

Leonel Exequiel Gómez

August 17, 2016

1 Introducción

Implementaremos un experimento basado en el paper de ? y ?. El objetivo del experimento es dado un conjunto de imágenes de difusión de alta calidad, usar la información presente en las mismas para mejorar la resolución de imágenes de difusión de menor calidad. Esto es conocido como transferencia de calidad de imágenes (o *image quality transfer* del término en ingles).

Usaremos el conjunto de datos *standford hardi*, provisto por la librería *Dipy* como nuestra imagen de alta calidad. El mismo tiene una resolución espacial de $2 \times 2 \times 2 \text{ mm}^3$ con 160 direcciones gradientes con $b = 2000 \text{ s/mm}^2$. Artificialmente generaremos (utilizando el método *reslice* de *Dipy* que hace una interpolación cubica) una imagen de difusión de baja calidad, a partir de la imagen provista por *standford hardi*. Luego mejoraremos la resolución de la misma y la compararemos con la original (figura 1).



Figure 1: Corte coronal del conjunto de datos S_0 . A la izquierda la imagen original de $81 \times 106 \times 76$ y a la derecha su equivalente disminuida en resolución de $43 \times 55 \times 40$. El rectángulo muestra el sub volumen con el que trabajaremos

2 Experimento

Dada una imagen en baja resolución predeciremos su equivalente en alta resolución. Para ello denotaremos como Y^{LR} a la imagen en baja resolución representada como un vector columna de dimensión N^{vrb} con N^{vrb} la cantidad de voxels por la cantidad de gradientes de la imagen. Denotaremos como Y^{HR} a la imagen en alta resolución representada como un vector columna de dimensión N^{vhrb} con N^{vhrb} la cantidad de voxels por la cantidad de gradientes de la imagen. Al igual que [?] consideramos a las imágenes en baja resolución como la versión sub muestreada de su equivalente en alta resolución. Luego el modelo de adquisición de la imagen de alta resolución lo podemos expresar como

$$Y^{LR} = GY^{HR}$$

Donde G es la matriz de sub muestreo de la resolución espacial. En nuestra implementación obtendremos G entrenando un algoritmo de *machine learning* con pares de sub volúmenes en baja resolución con su correspondiente en alta resolución. Para eso construiremos el conjunto de entrenamiento $T = \{x_i, y_i\}_i^{|T|}$, donde cada x_i tiene dimensión $N^{v_{lr}b}$ y los y_i dimensión $N^{v_{hr}b}$. En este caso lo construimos a partir de un mismo sub volumen pero con ruido *gauseano* agregado artificialmente .

Usaremos el algoritmo de *machine learning LinearRegression*, provisto por la librería *Scikit Learn*. Para entrenar el algoritmo dispusimos 10 pares de sub volúmenes. Los tamaños de los sub volúmenes fueron de $6 \times 6 \times 6$ y $12 \times 12 \times 12$ para la baja y para la alta resolución respectivamente. Ambos con 6 gradientes diferentes (todos ellos con $b = 2000 \text{ s/mm}^2$).

Con el algoritmo *machine learning* computaremos la transformación lineal $G = YX^\dagger$ donde Y tiene como columnas los y_i , X tiene como columnas los x_i y X^\dagger es la pseudo inversa de X . Luego de obtener G , planteamos el siguiente problema de optimización convexa para obtener la imagen en alta calidad buscada

$$\min_{Y^{HR}} \{ \|GY^{HR} - Y^{LR}\|^2 \}$$

Dicha optimización la calculamos usando la librería *CVXPY*.

Probamos el algoritmo para diferentes conjuntos de entrenamiento. Para ello, agregamos aleatoriamente ruido *gauseano* a cada par del conjunto de entrenamiento. Es decir, para cada par en el conjunto de entrenamiento T^{SNR} agregamos ruido con una relación señal ruido igual a SNR (lo hicimos con $SNR^1 = 25, 50, 75, 100$ y Inf). A partir de estos conjuntos obtuvimos distintas G^{SNR} . En la figura 2 podemos ver los sub volumen en alta y baja resolución originales y uno de los pares usados para entrenar al azar.

¹SNR lo definimos aquí como S_0/σ , donde σ es el desvío estándar de la distribución gauseana.

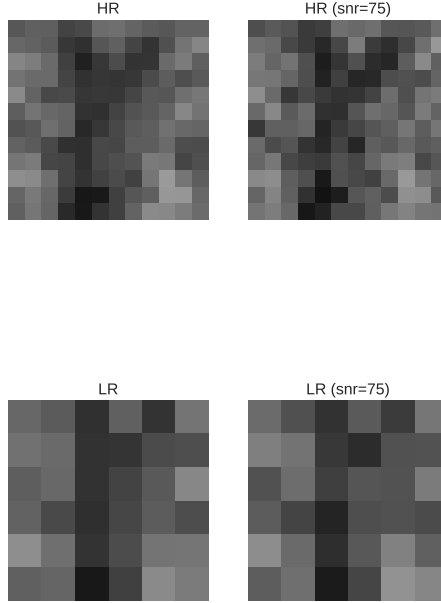


Figure 2: En la fila arriba de izquierda a derecha la imagen en alta resolución (original) y la equivalente con el ruido adicionado respectivamente. En la fila de abajo, de izquierda a derecha la imagen en baja resolución sin ruido y la equivalente con el ruido adicionado respectivamente. El par con ruido pertenece al conjunto de entrenamiento T^{75}

En la sección de resultados compararemos la imagen original con la reconstruida por el experimento. Luego compararemos los resultados según el ruido agregado artificialmente al conjunto de entrenamiento con el que estimamos G . De esta manera veremos como se comporta nuestro método cuando el conjunto de datos con el que contamos para hacer la transferencia de calidad no es tan bueno.

