

Analiza sistemelor hybrid de recunoaștere a formelor cu metodele HMM și SVM

Ștefania – Iuliana ȘOIMAN, Ștefan Gheorghe PENTIUC

Abstract—Probabilistic model such as Hidden Markov Models and Support Vector Machine classifier are representing the subject of our study. By combining the two above classifiers, in order to realize a pattern recognition system, we could obtain better classification performance. Throughout this paper we will present an overview of these classifiers and illustrate their usage inside a pattern recognition system. The goal is to highlight the method in which HMM and SVM can be merged together.

Index Terms—Hidden Markov Models, Support Vector Machine

I. INTRODUCERE

SCOPUL acestei lucrări este studierea metodelor de clasificare automată a formelor bazate pe tehnica vectorilor suport (SVM – Support Vector Machine) și pe Modelul Markov Ascuns (HMM – Hidden Markov Model). Rezultatul acestui studiu este concretizat sub forma unei analize comparativă asupra sistemelor de recunoaștere a formelor existente ce folosesc HMM și SVM. O analiză matematică a clasificatorilor generativi (HMM) și discriminativi (SVM) este făcută de Bouchard și Triggs [1]. La sistemul online de recunoaștere a scrisului de mână propus de autorii [2], HMM aplicat pentru clasificarea multi-clasă iar SVM cu Fisher kernel, pentru a clasifica formele rămase neclasificate după prima trecere HMM.

II. MODELE MARKOV ASCUNSE (HMM)

Teoria MMA și aplicațiile în care au fost utilizate nu reprezintă o noutate. Publicată inițial de matematicianul Baum, teoria de bază a început foarte târziu să fie înțeleasă și aplicată în recunoașterea formelor [3],[4],[5],[6]. Motivele pentru aceste întârzieri sunt în primul rând legate de faptul că inițial teoria pentru MMA a fost publicată în jurnale de matematică care nu erau citite de către inginerii interesați în problematica recunoașterii formelor; în al doilea rând nu a existat un tutorial detaliat și bine documentat pentru teoria legată de recunoașterea formelor și implementarea unei soluții având la bază un model MMA [3].

Ștefania – Iuliana ȘOIMAN, drd. ing. – Universitatea Ștefan Cel Mare Suceava, str.Universității nr.13, RO-720229 Suceava (e-mail: stefania_soiman@hotmail.com)

Ștefan Gheorghe PENTIUC, prof. dr.ing. – Universitatea Ștefan Cel Mare Suceava, str.Universității nr.13, RO-720229 Suceava (e-mail: pentiuc@eed.usv.ro)

A. Elementele unui MMA

Un MMA este caracterizat de următoarele componente:

- 1) N, numărul de stări ascunse: $Q = \{1, 2, \dots, N\}$
- 2) M, numărul de simboluri observabile: $V = \{v_1, v_2, v_3, \dots, v_M\}$
- 3) Distribuția probabilităților de tranziție dintre stări: $A = \{a_{ij}\}$, unde $a_{ij} = P[q_t = j | q_{t-1} = i], 1 \leq i, j \leq N$

Un exemplu de model Markov (Fig.1), sub forma unui graf orientat cu patru stări, nodurile reprezintă stările iar săgețile tranzițiile posibile de la o stare la alta.

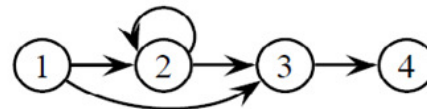


Fig. 1. Model Markov stânga – dreapta.

Modelul din Figura 1 are în starea 2 o tranziție în aceeași stare ceea ce înseamnă $a_{22} = P[X_{t-1} = 2 | X_t = 2] > 0$. O asemenea structură este denumită model Markov stânga – dreapta. Toate probabilitățile dintre două stări care nu sunt legate cu arce sunt nule.

