

طبقه‌بندی داده‌ها توسط مورچه‌های فازی

علیرضا نقش^۱ مه‌کام کاهکش^۲

شرکت سهامی ذوب آهن اصفهان

دانشگاه آزاد اسلامی واحد نجف آباد

Naghsh_a@yahoo.com

چکیده

در این مقاله به بررسی روشی برای طبقه‌بندی داده‌ها می‌پردازیم. در این روش، بدون نیاز به دانستن تعداد اولیه دسته‌ها، داده‌ها بر اساس شباهتهای ذاتی دسته‌بندی می‌شوند. الگوریتم مورچه در آغاز دسته‌ها را تولید کرده و سپس این دسته‌ها با الگوریتم فازی تصحیح می‌شوند. مورچه‌ها در ابتدا اشیای منحصر به فردی را برای تشکیل توده‌ها به کار می‌برند سپس داده‌هایی که به این اشیای شباهت دارند را درون توده‌ها قرار می‌دهند. پس از تصحیح توسط الگوریتم فازی دوباره مورچه‌ها طبقه‌بندی را بین توده‌های جدید انجام می‌دهند.

واژه‌های کلیدی: « الگوریتم فازی » « طبقه‌بندی » « الگوریتم مورچه »

مقدمه

جهت درک بهتر طبقه‌بندی هوشمند، لازم است که ابتدا طبقه‌بندی‌های کلاسیک را تجسم کنیم فرض کنید پارامترهای A, B, C, D داده‌های راتوصیف می‌کنند. به ازای هر سری پارامتر که یک داده خاص را تعریف می‌کند یک پاسخ R وجود دارد که می‌تواند عنوان طبقه‌ای که آن داده مربوط به آن است باشد. مدلسازی کلاسیک از نخستین قدم خطای بزرگی را مرتکب می‌شود که فقط در سیستمهای ساده (خطی یا نزدیک به خطی) قابل پیاده‌سازی است. نخستین قدم در روش کلاسیک برای بررسی داده‌ها بررسی شاخصهای متمایل به مرکز و شاخصهای پراکندگی (انحراف معیار، ...) است. از این مرحله به بعد در روش کلاسیک کاری با تک تک نمونه‌ها نداریم و اهمیت فردی آنها از بین می‌رود در صورتی که نگاه همزمان به پارامترهای A و B برای درک تاثیر آن دو بر روی R ضروری است، در حالیکه ما با گرفتن میانگین (یا دیگر مشتقات آماری) از کل ستونهای A و B و سایر مشخصات، اثر همراهی مقادیر A و B را از صورت مساله پاک کرده ایم. در هیچ یک از روشهای کلاسیک مدلسازی با داده‌های فردی (تک تک نمونه‌ها) کاری نداریم و این یک اشکال مهم است. در واقع روش کلاسیک با عملی شبیه به هوموژن کردن یا آسیاب کردن داده‌ها، پیچیدگی روابط آنها را محو می‌کند و به این دلیل از کشف این پیچیدگیها باز می‌ماند.

^۱ کارشناسی ارشد مهندسی مخابرات

^۲ مهندسی برق الکترونیک

در نهایت نیز، در روش کلاسیک، شما یک معادله سیستم خواهید داشت که داده های جدید را بدون در نظر گرفتن اثر همراهی پارامترهایش با هم استفاده می کند و مجدداً این خطا در پیشگویی اثر R توسط سیستم شما تاثیر خواهد داشت. به این ترتیب سیستم کلاسیک در "استخراج" معنی از داده ها ضعیف و با بازده پایین عمل می کند و در بسیاری از موارد از طبقه بندی داده ها ناکام می ماند. در صورتی که در یک طبقه بندی هوشمند پارامترهای ورودی، هر کدام به شکل یک سیگنال الکتریکی تحریک به کانالهای ورودی مدل ریاضی سلول عصبی وارد میشوند.

اصولاً هدف از هرگونه طبقه بندی داده ها، تشکیل گروه هایی است که در آنها نقاط مشترکی وجود دارد که در سایر گروه ها کمتر دیده می شود. هر گروه اصطلاحاً یک کلاستر نامیده می شود.

الگوریتم های طبقه بندی از قوانین تجمع هوشمند استفاده می کنند که تاکید بر توزیع، قابلیت انعطاف و قدرتمند بودن دارند. این الگوریتم ها مستقیم یا غیر مستقیم با پارامتر های ساده در ارتباط بوده و از آن ها فیدبک می گیرند.

