Large Margin Methods for Structured and interdependent Output Variables

Hendrik Blom

TU Dortmund

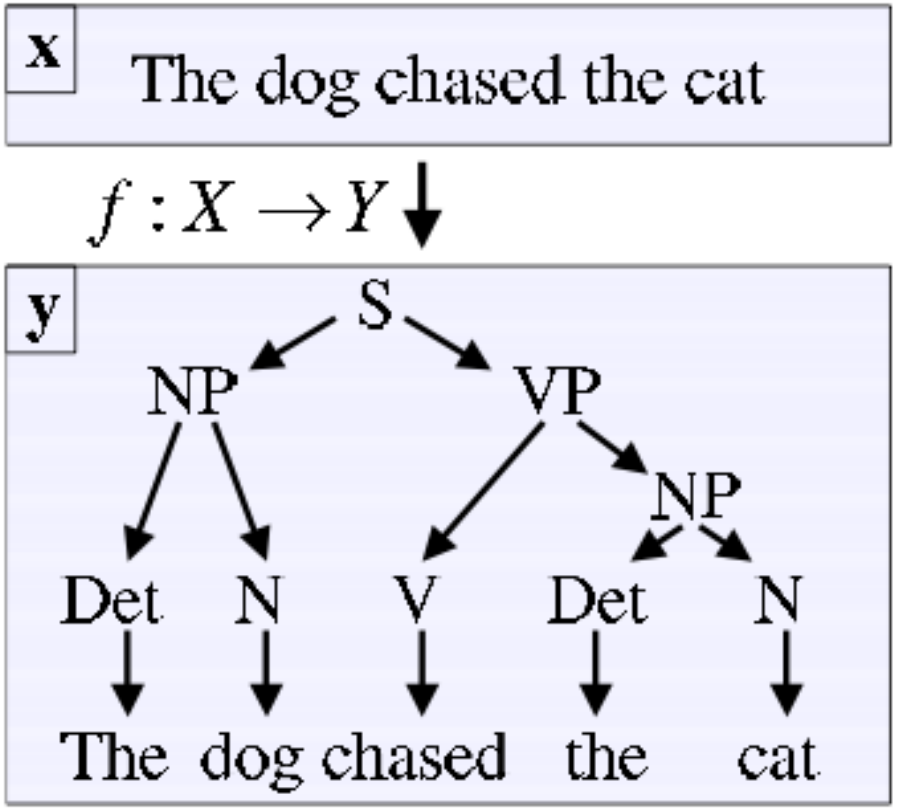
16. Juni 2009

Übersicht

Vortrag

1. Lernaufgabe
2. Primales Problem
3. Duales Problem
4. Optimierungsalgorithmus der SVM
5. Spezielle Ausprägungen und Anwendungen

Einleitendes Beispiel



Lernaufgabe

Geg.:
- Eingabevektoren oder - muster $x = (x_1, \ldots, x_k) \in \mathcal{X}$
- Ausgabevariablen $y = (y_1, \ldots, y_q) \in \mathcal{Y}$
- Eine Trainingsmenge $(x_1, y_1), \ldots, (x_n, y_n) \subset \mathcal{X} \times \mathcal{Y}$

Ziel: Finde eine Diskriminantenfunktion $F : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, so dass

$$f(x; w) = \arg\max_{y \in \mathcal{Y}} F(x, y; w), \text{ mit}$$

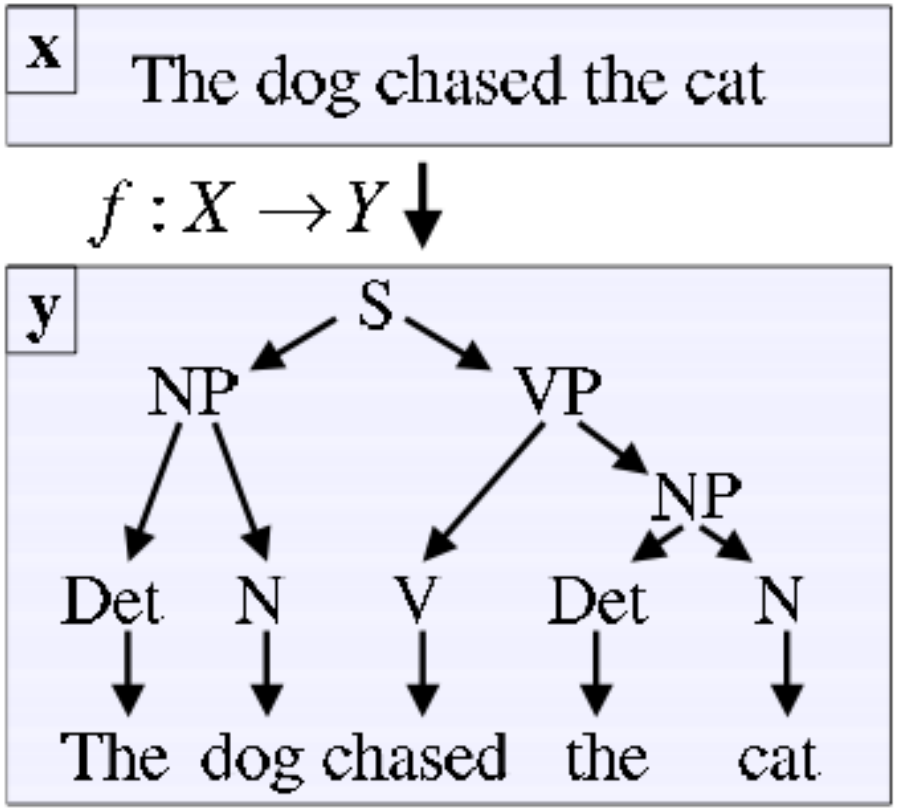
$$F = \langle w, \Psi(x, y) \rangle$$

Diskriminantenfunktion F

$$f(x, w) = \arg\max_{y \in \mathcal{Y}} F(x, y; w)$$

- F kann als w-parametrisierte Familie von Kompabilitätsfunktionen oder $-F$ als Familie von Kostenfunktionen verstanden werden
- Eine Vorraussage kann durch die Maximierung von F getroffen werden
- Finde also das w^*, so dass $f(x; w)$ das am besten zu x passende $y \in \mathcal{Y}$ liefert

Feature map Ψ: Beispiel



$$\Psi(\mathbf{x},\mathbf{y}) = \begin{pmatrix} 1 & S \rightarrow NP\,VP \\ 0 & S \rightarrow NP \\ 2 & NP \rightarrow Det\ N \\ 1 & VP \rightarrow V\ NP \\ \vdots & \\ 0 & Det \rightarrow dog \\ 2 & Det \rightarrow the \\ 1 & N \rightarrow dog \\ 1 & V \rightarrow chased \\ 1 & N \rightarrow cat \end{pmatrix}$$

$$F = \langle w, \Psi(x, y) \rangle$$

Verlustfunktion und Risikominiemierung

Verlustfunktion Δ

$\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ quantifiziert den Fehler. Es gilt:
$\Delta(y, y) = 0, \Delta(y, y') > 0$ mit $y \neq y'$ und für $y^* \exists \max_y \{y^*, y\}$

Risiko $\mathcal{R}^{\Delta}_{\mathcal{P}}(f)$

$$\mathcal{R}^{\Delta}_{\mathcal{P}}(f) = \int_{\mathcal{X}\times\mathcal{Y}} \Delta(y, f(x)) dP(x, y)$$

