Implementación y evaluación de kernels en la predicción de actividad biológica

by Yoamel Acosta Morales - Thursday, February 05, 2015

https://vinculando.org/beta/implementacion-evaluacion-kernels-prediccion-actividad-biologica.html

Título completo: Implementación y evaluación de los kernels Stump y Perceptron, para la predicción de actividad biológica, aplicando Máquinas de Soporte Vectorial

Marco

Las bases de la presente investigación surgen en el marco de trabajo del Proyecto “Plataforma Inteligente para la Predicción de Actividad Biológica de Compuestos Orgánicos” (GRAph TOol), una herramienta que se desarrolla conjuntamente por el Centro de Química Farmacéutica de Cuba y el Grupo de Bioinformática de la Universidad de las Ciencias Informáticas. Este proyecto consta de varias aristas, entre las que destacan: la visualización y edición de fármacos a nivel molecular y el abordado en la presente investigación: la predicción de actividad biológica empleando varias técnicas de inteligencia artificial y minería de datos.

Introducción

El descubrimiento de nuevas moléculas de interés farmacéutico está estrechamente relacionado con la búsqueda de

información en grandes bases de datos y con la determinación y estimación de estructuras, es por eso que el desarrollo de aplicaciones informáticas para extraer conocimiento de la información generada en los laboratorios se manifiesta como una de las necesidades importantes en la Bioinformática.

Dada la amplitud de compuestos conocidos y el alto costo de los ensayos que deben realizarse para determinar la utilidad o no de uno en particular, en los últimos tiempos la industria farmacéutica viene prestando especial atención a métodos que permitan una selección racional de compuestos con propiedades deseadas

Muchos de los más exitosos enfoques que se le ha dado al diseño de fármacos asistidos computacionalmente están basados en la correlación entre estructura química y propiedades de las moléculas. La búsqueda y comparación de estas estructuras y comportamientos sin la asistencia de un medio de cómputo y el software adecuado tardaba varios años, tras los que solo un pequeño por ciento de las moléculas llega siquiera a la fase de ensayos clínicos. Es para evitar o minimizar estos grandes gastos de tiempo y recursos, que se aspira a brindar una herramienta que permita la predicción con altos grados de efectividad de los compuestos candidatos con mayores posibilidades de resultar útiles a un fin determinado.

Se deseaba la funcionalidad de predecir actividad biológica en compuestos orgánicos, para lo que entre otros enfoques, se planteó el empleo de Máquinas de Soporte Vectorial (MSV). Se desarrolló entonces una versión base empleando un solo kernel, en busca de probar la efectividad de dicha técnica ante el problema en cuestión.

Los resultados obtenidos, empleando el kernel Polynomial, oscilaban entre un 60-70% de efectividad en las predicciones. Estos valores, aunque se consideran aceptables en el marco general de la Inteligencia Artificial, son insuficientes para representar un aporte sólido a la toma de decisiones respecto a elegir o descartar compuestos para investigaciones. Sin embargo, estos resultados probaban que la técnica era válida para solucionar la problemática.

Dentro de los objetivos de la investigación se encuentra entonces implementar kernels capaces de generar modelos predictivos de actividad biológica, especialmente en compuestos orgánicos, aplicando Maquinas de Soporte Vectorial, con capacidades de predicción superiores al 75%. Para cumplir este objetivo se emplean además algoritmos de aprendizaje combinados, en especial los del tipo infinito. Se trabaja en los kernels de procesamiento Stump y Perceptron.

Máquinas de Soporte Vectorial:

Las Máquinas de Soporte vectorial (MSV) son un sistema de aprendizaje, dentro de las más populares técnicas de Inteligencia Artificial. Es un sistema de aprendizaje basado en el uso de un espacio de hipótesis de funciones lineales en un espacio de mayor dimensión inducido por un Kernel, en el cual las hipótesis son entrenadas por un algoritmo.

Además es un sistema para entrenar máquinas de aprendizaje lineal eficientemente, y tanto para la clasificación como para la regresión se han encontrado muchas aplicaciones: clasificación de imágenes, reconocimiento de caracteres, detección de proteínas, clasificación de patrones, identificación de funciones, etc.

