

근사적 클러스터링에 의한 다중 전극 활동 전위 분류

안 중 훈

연세대학교 컴퓨터과학과

jonghune@hanmail.net

Multi-electrode Spike Sorting by Approximate Clustering

Jong-Hoon Ahn

Yonsei University, Department of Computer Science

요 약

다중 전극으로 측정된 활동 전위의 분류(Multi-electrode spike sorting)는 단일 전극(single-electrode)보다 더 정확한 결과를 보여준다. 그러나 다중 전극에서 주어지는 활동 전위 크기들의 클러스터는 일반적으로 분류하기 쉽지 않은 문제이다. 이 논문에서는 고전적인 클러스터링 알고리즘 중의 하나인 Mountain method 를 수정하여 다중 전극 활동 전위의 분류에 적합한 알고리즘을 제안한다. 통상적인 데이터 클러스터링이 아닌 공간 분할을 통해 신경 데이터의 다양한 클러스터에 대해서 적응도가 높아지고 빠른 분류를 하게 된다.

1. 서 론

인간의 행동과 인지 활동들은 기본적으로 뇌의 작용과 밀접한 관련을 맺고 있다. 뇌의 작용이라는 것은, 세포 수준에서 보면, 무수히 많은 신경 세포들 사이에서 일어나는 활동 전위(neural spike)들의 복잡한 처리 과정이다. 활동 전위는 신경 세포막에 전도성이 있어서 시냅스(synapse)를 통해 다른 신경 세포에 영향을 줄 수 있는 특별한 전위이고 형태는 세포마다 비교적 안정적이고 고유하다. 이 활동 전위들이 뇌의 정보 처리에 중요한 기반이 되고 있음은 일반적으로 받아들여지고 있다.[13] 뇌 연구 작업은 일차적으로 현상론적인 이해에 목표를 두고 있다. 뇌의 특정 영역들의 역할 및 그것들 사이의 정보 처리 관계를 규명하고 이론적 해석 및 모델을 만들어 본질적인 면을 알고자 하는 것이다. 하지만 뇌 연구의 기본적인 작업은 측정을 통해 이루어진다. 신경 생리 실험과 관련한 측정의 문제는 측정 도구, 방법, 정량화된 관찰, 적절한 알고리즘이 요구된다. 측정의 방법에 따라 우리가 볼 수 있는 뇌의 모습은 다르다. 바꾸어 말하면 우리가 연구하려는 대상에 따라 측정 방법들을 달리 해야 한다. 세포 크기의 수준에서 접근할 때 많이 사용되는 도구는 단일 전극(single electrode)이다. 끝이 뾰족하고 매우 가는 모양이고 100 마이크로미터 이내에 있는 3~7개의 신경 세포 활동 전위들을 볼 수 있다. 여기서 관찰하려는 것은 특정 신경 세포가 활동 전위를 일으킨 시간들이다. 하지만 전극을 통해 기록되는 것은 비슷한 활동 전위와 전기적인 노이즈이기 때문에 관측된 신호가 어떤 세포에서 온 것인지 알 수가 없다. 따라서 분류(spike sorting) 작업이 필요하다. 분류

작업은 세포마다 만들어내는 활동 전위는 어떤 고유한 성질을 가진다는 사실에 근거를 두고 특징 벡터 분석(feature vector analysis)과 클러스터링(clustering)을 공통적으로 사용하여 분류한다.[1,4,9] 그러나 어떤 특징 벡터와 클러스터링 알고리즘을 쓰더라도 노이즈로 인해 정확도에는 한계가 있다. 이 한계를 극복할 수 있는 한 가지 방법은 다중 전극(multi-electrode)을 쓰는 것이다. 그 중 4개를 묶어 쓰며, 테트로우드(tetrode)라고 불린다. 이 때 쓰는 특징 벡터는 각 전극을 통해 기록된 전위의 크기(amplitude)들이 된다. 측정 공간에서 신경 세포들과 전극의 각 끝들은 고정되어 있고 서로 상대적인 위치에 따라 각 세포의 신호는 다중 전극에 대해 고유한 신호 값의 비(ratio)를 갖게 되고 전위 데이터 공간에서 고유한 영역을 차지한다. 즉 특정 세포에 가까이 있는 전극은 다른 전극보다 큰 전위 값을 기록하게 될 것이며, 클러스터링을 통해 같은 집합으로 분류된 신호들은 동일한 신경 세포에서 온 것으로 가정한다. 따라서 단일 전극을 쓸 때 필요했던 특징 벡터 분석(feature vector analysis) 과정을 생략하거나 축소할 수 있게 되며, 상대적으로 매우 높은 정확도를 보인다.[2,6] 테트로우드를 통해 얻어진 전위의 값(amplitude)들이 형성한 클러스터(cluster)는 일반적으로 정상 분포(normal distribution)와 비슷하다. 하지만 비정상(non-Gaussian) 분포이면서 동시에 비등방형(anisotropic) 분포인 경우도 많다.[3] 흔히 클러스터 모양이 길며, 일부 겹친다. 이러한 클러스터들을 잘 분류하기 위해 흔히 쓰이는 방법은 실험자가 직접 눈으로 보고 판단하여 분류하는 방법이다. 사용자-컴퓨터 인터페이스에서 주어진 클러스