Empirisches Risiko $\mathcal{R}^{\Delta}_{\mathcal{S}}(f)$

$$\mathcal{R}^{\Delta}_{\mathcal{S}}(f) = \frac{1}{n} \sum_{i=1}^{n} \Delta(y_i, f(x_i))$$

Beschränkung des Risikos

$$\mathcal{R}^{\Delta}_{\mathcal{P}}(f) \leq \mathcal{R}^{\Delta}_{\mathcal{S}}(f) + \sqrt{\frac{h(\log(2l/h)+1) - log(\eta/4)}{l}}$$

Übersicht

Vortrag

1. Lernaufgabe
2. Primales Problem
3. Duales Problem
4. Optimierungsalgorithmus der SVM
5. Spezielle Ausprägungen und Anwendungen

SVM_0 : Separierbarer Fall

Herleitung der Nebenbedingungen

Annahme: Es existiert kein Trainingsfehler

$$\Rightarrow \forall i \in \{1, \ldots, n\} : max_{y \in \mathcal{Y} \setminus y_i} \{\langle w, \Psi(x_i, y) \rangle\} \leq \langle w, \Psi(x_i, y_i) \rangle$$

$$\Leftrightarrow \forall i \in \{1, \ldots, n\}, \forall y \in \mathcal{Y} \setminus y_i : \langle w, \Psi(x_i, y_i) - \Psi(x_i, y)) \rangle \geq 0$$

wir definieren: $\delta\Psi_i(y) \equiv \Psi(x_i, y_i) - \Psi(x_i, y)$

SVM_0

$$\text{SVM}_0 : \min_w \frac{1}{2} \parallel w \parallel^2$$

$$\text{u. d. Nb. } \forall i, \forall y \in \mathcal{Y} \setminus y_i : \langle w, \delta\Psi_i(y) \rangle \geq 1$$

Wenn es mehr als eine Lösung gibt, wähle die Lösung bei der die zweitbeste Lösung am weitesten Weg liegt.

SVM_1 und SVM_2 : Soft-Margin Maximization

SVM_1

$$\min_{w,\xi} \frac{1}{2} \parallel w \parallel^2 + \frac{C}{n} \sum_{i=1}^{n} \xi_i$$

u. d. Nb. $\forall i, \forall y \in \mathcal{Y} \setminus y_i : \langle w, \delta\Psi_i(y) \rangle \geq 1 - \xi_i, \xi_i \geq 0$

SVM_2

$$\min_{w,\xi} \frac{1}{2} \parallel w \parallel^2 + \frac{C}{2n} \sum_{i=1}^{n} \xi_i^2$$

u. d. Nb. $\forall i, \forall y \in \mathcal{Y} \setminus y_i : \langle w, \delta\Psi_i(y) \rangle \geq 1 - \xi_i$

In beiden Fällen kontrolliert die Konstante $C > 0$ den Trade-off zwischen Trainingsfehler und Margin-Maximierung.

$\text{SVM}_1^{\Delta s}$ und $\text{SVM}_2^{\Delta s}$: Slack Re-Scaling

$\text{SVM}_1^{\Delta s}$

$$\min_{w,\xi} \frac{1}{2} \parallel w \parallel^2 + \frac{C}{n} \sum_{i=1}^{n} \xi_i$$

u. d. Nb. $\forall i, \forall y \in \mathcal{Y} \setminus y_i : \langle w, \delta\Psi_i(y) \rangle \geq 1 - \frac{\xi_i}{\Delta(y_i, y)}$

$\text{SVM}_2^{\Delta s}$

$$\min_{w,\xi} \frac{1}{2} \parallel w \parallel^2 + \frac{C}{2n} \sum_{i=1}^{n} \xi_i^2$$

u. d. Nb. $\forall i, \forall y \in \mathcal{Y} \setminus y_i : \langle w, \delta\Psi_i(y) \rangle \geq 1 - \frac{\xi_i}{\sqrt{\Delta(y_i, y)}}$

$\text{SVM}_1^{\Delta s}$ und $\text{SVM}_2^{\Delta s}$: obere Schranke des empirischen Risikos

Proposition 1

Sei durch $\xi^*(w)$ die optimale Lösung der Slack-Variablen in $\text{SVM}_1^{\Delta s}$ für gegebenen Gewichtsvektor w bezeichnet. Dann ist mit $\frac{1}{n}\sum_{i=1}^{n}\xi_i^*$ eine obere Grenze für das empirische Risiko $\mathcal{R}_{\mathcal{S}}^{\Delta}(w)$ gegeben.

Beweis.

Es ist $\xi_i^* = \max\left\{0, \max_{y \neq y_i}\{\Delta(y_i, y)(1 - \langle w, \delta\Psi_i(y)\rangle)\}\right\}$

1. Fall: Wenn $f(x_i; w) = y_i$, dann ist $\xi_i^* \geq 0 = \Delta(y_i, f(x_i; w))$ und damit trivialer weise nach oben beschränkt.

2. Fall: Wenn $\hat{y} \equiv f(x_i; w) \neq y_i$, dann $\langle w, \delta\Psi_i(\hat{y})\rangle \leq 0$ und damit $\frac{\xi_i^*}{\Delta(y_i, y)} \geq 1$ was äquivalent zu $\xi_i^* \geq \Delta(y_i, y)$ ist. □

$\text{SVM}_1^{\Delta m}$ und $\text{SVM}_2^{\Delta m}$: Margin Re-Scaling

$\text{SVM}_1^{\Delta m}$

$$\min_{w,\xi} \frac{1}{2} \parallel w \parallel^2 + \frac{C}{n} \sum_{i=1}^{n} \xi_i$$

u. d. Nb. $\forall i, \forall y \in \mathcal{Y} : \langle w, \delta\Psi_i(y) \rangle \geq \Delta(y_i, y) - \xi_i$

$\text{SVM}_2^{\Delta m}$

$$\min_{w,\xi} \frac{1}{2} \parallel w \parallel^2 + \frac{C}{n} \sum_{i=1}^{n} \xi_i^2$$

u. d. Nb. $\forall i, \forall y \in \mathcal{Y} : \langle w, \delta\Psi_i(y) \rangle \geq \sqrt{\Delta(y_i, y)} - \xi_i$

$\text{SVM}_1^{\Delta m}$ und $\text{SVM}_2^{\Delta m}$: obere Schranke des empirischen Risikos

Proposition 2

Sei durch $\xi^*(w)$ die optimale Lösung der Slack-Variablen in $\text{SVM}_1^{\Delta m}$ für gegebenen Gewichtsvektor w bezeichnet. Dann ist mit $\frac{1}{n}\sum_{i=1}^{n}\xi_i^*$ eine obere Grenze für das empirische Risiko $\mathcal{R}_s^{\Delta}(w)$ gegeben.

Beweis.

Es ist $\xi_i^* = \max\{0, \max_y\{\Delta(y_i, y) - \langle w, \delta\Psi_i(y)\rangle\}\}$, was garantiert $\Delta(y_i, y)$ für y mit $\langle w, \delta\Psi_i(y)\rangle) \leq 0$ nach oben beschränkt.

□

Slack Re-Scaling vs. Margin Re-Scaling

Proposition 3

Angenommen $\Delta' \equiv \eta\Delta$ mit $\eta > 0$. Dann sind $\text{SVM}_1^{\Delta s}(C)$ und $\text{SVM}_1^{\Delta' s}(C')$ mit $C' = \frac{C}{\eta}$ äquivalent bzgl. w. Es ist sogar der optimale Gewichtsvektor w^* in beiden Fällen gleich.