Dacă avem secvența de timp $t = \{1, 2, \dots, T\}$ și N stări $Q = \{1, 2, \dots, N\}$, o singură stare j la momentul t este q_t , pentru orice $N \geq 1$ avem [3]:

$$a_{ij} = P[q_t = j | q_{t-1} = i], \forall 1 \leq i, j \leq N$$

Funcția P este numită funcție de tranziție și poate fi reprezentată prin intermediul unei matrici de dimensiune $N \times N$ ce îndeplinește condițiile:

$$\sum_{j=1}^N a_{ij} = 1 \quad \forall i \quad (1)$$

Distribuția stării inițiale $\pi = \{\pi_i\}$, unde $\pi_i = P(q_1 = i), 1 \leq i \leq N$ reprezintă probabilitatea ca starea de pornire a modelului să fie j, $\sum_{j=1}^N \pi_j = 1$.

4) Distribuția probabilităților simbolurilor observabile $B = \{b_j(k)\}$, unde $b_j(k) = P[v_k = v_j | q_t = j], 1 \leq k \leq M$

5) Distribuția inițială a stărilor $\pi = \{\pi_i\}$, unde probabilitatea inițială este $\pi_i = P(q_1 = i), 1 \leq i \leq N, \sum_{j=1}^N \pi_j = 1$.

Matematic MMA poate fi descris astfel:

$$\lambda = (A, B, \pi). \quad (2)$$

Clasificare cu MMA

Problema recunoașterii formelor cu ajutorul MMA este găsirea modelului λ pentru care probabilitatea de observație $P(O|\lambda)$ să fie maximă. Cel mai simplu mod este enumerarea secvențelor de stări $q = q_1 q_2 \dots q_T$ posibile din model. Lungimea lor trebuie să fie aceeași cu lungimea secvenței de observații $O = \{o_1, o_2, \dots, o_T\}$.

Probabilitatea stărilor q :

$$P(q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \dots a_{q_{T-1} q_T} = \pi_{q_1} \prod_{i=1}^{T-1} a_{q_i q_{i+1}}$$

Probabilitatea secvenței de observații O cu sevența de stări q este:

$$P(O|q, \lambda) = \prod_{i=1}^T P(o_i | q_i, \lambda) = \prod_{i=1}^T b_{q_i}(o_i)$$

Prin însumarea celor două probabilități condiționate de mai sus se obține probabilitatea căutată.

$$P(O, \lambda) = \sum_q P(O|q, \lambda) P(q|\lambda) \quad (3)$$

Acest calcul necesită un volum ridicat de calcul $O(2TNT)$.

Într-un sistem în care sunt mai multe modele, câte unul pentru fiecare unitate, probabilitatea ca un anumit model λ care trece prin secvența de stări Q se poate calcula conform relației (3), înmulțind probabilitățile de tranziție cu probabilitățile de apariție a unui simbol $b_i(o_i)$ astfel:

$$P(O, q|\lambda) = \pi_{q_1} \cdot b_{q_1}(o_1) a_{q_1 q_2} b_{q_2}(o_2) a_{q_2 q_3} \dots b_{q_T}(o_T) a_{q_{T-1} q_T} \quad (4)$$

Probleme specifice Modelelor Markov Ascunse.

Algoritmul Viterbi: Având o secvență observabilă O într-un model $\lambda = (A, B, \pi)$ trebuie calculată cea mai probabilă secvență de stări $Q = q_1 q_2 \dots q_T$, astfel încât să determine generarea secvenței observabile. Aceasta problemă, a recunoașterii modelului Markov ascuns, este rezolvată cu ajutorul Algoritmului lui Viterbi. Prima descriere a algoritmului Viterbi apare în [7]. Dându-se o secvența de observații O , algoritmul va trebui să calculeze cea mai probabilă secvență de stări $Q = q_1 q_2 \dots q_T$, astfel încât să determine generarea secvenței observabile. Se definește $\delta_t(j)$ care este probabilitatea celui mai bun drum care duce la starea j în momentul t trecând prin primele t observații. Pentru aceasta definim probabilitatea $\delta_t^{(i)}$ maximă de-a lungul unui singur drum, la momentul t , definită pentru primele t observații, terminată în starea j :

$$\delta_t(j) = \max_{q_1 q_2 \dots q_{t-1}} P[q_1 q_2 \dots q_{t-1}, q_t = j, o_1 o_2 \dots o_t | \lambda]$$

