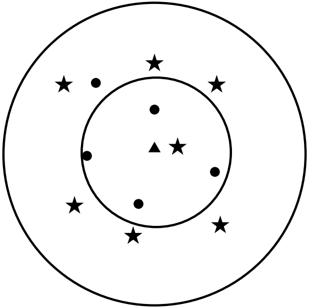

Figure: Graph of KNN

K-D Tree Graph

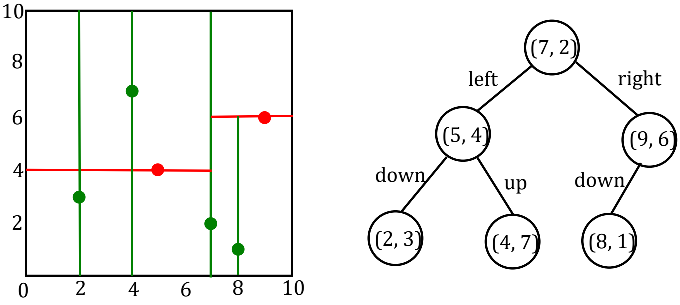

Figure: An illustration of K-D tree

- It divides the plane into different parts, first left/right, then up/down, then left/right, ...

Other details

- Different inserting orders will lead to different trees. Sometimes the tree needs to be balanced.
- In some cases, the trees could be pruned.
- Compared with ordinary methods (just enumerate with time complexity $\mathcal{O}(n)$, K-D tree could provide a better performance if $k << n$ (Why?).
- Python Code: Here

Section 2

SVM: An Application of KKT Conditions

Mathematical Foundation of SVM Proof

Suppose that the general nonlinear programming problem is

$$\min_{x} f(x)$$

$$c_i(x) = 0, i \in \mathcal{E}, c_i(x) \leq 0, i \in \mathcal{I}$$

Mathematical Foundation of SVM Proof

Theorem 1: Karush-Kuhn-Tucker Conditions

Suppose that x^* is a local minimizer, that the funciton f and c_i are continuously differentiable, and that x^* is a regular point. Then there is a Lagrange multiplier vector λ^* with components $\lambda_i^*, i \in \mathcal{E} \cup \mathcal{I}$ such that the following conditions are satisfied

$$\nabla_x \mathcal{L}(x^*, \lambda^*) = 0$$

$$c_i(x^*) = 0, i \in \mathcal{E}, c_i(x^*) \geq 0, i \in \mathcal{I}$$

$$\lambda_i^* \geq 0, i \in \mathcal{I}, \lambda_i^* c_i(x^*) = 0, i \in \mathcal{I}$$

Section 3

Adaboost: Mathematical Principles

Pseudo-Code of Adaboost

输入: 训练集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$;
基学习算法 $\mathfrak{L}$;
训练轮数 T.

过程:

$\mathcal{D}_1(\boldsymbol{x}) = 1/m.$

for $t = 1, 2, \ldots, T$ **do**

$h_t = \mathfrak{L}(D, \mathcal{D}_t);$

$\epsilon_t = P_{\boldsymbol{x} \sim \mathcal{D}_t}(h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}));$

if $\epsilon_t > 0.5$ **then break**

$\alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right);$

$$\mathcal{D}_{t+1}(\boldsymbol{x}) = \frac{\mathcal{D}_t(\boldsymbol{x})}{Z_t} \times \begin{cases} \exp(-\alpha_t), & \text{if } h_t(\boldsymbol{x}) = f(\boldsymbol{x}) \\ \exp(\alpha_t), & \text{if } h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}) \end{cases}$$

$$= \frac{\mathcal{D}_t(\boldsymbol{x}) \exp(-\alpha_t f(\boldsymbol{x}) h_t(\boldsymbol{x}))}{Z_t}$$

end for

输出: $H(\boldsymbol{x}) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x}) \right)$

Figure: Adaboost Algorithm

Problem 1: Why such classifier?

- Consider a binary classification problem with $f(\mathbf{x})$ the true answer and $y_i = \{1, -1\}$.
- Note that we could write the whole classifier as

$$H(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$$

where $\{h_t(x)\}, t = 1, 2, \cdots, T$ are base classifiers. Then we could optimize exponential loss function

$$l_{\exp}(H|\mathcal{D}) = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[e^{-f(\mathbf{x})H(\mathbf{x})}]$$

Explanation

It is easy to see $f(\mathbf{x})H(\mathbf{x})$ will be maximized when $H(\mathbf{x})$ achieves 100% accuracy.

Problem 1: Why such classifier?

Take derivatives w.r.t. $H(\mathbf{x})$ yields

$$\frac{\partial \ell_{\exp}(H|\mathcal{D})}{\partial H(\boldsymbol{x})} = -e^{-H(\boldsymbol{x})}P(f(\boldsymbol{x}) = 1|\boldsymbol{x}) + e^{H(\boldsymbol{x})}P(f(\boldsymbol{x}) = -1|\boldsymbol{x})$$

F.O.C could get

$$H(\boldsymbol{x}) = \frac{1}{2}\ln\frac{P(f(x) = 1|\boldsymbol{x})}{P(f(x) = -1|\boldsymbol{x})}$$

Explanation

What is the definition of expectation with discrete cases?

Problem 1: Why such classifier?

