IIC 3633 - Recommender Systems
Proyecto - Entrega 1
WineRec

Paula Navarrete Campos & Astrid San Martín
Department of Computer Science
School of Engineering
Pontifical Catholic University
pcnavarr@uc.cl, aesanmar@uc.cl

I. Contexto del problema

Reflexionando en torno a la propuesta inicial entregada y tomando en cuenta el feedback proporcionado, vemos que el proyecto no se enmarca como uno de predicción de recomendaciones personalizadas en base a preferencias del los usuarios si no que como uno de apoyo a la toma de decisiones. Junto con esto, nos vimos enfrentadas a temas de confidencialidad de los datos ya que no está disponible información desagregada respecto de las calificaciones de los jueces y tampoco información adicional de cada uno de ellos (debido a cómo se lleva a cabo el proceso de selección) y no nos sería posible presentarla en detalle en este trabajo ya que es confidencial de Corfo.

En torno a esto decidimos cambiar el dataset y realizar la tarea de recomendación con un dataset de vinos y licores recopilado desde el sitio *cellartracker.com*

Las recomendaciones en torno al rubro vitivinícola están enmarcadas dentro de las tareas donde las preferencias del usuario son difíciles de predecir, por ejemplo algunos usuarios tienen preferencia por los sabores ácidos y valorarán estas características subjetivas por sobre las de otros tintos, mientras que otros discriminan por región productiva. Con esto, vislumbramos que un sistema recomendador es un *must-have* para el comercio vitivinícola on-line. Buenas recomendaciones entimularán el comportamiento de compra de los usuarios. Basándonos en esto, apuntamos a desarrollar un modelo que nos permita hacer recomendaciones de vinos y licores a usuarios nuevos y antiguos.

II. Problema y su justicación

En el rubro vitivinícola nos encontramos con una amplia variedad de características que definen las cualidades de un vino o licor, por ejemplo sus cepas, denominación de origen, viña productora, etc. Quienes se especializan como catadores, al evaluar un vino toman en cuenta una diversidad de características objetivas y percepciones organolépticas que califican según su experiencia y gusto desarrollado a través la experiencia. Junto con esto, una componente primordial en el contexto de valorar un vino o tomar en cuenta sus recomendaciones tiene que ver con su temporalidad, es decir, año de cosecha, antiguedad de la recomendación, incluso tiene que que ver con la antiguedad como catador.

Para un consumidor común, los parámetros que los llevan escoger un vino varían según la valoración personal que dan a ciertas características, algunos consumidores prefieren los vinos blancos, otros los ensamblajes, otros valoran aquellos que son orgánicos por sobre otros, etc. Así, cuando un consumidor poco experimentado ingresa al círculo vitivinícola, tomar la decisión sobre qué vino probar se hace dificíl, y probablemente tome en consideración la sugerencia de algún consumidor más ”experimentado” o de un usuario que considere ideosincráticamente similar a ella o él.
Es aquí donde los sistemas recomendadores pueden entregar una buena solución a la hora de recomendar personalizadamente un vino o estimar la valoración que porbablemente conferirá a un consumidor no tan docto en la materia.
Existen sitios web dedicados a la revisión de vinos de todo el mundo de la más amplia gamma de variedades, con comentarios de expertos catadores y de usuarios comúnes. Con esto encontramos los ingredientes perfectos para poder entregar un servicio de útilidad con recomendaciones basadas en lo que conocemos sobre los gustos de los consumidores y usuarios de dichas páginas.

III. Objetivos

El objetivo de este trabajo es desarrollar un recomendador que pueda explotar tanto los ratings como la información del contenido de los items. A diferencia del enfoque tradicional de *collaborative filtering*, enmarcamos el problema como uno de calificación de itemes ayudándonos de la calificación de estos. Por otro lado, diferimos de los métodos *content based*, en que utlizaremos la información social, en forma de calificaciones de otros usuarios, en el proceso de aprendizaje inductivo. En particular, formalizaremos el problema de recomendar un *vino* como un problema de aprendizaje; específicamente, el problema de aprender una función f que toma como entrada a un $usuario$ y un $vino$ y produce como resultado un score que indica si el *vino* sería de su agrado (y por lo tanto lo comprará) o no, es decir:

$$f(\langle user, wine \rangle) \longrightarrow \{score\} \quad (1)$$

IV. SOLUCIÓN PROPUESTA

A. Obtención del Dataset

Usamos web scraping para extraer datos del sitio web *CellarTracker.com* a través de un proceso automatizado. CellarTracker fue creado en marzo de 2003 por Eric LeVine [1] como una forma de mantener un registro de su propia bodega mientras estaba trabajando en Microsoft, lanzando públicamente el sitio en abril de 2004. Hoy en día CellarTracker es la principal herramienta de gestión de bodegas personales con cientos de miles de coleccionistas que rastrean más de 75 millones de botellas. CellarTracker también se ha convertido en la base de datos más grande de notas de cata con más de 5.8 millones de notas registradas a fin de 2016. Anualmente millones de entusiastas del vino visitan el sitio para leer reseñas y obtener recomendaciones de vinos. Aún así, CellarTraker al día de hoy no presta ningún tipo de inteligencia que le permita recomendar personalizadamente un vino a algún usuario, solo es capaz de entregar estadísticas básicas para cada vino (rating promedio, y promedio like/dislike). El panorama local es menos prometedor, los sitios chilenos de venta de vino online son muy rudimentarios, disponen un sistema simple de reviews y no cuentan con una cantidad de reviews significativa.

B. Dataset

Para obtener el dataSet recopilamos datos de vinos y licores para todas las regiones de aproximadamente 80 paises[2]. Para cada país recolectamos las primeras 5 páginas de resultados con información de cada item (aprox 125 items) para las regiones de cada país[3]. También recopilamos los reviews de la comunidad activa para cada uno de los vinos (items) recolectados, Al momento, hemos recopilado la información señanada para items en un rango de precios de 20 a 40 dólares la botella[4].

Contamos con dos archivos, uno que contiene la información relativa a cada item o *vino* y otro que contiene los reviews de los usuarios a cada vino de la lista. A la fecha contamos con la infomación de aproximadamente 18,300 items, de los cuales al rededor 10.900 han sido evaluados por al menos un usuario; en total contamos con al rededor de 49.550 reviews. Pretendemos recopilar la información para items en los rangos de precio 20 usd o menos, 40 a 80 usd y más de 80 usd.

1) Items: Para cada vino o licor (*item*) contamos con la información referente a:

- **Score** - puntuación de 1 a 100.
- **Reviews** - url que aloja todos los reviews del item.
- **Vintage** - año de producción.
- **Type** - tipificación del vino, Ej: rojo, blanco, rosé, etc.
- **Producer** - nombre del productor (quien hizo el vino).
- **Variety** - cepa(s) utilizadas para producir el vino. Ej: Cabernet, Syrah, merlot, etc.
- **Designation** - designación dentro de la viña de donde provienen las uvas que elaboraron el vino.
- **Vineyard** - viña donde se produjo el vino.
- **Country** - país de producción.
- **Region** - indica de dónde se obtuvieron las uvas para producir el vino.
- **Subregion** - area específica dentro de una Región de dónde se obtuvieron las uvas para producir el vino.
- **Appellation** - denominación de origen. Indica el valle de dónde se obtuvieron las uvas para producir el vino.

La Tabla I ilustra la estructura mencionada.

2) Reviews: Recopilamos la siguiente información relativa a los reviews disponibles para los items:

- **UserID**: Identificador del usuario
- **Review**: texto con el comentario con la experiencia de cata del vino.
- **Web_Page_URL**: url que aloja el comentario
- **Timestamp**: fecha en que fue referido el comentario.
- **Score**: puntuación que el usuario ha dado al vino.

La Tabla II ilustra la estructura mencionada.

3) Users: Recopilamos la siguiente información relativa a los usuarios:

- **User_ID**: Identificador del usuario
- **Intake_qty**: cantidad de botellas consumidas.
- **Reviews_qty**: cantidad de reviews del usuario. Para cada review del usuario rescatamos:
  - **Review**: texto con el comentario con la experiencia de cata.
  - **Score**: puntuación que el usuario sobre esa experiencia.

La Tabla III ilustra la estructura mencionada.

C. Modelo

Las técnicas de Machine Learning detectan patrones en los datos y usan esos patrones para predecir resultados futuros, ayudando así a la toma de decisiones bajo incertidumbre [2]. Por lo tanto, del mismo modo que los expertos, al confiar en sus observaciones de correlaciones entre experiencias pasadas y sus resultados subsecuentes, desarrollan y adaptan de forma colectiva reglas simples y únicas para dirigir la toma de decisiones, también lo hacen las reglas supervisadas de Machine Learning, basadas en el reconocimiento de patrones entre multiples variables y su variable de respuesta [1].