$\Rightarrow$ Das Slack Re-scaling ist invariant bzgl. einer Skalierung der Verlustfunktion Δ.

- Im Gegensatz dazu ist Margin Re-scaling nicht invariant bzgl. einer Skalierung der Verlustfunktion Δ, den die Featuremap $\Psi(x, y)$ müsste auch skaliert werden.
- Ein weiterer Nachteil ist, dass das Margin Re-Scaling möglicherweise schlechten y ein hohes Gewicht gibt, denn jeder Anstieg des Verlusts verbreitert die erforderliche Margin.

Übersicht

Vortrag

1. Lernaufgabe
2. Primales Problem
3. Duales Problem
4. Optimierungsalgorithmus der SVM
5. Spezielle Ausprägungen und Anwendungen

Duales Problem

Duales Problem der Standard SVM

$$L_D(\alpha) = \Theta(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle$$

Bemerkungen

- y_i, y_j sind nicht mehr einfache Werte, sondern Strukturen.
- Da es nun $n|\mathcal{Y}| - n$ Nebenbedingungen gibt, existieren auch so viele α. Es wird für jedes Paar aus $\mathcal{X} \times \mathcal{Y}$ ein α benötigt.

SVM_0: Duales Problem

SVM_0

$$\alpha^* = \arg\max_{\alpha} \Theta(\alpha)$$

$$\Theta(\alpha) \equiv -\frac{1}{2} \sum_{i,y \neq y_i} \sum_{j,\bar{y} \neq y_j} \alpha_{(iy)} \alpha_{(j\bar{y})} J_{(iy)(j\bar{y})} + \sum_{i,y \neq y_i} \alpha_{(iy)},$$

u. d. N. $\alpha \geq 0$

wobei $J_{(iy)(j\bar{y})} = \langle \delta\Psi_i(y), \delta\Psi_j(\bar{y}) \rangle$

Herleitung duales Problem

1.) Bilde primale Lagrange-Funktion

$$\Theta_p = \tfrac{1}{2} \parallel w \parallel^2 - \sum_{i}^{|\mathcal{Y}|} \sum_{y \neq y_i} \alpha_{(iy)} (\langle w, \delta\Psi_i(y) \rangle - 1)$$

2.) Ersetze w in Θ_p durch $w^*(\alpha) = \sum_{i}^{|\mathcal{Y}|} \sum_{y \neq y_i} \alpha_{(iy)} \delta\Psi_i(y)$

SVM$_0$: Duales Problem $(J_{(iy)(j\bar{y})})$

$$\mathbf{J_{(iy)(j\bar{y})}} = \langle \delta \mathbf{\Psi_i}(\mathbf{y}), \delta \mathbf{\Psi_j}(\bar{\mathbf{y}}) \rangle$$

- Bei der klassichen SVM wird für $i, j = 1, \ldots, |\mathcal{Y}|$ das Skalarprodukt $\langle x_i, x_j \rangle$ berechnet
- Hier wird für $i, j = 1, \ldots, |\mathcal{Y}|$ das Skalarprodukt $\langle \delta\Psi_i(y), \delta\Psi_j(\bar{y}) \rangle$ berechnet

J	x_1	$\ldots$	x_j	$\ldots$	x_N
x_1	$-$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
x_i	$\ldots$	$\ldots$	$\langle \delta\Psi_i(y), \delta\Psi_j(\bar{y}) \rangle$	$\ldots$	$\ldots$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
x_N	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$-$

- Jeder Eintrag in diese Matrix ist eine $|\mathcal{Y}| \times |\mathcal{Y}|$ Matrix, d.h. J ist ein Tensor!

SVM_0: Duales Problem

SVM_0

$$\alpha^* = \arg\max_{\alpha} \Theta(\alpha)$$

$$\Theta(\alpha) \equiv -\frac{1}{2} \sum_{i,y \neq y_i} \sum_{j,\bar{y} \neq y_j} \alpha_{(iy)} \alpha_{(j\bar{y})} J_{(iy)(j\bar{y})} + \sum_{i,y \neq y_i} \alpha_{(iy)},$$

u. d. N. $\alpha \geq 0$

wobei $J_{(iy)(j\bar{y})} = \langle \delta\Psi_i(y), \delta\Psi_j(\bar{y}) \rangle$

Erklärung des $\alpha_{(iy)}$

$\alpha_{(iy)}$ hat einen Eintrag für jedes $y \neq y_i$, z.B. $\alpha_{(iy)} = \begin{pmatrix} \alpha_{(y_i \neq y_1)} \\ \dots \\ \alpha_{(y_i \neq y_{|\mathcal{Y}|})} \end{pmatrix}$

SVM_1: Duales Problem

SVM_1

$$\alpha^* = \arg\max_{\alpha} \Theta(\alpha)$$

$$\Theta(\alpha) \equiv -\frac{1}{2} \sum_{i, y \neq y_i} \sum_{j, \bar{y} \neq y_j} \alpha_{(iy)} \alpha_{(j\bar{y})} J_{(iy)(j\bar{y})} + \sum_{i, y \neq y_i} \alpha_{(iy)},$$

$$\text{u. d. Nb. } \alpha \geq 0, \sum_{y \neq y_i} \alpha_{(iy)} \leq \frac{C}{n} \forall i = 1, \ldots, n$$

SVM_2: Duales Problem

SVM_2

$$\alpha^* = \arg\max_{\alpha} \Theta(\alpha)$$

$$\Theta(\alpha) \equiv -\frac{1}{2} \sum_{i,y \neq y_i} \sum_{j,\bar{y} \neq y_j} \alpha_{(iy)} \alpha_{(j\bar{y})} J_{(iy)(j\bar{y})} + \sum_{i,y \neq y_i} \alpha_{(iy)},$$

u. d. Nb. $\alpha \geq 0$

$$J_{(iy)(j\bar{y})} = \langle \delta\Psi_i(y), \delta\Psi_j(\bar{y}) \rangle + \delta(i,j)\frac{n}{C}$$

$\text{SVM}_1^{\Delta s}$: Duales Problem

$\text{SVM}_1^{\Delta s}$

$$\alpha^* = \arg\max_{\alpha} \Theta(\alpha)$$

$$\Theta(\alpha) \equiv -\frac{1}{2} \sum_{i,y \neq y_i} \sum_{j,\bar{y} \neq y_j} \alpha_{(iy)} \alpha_{(j\bar{y})} J_{(iy)(j\bar{y})} + \sum_{i,y \neq y_i} \alpha_{(iy)},$$

$$\text{u. d. Nb. } \alpha \geq 0, \sum_{y \neq y_i} \frac{\alpha_{(iy)}}{\Delta(y_i, y)} \leq \frac{C}{n}, \forall i = 1, \ldots, n$$

$\text{SVM}_2^{\Delta s}$: Duales Problem

$\text{SVM}_2^{\Delta s}$

$$\alpha^* = \arg\max_{\alpha} \Theta(\alpha)$$

$$\Theta(\alpha) \equiv -\frac{1}{2} \sum_{i,y \neq y_i} \sum_{j,\bar{y} \neq y_j} \alpha_{(iy)} \alpha_{(j\bar{y})} J_{(iy)(j\bar{y})} + \sum_{i,y \neq y_i} \alpha_{(iy)},$$

u. d. Nb. $\alpha \geq 0$

$$J_{(iy)(j\bar{y})} = \langle \delta\Psi_i(y), \delta\Psi_j(\bar{y}) \rangle + \delta(i,j) \frac{n}{C\sqrt{\Delta(y_i, y)}\sqrt{\Delta(y_j, \bar{y})}}$$

