Reprodução do Algoritmo Evolutionary Design of Nearest Prototype Classifiers

Tiago J. dos Santos, Tiago N. Bastos

Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Pernambuco – PE – Brasil
{tjs2,tnb}@cin.ufpe.br

Abstract. *In pattern classification problems, many works have been carried out with the aim of designing good classifiers from different perspectives. These works achieve very good results in many domains. However, in general they are very dependent on some crucial parameters involved in the design. For instance, in nearest prototype approaches, main parameters are the number of prototypes to use, the initial set, and a smoothing parameter. In this work, an evolutionary approach based on Nearest Prototype Classifier (ENPC) is introduced where no parameters are involved.*

Resumo. *Em problemas de classificação de padrões muitos trabalhos foram realizados com o objetivo de criar bons classificadores tendo em vista diferentes perspectivas. Estes trabalhos alcançaram resultados muito bons em vários domínios. Entretanto, em geral, eles são muito dependentes em alguns parâmetros cruciais envolvendo a criação. Por exemplo, em abordagens do tipo, Nearest Prototype, os parâmetros principais são o número de protótipos a serem utilizados no conjunto inicial, e um parâmetro de amortecimento. Nesse trabalho, uma abordagem evolucionária baseada em Nearest Prototype Classifier (ENPC) é introduzida sem nenhum parâmetro envolvido.*

1. Introdução

Nearest Neighbour Classifiers são definidos como os tipos de classificadores que assimilam para cada novo exemplo não rotulado, *v*, o rótulo do protótipo mais próximo, r_i, de um conjunto, C, de N diferentes protótipos previamente classificados (Duda e Hart, 1973). Quando o conjunto C é muito reduzido estes tipos de classificadores podem ser chamados de *Nearest Prototype Classifier* (Bezdek e Kuncheva, 2001) (NPC).

Neste trabalho foi utilizado uma abordagem evolucionária chamada de *Evolutionary Nearest Neighbour classifier* (ENPC). Com esta abordagem é possível, em geral, encontrar o número ótimo de protótipos do classificador bem como a localização destes protótipos. Neste trabalho, os operadores para modificar o tamanho do classificador e o algoritmo de aprendizagem são completamente integrados e não podem ser utilizados separadamente.

Nesse contexto, este método é capaz de obter um bom mapeamento de protótipos sem precisar de nenhuma configuração inicial ou número de protótipos definidos inicialmente. Esta abordagem permite que os protótipos executem vários operadores, como introduzir novos protótipos, mudar a classe do protótipo, com o objetivo de melhorar a precisão global do classificador. Além disso, a execução desses operadores é gerida por probabilidades e pelas características extraídas dos próprios protótipos.

2. Algoritmo

2.1. Conceitos

Para um melhor entendimento deste trabalho, é necessário o conhecimento de alguns conceitos descritos abaixo.

Protótipo Rotulado, r_i. Define cada um dos protótipos do sistema. O protótipo é composto pelo espaço do protótipo(*p*) e pela classe(*s*) a qual o protótipo pertence. Desta forma o protótipo r_i será denotado por (*p*, *s*).

Classificador, C. Conjunto de *N* protótipos.

Padrão, v_r. São os exemplos de treinamento ou teste do sistema. Os padrões juntos compõem o conjunto *V*, assim como os protótipos, são compostos pela tupla espaço e classe.

Classe, s_j. Ambos, os protótipos e os padrões pertencem a uma classe do conjunto *S*.

Qualidade do protótipo, qualidade (r_i*).* É a medida do quanto um protótipo é representativo ao substituir um subconjunto de padrões da base de dados. Para determinar essa informação, é levando em consideração o número de padrões em sua região e quanto desses padrões pertence à mesma classe do protótipo.

2.2. Estrutura

O Sistema pode ser representado por uma matriz bidimensional onde cada linha é associada a um protótipo r_i que pertencem ao conjunto *C* e cada coluna é associada à classe s_j que pertence ao conjunto *S*. Tendo em vista isto, cada posição (*i*, *j*) da matriz é uma estrutura que contem características sobre o exemplo de conjunto de treinamento (V_{ij}) que estão na região r_i e pertence à classe s_j.

A figura 1 mostra um resumo da estrutura utilizada pra manter as informações.

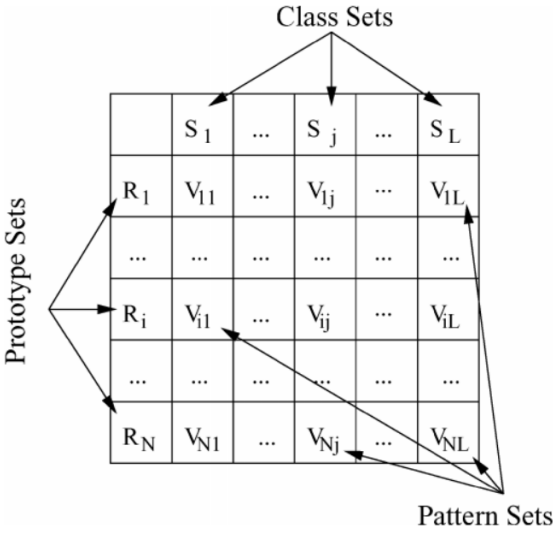

Figura 1. Estrutura do ENPC

2.2.1. Conjunto de classes. A classe S_j é definida como o conjunto de padrões que pertencem à classe s_j. A função de pertinência deste conjunto é a equivalência entre a classe do padrão e a classe associada ao conjunto.