터들의 그림을 마우스로 클릭하여 각 클러스터의 영역을 잘라내어 분류하는 방법이 주로 사용되었다. 이 경우 신뢰도는 높지만, 부수적인 일을 줄이고 보다 객관적으로 분류하기 위해 자동화 알고리즘이 필요하게 된다. 기존에 많이 사용되는 알고리즘으로는 베이저안 클러스터링 알고리즘 (Bayesian clustering algorithm)이 있다. 하지만 많은 경우 클러스터가 비정상 (abnormal) 구조를 이루고 있어서 한계를 지니고 있다. 또 클러스터의 모양이 길고 중심 사이의 거리가 가깝다면(그림 3), K-means와 같은 고전적인 클러스터링 방법은 사용될 수 없다. 특수한 모양의 클러스터에 대해 적용될 수 있는 클러스터링 알고리즘들이 있기는 하나 계산 시간이 길어져서, 빠른 분류 결과를 얻어야 하는 경우에는 쓸 수 없다. 더구나 실시간 분류가 요구되는 때도 있어 보편적으로 이용하기는 어렵다.[4,5] 이 논문에서는 특수한 모양의 클러스터 구조와 실험자들의 요구에 잘 맞는 새로운 활동 전위 분류 알고리즘을 제안한다. 계산 시간이 적고 클러스터 모양에 대한 파라미터(parameter) 모형을 가정하지 않으며, 사람의 시각에 의한 분류와 그 결과가 흡사하다. 논문의 나머지 부분에서는 제안되는 알고리즘을 설명하고 수동으로 분류할 때와의 비교 결과를 분류하기 쉬운 클러스터와 어려운 클러스터의 경우로 나누어 결과를 제시하고 설명하겠다.

2. Mountain Method(MM)

(1) Mountain 함수의 개념

s 차원의 연속 공간 R^s 에 n 개의 점으로 이루어진 집합을 생각하자. 그리고 k 번째 점의 j 번째 좌표를 x_{kj} 라고 하자. 전체 공간 R^s 대신 n 개의 점 집합을 포함하는 s 차원 공간 $I_1 \times \dots \times I_s$ 위로 작업 공간을 축소하여도 일반성을 잃지 않는다. $I_j = [\min(x_{kj}), \max(x_{kj})]$ 이다. 각 구간 I_j 를 r_j 개로 같은 간격으로 나누면 s 개의 격자를 형성하게 되고, 그 점들의 새로운 좌표 X_{kj} 은 $\{1, \dots, r_j\}$ 중의 하나이다. 그러면 Mountain 함수의 연산은 연속 공간 R^s 에서 불연속 공간 G^s 로 옮겨지게 된다. G^s 는 점 집합들을 포함하게 되는 최소 크기로 정의가 되며 각 격자점은 자연수 집합인 $N_{(i_1, \dots, i_s)}$ 으로 표시된다. 이러한 불연속적 변환을 하게 됨으로써 계산 시간이 줄어들게 되지만 기본적으로 근사적인 방법이 될 수밖에 없음을 의미한다.[10,11]