$\text{SVM}_1^{\Delta m}$: Duales Problem

$\text{SVM}_1^{\Delta m}$

$$\alpha^* = \arg\max_{\alpha} \Theta(\alpha)$$

$$\Theta(\alpha) \equiv -\frac{1}{2} \sum_{i,y \neq y_i} \sum_{j,\bar{y} \neq y_j} \alpha_{(iy)} \alpha_{(j\bar{y})} J_{(iy)(j\bar{y})} + \sum_{i,y \neq y_i} \alpha_{(iy)} \Delta(y_i, y),$$

$$\text{u. d. Nb. } \alpha \geq 0, \sum_{y \neq y_i} \alpha_{(iy)} \leq \frac{C}{n}, \forall i = 1, \ldots, n$$

$\text{SVM}_2^{\Delta m}$: Duales Problem

$\text{SVM}_2^{\Delta m}$

$$\alpha^* = \arg\max_{\alpha} \Theta(\alpha)$$

$$\Theta(\alpha) \equiv -\frac{1}{2} \sum_{i,y \neq y_i} \sum_{j,\bar{y} \neq y_j} \alpha_{(iy)} \alpha_{(j\bar{y})} J_{(iy)(j\bar{y})} + \sum_{i,y \neq y_i} \alpha_{(iy)} \sqrt{\Delta(y_i, y)},$$

u. d. Nb. $\alpha \geq 0$

$$J_{(iy)(j\bar{y})} = \langle \delta\Psi_i(y), \delta\Psi_j(\bar{y}) \rangle + \delta(i,j) \frac{n}{C\sqrt{\Delta(y_i, y)}\sqrt{\Delta(y_j, \bar{y})}}$$

Übersicht

Vortrag

1. Lernaufgabe
2. Primales Problem
3. Duales Problem
4. Optimierungsalgorithmus der SVM
5. Spezielle Ausprägungen und Anwendungen

Die SVM stellt ein schwieriges Optimierungsproblem

- Bei $n|\mathcal{Y}| - n$ Nebenbedingungen und vermutlich großem $|\mathcal{Y}|$ eine normale Optimierung durch quadratische Programmierung nicht möglich.
- Idee: Es sollen nur deutlich weniger Nebenbedingungen bearbeitet werden.
- Beobachtung: Es existiert immer eine Teilmenge von Nebenbedingungen, so dass die damit errechnete Lösung auch alle Nebenbedingungen mit einer Ungenauigkeit von nur ϵ erfüllt.

Algorithmus: Idee

- Für jedes Beispiel x_i wird ein working set S_i verwaltet, in dem die verletzen Nebenbedingungen gespeichert werden. Zunächst sind die S_i leer, und damit ist das Problem unbeschränkt.
- Für jedes Beispiel x_i wird die am schlimmsten verletzte Nebenbedingung bzgl. y_i^* festgestellt und S_i hinzugefügt. Damit wird das Problem zunehmend beschränkt.
- In jedem Schritt wird dann α optimiert:
  a. bzgl. $S = \bigcup S_i$
  b. bzgl. des jeweiligen S_i, wobei die anderen $\alpha_{(jy)}$ unverändert bleiben.
- Der Algorithmus terminiert, wenn sich in einer gesamten Iteration über die Beispiele kein S_i mehr verändert.

Algorithmus

Algorithm 1 Algorithm for solving SVM_0 and the loss re-scaling formulations SVM_1^* and SVM_2^*.

Input: $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n), C, \varepsilon$
$S_i \leftarrow \emptyset$ for all $i = 1, \ldots, n$
repeat
  for $i = 1, \ldots, n$ **do**
    / prepare cost function for optimization */*
    set up cost function

$$H(\mathbf{y}) \equiv \begin{cases} 1 - \langle \delta\Psi_i(\mathbf{y}), \mathbf{w} \rangle & (\mathrm{SVM}_0) \\ (1 - \langle \delta\Psi_i(\mathbf{y}), \mathbf{w} \rangle) \triangle(\mathbf{y}_i, \mathbf{y}) & (\mathrm{SVM}_1^{\triangle s}) \\ \triangle(\mathbf{y}_i, \mathbf{y}) - \langle \delta\Psi_i(\mathbf{y}), \mathbf{w} \rangle & (\mathrm{SVM}_1^{\triangle m}) \\ (1 - \langle \delta\Psi_i(\mathbf{y}), \mathbf{w} \rangle) \sqrt{\triangle(\mathbf{y}_i, \mathbf{y})} & (\mathrm{SVM}_2^{\triangle s}) \\ \sqrt{\triangle(\mathbf{y}_i, \mathbf{y})} - \langle \delta\Psi_i(\mathbf{y}), \mathbf{w} \rangle & (\mathrm{SVM}_2^{\triangle m}) \end{cases}$$

    where $\mathbf{w} \equiv \sum_j \sum_{\mathbf{y}' \in S_j} \alpha_{(j\mathbf{y}')} \delta\Psi_j(\mathbf{y}')$.
    / find cutting plane */*
    compute $\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{Y}} H(\mathbf{y})$
    / determine value of current slack variable */*
    compute $\xi_i = \max\{0, \max_{\mathbf{y} \in S_i} H(\mathbf{y})\}$
    if $H(\hat{\mathbf{y}}) > \xi_i + \varepsilon$ **then**
      / add constraint to the working set */*
      $S_i \leftarrow S_i \cup \{\hat{\mathbf{y}}\}$
      / Variant (a): perform full optimization */*
      $\alpha_S \leftarrow$ optimize the dual of SVM_0, SVM_1^* or SVM_2^* over S, $S = \cup_i S_i$.
      / Variant (b): perform subspace ascent */*
      $\alpha_{S_i} \leftarrow$ optimize the dual of SVM_0, SVM_1^* or SVM_2^* over S_i
    end if
  end for
until no S_i has changed during iteration

Schritt 5:

- Es sei ξ_i maximal für die Nebenbedingung.
- Damit herrscht Gleichheit und es kann umgeformt werden.
- Zur Lösung wird das optimale w eingesetzt.
- Damit erhalten wir das maximale ξ_i

$\Rightarrow$ wende den Cutting Plane Algorithmus an(Schritt 6).

Cutting Plane Algorithmus

Idee:

- Es soll eine konvexe Funktion $f(x)$ minimiert werden von der die analytische Form unbekannt ist.
- Es ist aber möglich für einen Punkt x^k den Funktionswert $f(x^k) = v_k$ und sogar den Subgradienten γ_k, welcher existiert da die Funktion konvex ist, zu berechnen.
- Es ist also möglich eine affine Funktion $g(x^k) = v_k + \gamma_k' x^k$ zu finden.
- Damit wird dann die Funktion $f(x)$, durch $f(x) \geq g(x)$ beschränkt.

Cutting Plane Algorithmus

Algorithmus:

0: Sei $x^1 \in S$ eine initiale zulässige Lösung, k = 0, obere Schranke $u_0 = f(x^1)$, untere Schranke $l_0 = -\infty$ und die untere Grenzfunktion $\beta_0(x) = -\infty$

1: Erhöhe k = k +1 und finde einen Subgradienten von f an der Stelle x^k

2: Aktualisiere die obere Schranke $u_k = \min\{u_{k-1}, f(x^k)\}$ und die untere Grenzfunktion $\beta_k(x) = max\{\beta_{k-1}(x), v_k + \gamma_k' x\}$

3: Löse das Problem $l_k = \min\limits_{x \in S} \beta_k(x)$ und x^{k+1} sei die optimale Lösung

4: Wenn $u_k - l_k < \epsilon$, dann stoppe, sonst gehe zu 1.

