Aceleración del algoritmo Fast-mRMR para selección de características en sistemas multinúcleo

Borja Freire Castro

Universidad de la Coruña

October 16, 2017

¿Por qué Fast-mRMR?

- El número de características de los conjuntos está creciendo enormemente → necesario desarrollar técnicas para discriminar cuales de dichas características son realmente útiles.
- Mrmr es uno de los métodos mas valorados debido a sus excepcionales resultados pero ha sido criticado por su complejidad computacional:
  - Cuadrático sobre el número de características.
  - Lineal sobre el número de instancias.
- Fast-mRMR mejora la complejidad del algoritmo, creciendo linealmente tanto con el número de características como en el número de características seleccionadas.

UNIVERSIDADE DA CORUÑA

¿Qué se pretende?

El propósito fundamental de este estudio es el desarrollo de una optimización paralela sobre Fast-mRMR dirigida a sistemas de memoria compartida o multiprocesador haciendo uso de OpenMP.

UNIVERSIDADE DA CORUÑA

¿Por que OpenMP?

- OpenMP permite una representación implícita del paralelismo con distribución explícita.
- Se delega la asignación, distribución, comunicación y sincronización en la propia interfaz de aplicaciones.
- Está presente en C,C++ y Fortran.

Cambios sobre Fast-mRMR: Secuenciales

Se tuvieron que realizar ligeros cambios sobre Fast-mRMR buscando posibles mejoras de tiempos y adecuando el problema a un planteamiento paralelo.

- Estructuras de datos:
  - Se eliminaron los objetos predefinidos tipo: vectores, colas o pilas.
  - Se añadieron estructuras tipo: array.
- Modificaciones menores sobre el código de Fast-mRMR realizando cambios de tipo secuenciales para tratar de agilizar el código adaptándolo a las nuevas estructuras de datos empleadas.

UNIVERSIDADE DA CORUÑA

Cambios sobre Fast-mRMR: Paralelización

- Búsqueda de secciones paralelizables → en este caso bucles en su totalidad:
  - Selección de la distribución que mejores resultados ofrecía: dinámica.
  - Selección del tamaño de bloque de interés: max(1,0.2*NumberOfFeatures)
- Gestión de las regiones de race condition.

Modificación

```
1: while selectedFeatures.size() <numFeaturesWanted do
2:     @For parallel, dynamic/static, private variables (relevance,redundancy,mrmr)
3:    for for feature fc in candidates do
4:        relevance = relevancesVector[fc];
5:        acumulatedRedundancy[fc] += mutualInfo(fc, lastFeatureSelected);
6:        redundancy = acumulatedRedundancy[fc] / selectedFeatures.size();
7:        mrmr = relevance - redundancy;
8:        if if mrmr is maximum[thread] then
9:            lastFeatureSelectedThread[thread] = fc;
10:       end if
11:    end for
12:    pos = getMaximum(maximum);
13:    lastFeatureSelected = lastFeatureSelectedThread[pos];
14:    selectedFeatures.add(lastFeatureSelected);
15:    candidates.remove(lastFeatureSelected);
16: end while
```

Conjuntos de datos

Dataset	Features	Samples	Classes
Lung	326	73	3
Leukemia	7071	72	3
NCI	9173	60	3
Lymphoma	4027	96	3
Colon	2001	62	3
BreastCancer	47293	128	2
Epsylon	2000	400000	2