Mountain 함수는 각 점들이 에너지원처럼 작용하는 것

을 가정하고 착안한 것이다. 질량이나 전하량과 같은 물리량을 갖는 물질 근처는 퍼텐셜이 높은 것처럼 임의의 점 근처에 가까이 갈수록 퍼텐셜은 높아진다. 바꾸어 말하면 특정 격자점 근처에 점들이 많아지면 그 격자점의 퍼텐셜은 높아진다. 하지만 Mountain 함수는 질량이나 전하량을 갖는 물질처럼 역 제곱의 힘을 정의하지 않고 지수 함수를 쓴다. 그리고 격자점과 주어진 점들의 거리는 p -norm을 쓴다. 따라서 격자 $N_{(i_1, \dots, i_s)}$ 에서 Mountain 함수는 다음과 같이 정의된다.

$$M(N_i) = \sum_{k=1}^n \exp[-\alpha d(x_k, N_i)],$$

$$d(x_k, N_i) = \left(\sum_{k=1}^s |x_k - N_i|^p \right)^{1/p}$$

(2) 클러스터 중심 계산

클러스터들의 중심은 Mountain 함수의 국소 최대값의 개수가 전체 클러스터의 수와 같다는 가정으로부터 찾는다. 클러스터가 단일 구조를 갖는 문제에서라면 타당한 가정이 된다. 그래서 다른 클러스터링 알고리즘과 결합되었을 때 클러스터 수에 대한 초기값을 잘 결정해줄 수 있다. 구체적인 과정은 다음과 같다.

우선 Mountain 함수에서 전체 최대값 $M_1^* = \max[M(N_i^*)]$

을 가지는 N_1^* 를 찾아낸다. 여러 개의 국소 최대값이 있지만 그 중 최대가 되는 격자점을 찾아낸다. 특정 격자가 국소 최대가 되는지 판단하는 것은 주위의 함수값들과 비교하는 매우 간단한 연산이다. 첫 번째 N_1^* 는 하나의 클러스터 중심의 인덱스로 근사된다. 그리고 첫 번째 클러스터의 퍼텐셜에 의한 영향을 상쇄시키기 위해 Mountain 함수를 새롭게 갱신할 필요가 있다.

$$M_{new}(N_i) = M(N_i) - M_1^* \sum_{k=1}^n \exp[-\beta d(N_1^*, N_i)]$$

Mountain 함수가 소멸해버릴 때까지 위의 과정을 반복해서 클러스터의 중심을 얻는다.

MM(Mountain Method)은 자동 패턴 분류기를 제작할 때 이용되는 고전적인 개념의 근사적 클러스터링 방법이다. MM만으로는 완성도가 낮은 알고리즘이기 는 하지만 클러스터의 위치를 찾기 위한 에너지 함수(objective function)를 정의하는데 유용하게 쓰일 수 있다. 클러스터의 중심을 찾는 과정이 사람의 시각 작용과 효과가 비슷하고 적은 개수의 데이터에 대해서는 상당히 빠른 직관적인 알고리즘이다.

3. Modified Mountain Method (MMM)

표1 modified Mountain function

	potential range	potential form
Mountain function	all region	exponential
discrete Parzen window	pixel	Dirac delta

MM은 독립된 클러스터링 알고리즘이라기 보다 FCM(fuzzy C-means)과 같은 보편적으로 사용되는 알고리즘의 보조적인 알고리즘으로써 사용되어졌다. 하지만 MMM(Modified Mountain Method)는 스파이크 분류와 같은 대용량 데이터의 분류에 적합한 알고리즘이다. 다음과 같이 MMM으로 수정하도록 제안한다.

(1)discrete Parzen window

데이터 밀도를 연속적으로 부드럽게 얻을 수 있는 정형화된 방법이 Parzen window이다. 이것을 격자 위에서 정의하게 되면 discrete Parzen window가 된다. MM에서는 각 점들에 의한 퍼텐셜을 지수 함수로 계산하였는데 MMM에서는 점들의 퍼텐셜을 갖게 되는 범위를 Parzen window 함수로 제한하도록 했다. MM에서의 퍼텐셜 계산은 데이터 밀도가 연속적으로 부드럽게 연결되고 국소 최대값 수가 가능한 적어지도록 하기 위한 과정에 불과하므로 MMM에서는 계산 시간을 보다 줄일 수 있도록 퍼텐셜 범위를 Parzen window 함수로 수정되게 한다.