Por lo tanto, al igual que el desarrollo de heurísticas individuales (ej. los vinos de cierta zona generalmente son

[1] https://www.cellartracker.com/content.asp?iContent=3

[2] Recopilamos la información para todos los paises donde CellarTraker tiene información.

[3] Recolectamos las 5 primeras páginas por restricciones de tiempo. La base de datos es extensa y para ciertas regiones que se destacan por su producción vitivinícola, muchas paginas quedaron sin ser consideradas en este dataset.

[4] Uno de los contratiempos a los que nos vimos expuestas fue la politica que cellartracker de bloquear IP que sospecha están realizando scrapping de sus datos, lo cual ralentizó en gran medida el proceso de obtención del dataset.

mas ácidos) puede ser modelado como un proceso de actualización Bayesiana, es decir, observaciones recurrentes de ciertos eventos dados, que preceden a la ocurrencia específica de un outcome, son 'aprendidos' como un eventos predictivos, las técnicas de machine Learning realizan iteraciones que prueban relaciones potenciales entre parámetros y un resultado, 'aprendiendo' qué parámetros son predictores más consistentes del resultado especificado.

El uso de sistemas recomendadores para entregar recomendación de productos o predecir cuál será la valoración de un usuario a cierto item de acuerdo a con las calificaciones entregadas a distintos items o respecto del comportamiento de usuarios similares, es uno de los problemas que podemos encontrar en ésta área [6]. El objetivo de obtener el perfil de un usuario, se puede abordar de dos formas: ya sea que el usuario entregue explicitamente la información ó reunir información de manera implicita que se relacione con el usuario.
Tenemos según Burke et al. 2007b [7] seis tipos diferentes de enfoques para recomendacion:

- Recomendación basada en contenidos: sugerir items basados en las preferencias historicas del usuario.
- Filtrado colaborativo: recomendación basada en usuarios que presentaron gustos similares.
- Recomendación basada en conocimiento: se recomiendan items basedós en el conocimiento de un área específica sobre cierta carácteristica que aborda las necesidades y preferencias del usuario, un sentido de utilidad para el ususario.
- Demográfico
- Recomendación basada en la comunidad
- Sitema híbrido de recomendación

Los sistemas de recomendación son una de las aplicaciones más exitosas y extendidas de las tecnologías de Machine Learning en negocios y proveen sugerencias a un usuario para apoyar su toma de decisiones. En particular, pueden ayudar a determinar si un usuario estaría interesado en adquirir un *vino o licor* en particular o no. Los algoritmos de Machine Learning en sistemas de recomendación se clasifican generalmente en dos categorías: métodos de filtrado basados en contenido (*content based*) y colaboración (*collaborative filtering*), aunque los recomendadores modernos combinan ambos enfoques [3]. Los métodos basados en el contenido se basan en la similitud de los atributos de los itemes y los métodos colaborativos calculan la similitud de las interacciones delos usuarios con los itemes.

Los métodos de *collaborative filtering* recopilan calificaciones de itemes de muchas personas y utilizan técnicas de *Nearest Neighbor* para hacer recomendaciones a un usuario sobre nuevos itemes. Sin embargo, estos no toman en cuenta la cantidad significativa información que generalmente está disponible sobre la naturaleza de cada item ni su temporalidad, dejando irresuelta la pregunta de qué rol puede jugar el contenido en el proceso de recomendación. Por el contrario, métodos del tipo *content-based* aceptan información que describe la naturaleza de un ítem, y basadose en una muestra de las preferencias del usuario, aprenden a predecir qué elementos gustarán al usuario. Ambos se pueden considerar problemas de aprendizaje cuyo objetivo es aprender una función que pueda tomar una descripción de un usuario y un item y predecir las preferencias del usuario con respecto a ese item.

Para modelar el proceso en el que los individuos desarrollan reglas de decisión utilizaremos un sistema recomendador, una técnica de aprendizaje supervizado que utiliza un set preclasificado de observaciones (resultados de interés) como un set de entrenamiento, que genera recomendaciones correlacionadas con un resultado de *performance* de interés que es este caso es su calificacion. Con este set de entrenamiento, el sistema computa interralaciones entre usuarios e itemes para clasificarlos, maximizando la capacidad de un conjunto de reglas iniciales (o features) para predecir la clasificación correcta de los resultados.