3 Resultados

Comparamos la imagen original con la reconstruida usando las siguientes métricas:

- Error cuadrático medio por voxel: Tomamos la imagen en su representación DWI y calculamos el error cuadrático medio (ECM) por cada voxel del volumen.
- Error cuadrático medio por gradiente: Tomamos la imagen en su representación DWI y calculamos el error cuadrático medio de todos los voxels por cada gradiente. Esta métrica nos permite observar cuanto dista la imagen reconstruida con la original según el gradiente.
- Coeficiente de determinación: El Coeficiente de Determinación, es el cuadrado del coeficiente de correlación de Pearson. Nos da la proporción de variación de la variable a predecir. Si la proporción es igual a 0, quiere decir que el modelo no es bueno. Si esta cerca de uno quiere decir que el modelo es bueno y para valores negativos significa que nuestro conjunto de datos no se comporta de manera lineal.

En la tabla 1 podemos ver el valor mínimo alcanzado por el algoritmo de minimización, el tiempo de ejecución del mismo, y las normas de la imagen estimada y la diferencia entre la estimada y la

original (siendo $\|Y_{original}^{HR}\| = 18093$), según la señal ruido agregada al conjunto de entrenamiento. Además por cada conjunto de entrenamiento probamos el algoritmo de minimización con distintos sub volúmenes $(Y^{LR})^{SNR}$ con distintos valores señal ruido.

SNR	Valor Óptimo	Tiempo	$\ Y^{HR}\ $	$\ Y^{LR}\ $	$\ Y^{HR} - Y_{original}^{HR}\ $
inf	3034 ± 494	5'07"	17908 ± 16	6427 ± 5	859 ± 3
200	3058 ± 530	5'02"	17926 ± 20	6433 ± 15	811 ± 3
100	3054 ± 556	5'08"	17917 ± 3	6430 ± 9	723 ± 1
75	3063 ± 540	5'33"	17948 ± 38	6441 ± 21	653 ± 6
50	3083 ± 559	5'04"	17893 ± 46	6420 ± 9	713 ± 14
25	3051 ± 500	5'09"	17931 ± 18	6435 ± 13	770 ± 3

Table 1: El valor alcanzado por el algoritmo de minimización, el tiempo de ejecución, la norma de la imagen estimada y la norma de la diferencia entre la estima y la original, en cada caso según el valor de señal ruido agregada al conjunto de entrenamiento. El algoritmo fue probado con disitintas Y^{LR} de entradas. Todas ellas con un valor diferente de señal ruido.

La figura 3 muestra una rebanada coronal de la imagen de difusión original, la producida por el método y el error cuadrático medio entre las últimas dos. Como podemos apreciar solo algunos voxels toman valores relativamente altos en la imagen ECM .

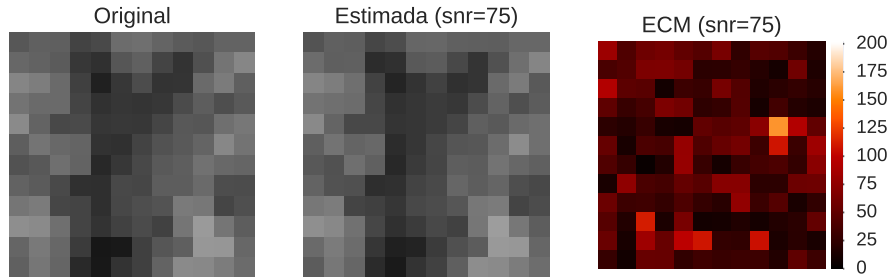


Figure 3: De izquierda a derecha una rebanada coronal de un gradiente arbitrario de la imagen original, la producida por el método y el error cuadrático medio por voxel de la representación DWI de la imagen. La imagen fue resultado del método con T^{75} y $(Y^{LR})^{75}$.

La figura 4 muestra también una rebanada coronal de la imagen original, la producida por el método y el coeficiente de determinación entre las últimas dos. Dicho coeficiente nos aporta información de cuan bueno fue el modelo que utilizamos para fitear los datos. En nuestro caso el modelo usado fue lineal y gran parte de los voxels presentan un coeficiente de determinación por encima del 95%. Esto significa que el modelo elegido es bueno para este conjunto de entrada.