격자 위의 Parzen window는 s-차원에서 한 픽셀을 이룬다. 그 픽셀의 한 모서리의 길이를 h_n 이라 하면 부피는 $V_n = h_n^s$ 가 된다. 하나의 픽셀에 포함되는 점의 개수 k_n 은 다음과 같은 window 함수를 정의함으로써 나타낼 수 있다.

$$\phi(\vec{u}) = \begin{cases} \delta(\vec{u}) & \left| \vec{u}_j \right| \leq \frac{1}{2} \\ 0 & otherwise \end{cases}$$

그러면 한 픽셀 내부의 점의 수 k_n 과 격자 위의 데이터 밀도 함수는 다음과 같이 주어진다.

$$k_n(\vec{X}) = \sum_{i=1}^n \int_{cell} \phi\left(\frac{\vec{X} - \vec{x}_i}{h_n}\right),$$

$$p_n(\vec{X}) = \frac{1}{n} \sum_{i=1}^n \frac{1}{V_n} k_n(\vec{X}).$$

Mountain 함수에서 지수함수 대신에 델타 함수를 써서



그림 1 2차원 평면 위로 투영된 전위 공간의 초기 데이터 분포 (N=32000)

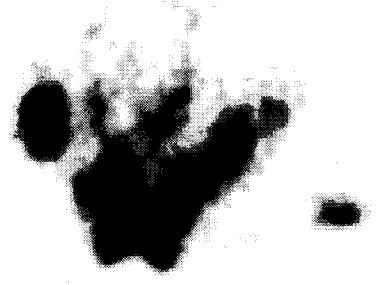


그림 2 Discrete Parzen Window를 통한 클러스터 밀도 분포

한 픽셀 내의 점의 개수를 계산하였고 h_n 의 값이 커짐에 따라 분포는 매끄러워지고 작아지면 거칠게 드러난다. 결론적으로, 수정된 부분을 정리하면 다음과 같다. 퍼텐셜 범위를 모든 영역에 대해 계산하던 것을 길이 h_n 의 픽셀 내부로 축소시킴으로써 계산량이 대폭 줄었다.

퍼텐셜 형태를 지수 함수에서 델타 함수로 변경했기 때문에 단순히 픽셀 내부의 점의 수만 고려하면 된다. 따라서 Mountain 함수가 각 점을 에너지원으로 보고 격자 위에서 퍼텐셜을 계산하였다면 discrete Parzen window는 사실상 점들의 분포에 대한 확률 밀도 함수를 계산하였다고 할 수 있다. 이러한 수정이 신경 데이터 분류를 위해 타당한 이유는 다음과 같다.

신경 세포들의 활동 전위를 측정하는 실험에서 각 클러스터는 서로 독립이다. 임의의 시간에 한 세포에서 스파이크가 생긴 것은 다음 스파이크에 어떤 영향도 끼치지 못한다. discrete Parzen window는 델타 함수를 쓰고 있으므로 각 점들은 서로 의존도가 없음을 의미하고 있다.

하지만 두 방법 모두 간격 h_n 격자 위에서 계산되므로 근사적인 clustering이다.

(2)국소 최대점

다음 과정은 Mountain method의 기본적인 생각과 동일하다. 즉 격자 집합 위에서 국소 최대점의 개수가 곧 클러스터 또는 클래스(class)의 수라는 가정이다. 그리고

국소 최대점들 사이의 골짜기 부분이 클러스터들 사이의 경계가 된다. 이 기본적인 생각으로부터 매우 빨리 클러스터 중심과 개수를 찾을 수 있게 된다. 하지만 첫 번째 수정으로 인한 추가적인 수정이 뒤따르게 된다. 지수 함수적 의존에서 완전 독립으로 바꾸려는 목적에서 discrete Parzen window로 수정하게 되었으므로 직렬적인(one-by-one) 클러스터 찾기는 그대로 병렬적인 클러스터 찾기로 바뀔 수 있다. MM에서는 클러스터 중심을 하나 찾고 퍼텐셜을 수정하고 또다른 클러스터의 중심을 찾는 방식이었는데, MMM에서는 퍼텐셜을 수정할 필요가 없으니 동시에 클러스터들의 중심을 찾는다.