UNIVERSIDADE DA CORUÑA

Experimentos

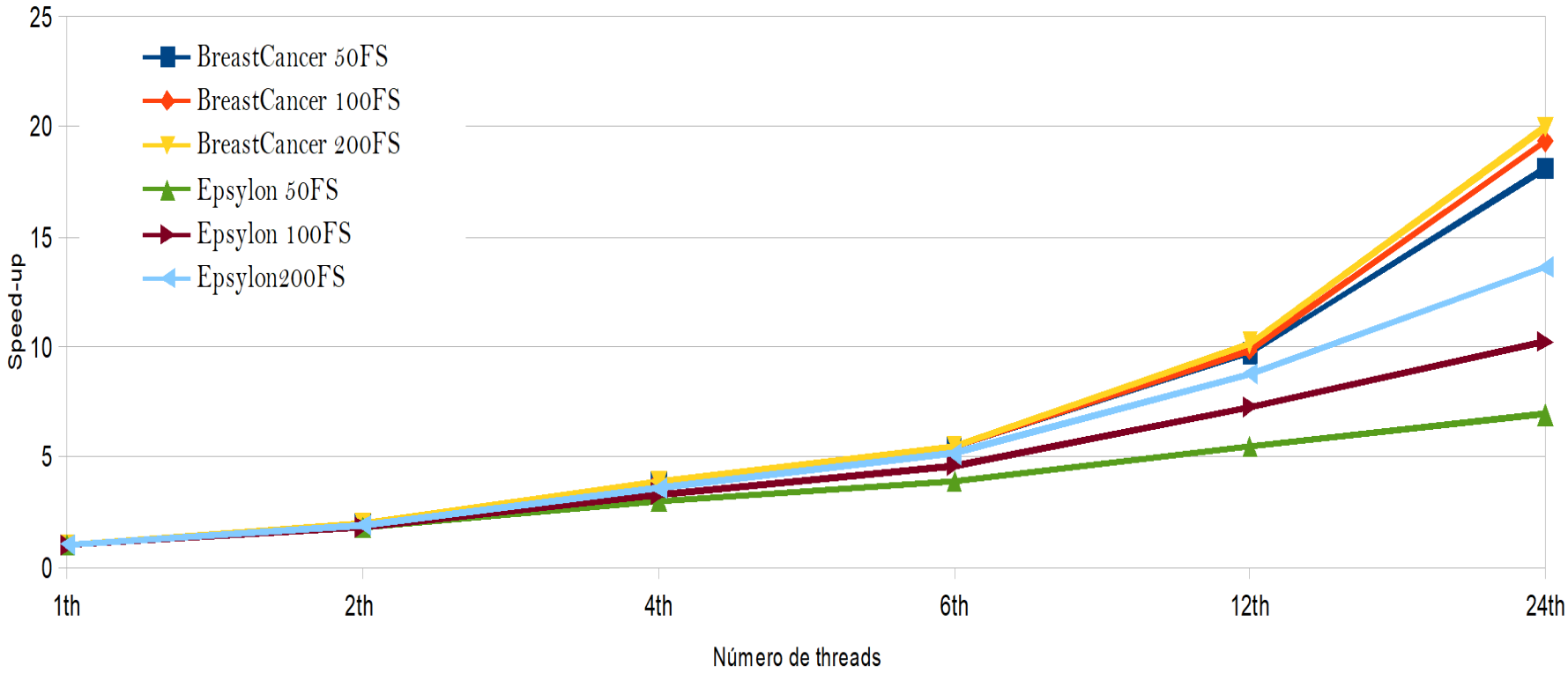
Experimentos realizados:

- mRMR vs Fast-mRMR/Fast-mRMR-24Threads
- Fast-mRMR vs Fast-mRMR-Threaded:
  - Fast-mRMR 1 threads.
  - Fast-mRMR 2 threads.
  - Fast-mRMR 4 threads.
  - Fast-mRMR 6 threads.
  - Fast-mRMR 12 threads.
  - Fast-mRMR 24 threads.

Resultados: Datasets Pequeños

Dataset	mRMR	Fast-mRMR	Fast-mRMR-24Threads
Lung	19676,41	24,84	7,29
Colon	34144,15	202,68	21,29
Leukemia	37283,49	588,39	38,33
Lymphoma	42951,11	476,63	33,21
NCI	32690,29	999,68	61,89

UNIVERSIDADE DA CORUÑA



UNIVERSIDADE DA CORUÑA

Conclusión

- El propósito principal era cuantificar el beneficio de la paralelización para sistemas de memoria compartida de un algoritmo de selección de características.
- El resultado obtenido muestra que se trata de un problema escalable, llegando a alcanzar hasta speed-ups de 22.
- Los resultados generales fueron los esperados y realmente aceptables.
- En base a los resultados obtenidos se plantea la posibilidad de extender el estudio a sistemas híbridos y sobre diferentes algoritmos de selección de características.
- El código se encuentra disponible en:
  $https://github.com/borjaf696/Fast-mRMR-Threaded$

UNIVERSIDADE DA CORUÑA