Características que pode ser adquiridas a partir deste conjunto:

- *Regiões* (s_j) É o número de regiões cuja classe do protótipo é s_j. Esta função pode ser calculada a partir da equação abaixo.

$$regiões(s_j) = \sum_{i=1}^{N} \alpha(r_i, s_j), \quad onde\ \alpha = \begin{cases} 1, sse\ r_i = (p, s_j) \\ 0, caso\ contrário \end{cases}$$

- *Expectativa (s_j)* É o número de padrões que qualquer protótipo r_i da classe s_j espera corretamente classificar. A expectativa é dada pela seguinte equação:

$$\text{expectativa}\ (s_j) = \frac{|S_j|}{regiões\ (s_j)}$$

2.2.2. Conjunto de padrões: O conjunto V_{ij} é definido como o conjunto de padrões que estão na região r_i e pertence à classe s_j. Cada conjunto V_{ij} é a intersecção destes conjuntos R_i e S_j.

2.2.3. Conjunto de protótipos: O conjunto R_i é definido como o conjunto de padrões que estão localizados na região r_i. A função de aptidão dessa função é a *Nearest Neighbour rule*.

Para cada protótipo r_i = (*p*, s_j) ∈ *R*, seu classificador de precisão pode ser calculado com a equação que se segue.

$$precisão(r_i) = \frac{|V_{ij}|}{|R_i|}$$

Onde $|V_{ij}|$ é o número de protótipos localizados na região r_i que pertencem a mesma classe que este protótipo.

Da precisão, o valor final da qualidade do protótipo é computado. A qualidade é a relação entre a precisão e o tamanho (em número de padrões) do protótipo. Para calcular o número de protótipos foram levados em conta os padrões colhidos e a expectativa de classe para a qual esse protótipo pertence, essa relação se chama *apportation* e está descrita a seguir.

$$apportation(r_i) = \frac{|V_{ij}|}{\frac{\text{expectativa}\ (s_j)}{2}}$$

Com estes conceitos, é possível calcular a qualidade do protótipo, que será alta se o protótipo classificar corretamente e se classificar uma quantidade suficiente de padrões.

$$qualidade(r_i) = \min\ (1, precisão(r_i) \times\ apportation(r_i))$$

2.3. O Algoritmo

O fluxograma, resumindo os passos que serão explicados nesta seção, do algoritmo *Evolutionary Nearest Prototype Classifiers* é mostrado na figura 2.

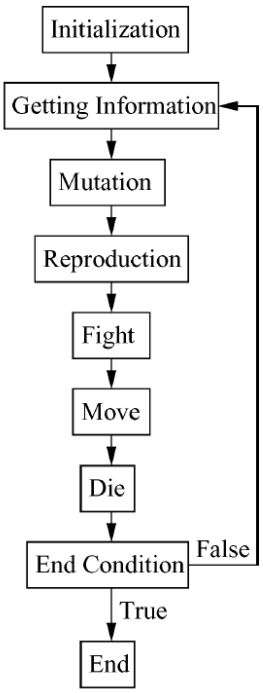

Figura 2. Fluxograma do algoritmo ENPC

2.3.1. Inicialização:

Um das principais características desse algoritmo é a flexibilidade da condição inicial, o que o torna bastante genérico. Tendo em vista este fator, o número inicial de protótipos pode ser um. O local que este protótipo se encontra não é relevante. Isso corrobora com o fato do algoritmo ENPC não possui variáveis de aprendizagem.

2.3.2. Coletando Informações:

No começo de cada interação o algoritmo deve computar a informação requerida para executar os operadores. No final dessa fase todos os padrões devem ter sido introduzidos em conjuntos, e a informação requerida sobre eles deve ter sido computada. É importante destacar que conjuntos anteriores não são realmente introduzidos, então, em vez de introduzir instancias nos conjuntos, a informação sobre esses conjuntos é atualizada como se as instancias tivessem sido introduzidas.

2.3.3. Operador de Mutação:

Este operador rotula cada protótipo com a classe predominante na região. Uma vez obtida as característica cada protótipo conhece o número de padrões de cada classe no local. Depois o protótipo muda de classe se for preciso e se torna da classe predominante. O algoritmo gera um conjunto de agrupamento que leva em consideração somente a distribuição dos dados. Em uma segunda fase, os agrupamentos recebem o rótulo da classe predominante da redondeza.

2.3.4. Operador de Reprodução:

O objetivo desse operador é introduzir novos protótipos no classificador. Essa decisão é feita por cada protótipo no sentido que cada protótipo tem a oportunidade de introduzir um novo protótipo, com o objetivo de aumentar sua própria qualidade. A razão para fornecer aos protótipos essa capacidade é fazer com que estes protótipos tenham a maior quantidade de padrões que pertencem a mesma classes.

Quando adicionar novos protótipos? Cada protótipo r_i, de classe S_j, executa uma roleta. Cada fatia da roleta representa um conjunto $V_{ij'}$, para todos $j' \in S$. O tamanho de cada fatia é proporcional ao número de elementos do conjunto $V_{ij'}$ que ele representa. Se na roleta j for igual a j', não haverá reprodução, caso contrário há uma reprodução e uma nova região $r_{i'}$ é criada para conter os padrões $V_{ij'}$ que é renomeada para $V_{i'j'}$.

2.3.5. Operador de Luta:

Esse operador prove a capacidade de pegar padrões de outras regiões. Formalmente, esse operador permite que um protótipo r_i modifique seus conjuntos V_{ij} a partir do conjunto $V_{i'j}$ de outro protótipo $r_{i'}$, para i diferente de i'.