Las Máquinas de Soporte Vectorial están basadas en el principio de minimización del riesgo estructural, principio originado de la teoría de aprendizaje estadístico desarrollada por Vapnik (*Statistical Learning Theory*, 1998) el cual es considerado superior al principio de minimización del riesgo empírico, utilizado por las redes neuronales convencionales.

Algunas de las razones por las que este método ha tenido éxito es que no padece de mínimos locales y el modelo solo depende de los datos con más información llamados vectores de soporte.

Algoritmos de aprendizaje ensemble

Entre los métodos más importantes aplicados en la investigación, se encuentran los algoritmos de aprendizaje ensemble, que representan toda una corriente de pensamiento en la creación de kernels de procesamiento.

El paradigma de aprendizaje ensemble denota una amplia colección de algoritmos de aprendizaje. En lugar de considerar un poderoso modelo de aprendizaje, un algoritmo de aprendizaje conjunto, maneja modelos de aprendizaje base que son generalmente simples.

Aunque la cantidad de modelos de aprendizaje base podría ser de tamaño infinito, en teoría, el conjunto tradicional de los algoritmos de aprendizaje suele ser finito y predefinido. Denominados algoritmos finitos de aprendizaje ensemble. Estos algoritmos suelen compartir otra característica común: eligen cada hipótesis llamando a otro algoritmo de aprendizaje, conocido como algoritmo de aprendizaje base.

Los algoritmos ensemble de aprendizaje son útiles por contar con algunas o todas las siguientes propiedades: estabilidad, precisión y eficiencia.

Estabilidad: Cuando un algoritmo de aprendizaje devuelve una muy diferente función de decisión ante un pequeño cambio en los datos, llamamos a este algoritmo inestable. Los algoritmos inestables no son deseables pues suelen ser susceptibles al ruido, medidas imprecisas o incluso pequeños errores en la entrada de los datos. Los algoritmos de aprendizaje ensemble son un acercamiento para brindarle estabilidad a los algoritmos base, dado que la combinación de varios de estos algoritmos base da como resultado una mayor generalización estabilizando los resultados.

Precisión: La precisión de los algoritmos de aprendizaje ensemble viene dada por la eliminación o reducción del error individualmente acumulado por cada algoritmo base, logrando que el posible error en el algoritmo ensemble sea eliminado o reducido.

Eficiencia: La eficiencia de los algoritmos de aprendizaje radica en que en vez de trabajar con un modelo complejo que generalmente complica el procesamiento y reduce la eficiencia, el tipo de algoritmos ensemble divide el problema en varios problemas pequeños, es decir trabaja con varios algoritmos base más simples.

Kernels

Las funciones kernel son funciones matemáticas que se emplean en las Máquinas de Soporte Vectorial. Estas funciones son las que le permiten convertir lo que sería un problema de clasificación no lineal en el espacio dimensional original, a un sencillo problema de clasificación lineal en un espacio dimensional mayor.

Para poder ser consideradas candidatas a kernels, las funciones deben cumplir tres condiciones iniciales fundamentales; deben ser continuas, simétricas y positivas. Estos son los requerimientos básicos para poder ser expresadas como un producto escalar en un espacio dimensional alto. El espacio dimensional simulado mediante las funciones kernel se define tomando a cada característica de los datos como una dimensión. Esto convierte a las entradas en un conjunto de puntos en un espacio euclidiano n-dimensional. Es mucho más fácil establecer relaciones entre los datos expresados en esta forma.

Con el objetivo de identificar los kernels óptimos a incorporar se analizaron numerosas opciones disponibles, entre las posibilidades estudiadas destacan los siguientes kernels: Gaussiano, Fisher, String, Hubert y Gaussiano de anchura modificada. Algunos de los anteriores kernels llegaron hasta la fase de implementación y prueba. No obstante se decidió finalmente trabajar sobre una familia de kernels muy vinculada a la teoría de algoritmos ensemble infinitos.

El kernel Stump es la base para una familia de funciones kernel relacionadas. Este encarna un número infinito de decisiones stump. La decisión stump es uno de los modelos de aprendizaje base más simples que se aplican. El kernel Perceptron, basado en el Stump y simulando una red neuronal de tamaño infinito, es otra popular opción para algoritmos de aprendizaje ensemble. Ambos serán implementados con posterioridad para una solución base alternativa a la actual y más eficiente en procesamiento.