(3) Tagging and Spreading η

MMM의 가장 큰 특징은 공간 분할이다. 그 결과가 우수하고 빠르며, 매우 많은 데이터에 대해서도 수행된다. 그리고 기존의 분류 방법과는 전혀 다른 근사적인 방법이다. 기존의 방법들이 점 사이의 거리에 근거한 체계적이고 정밀한 방법이라면 이 논문에서는 점 분포의 밀도에 근거한 근사적 방법이라 할 있다. 그리고 점들을 분류하기 전에 공간을 먼저 분할함으로써 데이터 분류를 자동적으로 수행한다. 공간 분할은 반드시 단일 위상적인 분할 (simply-connected topographic division)이 되고 하나의 분할된 공간은 대응되는 클러스터의 분포 영역과 동일하고 분할된 공간의 수는 클러스터의 수와 같다.

이런 작업을 위해서는 각 픽셀에 클러스터 번호 η 를 할당한다. η 는 자연수이고 1부터 번호가 시작하고 전체 공간의 픽셀들을 $\eta = 0$ 으로 초기화한다. 앞의 과정을 통해 국소 최대점을 찾았다면 그 픽셀의 η 에 대한 값을 1부터 순서대로 할당한다. 그리고 그 픽셀 주위의 낮은 확률 밀도값을 갖는 픽셀로 η 값을 퍼뜨리는 연산을 한다. 정확한 표현식은 다음과 같다:

$$\eta_N = \max(\eta_{N_{nei}}).$$

N_{nei} 는 픽셀 N주위의 픽셀들을 말한다. 3차원에서 계산하는 경우라면 N_{nei} 는 $2 * 3 = 6$ 개의 인접한 픽셀들이 있다. 그 주위 픽셀들 중 가장 큰 확률 밀도값을 가지는 픽셀의 η 값을 옮긴다. 이 연산을 통해 각 픽셀들은 서로 같거나 다른 η 를 갖게 되며 고유한 영역을 형성해 나간다. 확률 밀도 함수의 골짜기 부분에서 서로 다른

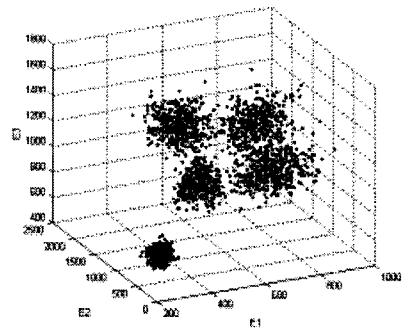


그림 3 Normal case: unclassified distribution (Az:-26 Po:64)

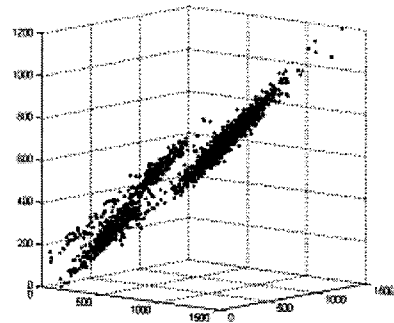


그림 4 Abnormal case: unclassified distribution (Az:39,Po:92)

영역이 마주하게 되고 그 경계까지 η 가 진행되어 더 이상의 확장이 없을 때 영역 분할은 끝나게 된다. 이 때 결정되는 경계는 오로지 확률 밀도 함수와 h_n 에만 의존한다. 유한한 크기의 픽셀 단위로 근사했으므로 경계는 톱니 모양의 들쭉날쭉한 모양이다. 그래서 충분히 작은 픽셀을 사용해서 경계를 부드럽고 연속적으로 만들 필요가 있다. 하지만 너무 작은 픽셀을 사용하면 확률 밀도 함수가 덜 부드러워져서 클러스터 수가 남발될 수 있다. 그래서 적절한 크기의 픽셀을 사용해야 한다.

