MUSTERERKENNUNG, HINGE LOSS

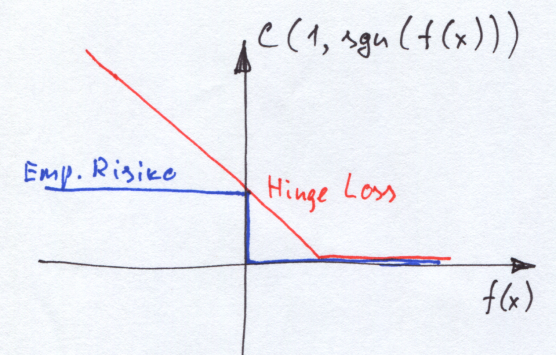
Zur Erinnerung: Eine Entscheidungsstrategie ist eine Abbildung $e : X \to Y$, die den Inputraum X in die Entscheidungsraum Y überführt. Mit $e(x) \in Y$ wird die Entscheidung für einen konkreten $x \in X$ bezeichnet. Gegeben sei eine Kostenfunktion $C : Y \times Y \to \mathbb{R}$, die zwei Entscheidungen vergleicht (eine „Strafe“ zuordnet). Das Problem der Suche nach der besten Entscheidungsstrategie wird wie folgt formuliert. Gegeben sei eine vollständig klassifizierte Lernstichprobe $L = \big((x_1, y_1), (x_2, y_2) \dots (x_l, y_l)\big)$. Das Empirische Risiko einer Entscheidungsstrategie e ist die auf der Lernstichprobe berechnete Strafe, d.h.

$$R(e) = \sum_l C\big(y_l, e(x_l)\big). \tag{1}$$

Die Aufgabe besteht in der Suche nach der Strategie (aus einer Klasse), die das Risiko minimiert.

Im weiteren wird der folgende Spezialfall betrachtet:

- die Menge der Entscheidungen ist $Y = \{-1, +1\}$, d.h. zwei Klassen;
- die Kostenfunktion ist die einfache Deltafunktion $C(y, y') = \mathbb{1}(y \neq y')$;
- die Entscheidungsstrategie lässt sich als $e(x) = \operatorname{sgn}(f(x))$ schreiben, wobei $f(x)$ eine „Evaluierungsfunktion“ $f : X \to \mathbb{R}$ ist, deren Vorzeichen am Ende der Klasse entspricht. Zum Beispiel entspricht $f(x) = \langle x, w \rangle - b$ einem linearen Klassifikator.



Das Problem bei der Minimierung von (1) besteht darin, dass die Zielfunktion nicht konvex ist. Das wird umgegangen, indem sie durch eine konvexe obere Schranke wie folgt ersetzt wird:

$$C(y, \operatorname{sgn}(f(x))) \leq \max(0, 1 - y \cdot f(x)) \tag{2}$$

Diese obere Schranke nennt man „Hinge Loss“. In der Abbildung oben ist die Vorgehensweise schematisch für $y = 1$ dargestellt (für $y = -1$ soll die Abbildung um die vertikale Achse gespiegelt werden).

Sei θ der unbekannte Parameter der gesuchten Entscheidungsstrategie, d.h. die Evaluierungsfunktion ist $f(x;\theta)$ (z.B. ist θ der Gewichtsvektor w bei den linearen Klassifikatoren). Die zu lösende Optimierungsaufgabe lautet dann

$$H(\theta) = \sum_l \max(0, 1 - y_l \cdot f(x_l;\theta)) \to \min_\theta . \tag{3}$$

Diese Aufgabe ist konvex aber nicht überall differenzierbar. Zur Lösung wird oft der Subgradientenalgorithmus verwendet (für Details siehe andere Quellen sowie Internet). Speziell für (3) ergibt sich das folgende iterative Schema:

1) Es werden Datenpunkte ermittelt, deren aktueller Hinge Loss größer Null ist:

$$L' = \{x_l : \ y_l f(x_l) < 1\}; \tag{4}$$

2) Der Subgradient ergibt sich somit als

$$\frac{\partial H}{\partial \theta} = - \sum_{x_l \in L'} y_l \frac{\partial f(x_l;\theta)}{\partial \theta}. \tag{5}$$

Beispiel: Handelt es sich um lineare Klassifikatoren, so ist der Subgradient

$$\frac{\partial H}{\partial w} = - \sum_{x_l \in L'} y_l x_l. \tag{6}$$

Das heißt, dass in jedem Schritt gewisse Datenpunkte zum Gewichtsvektor addiert werden. Dies erinnert an Perzeptron-Algorithmus.

Alles oben gesagte lässt sich auf Merkmalsräume verallgemeinern (siehe Vorlesung „Mustererkennung: Support Vector Machines“). Man erinnere, dass die Evaluierungsfunktion in diesem Fall wie folgt spezifiziert werden kann:

$$f(x;\alpha) = \sum_i \alpha_i y_i \kappa(x, x_i) \tag{7}$$

mit den Support Vektoren x_i und dem Kernel $\kappa(x, x')$. Die unbekannten Parameter sind dabei die Koeffizienten α_i. Der Hinge Loss ist somit

$$H(\alpha) = \sum_l \max\bigl(0, 1 - y_l \sum_i \alpha_i y_i \kappa(x_l, x_i)\bigr), \tag{8}$$

und der Subgradient ist

$$\frac{\partial H}{\partial \alpha} = - \sum_{x_l \in L'} y_l y_i \kappa(x_l, x_i). \tag{9}$$

Wie es bei den Kernels üblich ist, werden weder der Merkmalsraum $\mathscr{H}$, noch die Abbildung $\Phi : X \to \mathscr{H}$ explizit benötigt, wenn die Kernelfunktion $\kappa(x, x')$ angegeben ist.