Algorithmus

Algorithm 1 Algorithm for solving SVM_0 and the loss re-scaling formulations SVM_1^* and SVM_2^*.

Input: $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n), C, \varepsilon$
$S_i \leftarrow \emptyset$ for all $i = 1, \ldots, n$
repeat
  for $i = 1, \ldots, n$ **do**
    / prepare cost function for optimization */*
    set up cost function

$$H(\mathbf{y}) \equiv \begin{cases} 1 - \langle \delta\Psi_i(\mathbf{y}), \mathbf{w} \rangle & (\mathrm{SVM}_0) \\ (1 - \langle \delta\Psi_i(\mathbf{y}), \mathbf{w} \rangle) \triangle(\mathbf{y}_i, \mathbf{y}) & (\mathrm{SVM}_1^{\triangle s}) \\ \triangle(\mathbf{y}_i, \mathbf{y}) - \langle \delta\Psi_i(\mathbf{y}), \mathbf{w} \rangle & (\mathrm{SVM}_1^{\triangle m}) \\ (1 - \langle \delta\Psi_i(\mathbf{y}), \mathbf{w} \rangle) \sqrt{\triangle(\mathbf{y}_i, \mathbf{y})} & (\mathrm{SVM}_2^{\triangle s}) \\ \sqrt{\triangle(\mathbf{y}_i, \mathbf{y})} - \langle \delta\Psi_i(\mathbf{y}), \mathbf{w} \rangle & (\mathrm{SVM}_2^{\triangle m}) \end{cases}$$

    where $\mathbf{w} \equiv \sum_j \sum_{\mathbf{y}' \in S_j} \alpha_{(j\mathbf{y}')} \delta\Psi_j(\mathbf{y}')$.
    / find cutting plane */*
    compute $\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{Y}} H(\mathbf{y})$
    / determine value of current slack variable */*
    compute $\xi_i = \max\{0, \max_{\mathbf{y} \in S_i} H(\mathbf{y})\}$
    if $H(\hat{\mathbf{y}}) > \xi_i + \varepsilon$ **then**
      / add constraint to the working set */*
      $S_i \leftarrow S_i \cup \{\hat{\mathbf{y}}\}$
      / Variant (a): perform full optimization */*
      $\alpha_S \leftarrow$ optimize the dual of SVM_0, SVM_1^* or SVM_2^* over S, $S = \cup_i S_i$.
      / Variant (b): perform subspace ascent */*
      $\alpha_{S_i} \leftarrow$ optimize the dual of SVM_0, SVM_1^* or SVM_2^* over S_i
    end if
  end for
until no S_i has changed during iteration

Algorithmus : Bemerkungen

- Im Fall des Slack Re-Scalings muss $\hat{y} = \arg\max_{y \in \mathcal{Y}} \{(1 - \langle w, \delta\Psi_i(y) \rangle) \Delta(y_i, y)\}$ maximiert werden.
- In dem Fall, wo $\Delta(y_i, \cdot)$ nur eine endliche Anzahl von Werten annehmen kann, kann in Schritt 6 wie folgt vorgegangen werden:
  1.) Maximiere über die y für die der Verlust konstant ist
  2.) Maximiere über eine die restlichen y
- Im Fall des Margin Re-Scalings muss $\hat{y} = \arg\max_{y \in \mathcal{Y}} \{\Delta(y_i, y) - \langle w, \delta\Psi_i(y) \rangle\}$ maximiert werden.
- In den Fällen, wo die Verlustfunktion eine additive Dekomposition hat, die kompatibel mit der Featuremap ist, kann die Verlustfunktion in den Vektor w gefalten werden. $\langle w', \delta\Psi_i(y) \rangle = \langle w, \delta\Psi_i(y) \rangle - \Delta(y_i, y)$ für ein bel. w'.

Algorithmus : Bemerkungen

- Für den Fall von der Zero-One Verlustfunktion muss das höchste Scoring y das inkorrekt ist gefunden werden: $\hat{y} = \arg\max_{y \in \mathcal{Y}} \{1 - \langle w, \delta\Psi_i(y) \rangle\}$
- Es sollte also die beste Lösung $\hat{y}$ und die zweitbeste Lösung $\tilde{y}$ gefunden werden. Die zweite Lösung wird benötigt, um Margin Verletzungen in den Fällen, wo $\hat{y} = y_i$ aber $\langle w, \delta\Psi_i(\tilde{y}) < 1$ ist, aufzufinden.
- Das heißt, das für Probleme für die das Ausgangsproblem $f(x; w) = F(x, y; w)$ für die ersten zwei gelöst werden kann, kann die Zero-One Verlustfunktion angewendet werden.

Beweisidee:

- Finde eine untere Grenze für die Verbesserung, wenn eine Variable $\alpha_{i\hat{y}}$ hinzugefügt wird.
- Durch die zusätzliche Nebenbedingung in SVM_1 kann es aber zu Problemen kommen.
- $\Rightarrow$ Finde eine untere Grenze für die Verbesserung in nur eine Richtung η. Wenn es eine Verbesserung in eine Richtung gibt, gibt das eine untere Grenze für die gesamte Verbesserung.

Lemma 10

Sei J eine symmetrische, positiv semi-definite Matrix. Außerdem definiere mit

$$\Theta(\alpha) = -\frac{1}{2}\alpha' J\alpha + \langle h, \alpha \rangle$$

eine konkave Zielfunktion, bei der wir die Beschränktheit annehmen. Angenommen, es existiere eine Lösung α^0 und eine Optimierungsrichtung η, so dass $\langle \nabla\Theta(\alpha^0), \eta \rangle > 0$. Dann wird die Optimierung von Θ, beginnend bei α^0 und entlang der Richtung η, den Wert der Zielfunktion um

$$\max_{\beta>0}\{\Theta(\alpha^0 + \beta\eta)\} - \Theta(\alpha^0) = \frac{1}{2}\frac{\langle \nabla\Theta(\alpha^0), \eta \rangle^2}{\eta' J\eta} > 0$$

verbessern.

Korrektheit und Komplexität des Algorithmus: Beweis Lemma 10

Beweis:
Die Differenz die durch ein bestimmtes β erreicht werden kann ist

$$\delta\Theta(\beta) \equiv \beta \left[\langle \nabla\Theta(a^0), \eta \rangle - \frac{\beta}{2} \eta' J \eta \right].$$

Die Lösung für ein bestimmtes β ist gegeben als

$$\frac{d}{d\beta}\delta\Theta = 0 \Leftrightarrow \beta* = \frac{\langle \nabla\Theta(a^0), \eta \rangle}{\eta' J \eta}$$

Dazu wird $\eta' J \eta$ benötigt. Da J semi-definiet ist, ist $\eta' J \eta \geq 0$ für jedes η garantiert. Weiter gilt, das $\eta' J \eta = 0$ zusammen mit $\langle \Theta(\alpha^0), \eta \rangle > 0$ bedeuten würde, daß $\lim_{\beta \to \infty} \Theta(\alpha^0 + \beta\eta) = \infty$, was die Annahme, daß Θ beschränkt ist, verletzen würde. Wird $\beta*$ in den Ausdruck für $\delta\Theta$ eingesetzt ergibt sich die Behauptung.