Prin inducție pentru orice valoare t obținem

$$\delta_t(j) = \left[\max_i \delta_{t-1}(i) a_{ij} \right] \cdot b_j(o_t) \quad 2 \leq t \leq T, 1 \leq i, j \leq N \quad (6)$$

Terminarea algoritmului se face prin calcularea la fiecare moment de timp a probabilităților $\delta_t(j)$ și păstrarea a celei mai bune căi a stării anterioare ($t-1$) în variabila $\Psi_t(j)$.

Pentru un model $\lambda = (A, B, \pi)$, etapele algoritmului Viterbi sunt următoarele:

(1) Inițializare

$$\delta_1(i) = \pi_i b_i(o_1), 1 \leq i \leq N, \Psi_1(i) = 0$$

(2) Inducție

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] \cdot b_j(o_t) \quad \Psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i)]$$

$$(3) \text{ Terminare } P^* = \max_{1 \leq i \leq N} [\delta_T(i)] \quad q_T^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)]$$

(4) Backtracking

$$q_t^* = \Psi_{t+1}(q_{t+1}^*), t = T-1, \dots, 1$$

Secvența de stări generată de secvența de observații O se calculează recursiv prin înlocuirea celei mai probabile căi păstrate în $\Psi_{t-1}(j)$. Complexitatea algoritmului $O(N^2T)$.

Soluția pentru problema volumului ridicat de calcul din (4) este soluționată cu algoritmul Forward care stochează date intermediare pe care le reutilizează. Observarea în cazul modelului Markov ascuns λ , este formată din două etape: primele t observații $O(1:t)$ și emisia sfârșitului de observare $O(t+1:T)$ și egală cu:

$$P(O|\lambda) = \sum_{q_t} \alpha(t|q_t) \cdot \beta(t|q_t) \quad (7)$$

Definim variabila $\alpha_t(j) = P(o_1 o_2 \dots o_t, q_t = i | \lambda)$ probabilitatea de înaintare, a secvenței de observații de la începutul o_1 până la o_t cu condiția de a se ajunge la momentul t în starea i . Calculul funcției forward se face cu relația:

$$\alpha_t(j) = \left[\sum_{i=1}^N \alpha_{t-1}(i) a_{ij} \right] \cdot b_j(o_t), \quad 1 \leq j \leq N, 2 \leq t \leq T \quad (8)$$

Probabilitatea $P(O|\lambda)$ se obține însumând produsul $\alpha_{t-1}(i) a_{ij}$ pentru toate stările j și înmulțind cu $b_j(o_t)$ la fiecare t (α_1 la momentul $t=1$; α_2 la momentul $t=2 \dots \alpha_T$ la momentul $t=T$). Algoritmul de estimare a parametrilor Forward, complexitate $O(NT^2)$, este asemănător algoritmului Viterbi cu deosebirea că se calculează probabilitatea totală ca modelul să fie într-o anumită stare, pe când algoritmul Viterbi o calculează pe cea maximă. În fig.2 este ilustrat modul de calcul al probabilităților de înaintare/revenire.