That is to say

$$\operatorname{sign}(H(\boldsymbol{x})) = \operatorname{sign}\left(\frac{1}{2}\ln\frac{P(f(x)=1|\boldsymbol{x})}{P(f(x)=-1|\boldsymbol{x})}\right)$$

$$= \underset{y\in\{-1,1\}}{\arg\max}\, P(f(x)=y|\boldsymbol{x})$$

This means it achieves Bayesian optimal error rate, which means any other models could not lead to better results. In other words, we need to focus on $H(\mathbf{x})$.

Problem 2: How to choose α_i?

In Adaboost, we use greedy algorithm, which means we only consider the t-th base classifier. In fact, we could write the formula as

$$\begin{aligned}
\ell_{\exp}\left(\alpha_t h_t | \mathcal{D}_t\right) &= \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_t}\left[e^{-f(\boldsymbol{x})\alpha_t h_t(\boldsymbol{x})}\right] \\
&= \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_t}\left[e^{-\alpha_t}\mathbb{I}\left(f(\boldsymbol{x}) = h_t(\boldsymbol{x})\right) + e^{\alpha_t}\mathbb{I}\left(f(\boldsymbol{x}) \neq h_t(\boldsymbol{x})\right)\right] \\
&= e^{-\alpha_t}P_{\boldsymbol{x}\sim\mathcal{D}_t}\left(f(\boldsymbol{x}) = h_t(\boldsymbol{x})\right) + e^{\alpha_t}P_{\boldsymbol{x}\sim\mathcal{D}_t}\left(f(\boldsymbol{x}) \neq h_t(\boldsymbol{x})\right) \\
&= e^{-\alpha_t}\left(1 - \epsilon_t\right) + e^{\alpha_t}\epsilon_t
\end{aligned}$$

Take derivative of α_i could get the final result

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$

Problem 3: How to choose right $\mathcal{D}$?

- Ideally, we want every distribution to minimize the loss function. In this case, we want to minimize $\ell_{\exp}\left(H_{t-1}+h_t|\mathcal{D}\right)$ at time t, so we have

$$\begin{aligned}\ell_{\exp}\left(H_{t-1}+h_t|\mathcal{D}\right) &= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[e^{-f(\mathbf{x})(H_{t-1}(\mathbf{x})+h_t(\mathbf{x}))}\right]\\ &= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[e^{-f(\mathbf{x})H_{t-1}(\mathbf{x})}e^{-f(\mathbf{x})h_t(\mathbf{x})}\right]\end{aligned}$$

$$\begin{aligned}&\simeq \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[e^{-f(\mathbf{x})H_{t-1}(\mathbf{x})}\left(1-f(\mathbf{x})h_t(\mathbf{x})+\frac{f^2(\mathbf{x})h_t^2(\mathbf{x})}{2}\right)\right]\\ &= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[e^{-f(\mathbf{x})H_{t-1}(\mathbf{x})}\left(1-f(\mathbf{x})h_t(\mathbf{x})+\frac{1}{2}\right)\right]\end{aligned}$$

Explanation

Note that $f^2(\mathbf{x})=h_t^2(\mathbf{x})=1$.

Problem 3: How to choose right $\mathcal{D}$?

- Because we only consider the time t, the objective goal is to maximize

$$\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[e^{-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})}f(\boldsymbol{x})h(\boldsymbol{x})\right]$$

- Consider (Why?)

$$\mathcal{D}_t(\boldsymbol{x}) = \frac{\mathcal{D}(\boldsymbol{x})e^{-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})}}{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[e^{-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})}\right]}$$

we will get

$$\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[e^{-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})}f(\boldsymbol{x})h(\boldsymbol{x})\right] = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_t}[f(\boldsymbol{x})h(\boldsymbol{x})]$$

Problem 3: How to choose right $\mathcal{D}$?

For we have

$$f(\boldsymbol{x})h(\boldsymbol{x}) = 1 - 2\mathbb{I}(f(\boldsymbol{x}) \neq h(\boldsymbol{x}))$$

So the optimal base classifier at time t is

$$h_t(\boldsymbol{x}) = \arg\min_h \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_t}[\mathbb{I}(f(\boldsymbol{x}) \neq h(\boldsymbol{x}))]$$

This means we could get the optimal solution with that $\mathcal{D}$

Problem 3: How to choose right $\mathcal{D}$?

At last, by definition of $\mathcal{D}_t$, we have

$$
\begin{aligned}
\mathcal{D}_{t+1}(\boldsymbol{x}) &= \frac{\mathcal{D}(\boldsymbol{x}) e^{-f(\boldsymbol{x})H_t(\boldsymbol{x})}}{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[e^{-f(\boldsymbol{x})H_t(\boldsymbol{x})}\right]} \\
&= \frac{\mathcal{D}(\boldsymbol{x}) e^{-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})} e^{-f(\boldsymbol{x})\alpha_t h_t(\boldsymbol{x})}}{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[e^{-f(\boldsymbol{x})H_t(\boldsymbol{x})}\right]} \\
&= \mathcal{D}_t(\boldsymbol{x}) \cdot e^{-f(\boldsymbol{x})\alpha_t h_t(\boldsymbol{x})} \frac{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[e^{-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})}\right]}{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[e^{-f(\boldsymbol{x})H_t(\boldsymbol{x})}\right]}
\end{aligned}
$$

So we get the updated formula.

Section 4

Bagging versus Boosting: Other Properties

- The goal of bagging is to make each base classifier achieve a better performance because the data could be used in more than one classifier.
- Bagging: Lower Variance. Boosting: Lower Bias.
- Random Forest is an extension of bagging, exploiting turbulence of features, so it usually behaves better than bagging.

Thank you!