Kernels seleccionados

Laplacian kernel.

Aplicando sobre el kernel stump, un grupo de transformaciones matemáticas, obtenemos una expresión del kernel stump como árbol infinito, dada la combinación de varias funciones stump, utilizando la expansión de la serie de Taylor, obtenemos un equivalente en la teoría ensemble infinita, del tradicional kernel de base radial: Laplace, expresado mediante la fórmula:

$k(x,x')=\exp(-?||x-x'||_1),?>0.$

Exponencial kernel.

Igualmente aplicando un grupo de transformaciones similares, a partir del kernel Perceptron obtenemos un equivalente del popular kernel exponencial, expresado como árbol de decisiones Perceptron infinitas, expresado mediante la fórmula:

$k(x,x')=\exp(-?||x-x'||_2),?>0.$

Implementación de la solución

La solución desarrollada empleando Java como lenguaje de programación, siguiendo el popular enfoque orientado a objetos, dentro de la metodología OpenUP para el desarrollo ágil. Resultó muy eficiente en cuanto a consumo de recursos de cómputo y tiempos de respuesta, el enfoque de los algoritmos ensemble, dio frutos en este sentido; garantizando que la complejidad de casi todas las funciones fuese O(n).

Ejemplo en pseudocódigo de la solución:

El algoritmo dist_1 es creado para el tratamiento de los datos a procesar empleando el kernels Laplace (aplicable al Stump) que requieren de un manejo más lineal de los datos de entrada que sus homólogos.

```
Función dist_1(svm_node[] x, svm_node[] x1)

real sum <- 0

entero xlen <- longitud de x

entero x1len <- longitud de x1

entero i <- 0

entero j <- 0

Mientras (i < xlen-1 y j < x1len-1) hacer
```

```
Si (x[i]índice = x1[j]índice)
entonces
Si ((x[i]valor – x1[j]valor)>0)
entonces
sum += x[i]valor – x1[j]valor
Si no
sum += x1[j]valor – x[i]valor
fin Si
incrementar i
incrementar j
Si no
Si(x[i]índice > x1[j]índice)
entonces
Si (x1[j]valor > 0)
entonces
sum += x1[j]valor
incrementar j
Si no
sum = sum – x1[j]valor
incrementar j
fin Si
Si no
Si(x[i]valor > 0)
entonces
sum += x[i]valor
```

incrementar i

Si no

sum = sum – x[i]valor

incrementar i

fin Si

fin Si

fin Si

fin Mientras

Mientras (i < xlen-1) hacer

Si(x[i]valor > 0)

entonces

sum += x[i]valor

incrementar i

Si no

sum = sum – x[i]valor

incrementar i

fin Si

fin Mientras

Mientras (j < x1len-1) hacer

Si (x1[j]valor > 0)

entonces

sum += x1[j]valor

incrementar j

Si no

sum = sum – x1[j]valor

incrementar j

fin Si

fin Mientras

Devolver sum

Fin función

Como se puede apreciar, la complejidad computacional de la función es extremadamente baja, lo que trasciende para el resto de las que componen la solución; brindando un extra de eficiencia. Este detalle abre además las puertas incluso a futuras versiones parcialmente distribuidas de la misma, lo que reduciría aún más los requerimientos de hardware para el procesamiento de grandes volúmenes de datos.

Resultados experimentales

Se describirán a continuación los más importantes experimentos realizados sobre la solución que el presente trabajo aportó.

Para probar la efectividad de la solución se empleó una muestra de 81 Cefalosporinas determinadas por 11 descriptores cada una. La variable dependiente empleada en las pruebas fue el nivel de actividad biológica de cada una de las Cefalosporinas sobre la Escherichia coli (POT_EC) y la Staphylococcus aureus (POT_SA).

También se realizó una modelación y predicción de actividad biológica empleando datos reales obtenidos del ensayo NCI Yeast Anticancer Drug Screen. Esta muestra de trabajo tiene una relación de 1×1 entre moléculas activas e inactivas. Los datos están dados por 4000 compuestos definidos por 25 descriptores cada uno. Para comprobar la calidad de las predicciones se tomaron 3500 datos para crear los modelos y los 500 restantes para predecir sobre ellos.