În calculul funcției backward $\beta_t(j) = P(o_{t+1} o_{t+2} \dots o_T, q_t = i | \lambda)$, probabilitatea de revenire, a secvenței de observații de la o_{t+1} până la o_T , la momentul t fiind în starea i , se calculează astfel:

$$\beta_t(i) = \sum_{j=1}^N a_{ij} \cdot b_j(o_{t+1}) \cdot \beta_{t+1}(j) \quad (9)$$

Valorile pentru două cazuri particulare $t=0$ și $t=T$:

$$P(O|\lambda) = \sum_i \alpha_T(i) = P(O|\lambda) = \sum_i \pi_i \beta_0(i)$$

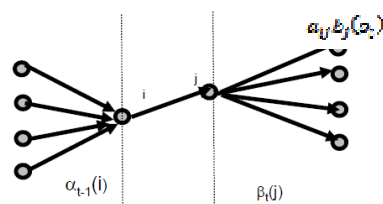


Fig. 2. Nod de calcul a probabilității forward-backward.

În practică [8], valorile probabilităților α , β din (8) și (9), la un număr mare de termeni tind repede către zero. O soluție este scalarea probabilității α proporțional cu numărul de stări, același factor de scalare se aplică și probabilității β , efectul se anulează la etapa 2 a algoritmului.

Modelele Markov Ascunse pot fi clasificate după structura grafului de stări și după tipul secvențelor observabile. Conform primului tip de clasificare, există două tipuri de modele Markov ascunse cel mai des utilizate: HMM ergodic, HMM stânga – dreapta (left – right model) și mixt.

În modelele HMM ergodice din fiecare stare se poate ajunge în orice altă stare (chiar în aceeași stare), într-un număr finit de tranziții fig.3.

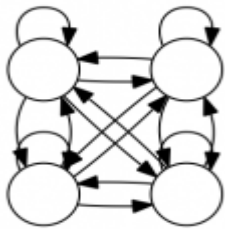


Fig. 3. HMM ergodic.

La modelul Markov Bakis sau modelul stânga – dreapta (left – right model) stările se desfășoară de la stânga la dreapta, la acest tip de modele matricea de tranziție respectă condiția: $a_{ij}=0, \forall j < i$, ceea ce înseamnă că nu sunt permise tranziții dreapta - stânga. Exemplu în fig.4.

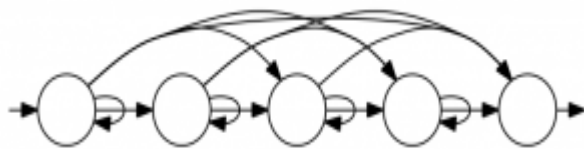


Fig. 4. MMA Bakis.

III. MAȘINA CU VECTORI SUPT (SVM)

A. Hyperplanul de separare a formelor

SVM (Support Vector Machine) sunt clasificatori liniari care funcționează foarte bine pentru un volum mare de date (analiza textelor), popularizați după 1992, dezvoltați de Vapnik [9].

SVM este un model de clasificator care găsește hiperplanul de separare care împarte formele în două clase. SVM construiește soluția pornind de la o mulțime de date de antrenament care se numesc vectori suport.

Clasificatorii liniari identifică un plan de separație dintre două clase și folosesc pentru antrenament un set de date $\{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ unde $x_i = (x_1, x_2, \dots, x_m)$ este un vector de intrare (sau formă) într-un spațiu real $X \subseteq \mathbb{R}^m$ și y_i este eticheta clasei (valoarea de ieșire), $y_i \in \{-1, 1\}$. SVM găsește o funcție liniară de forma:

$$f(x) = \langle w, x \rangle + b \quad (10)$$

unde w , este numit vector pondere. Hiperplanul de decizie ce separă cele două clase este

$$\langle w, x \rangle + b = 0;$$

cu cele două regiuni: regiunea 1, $\langle w, x \rangle + b \geq 0$ și regiunea -1, $\langle w, x \rangle + b \leq 0$ arată ca în figura 5.

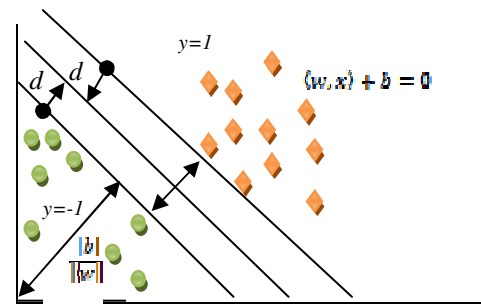


Fig. 5. Hyperplanul de separare a formelor cu SVM.