مثلاً در الگوریتم مورچگان هر مورچه منحصر به فرد همانند یک المان ساده با حافظه محدود رفتار می کند. این مورچه ها، با رفتار تصادفی و حافظه محدود خود طوری عمل می کنند که می توانند کارهای پیچیده ای انجام دهند. طبقه بندی مبتنی بر رفتار مورچه، بر مبنای سیستم قبرستان و مرتب سازی لاروهایشان استوار است. در کلونی های مورچه، کارگران برای تمیز کردن لانه توده هایی از اجساد تشکیل می دهند. وجود این توده ها کارگران را تشویق می کند که اجساد بیشتری جمع کرده و توده های کوچک را بزرگ تر کنند و این فیدبک مثبت در نهایت سبب تشکیل کلاسترهای بزرگ تر می شود. یکی از عظیم ترین امور در کلونی های مورچه منظم کردن نوزادان است. مورچه های کارگر لاروها را بر اساس اندازه شان جمع آوری می کنند، به این صورت که آنقدر آنها را جابجا می کنند تا لاروهای با اندازه های مشابه در یک دسته قرار گیرند.

داده ها با استفاده از نسخه تکرار شونده الگوریتم فازی (FCM) طبقه بندی می شوند، ولی اشکال این روش حساسیت زیاد آن نسبت به مرکز کلاستر اولیه است. برای رفع این مشکل و عمومی تر کردن عملیات از مراکز دسته هایی که توسط الگوریتم مورچه به دست می آیند به جای خود دسته ها استفاده می کنیم.

مرحله اول الگوریتم اصلی مورد استفاده

در آغاز کار اشیاء بر روی یک صفحه دو بعدی پراکنده می شوند. صفحه را به صورت یک ماتریس $m \times n$ سلولی در نظر می گیریم. این ماتریس طوری است که به مورچه ها اجازه می دهد به راحتی جابجا شوند. اندازه صفحه بستگی به تعداد اشیاء دارد. در این جا صفحه ای $m \times n$ انتخاب شده که $m^2 = 4n$ است و n تعداد کل اشیایی است که می خواهیم طبقه بندی کنیم. ابتدا مورچه ها به طور تصادفی بر روی صفحه پراکنده می شوند. تعداد مورچه ها یک سوم تعداد اشیاء است. مورچه ها اشیاء را طبقه بندی کرده و توده ها را می سازند. هر توده، به صورت اجتماعی از دو یا بیش از دو شیء تعریف می شود و در یک سلول خاص قرار می گیرد. برای ساختن الگوریتم طبقه بندی از پارامترهای زیر برای هر توده استفاده می شود. اگر یک توده را H نامیده و n_H تعداد اشیاء آن باشد، می توان پارامترها را به این صورت تعریف نمود

بیشترین فاصله میان دو شیء در توده:

$$D_{\max}(H) = \max (x_i, x_j) ; \quad x_i, x_j \in H \quad (1)$$

مرکز جرم همه اشیاء در توده:

$$\sum O_i ; \quad O_i \in H \quad O_{\text{center}}(H) = (1/n_H) \quad (2)$$

شیء ای در یک توده که کمترین شباهت را به بقیه دارد، $O_{\text{dissim}}(H)$ است. در واقع این شیء دورترین شیء از مرکز توده می باشد.

میانگین فاصله میان اشیاء H و مرکز جرم توده:

$$\sum D(O_i, O_{center}(H)) ; \quad O_i \in HD_{mean}(H) = (1/n_H) \quad (3)$$

الگوریتم اصلی کار به صورت زیر است:

۱- مورچه‌ها و اشیاء به صورت تصادفی روی صفحه پراکنده می‌شوند و هر شیء حداکثر در یک سلول قرار می‌گیرد.

۲- برای هر مورچه مراحل زیر انجام می‌شود:

۱-۲ حرکت مورچه

۲-۲ در صورتی که مورچه باری حمل نکرده و اگر در ۸ سلول همسایه او شیءای وجود دارد، مورچه می‌تواند آن را بردارد.

۳-۲ در غیر این صورت با توجه به ۸ سلول اطراف خود می‌تواند بار را زمین نهد.