Durante las pruebas que se realizaron sobre los kernels incorporados (Laplace y Exponencial) se movieron los valores de las variables gamma, nu y costo dependiendo en cada caso del kernel empleado y buscando los extremos de la efectividad de los kernels.

Al realizar las pruebas se obtuvieron los resultados que muestran las siguientes tablas:

Variables, costo=100		POT_EC		POT_SA	
		Kernel % exactitud		Kernel % exactitud	
?u	gamma	Laplace	Exponencial	Laplace	Exponencial
0.5	0.1	100	100	100	100
0.1	0.1	100	100	100	100
0.5	0.5	100	100	100	100
0.1	0.5	100	100	100	100
0.01	0.1	100	100	100	100
0.01	0.5	100	100	100	100
0.01	0.99	100	100	100	100
0.01	0.01	97.53	98.76	98.07	98.07

0.5	0.01	97.53	98.76	98.07	98.07
0.99	0.01	97.53	98.76	98.07	98.07
0.8	0.01	97.53	98.76	98.07	98.07
0.9	0.01	97.53	98.76	98.07	98.07
0.95	0.01	97.53	98.76	98.07	98.07
0.88	0.01	97.53	98.76	98.07	98.07
0.1	0.9	100	100	100	100
0.5	0.9	100	100	100	100
0.3	0.9	100	100	100	100

Tabla 1:

Resultados obtenidos con los kernels de procesamiento Laplace y Exponencial, fijando la variable de entrada costo en 100 y haciendo oscilar los valores de nu y gamma entre 0,01 y 0,99 en ambos casos. POT_EC y POT_SA representan las pruebas hechas sobre los juegos de datos de la actividad biológica ante la Escherichia coli (POT_EC) y la Staphylococcus aureus (POT_SA), respectivamente.

Variables, nu=0.5		POT_EC		POT_SA	
		Kernel % exactitud		Kernel % exactitud	
Costo	gamma	Laplace	Exponencial	Laplace	Exponencial
10000	0.1	100	100	100	100
1000	0.1	100	100	100	100
1000	0.5	100	100	100	100
1000	0.9	100	100	100	100
10000	0.5	100	100	100	100
10000	0.9	100	100	100	100
10000	0.01	100	100	100	100
1000	0.01	100	100	100	100
10000	0.99	100	100	100	100
100	0.1	100	100	100	100
10	0.1	98.76	98.76	100	99.03
1	0.1	86.41	80.24	81.73	81.73
100	0.01	97.53	98.76	98.07	98.07
10	0.01	81.48	80.24	81.73	78.84
100	0.5	100	100	100	100
10	0.5	100	100	100	100
1	0.5	98.76	93.82	99.03	91.34
100	0.9	100	100	100	100
1	0.9	100	98.76	100	99.03

Tabla 2:

Resultados obtenidos con los kernels de procesamiento Laplace y Exponencial, fijando la variable de entrada un en 0.5 y haciendo oscilar los valores de costo entre 1 y 10000 y los de gamma entre 0,01 y 0,99. POT_EC y POT_SA representan las pruebas hechas sobre los juegos de datos de la actividad biológica ante la Escherichia coli (POT_EC) y la Staphylococcus aureus (POT_SA), respectivamente.

Variables, costo=100		NCI Yeast Anticancer Drug Screen Kernel % exactitud	
?u	gamma	Laplace	Exponencial
0.5	0.1	83.39	87.00
0.1	0.1	83.39	87.00
0.5	0.5	85.60	86.80
0.1	0.5	85.60	86.80
0.01	0.1	83.39	87.00
0.01	0.5	85.60	86.80
0.01	0.99	87.00	86.40
0.01	0.01	84.80	88.20
0.5	0.01	84.80	88.20
0.99	0.01	84.80	88.20
0.8	0.01	84.80	88.20
0.9	0.01	84.80	88.20
0.95	0.01	84.80	88.20
0.88	0.01	84.80	88.20
0.1	0.9	87.20	86.60
0.5	0.9	87.20	86.60
0.3	0.9	87.20	86.60
0.1	0.99	87.00	86.40
0.5	0.001	87.80	88.80
0.99	0.95	87.20	86.60

Tabla 3:

Resultados obtenidos con los kernels de procesamiento Laplace y Exponencial, separando la muestra original de datos, en una muestra de 3500 para entrenamiento y otra muestra de 500 para realizar sobre ella la predicción. Los valores de la variable costo se fijaron en 100, mientras los de gamma oscilaron entre 0,01 y 0.99; así mismo los de nu se movieron entre 0.01 y 0.99.