Korollar 11

Unter den gleichen Annahmen wie in Lemma 10 und für den speziellen Fall der Optimierungsrichtung $\eta = e_r$, verbessert sich die Zielfunktion um

$$\delta(\beta^*) = \frac{1}{2J_{rr}} \left(\frac{\delta\Theta}{\delta\alpha_r} \right)^2 > 0$$

Beweis:

Aus $\eta = e_r$ folgt $\langle \nabla\Theta, \eta \rangle = \frac{\delta\Theta}{\delta\alpha_r}$ und $\eta' J \eta = J_{rr}$

Korollar 12

Unter den gleichen Annahmen wie in Lemma 10 und der Beschränkung von $\beta \leq D$ für ein $D > 0$ verbessert sich die Zielfunktion um

$$\max_{0<\beta\leq D} \left\{\Theta(\alpha^0 + \beta\eta)\right\} - \Theta(\alpha^0)$$

$$= \begin{cases} \frac{\langle \nabla\Theta(\alpha^0),\eta\rangle^2}{2\eta' J\eta} & \text{Wenn } \langle \nabla\Theta(\alpha^0),\eta\rangle \leq D\eta' J\eta \\ D\langle \nabla\Theta(\alpha^0),\eta\rangle - \frac{D^2}{2}\eta' J\eta & \text{sonst} \end{cases}$$

Die Verbesserung kann sogar nach oben Beschränkt werden

$$\max_{0<\beta\leq D} \left\{\Theta(\alpha^0 + \beta\eta)\right\} - \Theta(\alpha^0)$$

$$\geq \frac{1}{2}\min\left\{D, \frac{\langle \nabla\Theta(\alpha^0),\eta\rangle}{\eta' J\eta}\right\}\langle \nabla\Theta(\alpha^0),\eta\rangle$$

Korrektheit und Komplexität des Algorithmus: Beweis Korollar 12

Beweis:

- Es muss in 2 Fälle unterschieden werden:
  - **1. Fall** $\beta* \leq D$: Es kann einfach Lemma 10 angewandt werden, da die zusätzliche Nebenbedingung inaktiv ist und damit die Lösung nicht ändert.
  - **2. Fall** $\beta* \leq D$ Da Θ konvav ist, wird bei $\beta = D$ das Maximum von $\delta\Theta$ über den eingeschränkten Raum erreicht.

Korollar 13

Unter den gleichen Annahmen wie in Korollar 12 und für den speziellen Fall der Optimierungsrichtung $\eta = e_r$, verbessert sich die Zielfunktion mindestens um

$$\max_{0<\beta\leq D} \left\{\Theta(\alpha^0 + \beta\eta)\right\} - \Theta(\alpha^0)$$

$$\geq \frac{1}{2} \min \left\{ D, \frac{\frac{\partial\Theta}{\partial\alpha_r}(\alpha^0)}{J_{rr}} \right\} \frac{\partial\Theta}{\partial\alpha_r}(\alpha^0)$$

Beweis:

- Aus $\eta = e_r$ folgt $\langle \nabla\Theta, \eta \rangle = \frac{\partial\Theta}{\partial\alpha_r}$ und $\eta' J \eta = J_{rr}$

Proposition 14 ($\text{SVM}_2^{\Delta s}$)

Für $\text{SVM}_2^{\Delta s}$ ist in Schritt 10 des Algorithmus' die Verbesserung $\delta\Theta$ der dualen Zielfunktion nach unten beschränkt durch

$$\delta\Theta \geq \frac{1}{2}\frac{\epsilon^2}{\Delta_i R_i^2 + \frac{n}{C}}, \text{ wobei}$$

$$\Delta_i \equiv \max_y \{\Delta(y_i, y)\} \text{ und } R_i \equiv \max_y \{\parallel \delta\Psi_i(y) \parallel\}$$

Korrektheit und Komplexität des Algorithmus: Beweis von Proposition 14 (1/2)

Beweis:
Für die Benutzung von Korollar 11 wähle man $r = (i\hat{y}), h = 1$ und $J_{(i\hat{y})(jy)} = \langle \delta\Psi_i(\hat{y}), \delta\Psi_j(y) \rangle + \frac{\delta(i,j)n}{C\sqrt{\Delta(y_i,\hat{y})}\sqrt{\Delta(y_i,y)}}$.

Die partielle Ableitung von Θ nach $\alpha_{(i\hat{y})}$ ist gegeben als

$$\frac{\partial\Theta}{\partial\alpha_{(i\hat{y})}} = 1 - \sum_{j,y} \alpha^0_{(jy)} J_{(i\hat{y})(jy)} = 1 - \langle w^*, \delta\Psi_i(\hat{y}) \rangle - \frac{\xi^*_i}{\sqrt{\Delta(y_i,\hat{y})}},$$

da die optimalen Werte für die primalen Variablen gegeben sind als

$$w^* = \sum \alpha^0_{(jy)} \delta\Psi_j(y) \text{ und } \xi^*_i = \sum_{y \neq y_i} \frac{n\alpha^0_{iy}}{C\sqrt{\Delta(y_i,y)}}.$$

Korrektheit und Komplexität des Algorithmus: Beweis von Proposition 14 (2/2)

Wenn nun die Bedingung $\sqrt{\Delta(y_i, \hat{y})}(1 - \langle w^*, \delta\Psi_i(\hat{y})\rangle) > \xi_i^* + \epsilon$ (10. Schritt) angewandt wird, liefert das die Grenze

$$\frac{\partial\Theta}{\partial\alpha_{i\hat{y}}}(\alpha^0) \geq \frac{\epsilon}{\sqrt{\Delta(y_i, \hat{y})}}.$$

Wird nun $J_{rr} = \| \delta\Psi_i(\hat{y}) \|^2 + \frac{n}{C\Delta(y,\hat{y})}$ und die obige Grenze in Korollar 11 eingesetzt liefert das:

$$\frac{1}{2J_{rr}}\left(\frac{\partial\Theta}{\partial\alpha_{(i\hat{y})}}\right)^2 \geq \frac{\epsilon^2}{2(\Delta(y_i, \hat{y}) \| \delta\Psi_i(\hat{y}) \|^2 + \frac{n}{C}} \geq \frac{\epsilon^2}{2(\Delta_i R_i^2 + \frac{n}{C})}$$

Die Behauptung folgt daraus, das bei einer Optimierung über eine Menge von Variablen, die α_r enthält, der Wert nur steigen kann.

Proposition 15 ($\text{SVM}_2^{\Delta m}$)

Für $\text{SVM}_2^{\Delta m}$ ist in Schritt 10 des Algorithmus' die Verbesserung $\delta\Theta$ der dualen Zielfunktion nach unten beschränkt durch

$$\delta\Theta \geq \frac{1}{2}\frac{\epsilon^2}{R_i^2 + \frac{n}{C}}, \text{ wobei } R_i \equiv \max_y \{\| \delta\Psi_i(y) \|\}$$

Korrektheit und Komplexität des Algorithmus: Beweis Proposition 15

Beweis:
Durch die Definition von $\delta\tilde{\Psi}_i(y) \equiv \frac{\Psi_i(y)}{\sqrt{\Delta(y_i,y)}}$ kann Proposition 14 direkt benutzt werden mit

$$\max_y \left\{ \Delta(y_i, y) \parallel \delta\tilde{\Psi}_i(y) \parallel^2 \right\} = \max_y \left\{ \parallel \delta\Psi_i(y) \parallel^2 \right\} = R_i^2, \text{ da}$$

$$\langle w, \delta\Psi_i(y) \rangle \geq \sqrt{\Delta(y_i, y)} - \xi_i \Leftrightarrow \langle w, \delta\tilde{\Psi}_i(y) \rangle \geq 1 - \frac{\xi_i}{\sqrt{\Delta(y_i, y)}}$$