Passos para o operador de luta:

1. Escolhe um protótipo $r_{i'}$ para lutar contra. Protótipos são escolhidos a partir do conjunto de *vizinhos* (r_i), que define como o conjunto de região que tem uma borda em comum com a região r_i. Para decidir quais protótipos escolher do conjunto de vizinhos, uma roleta é usada para assimilar cada região r_j que pertencem a *vizinhos* (r_i), uma fatia de tamanho proporcional à diferença entre sua qualidade e a qualidade de r_i.
2. Decide se haverá uma luta ou não. Essa probabilidade de luta entre r_i e $r_{i'}$ é proporcional à distância de suas qualidades.
$$PLuta(r_i, r_{i'}) = |qualidade(r_i) - qualidade(r_{i'})|$$
3. Se o protótipo r_i = (p_i, s_i) decide lutar com o protótipo $r_{i'}$ existem 2 possibilidades.
    - Se s_i for diferente de $s_{i'}$ (cooperação). Ambos os protótipos pertencem a classes diferentes, logo, eles dão os padrões que pertencem às classes que são do companheiro.
    - Se s_i for igual a $s_{i'}$ (competição). Nesse caso, padrões podem ser transferidos do conjunto $V_{i's'}$, para o conjunto $V_{ij'}$, ou vice versa, depende do resultado da luta. A luta acontece novamente pelo método da roleta com apenas 2 fatias, onde cada fatia é proporcional a qualidade de cada protótipo. E depois a quantidade de padrões que vão ser transferidos também é de acordo com a probabilidade proporcional a qualidade dos protótipos.

2.3.6. Operador de Movimento:

Realoca cada protótipo no melhor local esperado então cada protótipo $r_i = (p_i, s_j)$ irá se mover para o centroide do conjunto V_{ij}.

2.3.7. Operador de Morte:

A probabilidade de um protótipo ser descartado (morrer) é um menos o dobro da sua qualidade.

2.3.8. Condição de Parada

A condição de parada utilizada foi o número de interações. O usuário define um número máximo iterações e quando atingir esse valor a execução será concluída.

3. Codificação

A codificação do algoritmo ENPC foi realizada utilizando a linguagem Python. O motivo de a utilizarmos está em sua característica multiparadigma, o fato de não ser fortemente tipada, permitindo mais rapidez na codificação, e a possibilidade de uso das bibliotecas Numpy, Matplrolib e Sklearn, bibliotecas bem difundidas e de utilização intuitiva.

3.1. Dificuldades de codificação

Em reproduções de artigos científicos diversas vezes nos deparamos com passos em que não compreendemos a forma de codificar o que foi escrito. Isso aconteceu conosco na etapa de disputa (fight) do algoritmo ENPC. Não sabíamos ao certo como codificar a busca pelos protótipos que fazem fronteira com outro dado protótipo.

Com o intuito de reproduzir o algoritmo da maneira mais fiel, consultamos o código fonte disponível na linguagem Java em [1]. Constatamos que os autores realizam esse passo obtendo os k-ésimos protótipos mais próximos ao protótipo em questão. O valor de k utilizado pelos autores foi 2. Optamos por usar o mesmo valor de k utilizado no código fonte dos autores.

4. Resultados e Discussões

Para avaliação da reprodução da técnica ENPC, utilizamos quinze bases do UCI [2]. Dentre as quinze bases, treze foram escolhidas e divididas em nove *folds*. Cinco bases foram utilizadas para geração de doze bases desbalanceadas, cada uma com cinco *folds*.

Comparamos o algoritmo ENPC com dois algoritmos um de seleção e outro de geração de protótipos: CNN [3] e *Self-Generating Prototype* - SGP [4]. Também comparamos os resultados obtidos pelo ENPC com 1-NN. Utilizamos 1-NN como

método de classificação para comparação do ENPC com os algoritmos de geração e seleção de protótipos.

Tabela 1. Avaliação do algoritmo ENPC em bases desbalanceadas exibindo a média e o desvio padrão de cada análise.

Base	Iterações	Ger. Acurácia		Maj. Acurácia		Min. Acurácia		ASC. Acurácia		Redução	
glass1	100	0.80	0.03	0.85	0.04	0.71	0.04	0.78	0.03	0.89	0.04
	200	0.81	0.05	0.86	0.03	0.71	0.09	0.79	0.05	0.87	0.03
ecoli-0_vs_1	100	0.98	0.02	0.95	0.05	1.00	0.00	0.97	0.02	0.94	0.03
	200	0.96	0.02	0.91	0.07	0.99	0.01	0.95	0.03	0.95	0.02
iris0	100	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	0.94	0.02
	200	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	0.93	0.03
glass0	100	0.79	0.07	0.81	0.06	0.74	0.12	0.78	0.08	0.89	0.02
	200	0.81	0.08	0.83	0.04	0.76	0.25	0.80	0.12	0.84	0.06
ecoli1	100	0.91	0.04	0.95	0.05	0.78	0.13	0.87	0.06	0.94	0.03
	200	0.91	0.03	0.95	0.04	0.80	0.01	0.88	0.01	0.96	0.01
new-thyroid2	100	0.95	0.04	0.98	0.02	0.83	0.17	0.90	0.09	0.96	0.02
	200	0.97	0.02	0.98	0.02	0.94	0.11	0.96	0.05	0.97	0.01
new-thyroid1	100	0.98	0.02	0.98	0.02	1.00	0.00	0.99	0.01	0.95	0.03
	200	0.99	0.02	0.98	0.02	1.00	0.00	0.99	0.01	0.96	0.02
ecoli2	100	0.95	0.02	0.96	0.03	0.91	0.06	0.93	0.03	0.97	0.01
	200	0.96	0.02	0.97	0.02	0.91	0.06	0.94	0.03	0.96	0.01
glass6	100	0.90	0.06	0.91	0.07	0.83	0.15	0.87	0.08	0.99	0.00
	200	0.90	0.05	0.91	0.06	0.83	0.15	0.87	0.07	0.98	0.01
glass2	100	0.87	0.07	0.93	0.10	0.15	0.30	0.54	0.10	0.95	0.04
	200	0.89	0.04	0.97	0.05	0.00	0.00	0.48	0.02	0.97	0.01
shuttle-c2-vs-c4	100	0.99	0.02	1.00	0.00	0.90	0.20	0.95	0.10	0.92	0.01
	200	0.99	0.02	1.00	0.00	0.90	0.20	0.95	0.10	0.95	0.02
glass-0-1-6_vs_5	100	0.95	0.01	0.99	0.01	0.10	0.20	0.55	0.09	0.98	0.02
	200	0.94	0.03	0.98	0.03	0.10	0.20	0.54	0.08	0.98	0.02

A Tabela 1 exibe o resultado do algoritmo ENPC quando aplicado a bases desbalanceadas. Na primeira coluna temos o nome da base, na segunda o número de iterações utilizado na execução do ENPC, na terceira coluna tem a média da taxa de acerto geral, bem como o desvio padrão. Na quarta coluna temos a média e o desvio padrão da taxa de acerto da classe minoritária. Na quinta coluna temos a média da área sob a curva ROC (ASC) e o respectivo desvio padrão. Finalmente, na última coluna, temos a taxa média da redução no tamanho da base e o desvio padrão atrelado à média.