같은 분할 공간 G_i 에 있는 점들은 동일한 클러스터 번호 η_i 를 가지므로 공간 분할을 통해 최종적으로 분류가 된다. 이 모든 과정을 효율적으로 수행하기 위해서는 데이터의 배열 구조가 중요해진다. 구현할 때 처음에는 점들의 좌표가 주어지고 이것이 격자 위로 일차 변환된다. 이 과정은 discrete Parzen window에서 수행되고 각 격자 또는 픽셀은 확률 밀도값을 저장하게 된다. 그리고 이 확률 밀도값에 근거하여 클러스터 번호 η 를 퍼뜨려

영역을 분할하게 된다. 영역 위의 η 에 근거하여 원래 점들을 분류하게 되는데 이 전체적인 과정에서 두 가지 연결이 필요하다. 첫 번째 연결은 각 점과 픽셀 또는 격자의 인덱스이고 두 번째 연결은 그 인덱스와 η 의 연결이다. 이것을 자연스럽게 구현할 수 있는 방법은 데이터를 다음과 같은 배열 구조로 정리해두는 것이다.

...
coordinates of Nth point	index of pixel N_x	
...

4. 결과 및 토론

표2 normal case result

	cluster 1	cluster 2	cluster 3	cluster 4	cluster 5
exact	400	400	400	400	400
result	410	405	401	388	401

표3 abnormal case result

	cluster 1	cluster 2	cluster 3
manual sorting	1015	550	288
MMM	1018	546	289

지금까지 스파이크 분류 알고리즘으로서 수정된 Mountain method에 대해 설명하였다. 스파이크 분류 알고리즘에 적용되는 MMM은 매우 신속하게 수행되는 특징을 보인다. 그리고 비교적 정확한 분류 결과를 보여주며 클러스터의 구조에 대한 적응도가 높다. 먼저 정상 분포를 이루는 클러스터에 대한 결과는 그림 3~6과 같다. (N=2000, l=20, T=8sec) 그림 3은 분류 성능을 시험하기 위해 직접 만든 클러스터이다. 클러스터 수는 5개이고 각각 400개가 되도록 임의로 만들었다. 각 클러스터는 정상 분포를 이루고 있다. 그림 4은 분류된 클러스터를 색깔로 구분해놓은 것이다. 정상 분포의 경우에는 충분히 정확한 결과를 보여준다. 수치 비교는 표 2와 같다. 하지만 많은 경우에 신경 데이터의 클러스터의 모양은 길고 비정상 구조를 하고 있어 쉽지 않다. 그림 5와 6은 쥐(rat)의 대뇌 신피질(neocortex)에서 얻은 데이터를 분류한 것이다. 그림 7은 분류 시간(T)과 데이터 수(N) 그리고 1차원 당 픽셀 수(l)의 관계를 나타낸다. 데이터의 차원 수가 높지 않을 때 가장 많은 계산량을 요구하는 부분은 discrete Parzen window를 통해 PDF를 계산할 때이다. 하나의 점이 어떤 픽셀에 속하게 되는지 결정하는 것이 계산의 주요 기본 단위이므로 데이터의 수에 비례하게 된다. 그리고 1차원 당 픽셀 수는 상대적으로 계산량에 거의 영향을 끼치지 못한다. 기본

적으로 3D에서 알고리즘을 구현했으므로 1D 픽셀 수의 세제곱에 비례한다.

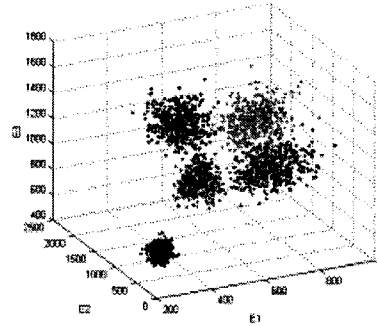


그림 5 Normal case: classified distribution (Az:-26,Po:64)

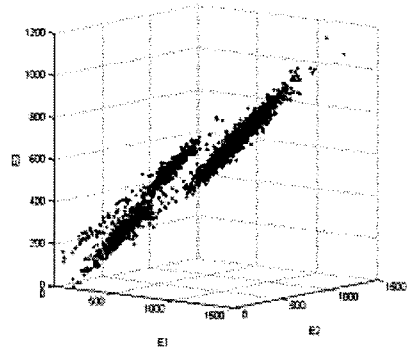


그림 6 Abnormal case: classified distribution (Az:39,Po:92)