En este caso, los datos de entrada para el sistema recomendador preceden en el tiempo a los datos de resultado, lo que se asemeja a la forma en que los individuos infieren causalidad a través de un proceso de actualización bayesiano. Además, de forma consistente con la que los individuos descifran interralaciones entre variables predictivas y su respuesta asociada. El modelo de aprendizaje elegido es una *factorization machine* [4].

V. Descripción de experimentos

Presentamos una descripción detallada del set de datos que utilizaremos en la realización del proyecto. A su vez explicamos los distintos experimentos que se proponen, con su respectiva estrategia de evaluación.

1) Preprocesamiento de los Datos: Separamos nuestro set de datos en un set de entrenamiento y un de testeo, La idea es tener un set para poder entrenar nuestro algoritmo y luego testear el entrenamiento bajo nuestro set de testeo. El enfoque a utilizar es con *k-fold cross validation*. Manipulamos los datos para obtener la respresentación ilustrada en la tabla IV. Utilizamos one-of-K or one-hot coding para binarizar las variables categóricas.

Luego de este preprocesamiento apuntaremos a realizar exploraciones en el set de datos que son útiles para revelar algún patron bajo las revisiones y calificaciones, esto nos ayudará a decidir qué características son mejores o peores para hacer la predicción. Apuntaremos a determinar las relaciones entre:

- Año de producción versus average rating, el criterio común indica que los vinos mas añejados tendrán una calidad mayor. También realizaremos este estudio sobre las carácterísticas geográficas y la tipificación.
- Tamaño del review (texto) vs average rating
- Experiencia en relación con el tiempo vs average rating, en particular exploraremos la relacion entre el tiempo entre ratings (time_since_last) vs average score junto con la expertiz del usuario (reviewer_expertice), medida como cantidad de reviews realizados, vs average score. Como bonus intentaremos explorar la madurez del

reviewer (review_maturity) en términos de su tiempo promedio entre reviews y su average score.
- Consumo(intake_qty) vs average rating.

2) Métodos: Aplicaremos dos algoritmos para luego compararlos en cuanto a su resultados. El primer método escogido para la realización de éste proyecto que es el de *Factorization Machines* [4]. Aquí usamos parámetros factorizados, con lo cual podemos conocer la interacción, aún cuando nos encontramos con poca información (*"sparse data"*), escenario donde escala en $O(kn)$

Existen distintas tareas de predicción que podemos realizar: regresión, clasificador binario y ranking [4]. Según nuestros objetivos fijados usaremos *factorization machine* con ranking, es decir entregar la puntuación que el usuario entregaria y a partir de cierto threshold en la puntuación recomendar el item o no.

Como puntos de comparación realizaremos:

1) **UserKnn:** Incrementando el número de vecinos en 5 en cada paso. Comenzaremos con 5 vecinos terminaremos con 30.
2) **ItemKnn:** Seguiremos la misma estrategia, comenzar con 5 e incrementar en 5 hasta llegar a 30 vecinos.
3) **SlopeOne:** Este método no requiere el seteo de parámetros en particular para mejorar el performance.
4) **SVD:** Cambiaremos dos parametros en este modelo, el número de factores y el número de iteraciones. Incrementaremos el número de factores en incrementos de 10 hasta llegar a 100. Haremos lo mismo con las iteraciones.
5) **ALS:** Siguiendo una etsategia similar que en *SVD*, cambiaremos 2 parámetros de peste método, el número de factores y las iteraciones.

3) Evaluación: Aquí varios puntos deben ser definidos de antemano: una recomendación exitosa será recomendar un vino que este usuario le daría más que su puntaje promedio para otros vinos en el pasado. Así será una falta si se recomienda un vino que el usuario es probable que dé baja puntuación. El modelo queda planteado por la ecuación (1).