۳- رسیدن به معیار توقف

گرچه در ابتدای کار مورچه‌ها به طور تصادفی پراکنده می‌شوند، اما حرکت آنها کاملاً تصادفی نیست. ابتدا مورچه یک مسیر تصادفی را برمی‌گزیند سپس با احتمال $P_{direction}$ این مسیر را ادامه داده و در غیر این صورت مسیر جدیدی را انتخاب می‌کند. در مسیر جدید مورچه می‌تواند چیزی را بلند کرده و یا اگر باری حمل می‌کند آن را بیاندازد. معیار توقف مورچه‌ها در این الگوریتم، رسیدن به حداکثر دفعات تکرار حلقه است.

الگوریتم برداشتن یک شیء :

همانطور که گفته شد مورچه‌ای که چیزی حمل نمی‌کند، در مسیر خود و با توجه به ۸ خانه اطراف می‌تواند باری را بردارد. روش خودآموز برداشتن اشیاء بستگی به تعداد اشیاء درون توده دارد. سه حالت ممکن است بوجود آید: فقط یک شیء باشد، یک توده دو عضوی باشد و یا یک توده با بیش از دو شیء. اگر یک شیء تنها موجود باشد، مورچه حتماً آن را برمی‌دارد. اگر دو تا باشد، به احتمال $P_{destroy}$ با برداشتن تصادفی یکی از دو شیء آن توده را خراب می‌کند. در مورد سوم، مورچه شیء ای را که شباهت کمتری به بقیه اشیاء دارد برمی‌دارد، مشروط بر اینکه عدم شباهت از حد آستانه مورد نظر (T_{remove}) بیشتر باشد. الگوریتم برداشتن شیء به صورت زیر است:

۱- سلول اطراف شیء به عنوان سلول‌های "جستجو نشده" نامیده می‌شوند.

۲- تکرار مراحل زیر

۱-۲. سلول "جستجو نشده" بعدی در اطراف مورچه در نظر گرفته می‌شود.

۲-۲ اگر سلول خالی نبود به صورت زیر عمل خواهد شد:

اگر یک شیء X وجود داشت به احتمال P_{load} آن را برمی‌دارد.

اگر دو شیء وجود داشت، توده به احتمال $P_{destroy}$ خراب خواهد شد.

اگر سلول بیش از دو شیء داشت، عضوی برداشته می‌شود که شباهت کمتری به بقیه اعضا دارد. به این شرط که رابطه زیر برقرار باشد:

$$T_{remove} > D(O_{dissim}(H), O_{center}(H)) / D_{mean}(H) \quad (4)$$

۳-۲ سلول را به عنوان "جستجو نشده" برچسب می‌نهم.

۳- این روند تا وقتی ادامه پیدا می‌کند که همه سلول‌های مجاور جستجو شده و یا یک شیء برداشته شود.

الگوریتم جایگذاری یک شیء :

وقتی مورچه چیزی را حمل می‌کند، ۸ سلول اطرافش را بررسی می‌کند، در این مورد ممکن است سه حالت اتفاق بیافتد: سلول خالی باشد، سلول یک عضو داشته باشد و حالت سوم اینکه سلول مورد نظر محتوی یک توده باشد. در حالت اول فقط

یک احتمال وجود دارد و آن انداختن شی در سلول خالی است، در حالت دوم ممکن است که با انداختن شی درون سلول یک توده بوجود بیاید و در حالت سوم مورچه بار خودش را، اگر از بقیه به توده H نزدیکتر باشد، به آن اضافه می کند.

الگوریتم جایگذاری شی به صورت زیر است:

۱- هشت سلول اطراف مورچه را " جستجو نشده " تلقی می کنیم.

۲- تکرار مراحل زیر

۱-۲ سلول " جستجو نشده " بعدی مجاور مورچه را در نظر می گیریم.

۲-۲ اگر سلول خالی باشد، شی حمل شده توسط مورچه به احتمال P_{drop} در آن انداخته می شود. در غیر این صورت:
اگر سلول یک شی X داشته باشد، مورچه با انداختن بار خود، X' ، درون آن سلول یک توده تشکیل خواهد داد، به این شرط که:

$$[D(x, x') / D_{max}] < T_{create} \quad (5)$$

اگر توده حاوی یک شی بوده و شرط زیر نیز برقرار باشد، شی X به این توده اضافه می شود.

$$D(x, O_{center}(H)) < D(O_{dissim}(H), O_{center}(H)) \quad (6)$$

۳- نام سلول (c) را به " جستجو شده " تغییر می دهیم.