Proposition 16 ($\text{SVM}_1^{\Delta s}$)

Für $\text{SVM}_1^{\Delta s}$ ist in Schritt 10 des Algorithmus' die Verbesserung $\delta\Theta$ der dualen Zielfunktion nach unten beschränkt durch

$$\delta\Theta \geq \min\left\{\frac{C\epsilon}{2n}, \frac{\epsilon^2}{8\Delta_i^2 R_i^2}\right\}, \text{ wobei}$$

$$\Delta_i = \max_y \{\Delta(y_i, y)\} \text{ und } R_i = \max_y \{\| \delta\Psi_i(y) \|\}.$$

Korrektheit und Komplexität des Algorithmus: Beweis Proposition 16

Wenn das Working Set kein Element (iy) enthält, dann kann über $\alpha_{(i\hat{y})}$ unter der Bedingung, dass $\alpha_{(i\hat{y})} \leq \Delta(y_i, \hat{y})\frac{C}{n} = D$ optimiert werden. Dabei ist zu beachten, dass $\frac{\partial\Theta}{\partial\alpha_{(i\hat{y})}}(\alpha^0) = 1 - \langle w^*, \delta\Psi_i(\hat{y})\rangle > \frac{\xi_i^* + \epsilon}{\Delta(y_i,\hat{y})} \geq \frac{\epsilon}{\Delta(y_i,\hat{y})}$, wobei die erste Ungleichung aus der Vorbedingung für die Selektion von $(i\hat{y})$ und letztere aus $\xi_i^* \geq 0$. Desweiteren ist $J_{(i\hat{y})(i\hat{y})} \leq R_i^2$. Wird nun Korollar 13 angewandt folgt

$$\delta\Theta \geq \frac{1}{2}\min\left\{D, \frac{1}{J_{rr}}\frac{\partial\Theta}{\partial\alpha_{(i\hat{y})}}(\alpha^0)\right\}\frac{\partial\Theta}{\partial\alpha_{(i\hat{y})}}(\alpha^0)$$

$$\geq \frac{1}{2}\min\left\{\frac{\Delta(y_i,\hat{y})C)}{n}, \frac{\epsilon}{\Delta(y_i,\hat{y})R_i^2}\right\}\frac{\epsilon}{\Delta(y_i,\hat{y})}$$

$$= \min\left\{\frac{C\epsilon}{2n}, \frac{\epsilon^2}{2R_i^2\Delta(y_i,\hat{y})^2}\right\}$$

Korrektheit und Komplexität des Algorithmus: $\text{SVM}_1^{\Delta m}$

Proposition 17 ($\text{SVM}_1^{\Delta m}$)

Für $\text{SVM}_1^{\Delta m}$ ist in Schritt 10 des Algorithmus' die Verbesserung $\delta\Theta$ der dualen Zielfunktion nach unten beschränkt durch

$$\delta\Theta \geq \frac{\epsilon^2}{8R_i^2}, \text{ wobei } R_i = \max_y \parallel \delta\Psi_i(y) \parallel .$$

Korrektheit und Komplexität des Algorithmus: Beweis Proposition 17

Durch die Definition von $\delta\tilde{\Psi}_i(y) \equiv \frac{\Psi_i(y)}{\Delta(y_i,y)}$ kann Proposition 16 direkt benutzt werden mit

$$\max_y \left\{ \Delta(y_i, y)^2 \parallel \delta\tilde{\Psi}_i(y) \parallel^2 \right\} = \max_y \left\{ \parallel \delta\Psi_i(y) \parallel^2 \right\} = R_i^2, \text{ da}$$

$$\langle w, \delta\Psi_i(y) \rangle \geq \Delta(y_i, y) - \xi_i \Leftrightarrow \langle w, \delta\tilde{\Psi}_i(y) \rangle \geq 1 - \frac{\xi_i}{\Delta(y_i, y)}$$

Korrektheit und Komplexität des Algorithmus:Theorem 18

Theorem 18

Mit $\bar{R} = \max_i R_i (= \max_y \parallel \delta\Psi_i(y) \parallel)$ und $\bar{\Delta} = \max_i \Delta_i (= \max_y \{\Delta(y_i, y)\})$ und für ein gegebenes $\epsilon > 0$ terminiert der Algorithmus nach dem inkrementellen hinzufügen von

$$max\left\{\frac{2n\bar{\Delta}}{\epsilon}, \frac{8C\bar{\Delta}^3\bar{R}^2}{\epsilon^2}\right\}, max\left\{\frac{2n\bar{\Delta}}{\epsilon}, \frac{8C\bar{\Delta}\bar{R}^2}{\epsilon^2}\right\},$$

$$\frac{C\bar{\Delta}^2\bar{R}^2 + n\bar{\Delta}}{\epsilon^2} \text{ und } \frac{C\bar{\Delta}\bar{R}^2 + n\bar{\Delta}}{\epsilon^2}$$

Nebenbedingungen zum working set S für $SVM_1^{\Delta s}$, $SVM_1^{\Delta m}$, $SVM_2^{\Delta s}$ und $SVM_2^{\Delta m}$.

Übersicht

Vortrag

1. Lernaufgabe
2. Primales Problem
3. Duales Problem
4. Optimierungsalgorithmus der SVM
5. Spezielle Ausprägungen und Anwendungen

3 Fragen

- **Modellierung:** Wie werden passende Featuremaps $\Psi(x, y)$ für spezifische Probleme definiert?
- **Algorithmus:** Wie kann die benötigte Maximierung über $\mathcal{Y}$ für ein gegebenes x berechnet werden?
- **Spärlichkeit:** Wie kann $\| \Psi(x, y) - \Psi(x, y') \|$ beschränkt werden ?

Gewichtete Kontext-freie Grammatiken

Problemstellung

- Sage, basierend auf einen String $x = (x_1, \ldots, x_k)$ von Terminalsymbolen, einen gelabelten Baum y vorraus.
- Jeder Knoten im Baum entspricht der Anwendung einer kontext-freien Grammatikregel.
- Die Blätter sind die Symbole aus x und die inneren Knoten sind nicht-terminal Knoten aus einem gegebenen Alphabet $\mathcal{N}$.
- Es wird davon ausgegangen, dass der Baum in Chomsky Normal Form ist.

Modellierung

- **Annahme:** WCFG modellieren die Abhängigkeit zwischen x und y.
- **Regeln:** $n_l\,[C_i \rightarrow C_j, C_k]$ oder $n_l\,[C_i \rightarrow x_t]$, mit $C_i, C_j, C_k \in \mathcal{N}$ (Nicht-Terminal Symbole) und $x_t \in \mathcal{T}$ (Terminalsymbole).
- **Gewichtung(PCFG):** Jede Regel n_l ist durch ein w_l parametrisiert, wobei w_l die logarithmierte W'keit dafür angibt , dass ein Knoten C_i mit Regel n_l expandiert wird.
- das führt zum folgenden Maximierungsproblem:

$$h(x) = \arg\max_{y \in \mathcal{Y}} P(y|x) = \arg\max_{y \in \mathcal{Y}} \left\{ \sum_{n_l \in rules(y)} w_l \right\}$$

- Damit ergibt sich $\langle w, \Psi(x,y) \rangle = \sum_{n_l \in rules(y)} w_l$

Gewichtete Kontext-freie Grammatiken: Algorithmus und Spärlichkeit

Algorithmus

- Die Lösung für $\arg\max_{y \in \mathcal{Y}} \langle w, \Psi(x, y)$kann mit dem CKY-Parser berechnet werden. Dieser liefert außerdem auch die zweitbeste Lösung.
- Für andere Verlustfunktionen als der Zero-One Loss kann der Algorithmus angepasst werden.