Na Tabela 2 verificasse o resultado da execução do algoritmo ENPC sobre treze bases balanceadas. A disposição das colunas obedece ao mesmo padrão da Tabela 1 removendo as colunas que diriam respeito à taxa de acerto da classe minoritária e majoritária.

Tabela 2. Avaliação do algoritmo ENPC em bases balanceadas exibindo a média e o desvio padrão de cada análise.

Base	Iterações	Ger. Acurácia		ASC. Acurácia		Redução	
glass	100	0.71	0.09	0.72	0.09	0.68	0.01
	200	0.70	0.09	0.71	0.10	0.68	0.02
image_segmentation	100	0.93	0.01	0.96	0.01	0.98	0.00
	200	0.92	0.02	0.95	0.01	0.98	0.00
ionosphere	100	0.88	0.05	0.86	0.06	0.98	0.01
	200	0.89	0.03	0.87	0.03	0.97	0.01
iris	100	0.97	0.04	0.96	0.06	0.95	0.01
	200	0.97	0.03	0.96	0.04	0.94	0.02
liver	100	0.58	0.08	0.57	0.09	0.66	0.02
	200	0.62	0.08	0.61	0.08	0.66	0.01
pendigits	50	0.94	0.03	0.92	0.03	1.00	0.00
	100	0.95	0.01	0.92	0.03	1.00	0.00
pima_diabetes	100	0.71	0.06	0.68	0.07	0.79	0.03
	200	0.70	0.06	0.68	0.08	0.76	0.02
sonar	100	0.88	0.05	0.87	0.05	0.88	0.01
	200	0.86	0.06	0.85	0.06	0.88	0.02
spambase	100	0.82	0.03	0.81	0.02	1.00	0.00
	200	0.82	0.03	0.81	0.04	1.00	0.00
vehicle	100	0.65	0.03	0.60	0.05	0.73	0.01
	200	0.66	0.05	0.61	0.06	0.73	0.01
vowel	100	0.95	0.04	0.96	0.05	0.84	0.00
	200	0.96	0.02	0.97	0.03	0.84	0.00
wine	100	0.95	0.05	0.96	0.05	0.94	0.02
	200	0.97	0.04	0.96	0.05	0.95	0.01
yeast	50	0.48	0.03	0.47	0.04	0.57	0.07
	100	0.50	0.03	0.49	0.03	0.51	0.01

Como podemos perceber, o algoritmo ENPC tende a convergir de forma rápida, visto que as execuções com cem iterações possuem valores muito próximos das execuções de duzentas iterações, quando não, possuem taxas de acerto superiores. Vale salientar também o bom resultado em discriminar as classes em bases desbalanceadas, embora tenhamos duas bases que a taxa de acerto da classe minoritária tendeu a zero.

Na Tabela 3 pode ser observada a comparação entre o ENPC e o 1-NN. Das doze bases analisadas o ENPC é superado em apenas quatro bases, ao analisarmos apenas a taxa média de acerto. Se levarmos em conta a redução da quantidade de dados, o uso do ENPC mostrasse tão bom quanto e por diversas vezes melhor que o 1-NN.

Tabela 3. ENPC x 1-NN - Bases desbalanceadas

Base	Algoritmo	Ger. Acurácia		Maj. Acurácia		Min. Acurácia		ASC. Acurácia		Redução	
glass1	ENPC - 200	0.81	0.05	0.86	0.03	0.71	0.09	0.79	0.05	0.87	0.03
	KNN	0.81	0.05	0.89	0.02	0.67	0.12	0.78	0.06	-	-
ecoli-0_vs_1	ENPC - 100	0.98	0.02	0.95	0.05	1.00	0.00	0.97	0.02	0.94	0.03
	KNN	0.97	0.02	0.95	0.07	0.98	0.03	0.96	0.03	-	-
iris0	ENPC - 200	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	0.93	0.03
	KNN	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	-	-
glass0	ENPC - 200	0.81	0.08	0.83	0.04	0.76	0.25	0.80	0.12	0.84	0.06
	KNN	0.84	0.06	0.86	0.08	0.80	0.08	0.83	0.06	-	-
ecoli1	ENPC - 200	0.91	0.03	0.95	0.04	0.80	0.01	0.88	0.01	0.96	0.01
	KNN	0.86	0.04	0.92	0.04	0.67	0.12	0.80	0.06	-	-
new-thyroid2	ENPC - 200	0.97	0.02	0.98	0.02	0.94	0.11	0.96	0.05	0.97	0.01
	KNN	0.99	0.01	0.99	0.01	0.97	0.06	0.98	0.03	-	-
new-thyroid1	ENPC - 200	0.99	0.02	0.98	0.02	1.00	0.00	0.99	0.01	0.96	0.02
	KNN	0.98	0.01	0.98	0.01	0.97	0.06	0.98	0.02	-	-
ecoli2	ENPC - 200	0.96	0.02	0.97	0.02	0.91	0.06	0.94	0.03	0.96	0.01
	KNN	0.95	0.04	0.96	0.04	0.87	0.12	0.92	0.07	-	-
glass6	ENPC - 200	0.90	0.05	0.91	0.06	0.83	0.15	0.87	0.07	0.98	0.01
	KNN	0.96	0.02	0.99	0.01	0.79	0.14	0.89	0.07	-	-
glass2	ENPC - 200	0.89	0.04	0.97	0.05	0.00	0.00	0.48	0.02	0.97	0.01
	KNN	0.87	0.05	0.92	0.05	0.22	0.19	0.57	0.11	-	-
shuttle-c2-vs-c4	ENPC - 100	0.99	0.02	1.00	0.00	0.90	0.20	0.95	0.10	0.92	0.01
	KNN	0.99	0.02	1.00	0.00	0.90	0.20	0.95	0.10	-	-
glass-0-1-6_vs_5	ENPC - 100	0.95	0.01	0.99	0.01	0.10	0.20	0.55	0.09	0.98	0.02
	KNN	0.96	0.02	0.97	0.02	0.80	0.24	0.89	0.12	-	-
glass1	ENPC - 200	0.81	0.05	0.86	0.03	0.71	0.09	0.79	0.05	0.87	0.03
	KNN	0.81	0.05	0.89	0.02	0.67	0.12	0.78	0.06	-	-