الگوریتم تا وقتی ادامه پیدا می کند که همه سلول ها جستجو شده و یا شی X جایگذاری شود.

۴- الگوریتم تا وقتی ادامه پیدا می کند که همه سلول ها جستجو شده و یا شی X جایگذاری شود.

مرحله دوم آگوریتم اصلی

در الگوریتم مورد بحث اگر در توده ای یک شی نامناسب وجود داشته باشد، مدت زمان زیادی طول می کشد تا به یک توده / کلاستر مناسب تر منتقل شود. یک مساله در الگوریتم مورچه با کمک الگوریتم FCM این است که تعداد کلاس های ایجاد شده بیش از حد انتظار است. توده های کوچک بسیاری شکل می گیرند که ما از آنها به عنوان بلوک هایی برای ساختن توده های بزرگ تر استفاده می کنیم.

در مرحله دوم فرض بر این است که توده هایی که در مرحله قبلی تشکیل شده اند به طور کامل در صفحه دو بعدی مورد نظر قرار دارند، پس مورچه ها کل توده و اشیای درون آن را حمل می کنند. الگوریتم برداشتن یک توده کاملاً مشابه الگوریتم برداشتن اشیاء است. مورچه ها توده را با همان احتمال P_{load} برمی دارند. اگر شرط زیر برقرار گردد مورچه ها توده H_1 را برداشته و روی توده H_2 می گذارند.

$$D(O_{center}(H_1), O_{center}(H_2)) / D_{max} < T_{createforheap} \quad (7)$$

اگر دو توده H_1 و H_2 با هم ترکیب شوند، توده H_3 را تشکیل داده و جدا نمی شوند. در ادامه الگوریتم این عمل را توضیح می دهیم:

۱- اشیاء را تصادفی روی صفحه پخش می کنیم.

۲- مورچه ها از مسیرها و مبدا های تصادفی به حرکت در می آیند.

۳- عملیات زیر ۱۰۰۰ بار تکرار می شود:

۱-۳ برای هر مورچه این عمل انجام می شود:

مورچه حرکت می کند.

اگر مورچه شی X را حمل کند، احتمال دارد آن را بیاندازد. در غیر این صورت:

شاید شیء X را بردارد.

۴- برای آغاز مراکز کلاسترهای الگوریتم FCM از مراکز کلاسترهای به دست آمده در دستورالعمل شماره سه (گام سوم همین الگوریتم) استفاده می کنیم.

۵- به کمک الگوریتم FCM داده ها را طبقه بندی می کنیم.

۶- برای تشکیل توده های جدید از الگوریتم FCM، تثبیت داده های به دست آمده و با معیار بیشترین عضویت کمک می گیریم.

مراحل ۱ تا ۶ را با فرض اینکه هر توده یک شیء تنهاست تکرار می کنیم

نتایج تجربی

الگوریتم مورد نظر برای سه مجموعه کوچک از داده ها آزمایش شده است: پایگاه داده گیاه زنبق، تشخیص نوشیدنی و شناسایی شیشه.

الف - بانک داده گیاه زنبق : این مجموعه داده شامل اطلاعات مختلفی در مورد انواع گل های زنبق است. این بانک ۱۵۰ نمونه دارد که هریک چهار صفت قابل پیشگویی دارند. داده ها در سه کلاس تنظیم شده که هر کلاس ۵۰ عضو دارد و در ضمن هر کدام از کلاس ها مربوط به یک گونه از گیاه زنبق می باشد. یکی از کلاس ها را می توان به طور خطی از دوتای دیگر جدا کرد، کلاس های دیگر تا حدودی اشتراک دارند.

ب - بانک داده تشخیص نوشیدنی: این داده ها نتیجه بررسی های شیمیایی بر روی نوشیدنیها است که در یک منطقه تولید شده، اما از سه کارخانه متفاوت خارج میشود. بانک داده های آن مجموعاً ۱۷۸ نمونه دارد که هریک دارای ۱۳ ویژگی می باشند. داده ها شامل سه کلاس با توزیع مقابل هستند: کلاس ۱: ۵۹ نمونه، کلاس ۲: ۷۱ نمونه و کلاس ۳: ۴۸ نمونه دارد.