Spärlichkeit

- Der Baum hat für eine Sequenz der Länge N $N-1$ interne Knoten.
- Der Baum hat N Pre-Terminalknoten.
- Damit ist die L_1-Norm von $\Psi(x, y) = 2N - 1$ und die L_2-Norm ist maximal $\sqrt{4N^2 + 4(N-1)^2} < 2\sqrt{2}N$

Label Sequence Learning

Problemstellung

- Sage, basierend auf einer Sequenz $x = (x^1, \ldots, x^T)$, eine Labelsequenz $y = (y^1, \ldots, y^T)$.
- Vereinfachung: Alle Sequenzen haben die Länge T.
- Σ bezeichnet die Menge der möglichen für jede Variable y^t, beispielsweise $\mathcal{Y} = \Sigma^T$.
- Jede Labelsequenz wird als eigene Klasse definiert, damit ex. $|\Sigma|^T$ Klassen.
-

Modellierung:

- Inspiriert durch Hidden Markov Models (HMM), beinhaltet Ψ Interaktionen von Inputfeatures und Labels. Das wird durch Kopien der Inputfeatures und durch Features, die die Interaktion zwischen eng zusammenliegenden Labels modellieren, erreicht.

$$F(x, y; w) =$$

$$\sum_{t=1}^{T} \sum_{\sigma \in \Sigma} \langle \bar{w}_\sigma, \Phi(x^t) \rangle \delta(y^t, \sigma) + \eta \sum_{t=1}^{T-1} \sum_{\sigma \in \Sigma} \sum_{\hat{\sigma} \in \Sigma} \hat{w}_{\sigma,\bar{\sigma}} \delta(y^t, \sigma) \delta(y^{t+1}, \bar{\sigma})$$

$$= \left\langle \bar{w}, \sum_{t=1}^{T} \Phi(x^t) \otimes \Lambda^c(y^t) \right\rangle + \eta \left\langle \bar{w}, \sum_{t=1}^{T-1} \Lambda^c(y^t) \otimes \Lambda^c(y^{t+1}) \right\rangle,$$

mit $w = (\bar{w}', \hat{w}')'$, λ^c bezeichnet die orthogonale Repräsentation der Labels über Σ und $\eta \geq 0$.

Modellierung:
Damit ist:

$$\Psi(x,y) = \begin{pmatrix} \sum_{t=1}^{T} \Phi(x^t) \otimes \Lambda^c(y^t) \\ \eta \sum_{t=1}^{T-1} \Lambda^c(y^t) \otimes \Lambda^c(y^{t+1}) \end{pmatrix}$$

Damit kann das Skalarprodukt von zwei Labelsequenzen geschrieben werden als:

$$\langle \Psi(x,y), \Psi(\bar{x},\bar{y}) \rangle =$$

$$\sum_{s,t=1}^{T} \delta(y^t,\bar{y}^s) K(x^t,\bar{x}^s) + \eta^2 \sum_{s,t=1}^{T-1} \delta(y^t,\bar{y}^s)\delta(y^{t+1},\bar{y}^{s+1})$$

Label Sequence Learning: Modellierung

Modellierung(mögliche Erweiterungen):

- Extrahiere nicht nur Features aus x^t sondern aus einem Fenster um x^t.
- Kombiniere mehrere Labelfeatures
- Kombiniere mehrereLabelfeatures mit Inputfeatures

Label Sequence Learning: Algorithmus und Spärlichkeit

Algorithmus

- $\langle w, \Psi(x, y) \rangle$ kann durch dynamische Programierung maximiert werden.
- Insbesondere bietet sich die Viterbi-Dekodierung an.
- Diese kann auch die zweitbeste Lösung im Falle der Zero-One Verlustfunktion finden.
- Viterbi-Dekodierung ist auch mit anderen Verlustfunktionen möglich.

Spärlichkeit

- Definiere $R_i \equiv \max_t \| \Phi(x_i^t) \|$, dann ist
  $\| \Psi(x_i, y) - \Psi(x_i, y') \|^2 \leq 2T^2(R_i^2 + \eta^2)$

Problemstellung

- Lerne das Alignment von einer Sequenz $x \in \Sigma^*$zu einer Sequenz $y \in \Sigma^*$, mit Σ^* die Menge aller Strings über ein endliches Alphabet Σ.
- Dazu wähle eine Sequenz von Alignmentoperationen, z.B. Einfügen, Entfernen oder Ersetzten, die x in y transformiert und eine lineare Zielfunktion maximiert

$$\hat{a}(x, y) = \arg\max_{a \in A} \langle w, \Psi(x, y, a) \rangle.$$

- Ψ ist der Histrogrammvektor der Operationen.
- der Wert von $\langle w, \Psi(x, y, \hat{a}(x, y)) \rangle$ kann benutzt werden um die Ähnlichkeit zwischen x und y zu messen.

Modellierung

- Lerne auf nativen Sequenz x_i. Für jede Sequenz existiert eine gleichwertige Sequenz y_i mit einem optimalen Alignment a_i.
- Des weiteren werden s.g. Decoy-Sequenzen $y_i^t =, t = 1, \ldots, n$ mit unbekanntem Alignment benutzt.
- Die Lernaufgabe besteht darin, den gleichwertigen Sequenzen einen hohen Score zu zuordnen und den Decoy-Sequenzen einen niedrigeren Score.
- Ausgaberaum $\mathcal{Y}_\rangle = \{y_i, y_i^1, \ldots, , y_i^k\}$ für das $i-$te Beispiel.
- Es wird ein w gesucht, so dass $\langle w, \Psi(x_i, y_i, a_i)\rangle > \langle w, \Psi(x_i, y_i^t, a)\rangle$ für alle t und a.
- Damit folgt ein Zero-One Verlust und

$$f(x_i) = \arg \max_{y \in \mathcal{Y}_i} \max_a \langle , \Psi(x, y, a)\rangle$$

Sequence Alignment: Algorithmus und Spärlichkeit

Algorithmus

- Um das Maximierungproblem zu lösen, kann dynamische Programmierung benutzt werden, z.B. der Smith-Waterman Algorithmus.
- Um das obige $\arg\max$ Problem zu lösen, wird davon ausgegangen, dass die Anzahl der Decoy-Sequenzen klein genug ist, um mit vollständiger Suche das Optimum zu finden.

Spärlichkeit

- Jede Operation liest einen Buchstaben in x oder y
- Wenn die maximale Sequenzlänge N ist, dann ist die L_1-Norm von $\Psi(x, y, a)$maximal $2N$ und die L_2-Norm von $\Psi(x, y, a) - \Psi(x, y', a')$ ist maximal $2\sqrt{2}N$

Zusammenfassung

- Durch die Featuremap $\Psi(x, y)$ haben wir die Möglichkeit eine Vielzahl von Problemen zu lösen.
- Durch beliebige Verlustfunktionen $\Delta(y_i, y)$ können wir problemspezifische Verluste modellieren.
- Der vorgestellte Algorithmus gibt uns die Möglichkeit mit sehr vielen Nebenbedingungen umgehen zu können.

Noch Fragen?