Fie un set de date de antrenare în clasificarea binară a formelor ce vor fi separate de hiperplanul Separating Hyperplane (S),

$\langle w, x \rangle + b = 0, b \in \mathbb{R}$ iar $\frac{|b|}{\|w\|}$ este distanța de la origine la hiperplan. Cele două mulțimi de vectori de intrare ce corespund etichetelor $y_i = +1$ și $y_i = -1$ sunt liniar separabile.

Datele sunt liniar separabile dacă există $w \in \mathbb{R}^n$ și $b \in \mathbb{R}$ astfel încât pentru $i=1,2,\dots,m$ avem:

$$\begin{cases} \langle w, x_i \rangle + b \geq +1, y_i = +1 \\ \langle w, x_i \rangle + b \leq -1, y_i = -1 \end{cases} \quad (11)$$

Punctele (x_i, y_i) pentru care sunt satisfăcute relațiile (11) se numesc vectori suport.

B. Orientarea și localizarea planului de separare

Problema este găsirea valorilor celor două variabile b și w , astfel încât să fie realizată separarea optimă a punctelor (cea mai largă bandă de separare). Ec. (11) se mai poate scrie sub forma:

$$y_i (\langle w, x_i \rangle + b) - 1 \geq 0, i = 1, 2, \dots, m \quad (12)$$

Prin scalarea relației (12) cu o constantă, fără schimbarea hiperplanului, marginea maximă poate fi calculată și este egală cu $1/2$ din w , orientarea planului [11]:

$$M = \frac{2}{\|w\|}$$

Datele de intrare pot fi liniar separabile, în acest caz eroarea este nulă, sau ne-separabile, există erori (Fig.6):

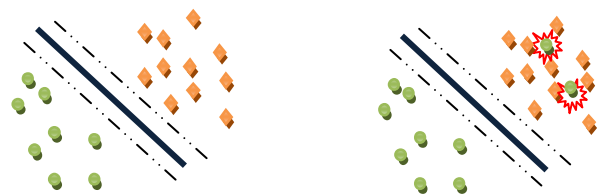


Fig. 6.1 SVM. Tipuri de date: liniar separabile, ne-separabile.

În cazul datelor de instruire ne-separabile (12) poate fi extinsă prin tratarea restricțiilor prin introducerea unei variabile ξ , numită de relaxare, astfel:

$$y_i (w \cdot x_i + b) \geq 1 - \xi_i, \quad i = 1, 2, \dots, m, \xi_i \geq 0$$

Metodologia este bazată pe teoria Vapnik – Chervonenkis, principiul de reducere a riscului [10]. Rezolvarea problemelor se face prin introducerea multiplicatorului Lagrange, $\alpha_i \geq 0$ și a Lagrangianului:

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 + \sum_{i=1}^m \alpha_i [1 - y_i (w \cdot x_i + b)]$$

Aplicând derivata parțială Lagrangianului L, se poate obține valoarea w [12]:

$$w = \sum_{i=1}^m \alpha_i y_i x_i$$

Localizarea planului, b se calculează aplicând condițiile Karush-Kuhn- Tucker1:

$$\alpha_i [y_i (w \cdot x_i + b) - 1] = 0, \quad b = \sum_{i=1}^m \alpha_i y_i (x_i \cdot w) - y_i$$

Pentru fiecare vector suport se va calcula o medie a valorii b.

C. Funcții nucleu

În cazul datelor non – linear separabile ele pot deveni linear separabile cu ajutorul unei funcții nucleu (kernel), spațiul de intrare se transformă într-un spațiu cu mai multe dimensiuni (feature space).