پ - بانک داده شناسایی شیشه : مجموعاً ۲۱۴ نمونه بوده و هریک ۹ صفت دارند. آنها در قالب شش کلاس از شیشه طبقه بندی شده اند.

نتایج به دست آمده حاصل ۵۰ بار اجرای آزمایش هستند که در هر تکرار، اشیاء و مورچه ها دوباره در محل های متفاوت و به صورت تصادفی روی صفحه قرار گرفته اند. این آزمایشات به کمک سه مجموعه از پارامترها انجام شده که تعداد توده های به دست آمده با تغییر پارامترها کنترل می شود. در هر آزمایش یک پارامتر را تغییر داده و بقیه را ثابت نگه می داریم. پارامترها و محدوده آنها در جدول ۱ نشان داده شده است. (بجز پارامتر $P_{createforheap}$ که برای هر آزمایش، جداگانه تعریف شده است). میانگین حاصل از ۵۰ بار اجرای آزمایش ها به کمک الگوریتم FCM نیز در جدول ۲ داده شده است.

جدول شماره ۱- پارامترها و محدوده آنها

پارامتر	محدوده
تعداد تکرار	۱۰۰۰
T_{create}	۰/۵
P_{drop}	۰/۲
$P_{destroy}$	۰/۳
P_{load}	۰/۳
T_{remove} (زنبق)	۱/۵
T_{remove} (نوشیدنی)	۳/۰
T_{remove} (شیشه)	۲/۰

جدول شماره ۲- میانگین حاصل از ۵۰ بار

خطاها	کلاس ها	مجموعه داده
۱۶	۳	زنبق
۵۶	۳	نوشیدنی
۴۸	۶	شیشه

آزمایش

نتایج به دست آمده در مورد زنبق در جدول شماره ۳ نشان داده شده است. همانطور که دیده می شود با افزایش $T_{createforheap}$ ، کلاسترهای کمتری ایجاد می شوند.

جدول شماره ۳- نتایج مربوطه در مورد زنبق

خطاهای بعد از FCM دوم	خطاهای قبل از FCM دوم	کلاس های ایجاد شده	$T_{createforheap}$
۴۴/۵۶	۵۰/۰۲	۲/۱۶	۰/۱۸
۲۹	۴۷/۲۶	۳	۰/۱۴
۱۷/۷	۳۸/۱	۴/۰۴	۰/۱۲

جدول شماره ۴ نتایج مربوط به پایگاه داده زنبق را با فرض دو کلاستر و سه تنظیم پارامتری مختلف، نشان می دهد.

جدول شماره ۴- نتایج مربوط به پایگاه داده زنبق

خطاهای بعد از FCM دوم	خطاهای قبل از FCM دوم	کلاس های ایجاد شده	$T_{createforheap}$
۲/۷	۵/۲۲	۱/۲	۰/۲
۲/۴۶	۱/۳۲	۲/۱۸	۰/۱۸
۰	۰	۴/۰۴	۰/۱۲

نتایج به دست آمده برای نوشیدنی در جدول ۵ آمده است.

جدول شماره ۵- مربوطه برای نوشیدنی

خطاهای بعد از FCM دوم	خطاهای قبل از FCM دوم	کلاس های ایجاد شده	$T_{createforheap}$
۶۱/۶۴	۶۵/۸۲	۲/۲	۰/۰۷۵
۵۴/۴۸	۵۸/۱	۳/۶	۰/۰۸۵
۵۱/۹۶	۵۶/۵	۴/۶۴	۰/۱

نتایج مربوط به بانک اطلاعاتی شیشه را می توان مطابق جدول شماره ۶ بیان نمود. این داده ها به دو گروه عمده تقسیم می شوند: شیشه های پنجره ای و غیر پنجره ای. اگر فرض کنیم که فقط همین دو کلاس وجود دارند، نتایج بهتری به دست خواهد آمد. الگوریتم FCM با شروع تصادفی در این مورد ۲۰ خطا دارد. نتایج، برای سه آزمایش با تنظیمات پارامتری مختلف در جدول ۷ ارائه شده است.