Variables, nu=0.5		NCI Yeast Anticancer Drug Screen Kernel % exactitud	
Costo	gamma	Laplace	Exponencial
10000	0.1	83.60	86.80
1000	0.1	83.60	86.80
1000	0.5	85.60	86.80
1000	0.9	87.20	86.60
10000	0.5	85.60	86.80
10000	0.9	87.20	86.60
10000	0.01	82.19	86.60
1000	0.01	82.00	87.00
10000	0.99	87.00	86.40
100	0.1	83.39	87.00
10	0.1	86.20	88.20
1	0.1	87.80	88.80
100	0.01	84.80	88.20
10	0.01	87.80	88.80

100	0.5	85.60	86.80
10	0.5	85.80	86.60
1	0.5	88.40	88.00
100	0.99	87.00	86.40
1	0.99	88.00	88.40
1000	0.99	87.00	86.40
1	0.01	87.80	88.80
1	0.9	88.00	88.40

Tabla 4:

Resultados obtenidos con los kernels de procesamiento Laplace y Exponencial, separando la muestra original de datos, en una muestra de 3500 para entrenamiento y otra muestra de 500 para realizar sobre ella la predicción. Los valores de la variable costo se movieron entre 1 y 10000, mientras los de gamma oscilaron entre 0,01 y 0.99; así mismo los de nu se fijaron en 0,5.

Conclusiones

Como se puede apreciar en las tablas los resultados son muy satisfactorios aunque se trate de un juego de datos bastante pequeño, lo que fortalece las predicciones, no se puede perder de vista que la fortaleza de las Máquinas de Soporte Vectorial se encuentra en capacidad de procesar grandes cantidades de datos. Por estos motivos el alcanzar resultados superiores al 78 % en la exactitud de todas las predicciones se trata de resultados muy alentadores.

Algo muy importante es además la estabilidad de todos los kernels, que al realizarse importantes variaciones en los parámetros de entrada, mientras se mantiene el mismo juego de datos, el sistema es capaz de lograr resultados aceptables de manera constante.

Se han realizado además pruebas sobre juegos de datos de hasta 4000 compuestos definidos por más de una veintena de descriptores y, con una correcta selección de variables, se han alcanzado predicciones con un muy alto grado de exactitud. Si bien los resultados han alcanzado picos de efectividad muy altos al emplearse sobre muestras menores (alrededor de un centenar), al aumentar la cantidad de compuestos de entrenamiento y a predecir, se han logrado resultados mucho más estables, todos superando el 82% de efectividad.

La adición de nuevos kernels, se presenta sola como la opción a seguir para mejorar las capacidades de la presente solución, principalmente kernels como el stump y el perceptron, para los que se puede reutilizar gran parte de las funciones desarrolladas y que pueden brindar relevantes resultados.

Autores: Ing. Yoamel Acosta Morales [1], Ing. Yixander Yero Tarancón [2].

1: Universidad de las Ciencias Informáticas, La Habana, Cuba. Departamento de Programación y Sistemas Digitales, Facultad # 6;

2: Universidad de las Ciencias Informáticas, La Habana, Cuba. Dirección de Calidad.

Bibliografía de interés

1. ESCALONA, J. C.; CARRASCO, R. Introducción al diseño de Fármacos. Disponible en: .
2. CHEN, N.; YANG, J., et al. Support Vector Machine in Chemistry. World Scientific, 2004.
3. R, B.; M, T., Drug design by machine learning: support vector machines for pharmaceutical data analysis.

Disponible en: http://citeseer.ist.psu.edu/528480.html.
4. HSU, C.-W.; CHANG, C.-C., et al. A Practical Guide to Support Vector Classification . 1-12. Disponible en: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.
5. Vapnik, Statistical Learning Theory, 1998.
6. The SVM algorithm can also be extended to cope with noise in the training set and with multiple classes (Cristianini and Shawe-Taylor, An Introduction to Support Vector Machines, 2000).