



**DEPARTAMENTO
DE COMPUTACION**

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico 1

“Si nos organizamos aprobamos todos...”

Metodos numericos
Primer Cuatrimestre de 2015

Integrante	LU	Correo electrónico
Gastón Zanitti	058/10	gzanitti@gmail.com
Ricardo Colombo	156/08	ricardogcolombo@gmail.com
Dan Zajdband	144/10	Dan.zajdband@gmail.com
Franco Negri	893/13	franconegri200@gmail.com



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

Índice

1. Introduccion	3
2. Desarrollo	4
2.1. Algoritmo de kNN	4
2.2. Optimizacion mediante Análisis de componentes principales	4
2.3. Cross-validation	4
3. Análisis	5
3.1. KNN	5
3.1.1. Cantidad de vecinos	5
3.1.2. Mejoras en kNN	5
3.2. PCA	5
3.2.1. Lambda inicial	5
3.2.2. Algo mas que no me acuerde	5
4. Resultados	6
4.1.	6
5. Conclusiones	7
5.1. Aca tiramos tiros	7
6. Apendice	8

1. Introduccion

INTRO!!

2. Desarrollo

2.1. Algoritmo de kNN

Como primera aproximación para la resolución del problema de OCR, implementamos el algoritmo de *Kvecinosmascercanos* (o *kNN* por sus siglas en ingles). Este algoritmo consiste basicamente en la idea de que entradas parecidas, a partir de una metrica definida en la implementacion (que para este caso podria variar desde, por ejemplo, desde la cantidad de puntos arriba de cierto valor para contabilizar la cantidad de valores *negros* y *blancos* hasta la norma de cada vector digito) presentaran características definidas y al ser ubicadas sobre un plano se agruparan de acuerdo a estas. Luego, para clasificar un nuevo objeto, basta con ubicarlo dentro de este plano y promediar la etiqueta de los k vecinos mas cercanos para obtener una clasificación. Sin embargo, y mas alla de las mejoras que puedan realizarse sobre los datos en crudo, este algoritmo es muy sensible a la variabilidad de los datos. Un conjunto de datos con un poco de dispersion entre las distintas clases de clasificacion, hace empeorar rapidamente los resultados. Analizaremos en la segunda etapa del trabajo practico, una forma de solucionar este inconveniente mediante el análisis de componentes principales.

//ACA UN PSEUDOCODIGO?

2.2. Optimizacion mediante Análisis de componentes principales

En esta segunda parte, utilizaremos una tecnica conocida como *analisisdecomponentesprincipales* como una forma de optimizar los resultados de la primera etapa. El analisis de componentes principales (o PCA) consiste basicamente en conseguir una descomposicion de los datos en sus matrices ortogonales de valores principales para obtener una transformacion lineal que resuma la información mas relevante de cada imagen, descartando aquellos valores que no aportan datos y resultan redundantes. Para realizar este procedimiento tomamos la matriz de covarianza como una forma de expresar la relacion de dependencia intrinseca entre cada variable. A partir de esta información y mediante el metodo de la potencia, obtenemos un vector P que, al multiplicarlo por nuestros valores originales, realiza el cambio de coordenadas minimizando la covarianza.

//ACA UN PSEUDOCODIGO?

2.3. Cross-validation

3. Análisis

Ahora vamos a analizar el algoritmo del PCA.

3.1. KNN

Análisis de KNN

3.1.1. Cantidad de vecinos

Mover el k y presentar un análisis

3.1.2. Mejoras en kNN

Cortar el dataset en x valor y comparar para ver si mejora

3.2. PCA

Análisis de PCA

3.2.1. Lambda inicial

3.2.2. Algo mas que no me acuerde

4. Resultados

4.1.

5. Conclusiones

5.1. Aca tiramos tiros

6. Apendice