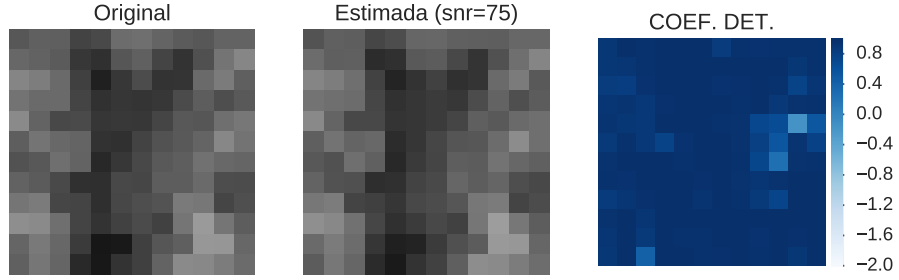


Figure 4: De izquierda a derecha una rebanada coronal de un gradiente arbitrario de la imagen original, la producida por el método y el el coeficiente de determinación por voxel de la representación DWI de la imagen. La imagen fue resultado del método con T^{75} y $(Y^{LR})^{75}$.

En la figura 5 graficamos el error cuadrático medio de todo el sub volumen por cada uno de los 6 gradientes utilizados para los distintos conjuntos de entrenamiento. Como se puede ver el error se hace marcadamente mas grande cuando la relacion señal ruido de la imagen disminuye.

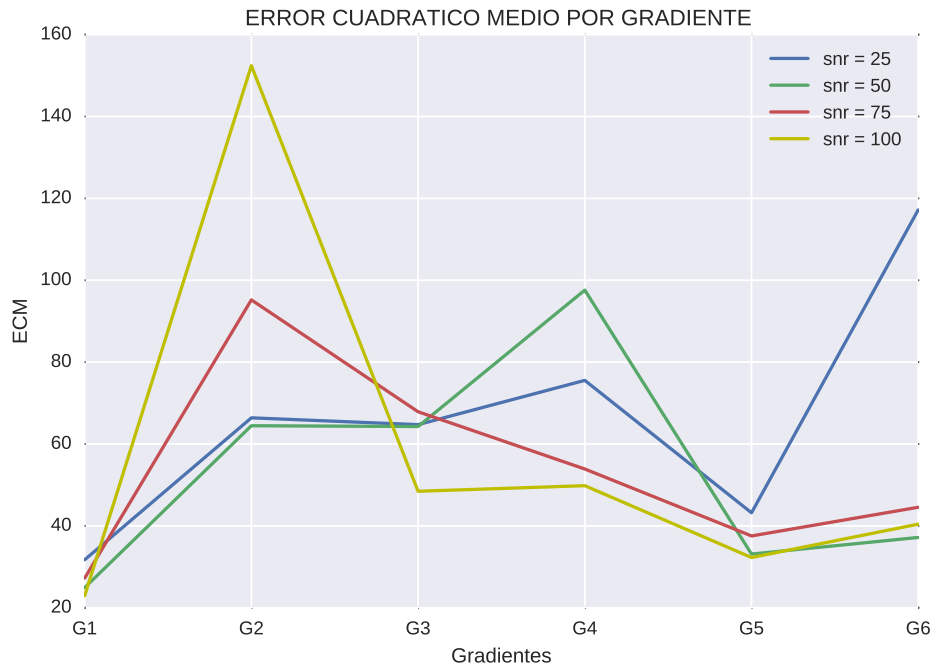


Figure 5: El error cuadrático medio por cada una de las 6 direcciones gradientes utilizadas por cada conjunto de entrenamiento con SNR igual a 25, 50, 75, 100 y 200.

4 Conclusiones

El modelo propuesto se comporta bien para valores no tan altos de relación señal ruido. Como se ve en la figura 5 cuando la relación señal ruido es inferior a 50 el error del método crece abruptamente.

A simple vista (incluso los resultados con $SNR = 25$) el modelo aproxima bien la imagen en cuanto a graficar la señal de difusión. Es decir, que respeta bastante bien el contraste entre voxels. Más allá de que las magnitudes de los mismos difieran con las de la original y eso cause diferencias sensibles al calcular el ECM . Habría que probar estos resultados al intentar usar los datos para hacer cosas mas complejas como por ejemplo un tractograma.

References

- Daniel C. Alexander, Darko Zikic, Jiaying Zhang, Hui Zhang, and Antonio Criminisi. Image quality transfer via random forest regression: Applications in diffusion MRI. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8675 LNCS(PART 3):225–232, 2014. ISSN 16113349. doi: 10.1007/978 – 3 – 319 – 10443 – 0₂₉.
- Lipeng Ning, Kawin Setsompop, Oleg Michailovich, Nikos Makris, Martha E. Shenton, Carl Fredrik Westin, and Yogesh Rathi. A joint compressed-sensing and super-resolution approach for very high-resolution diffusion imaging. *NeuroImage*, 125: 386–400, 2016. ISSN 10959572. doi: 10.1016/j.neuroimage.2015.10.061. URL <http://dx.doi.org/10.1016/j.neuroimage.2015.10.061>.