Uma comparação do ENPC com o 1-NN sobre bases balanceadas também é analisado e pode ser visualizado na Tabela 4.

Na comparação com o 1-NN sobre bases balanceadas o algoritmo ENPC tem um desempenho superior em 50% das bases. Nas bases que o 1-NN se mostra superior ao ENPC, no que diz respeito à taxa média de acerto, a diferença para o ENPC não supera o valor de 0.10.

Tabela 4. ENPC x 1-NN - Bases balanceadas

Base	Algoritmo	Ger. Acurácia		ASC. Acurácia		Redução	
glass	ENPC - 100	0.71	0.09	0.72	0.09	0.68	0.01
	KNN	0.70	0.05	0.71	0.06	-	-
image_segmentation	ENPC - 100	0.93	0.01	0.96	0.01	0.98	0.00
	KNN	0.97	0.01	0.98	0.00	-	-
ionosphere	ENPC - 200	0.89	0.03	0.87	0.03	0.97	0.01
	KNN	0.86	0.04	0.82	0.05	-	-
iris	ENPC - 200	0.97	0.03	0.96	0.04	0.94	0.02
	KNN	0.95	0.05	0.95	0.07	-	-
liver	ENPC - 200	0.62	0.08	0.61	0.08	0.66	0.01
	KNN	0.62	0.06	0.61	0.06	-	-
pendigits	ENPC - 100	0.95	0.01	0.92	0.03	1.00	0.00
	KNN	0.99	0.00	0.99	0.00	-	-
pima_diabetes	ENPC - 100	0.71	0.06	0.68	0.07	0.79	0.03
	KNN	0.70	0.05	0.66	0.06	-	-
sonar	ENPC - 100	0.88	0.05	0.87	0.05	0.88	0.01
	KNN	0.87	0.11	0.86	0.12	-	-
spambase	ENPC - 100	0.82	0.03	0.81	0.02	1.00	0.00
	KNN	0.91	0.01	0.91	0.01	-	-
vehicle	ENPC - 200	0.66	0.05	0.61	0.06	0.73	0.01
	KNN	0.70	0.05	0.63	0.07	-	-
vowel	ENPC - 200	0.96	0.02	0.97	0.03	0.84	0.00
	KNN	0.99	0.02	0.98	0.04	-	-
wine	ENPC - 200	0.97	0.04	0.96	0.05	0.95	0.01
	KNN	0.96	0.03	0.97	0.02	-	-
yeast	ENPC - 100	0.50	0.03	0.49	0.03	0.51	0.01
	KNN	0.52	0.04	0.51	0.05	-	-

Na comparação com o 1-NN sobre bases balanceadas o algoritmo ENPC tem um desempenho superior em 50% das bases. Nas bases que o 1-NN se mostra superior ao ENPC, no que diz respeito à taxa média de acerto, a diferença para o ENPC não supera o valor de 0.10.

Na Tabela 5 e Tabela 6 podemos visualizar os resultados comparativos entre o ENPC e o CNN sobre bases desbalanceadas e balanceadas, respectivamente. Por sua vez, na Tabela 7 e 8 verificamos, respectivamente, os dados comparativos entre o ENPC e o SGP sobre bases desbalanceadas e balanceadas.

Tabela 5. ENPC x CNN - Bases desbalanceadas

Base	Algoritmo	Ger. Acurácia		Maj. Acurácia		Min. Acurácia		ASC. Acurácia		Redução	
glass1	ENPC - 200	0.81	0.05	0.86	0.03	0.71	0.09	0.79	0.05	0.87	0.03
	CNN	0.68	0.03	0.77	0.06	0.52	0.14	0.64	0.05	0.81	0.03
ecoli-0_vs_1	ENPC - 100	0.98	0.02	0.95	0.05	1.00	0.00	0.97	0.02	0.94	0.03
	CNN	0.72	0.13	0.96	0.03	0.59	0.22	0.78	0.09	0.96	0.01
iris0	ENPC - 200	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	0.93	0.03
	CNN	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	0.98	0.00
glass0	ENPC - 200	0.81	0.08	0.83	0.04	0.76	0.25	0.80	0.12	0.84	0.06
	CNN	0.72	0.03	0.99	0.01	0.16	0.07	0.58	0.04	0.92	0.03
ecoli1	ENPC - 200	0.91	0.03	0.95	0.04	0.80	0.01	0.88	0.01	0.96	0.01
	CNN	0.71	0.05	0.80	0.09	0.41	0.11	0.61	0.04	0.92	0.03
new-thyroid2	ENPC - 200	0.97	0.02	0.98	0.02	0.94	0.11	0.96	0.05	0.97	0.01
	CNN	0.82	0.17	0.78	0.20	1.00	0.00	0.89	0.10	0.97	0.01
new-thyroid1	ENPC - 200	0.99	0.02	0.98	0.02	1.00	0.00	0.99	0.01	0.96	0.02
	CNN	0.96	0.02	0.97	0.02	0.91	0.07	0.94	0.03	0.94	0.01
ecoli2	ENPC - 200	0.96	0.02	0.97	0.02	0.91	0.06	0.94	0.03	0.96	0.01
	CNN	0.55	0.09	0.50	0.12	0.83	0.12	0.66	0.04	0.93	0.01
glass6	ENPC - 200	0.90	0.05	0.91	0.06	0.83	0.15	0.87	0.07	0.98	0.01
	CNN	0.66	0.14	0.61	0.15	0.97	0.07	0.79	0.10	0.95	0.01
glass2	ENPC - 200	0.89	0.04	0.97	0.05	0.00	0.00	0.48	0.02	0.97	0.01
	CNN	0.78	0.08	0.83	0.08	0.30	0.27	0.56	0.16	0.82	0.02
shuttle-c2-vs-c4	ENPC - 100	0.99	0.02	1.00	0.00	0.90	0.20	0.95	0.10	0.92	0.01
	CNN	0.98	0.02	1.00	0.00	0.70	0.40	0.85	0.20	0.97	0.01
glass-0-1-6_vs_5	ENPC - 100	0.95	0.01	0.99	0.01	0.10	0.20	0.55	0.09	0.98	0.02
	CNN	0.86	0.08	0.86	0.08	0.80	0.24	0.83	0.13	0.95	0.01