جدول شماره ۶- نتایج مربوط به بانک اطلاعاتی شیشه

خطاهای بعد از FCM دوم	خطاهای قبل از FCM دوم	کلاس های ایجاد شده	$T_{createforheap}$
۱۱۰/۶۶	۱۳۶/۵۶	۲/۷	۰/۸
۹۴/۶۲	۱۱۶/۷۶	۴/۷۸	۰/۱۲

۰/۱۶	۷/۷۸	۱۰۳/۸	۸۲/۳۶
------	------	-------	-------

جدول شماره ۷- نتایج مربوطه سه آزمایش با تنظیمات پارامتری

$T_{createforheap}$	کلاس‌های ایجاد شده	خطاهای قبل از FCM دوم	خطاهای بعد از FCM دوم
۰/۱۲	۴/۷۸	۲۵/۵۶	۱۹/۲۸
۰/۱۶	۲/۷	۴۹/۵۸	۲۱/۳
۰/۱۸	۲/۱۸	۴۹/۹	۲۷/۲۲

نتیجه و جمع‌بندی

استفاده از مورچه‌ها در فرایند طبقه‌بندی داده یکی از روش‌های یافتن تعداد کلاسترهاست. مورچه‌ها هنگامی که درمورد اضافه یا حذف کردن چیزی در توده‌ها تصمیم‌گیری می‌کنند، نسبت به آستانه بسیار حساس هستند. کارهای ابتدایی در این مبحث، مقادیر را تولید کرده اما هیچ توجیه یا روشی برای منظم کردن آنها ارائه نمی‌دهد. ما در اینجا بازه‌ای از مقادیر را به‌دست آوردیم (اگرچه برای یک پارامتر) و نشان دادیم که نتایج چگونه تغییر می‌کنند. هنوز روش سیستماتیکی برای تنظیم پارامترها به‌دست نیامده‌است. نتیجه به دست آمده برای داده‌های گیاه زنبق با وجود سه کلاس، در بخش آخر، همیشه معادل با چیزی است که از روش FCM به‌دست می‌آوریم. البته این مقدار متوسط بوده و ممکن است که گاهی ۲ و یا ۴ کلاس داشته باشیم. مورچه‌ها اصولاً تعداد کلاسترها و شروع FCM را می‌یابند و واقعاً نتیجه نهایی را ایجاد نمی‌کنند. یکی از مشکلات مورچه‌ها در ایجاد آنچه نتیجه نهایی نام دارد، این است که پس از ترکیب توده‌ها با هم نمی‌توان چیزی از آنها برداشت و این مشکل ساز است. آزمایشات نشان می‌دهند که طبقه‌بندی، شدیداً نسبت به مقادیر T_{create} و $T_{createforheap}$ حساس است و این موضوع بر تعداد کلاسترهایی که به‌دست می‌آیند اثر بسیار زیادی خواهد داشت. در آینده می‌توان به موضوع تنظیم خودکار پارامترها پرداخت، و یا اینکه پی برد از چه روشی می‌توان تعداد دقیق کلاسترها را پیدا کرد. همچنین به مورچه‌ها اجازه داد اجزایی از توده‌ها را بردارند و با این کار آنها را واداشت که آستانه‌ها را به‌صورت فازی تعریف کنند.

تشکر و قدردانی :

با سپاس فراوان از معاونت محترم شرکت سهامی ذوب آهن اصفهان مهندس قانونی که امکان این تحقیق را فراهم آوردند.

منابع

- [1] P. M. Kanade and L. O. Hall, "Fuzzy ants as a clustering concept," North American Fuzzy Information Processing Society, NAFIPS 2003, 22nd International Conference of the, pp. 227-232, 2003.
- [2] N. Monmarché, M. Slimane, and G. Venturini, "On improving clustering in numerical databases with artificial ants," in 5th European Conference on Artificial Life (ECAL'99), Lecture Notes in Artificial Intelligence, D. Floreano, J. Nicoud, and F. Mondala, Eds., vol. 1674. Lausanne, Switzerland: Springer-Verlag, Sep 1999, pp. 626-635.

- [3] N. Monmarché, M. Slimane, and G. Venturini , “AntClass : discovery of clusters in numeric data by an hybridization of an ant colony with k_means algorithm”, internal report no. 213, Laboratoire d’Informatique de l’Université de January 1999.
- [4]C. K. Tse, F. C. M. Lau, K. Y. Cheong, and S. F. Hau, "Return-map-based approaches for noncoherent detection in chaotic digital communications," IEEE Transactions on Circuits and Systems I, vol. 49, pp. 1495-1499, 2002
- [5] Wang professor Robert J. Scalkoff “Artificial Neural Networks” McGraw-Hill 1997