Găsirea hiperplanului se transformă în găsirea unei sfere de dimensiune maximă cu cost minim care grupează formele asemănătoare. Cu ajutorul funcției kernel datele de intrare vor fi transpuse într-un spațiu de dimensiune mai mare.

Definim funcția $K: X \times X \rightarrow \mathbb{R}, (x, x') \mapsto K(x, x')$ pentru toți $x, x' \in X$, ce satisface relația [13]:

$$K(x, x') = (\Phi(x), \Phi(x')) \quad (13)$$

unde Φ , este feature space. Funcția SVM liniară (10) cu kernel, are forma:

$$f(x) = \sum_{i=1}^m \alpha_i y_i K(x_i, x) + b \quad (14)$$

unde $K(.,.)$ este funcția kernel, $\sum_{i=1}^m \alpha_i y_i = 0$, $y_i \in \{-1, 1\}$, și $\alpha_i > 0$. Clasificarea formelor se va face în funcție de semnul lui f astfel:

$$f(x) = \text{sign} \left(\sum_{i=1}^m \alpha_i y_i K(x_i, x) + b \right) \quad (15)$$

Cele mai des utilizate nuclee care construiesc hiperplanul nelinear de separare, sunt:

nucleu polinomial, $K(x, x') = (\gamma(x, x') + r)^d$, $\gamma > 0$ și d sunt parametrii kernel care vor fi reglați pentru a obține cele mai bune performanțe.

nucleu Gaussian Radial Basis Function (RFB), $K(x, x') = e^{-\frac{\|x-x'\|^2}{2\sigma^2}}$, unde σ reprezintă dispersia.

Alte funcții nucleu: Fisher, lagrange exponențiale, Multi-Layer Perceptron.

D. Clasificarea cu SVM

Tehnicile avansate de clasificare automată a formelor includ Mașinile cu Vectori Suport. Algoritmul de clasificare va încerca să găsească hiperplanul optim de separare a formelor.

Alegerea corectă a parametrilor kernel-ului influențează performanțele obținute în clasificarea datelor. Sequential Minimal Optimization, este algoritmul propus [15] în antrenarea SVM: se caută hiperplanul de separare a claselor cu cea mai largă margine (maximum margin). Timpul de testare a algoritmului SMO crește linear cu numărul de vectori suport.

IV. ANALIZA SISTEMELOR DE RECUNOAȘTERE A FORMELOR CU

HMM ȘI SVM

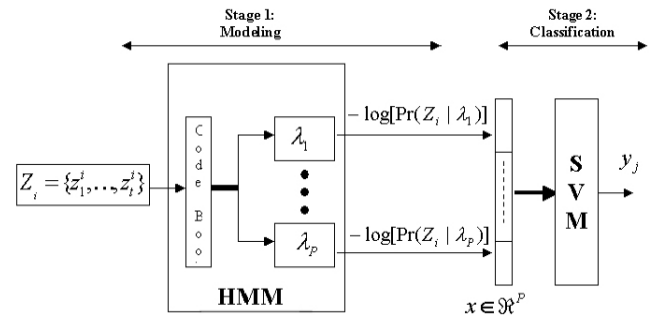


Fig. 7. Model hybrid HMM-SVM Handwriting Recognition [14].

Sistemul de recunoaștere a cifrelor scrise de mână autorii [14] combină avantajele modelului generativ Markov ascuns și modelul discriminativ SVM, utilizând date de test din baza de date MNIST [16]. Modelarea realizată cu 10 modele H-HMM și 10 modele V-HMM (Fig.7) obține performanțe 91.26%, respectiv, 91.02%. O rată de recunoaștere a caracterelor, 92.63% este obținută la clasificarea formelor cu pachetul SVM Light V.5.0 cu RFB kernel. Se observă performanțe îmbunătățite la combinarea celor două metode HMM și SVM la clasificarea formelor.

Sistemul de recunoaștere a simbolurilor chimice [17] folosind HMM obține o performanță 93,01% comparativ cu 98,08 a sistemului HMM-SVM.