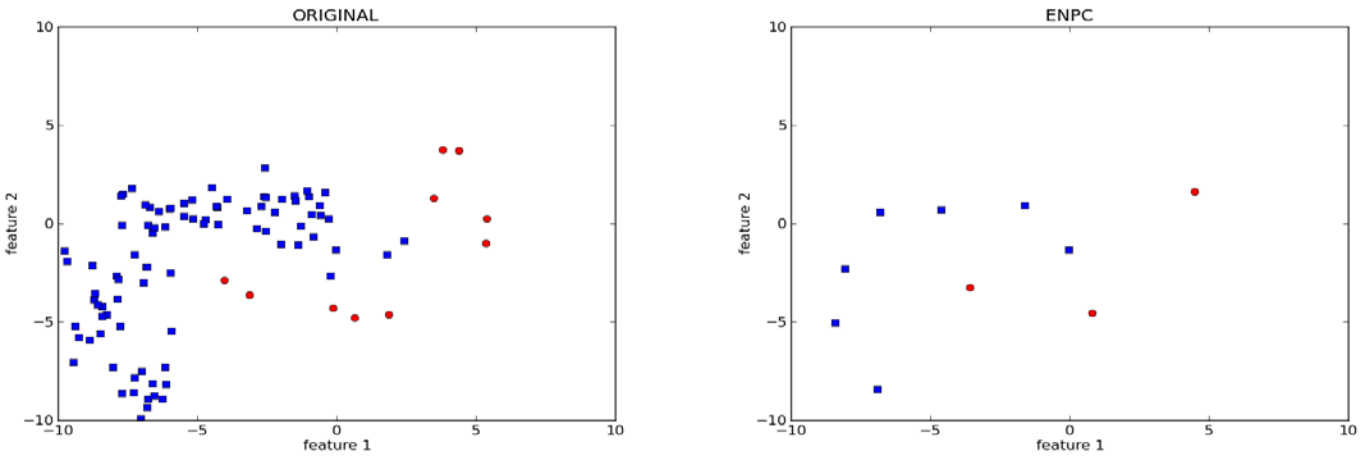

Figura 3. Base artificial Banana. Gráfico à esquerda, pontos iniciais. Gráfico à direita, protótipos gerados pelo algoritmo ENPC.

Tabela 6. ENPC x CNN - Bases balanceadas

Base	Algoritmo	Ger. Acurácia		ASC. Acurácia		Redução	
glass	ENPC - 100	0.71	0.09	0.72	0.09	0.68	0.01
	CNN	0.70	0.05	0.70	0.06	0.60	0.02
image_segmentation	ENPC - 100	0.93	0.01	0.96	0.01	0.98	0.00
	CNN	0.95	0.02	0.97	0.01	0.91	0.00
ionosphere	ENPC - 200	0.89	0.03	0.87	0.03	0.97	0.01
	CNN	0.82	0.07	0.79	0.08	0.83	0.02
iris	ENPC - 200	0.97	0.03	0.96	0.04	0.94	0.02
	CNN	0.92	0.06	0.91	0.07	0.88	0.02
liver	ENPC - 200	0.62	0.08	0.61	0.08	0.66	0.01
	CNN	0.61	0.04	0.60	0.05	0.56	0.02
pendigits	ENPC - 100	0.95	0.01	0.92	0.03	1.00	0.00
	CNN	0.98	0.01	0.97	0.01	0.97	0.00
pima_diabetes	ENPC - 100	0.71	0.06	0.68	0.07	0.79	0.03
	CNN	0.63	0.05	0.60	0.06	0.63	0.01
sonar	ENPC - 100	0.88	0.05	0.87	0.05	0.88	0.01
	CNN	0.82	0.12	0.82	0.11	0.74	0.02
spambase	ENPC - 100	0.82	0.03	0.81	0.02	1.00	0.00
	CNN	0.86	0.01	0.86	0.01	0.82	0.00
vehicle	ENPC - 200	0.66	0.05	0.61	0.06	0.73	0.01
	CNN	0.64	0.04	0.59	0.06	0.60	0.01
vowel	ENPC - 200	0.96	0.02	0.97	0.03	0.84	0.00
	CNN	0.95	0.03	0.96	0.03	0.78	0.01
wine	ENPC - 200	0.97	0.04	0.96	0.05	0.95	0.01
	CNN	0.93	0.05	0.95	0.05	0.89	0.01
yeast	ENPC - 100	0.50	0.03	0.49	0.03	0.51	0.01
	CNN	0.47	0.05	0.46	0.05	0.44	0.01