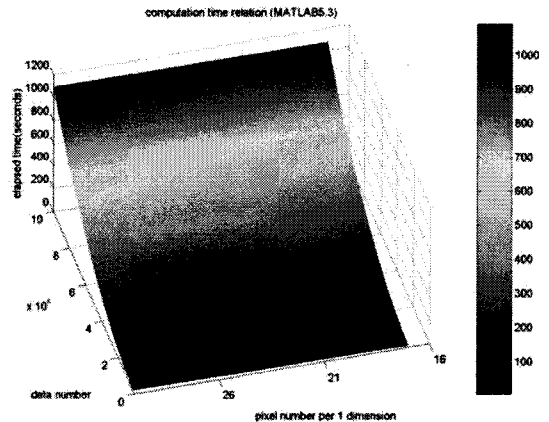


그림 7 Computational requirement

따라서 분류 시간은 다음과 같이 표현된다:

$$T = cN + kI^3$$

($c=0.0007\text{sec/개}$, $k=0.0008\text{sec/개}$, MATLAB5.3)

지금까지 이 논문에서는 스파이크의 분류에 적합한 새로운 클러스터링 알고리즘을 논의하였다. 기본적으로 공간 분할을 통해 데이터 분류를 하였고 근사적인 방법이라는 한계를 지닌다. 그러나 매우 신속하고 신경 데이터의 다양한 클러스터의 모양에 대해 높은 적응도를 보인다. 그리고 다음과 같은 전망과 과제를 동시에 남긴다. 스파이크를 기록하는 실험은 실시간으로 이루어진다. 오프라인에서 파일 입출력을 통한 분류는 큰 제약이 된다. 그러나 알고리즘이 신속하지 않으면 실시간 분류 알고리즘으로 이용될 수 없다. 공간 분할을 통한 분류 방법은 실시간 분류 알고리즘으로 사용될 수 있다.

감사의 글

이 논문은 교육인적자원부지원 연세대학교 BK21 지능형 모바일 서비스를 위한 차세대 단말 소프트웨어 사업단의 지원을 받아 연구되었음

참조 문헌

[1] Michael SL "A review of methods for spike sortig : the detection and classification of neural action potentials" *Network: Comput. Neural Syst.* 9(1998) 53-78

[2] Charles MG, Pedro EM, Mathew W, Bruce M "Tetrodes markedly improve the reliability and yield of multiple single-unit isolation from multi-unit recordings in cat striate cortex" *J. Neurosci. Meth.* 63(1995) 43-54

[3] Michale SF, Partha PM, David K "Automatic sorting of multiple unit neuronal signals in the presence of anisotropic and non-Gaussian variability" *J. Neurosci. Meth.* 69(1996) 175-188

[4] Patricio G, Carmen PS, Javier R, Manuel R "Unsupervised classification of neural spikes with a hybrid multilayer artificial neural network" *J. Neurosci. Meth.* 82(1998) 59-73

[5] R.K. Snider, A.B. Bonds "Classification of non-stationary neural signals" *J. Neurosci. Meth.* 84(1998) 155-166

[6] Sergei PR, Brian DW, Alfred AE, Kenneth DM "Cross-channel correlations in tetrode recordings: implications for spike-sorting" *Neurocomputing* 26-27(1999) 1033-1038

[7] M. Wehr, J.S. Pezaris, M. Sahani "Simultaneous paired intracellular and tetrode recordings for evaluating the performance of spike sorting algorithms" *Neurocomputing* 26-27(1999) 1061-1068

[8] Dmitry R, Hanan D, Naftali T "Multi-electrode spike sorting by clustering transfer functions" NIPS 98

[9] Xiaowei Y, Shihab AS "A totally automated system for the detection and classification of neural spikes" *IEEE Trans. Biomed. Engine.* 35. 10(1988) 806-816

[10] Rajesh ND, Raghu K "Robust clustering methods: A unified view" *IEEE Trans. Fuzzy syst.* 5. 2(1997) 270-293

[11] Ronald RY, Dimitar PF "Approximate clustering via the Mountain method" *IEEE Trans. Syst. Man and Cyber.* 24. 8.(1994) 1279-1284

[12] Eyal H, Ronen S, Yoash S, Morris B, Eshel BJ "Detection and Sorting of neural spikes using Wavelet packets" *Physical rev. lett.* 85. 21(2000) 4637-4640

[13] M. Barinaga "Listening in on the brain" *Science* 280(1998) 376-378