Modelul hybrid HMM – SVM propus de autorii [18] folosește metoda găsită de Platt în sistemul de recunoaștere a vorbirii. În sistemul hibrid HMM-SVM (SVM-RFB) fiecare clasificator este antrenat ca un clasificator „o clasă versus celelalte clase” cu un volum mare de date. Performanțele sistemului HMM-SVM cresc cu 10% față de sistemul de recunoaștere cu modelul HMM.

V. CONCLUZII

S-a făcut o analiză comparativă a performanțelor sistemelor hibrid de recunoaștere a formelor cu HMM și SVM. Modelele probabilistice existente cum ar fi Modelele Markov Ascunse, un model generativ utilizat pe scară largă, utilizate cu Mașina cu Vectori Suport îmbunătățesc performanțele clasificatorului. Probabilitatea generativă a modelelor Markov ascunse și metodele discriminative permit construirea unei mai bune clasificări comparativ cu abordarea modelelor separat.

Direcția viitoare de cercetare se orientează către proiectarea algoritmilor de clasificare a formelor HMM, respectiv SVM folosind calculul de înaltă performanță.

¹ <http://www.svms.org/kkt/>

REFERINȚE

- [1] Bouchard G., Triggs B.(2004) *The Trade-Off Between Generative and Discriminative Classifiers*, International Conference on Computational Statistics, pp. 721-728.
- [2] Sanguansat S, Asdornwised W, Jitapunkul S.(2004) Online Thai handwritten character recognition using hidden Markov models and support vector machines.
- [3] Rabiner, L.R. (1989). A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*.
- [4] Baum, L.E. (1972). An Inequality and Associated Maximization Technique in Statistical Estimation of Probabilistic Functions of a Markov Process. *Inequalities*.
- [5] Backer J.K. (1975). The dragon system – An overview. *IEEE Trans. Acoust. Speech Signal Processing*.
- [6] Yang J., Xu Y. (1994). Hidden Markov Model for Gesture Recognition, tech. report CMU-RI-TR-94-10, Robotics Institute, Carnegie Mellon University.
- [7] Forney G.D. (1973). The Viterbi Algorithm, *Proc. IEEE*, pp.268–278.
- [8] Theodoridis S, Koitroimbas K.(2006) *Pattern recognition*, Third Edition, pp.443-469.
- [9] Vapnik V. (1995) *The Nature of Statistical Learning Theory*, Springer Verlag.
- [10] Vapnik V., Asa Ben-Hur, Horn D., Sieglmann H.T.(2001). Support Vector Clustering, *Journal of Machine Learning Research*, pp.125-137.
- [11] Pentiu Ș., Radu G. (2010). Mașini cu vectori suport nuanțate – o nouă abordare în clasificarea datelor, *Sisteme Distribuite*, Suceava.
- [12] Zhang B. (2001). Is the Maximal Margin Hyperplane Special in a Feature Space?
- [13] Cristianini N. Shawe-Taylor J (2000). *An Introduction to Support Vector Machines and other kernel-based learning methods*, Cambridge University Press, pp.30-45
- [14] Moustafa K.A., Cheriet M., Ching Suen. (2004). Classification of Time-Series Data Using a Generative/Discriminative Hybrid, *IEEE Proc. of IWFHR*, pp. 51 – 56.
- [15] Platt J. (1999) Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods—Support Vector Learning*, pp.185–208.
- [16] <http://yann.lecun.com/exdb/mnist/>
- [17] Zhang Y., Guangshun Shi, Wang K. (2010) A SVM-HMM Based Online Classifier for Handwritten Chemical Symbols, *ICPR*.
- [18] Ganapathiraju A., Hamaker J., Picone J. (2000). Hybrid SVM/HMM Architectures for Speech Recognition, *Proceedings of the Speech Transcription Workshop*, College Park, Maryland, USA.