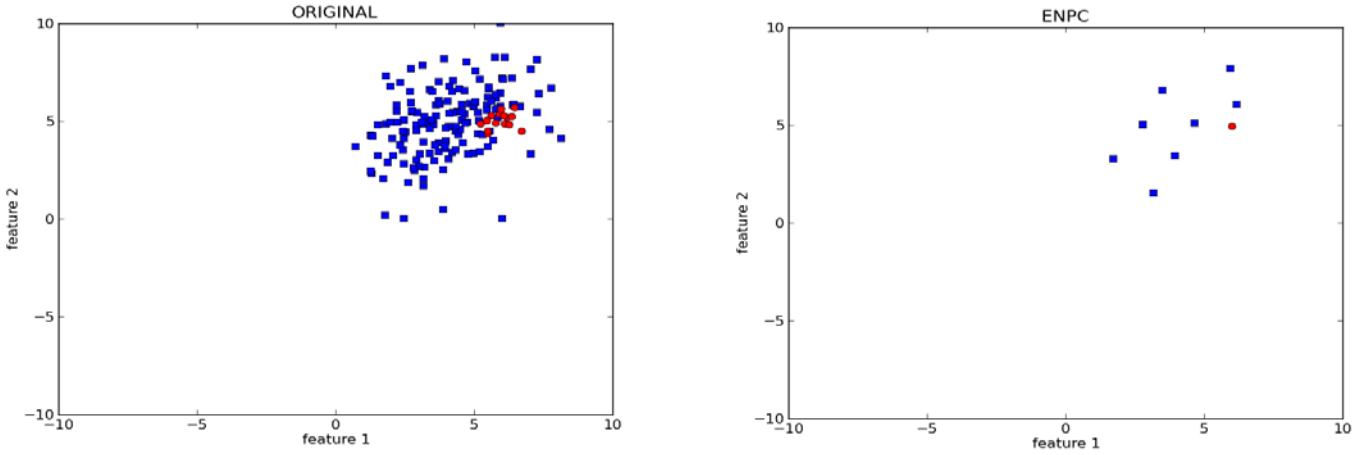

Figura 4. Base artificial Normal. Gráfico à esquerda, pontos iniciais. Gráfico à direita, protótipos gerados pelo algoritmo ENPC.

Tabela 7. ENPC x SGP - Bases desbalanceadas

Base	Algoritmo	Ger. Acurácia		Maj. Acurácia		Min. Acurácia		ASC. Acurácia		Redução	
glass1	ENPC - 200	0.81	0.05	0.86	0.03	0.71	0.09	0.79	0.05	0.87	0.03
	SGP	0.78	0.04	0.85	0.05	0.66	0.12	0.75	0.05	0.61	0.02
ecoli-0_vs_1	ENPC - 100	0.98	0.02	0.95	0.05	1.00	0.00	0.97	0.02	0.94	0.03
	SGP	0.95	0.04	0.93	0.07	0.97	0.05	0.95	0.04	0.87	0.02
iris0	ENPC - 200	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	0.93	0.03
	SGP	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	0.98	0.00
glass0	ENPC - 200	0.81	0.08	0.83	0.04	0.76	0.25	0.80	0.12	0.84	0.06
	SGP	0.80	0.04	0.79	0.03	0.83	0.12	0.81	0.06	0.62	0.02
ecoli1	ENPC - 200	0.91	0.03	0.95	0.04	0.80	0.01	0.88	0.01	0.96	0.01
	SGP	0.85	0.06	0.91	0.05	0.66	0.13	0.79	0.08	0.75	0.03
new-thyroid2	ENPC - 200	0.97	0.02	0.98	0.02	0.94	0.11	0.96	0.05	0.97	0.01
	SGP	0.97	0.03	0.97	0.03	0.97	0.06	0.97	0.03	0.90	0.01
new-thyroid1	ENPC - 200	0.99	0.02	0.98	0.02	1.00	0.00	0.99	0.01	0.96	0.02
	SGP	0.98	0.02	0.98	0.02	1.00	0.00	0.99	0.01	0.90	0.02
ecoli2	ENPC - 200	0.96	0.02	0.97	0.02	0.91	0.06	0.94	0.03	0.96	0.01
	SGP	0.92	0.03	0.93	0.02	0.85	0.15	0.89	0.07	0.79	0.02
glass6	ENPC - 200	0.90	0.05	0.91	0.06	0.83	0.15	0.87	0.07	0.98	0.01
	SGP	0.95	0.01	0.98	0.01	0.72	0.10	0.85	0.05	0.91	0.02
glass2	ENPC - 200	0.89	0.04	0.97	0.05	0.00	0.00	0.48	0.02	0.97	0.01
	SGP	0.86	0.07	0.90	0.06	0.45	0.27	0.67	0.15	0.71	0.03
shuttle-c2-vs-c4	ENPC - 100	0.99	0.02	1.00	0.00	0.90	0.20	0.95	0.10	0.92	0.01
	SGP	0.99	0.02	1.00	0.00	0.90	0.20	0.95	0.10	0.97	0.01
glass-0-1-6_vs_5	ENPC - 100	0.95	0.01	0.99	0.01	0.10	0.20	0.55	0.09	0.98	0.02
	SGP	0.95	0.03	0.97	0.02	0.70	0.24	0.83	0.13	0.91	0.01

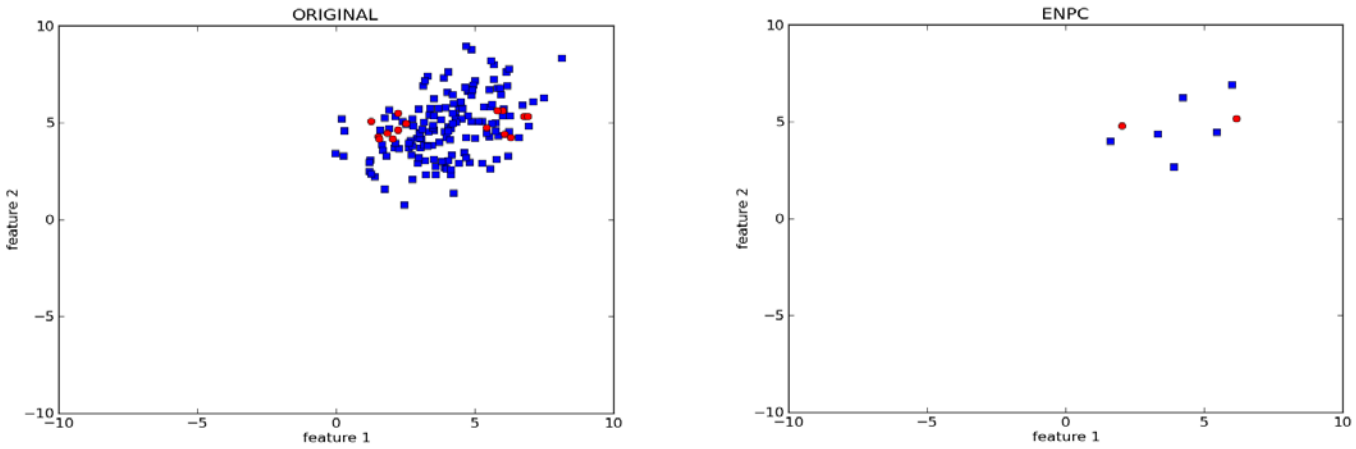

Figura 5. Base artificial Normal Multimodal. Gráfico à esquerda, pontos iniciais. Gráfico à direita, protótipos gerados pelo algoritmo ENPC.

Tabela 8. ENPC x SGP - Bases balanceadas

Base	Algoritmo	Ger. Acurácia		ASC. Acurácia		Redução	
glass	ENPC - 100	0.71	0.09	0.72	0.09	0.68	0.01
	SGP	0.71	0.07	0.71	0.08	0.54	0.02
image_segmentation	ENPC - 100	0.93	0.01	0.96	0.01	0.98	0.00
	SGP	0.96	0.01	0.98	0.01	0.89	0.00
ionosphere	ENPC - 200	0.89	0.03	0.87	0.03	0.97	0.01
	SGP	0.88	0.05	0.85	0.07	0.80	0.01
iris	ENPC - 200	0.97	0.03	0.96	0.04	0.94	0.02
	SGP	0.93	0.07	0.92	0.08	0.88	0.02
liver	ENPC - 200	0.62	0.08	0.61	0.08	0.66	0.01
	SGP	0.63	0.06	0.62	0.06	0.40	0.02
pendigits	ENPC - 100	0.95	0.01	0.92	0.03	1.00	0.00
	SGP	0.99	0.00	0.98	0.01	0.95	0.00
pima_diabetes	ENPC - 100	0.71	0.06	0.68	0.07	0.79	0.03
	SGP	0.68	0.05	0.65	0.06	0.51	0.01
sonar	ENPC - 100	0.88	0.05	0.87	0.05	0.88	0.01
	SGP	0.87	0.08	0.87	0.08	0.75	0.01
vehicle	ENPC - 200	0.66	0.05	0.61	0.06	0.73	0.01
	SGP	0.67	0.03	0.61	0.05	0.54	0.01
vowel	ENPC - 200	0.96	0.02	0.97	0.03	0.84	0.00
	SGP	0.96	0.02	0.97	0.03	0.77	0.01
wine	ENPC - 200	0.97	0.04	0.96	0.05	0.95	0.01
	SGP	0.95	0.05	0.96	0.04	0.90	0.01
yeast	ENPC - 100	0.50	0.03	0.49	0.03	0.51	0.01
	SGP	0.51	0.03	0.50	0.04	0.34	0.01

Percebemos através da Tabela 5 e Tabela 6 que o ENPC é em geral, superior ao CNN, tanto na taxa média de acerto como na taxa média de redução da base inicial, independente de a base está, ou não, balanceada. O SGP, por usa vez, se mostra mais competitivo, mas o ENPC ainda se sobressai, obtendo resultados melhores que o SGP em mais bases, tanto na taxa média de acerto como na taxa média de redução do tamanho inicial da base.

A Figura 3, Figura 4 e Figura 5, mostram o resultado visual da aplicação do algoritmo ENPC a três bases artificiais de duas dimensões. Como se pode observar através da Figura 5, mesmo com dados multimodais e com classe minoritária, o algoritmo ENPC ainda se sai razoavelmente bem.

6. Conclusão

No presente trabalho, realizamos uma análise da nossa codificação do algoritmo *Evolutionary Design of Nearest Prototype Classifiers* – ENPC em linguagem Python. O

trabalho onde o ENPC foi desenvolvido afirma que o algoritmo é livre de variáveis no que diz respeito a aprendizagem de máquina. Toda via, ao termos dificuldade na implementação de um dos passos do algoritmo, constatamos, em nível de implementação, a existência de um parâmetro para determinar os k vizinhos de um protótipo, que está para ser desafiado na etapa da disputa (fight).

No que diz respeito a qualidade, o algoritmo se mostrou bastante robusto a diferentes tipos de bases de dados, mostrando-se tão bom quanto o 1-NN e o SGP e melhor que o CNN.

7. Referências

[1] “KEEL: A software tool to assess evolutionary algorithms for Data Mining problems (regression, classification, clustering, pattern mining and so on)”, http://www.keel.es, Maio 2014.

[2] “UCI Machine Learning Repository: Data Sets”, https://archive.ics.uci.edu/ml/datasets.html, Julho 2014.

[3] P. E. Hart. The condensed nearest neighbor rule. IEEE Transactions on Information Theory, IT-4:515–516, 1968.

[4] Hatem A. Fayed, Sherif R. Hashem, Amir F. Atiya, Self-generating prototypes for pattern classification, Pattern Recognition, Volume 40, Edição 5, Maio 2007, Páginas 1498-1509, ISSN 0031-3203.