Al hacer eso, podemos obtener un punto con la calificación $rec(usuario, vino)$ para un par $(usuario, vino)$. Para evaluar el rendimiento, calcularemos el

- Mean Absolute Error (MAE)[8]
- Mean Squared Error (RMSE) [9]
- Mean Average Precision (MAP@10) [10]
- nDCG: normalized Discounted Cummulative Gain [11]
- Intra list Similarity [12]

También evaluaremos el tiempo de entrenamiento requerido. Con esto esperamos tener una evaluación sobre qué estrategia resulta con un mejor balance entre tiempo de entrenamiento y mejor predicción.

Para evaluar el desempeño de nuestro modelo, usaremos nuestro set de testeo. El set de test estará compuesto por 30% de los reviewsmas recientes y el training dataset estará compuesto por el 70% más antiguo.

4) Implementación: La implementación del modelo se realizará con tffm [5], una libreria de *factorization machine* en *Python*, usando de base *Tensorflow.* Una ventaja de usar esta libreria es que podemos usar una implementación especial en GPU, y asi aprovechar en su totalidad la capacidad que nos entrega *Google Collaboratory.*. La tabla IV ilustra un bosquejo de la estructura de datos propuesta como input. En el github del proyecto dejamos un bosquejo de la implementación de la factorization machine sobre un set reducido de los datos.

REFERENCES

[1] Leatherbee, M., del Sol, P. (2016). Predicting Entrepreneurial Performance: Simple Rules versus Expert Judgment. Working Paper. Available at: http://ctie.economia.cl/wp-content/uploads/2017/07/Predicting-Entrepreneurial-Performance-Simple-Rules-2016. pdf (accessed May 2017).

[2] Murphy, K. (2012). Machine learning: a probabilistic approach. Massachusetts Institute of Technology, 1-21.

[3] Schafer, J. B., Frankowski, D., Herlocker, J., Sen, S. (2007). Collaborative filtering recommender systems. In The adaptive web (pp. 291-324). Springer Berlin Heidelberg.

[4] Rendle, S. (2010). Factorization machines. In Data Mining (ICDM), 2010 IEEE 10th International Conference on (pp. 995-1000). IEEE.

[5] Trofimov, M. Novikov, A. (2016). tffm: TensorFlow implementation of an arbitrary order Factorization Machine. GitHub, GitHub repository, https://github.com/geffy/tffm

[6] Nilashi, M., Bagherifard, K., Ibrahim, O., Alizadeh, H. Collaborative Filtering Recommender Systems. Research Journal of Applied Sciences, Engineering and Technology 5(16): 4168-4182, 2013.

[7] Burke, R.D., 2007b. Hybrid web recommender systems. Lect. Notes Comput. Sc., 4321: 377-408.

[8] Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: 14th Conference on Uncertainty in Artificial Intelligence, pp. 43–52 (1998)

[9] Shardanand, U., Maes, P.: Social information filtering: algorithms for automating word of mouth. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 1995, pp. 210–217. ACM Press/Addison-Wesley Publishing Co., New York (1995)

[10] Manning, C.D., Raghavan, P., Schtze, H.: Introduction to Information Retrieval. Cambridge University Press, New York (2008)

[11] J'arvelin, K., Kek'al'ainen, J.: Cumulated gain-based evaluation of ir techniques. ACM Trans. Inf. Syst. 20, 422–446 (2002)

[12] Ziegler, C.-N., McNee, S.M., Konstan, J.A., Lausen, G.: Improving recommendation lists through topic diversification. In: Proceedings of the 14th International Conference on World Wide Web, WWW 2005, pp. 22–32. ACM, New York (2005)

Score	Reviews	Vintage	Type	Producer	Variety	Designation	Vineyard	Country	Region	Subregion	Appellation	Pricelevel
88	https://www.cellartracker.com/notes.asp?iWine=858606	2006	Red	'Tilda	Syrah Blend	Petulance	n/a	USA	Washington	Columbia Valley	Columbia Valley	1
	https://www.cellartracker.com/editnote.asp?iWine=2638585	2016	Rosé - Sparkling	14 Hands	Champagne Blend	n/a	n/a	USA	Washington	Columbia Valley	Yakima Valley	1
75.5	https://www.cellartracker.com/notes.asp?iWine=2630939	2015	Red	100 MILE VINEYARD	Merlot	n/a	n/a	USA	California	Central Valley	Lodi	1
86	https://www.cellartracker.com/notes.asp?iWine=2914131	2016	Red	100 MILE VINEYARD	Zinfandel	n/a	n/a	USA	California	Central Valley	Lodi	1
	https://www.cellartracker.com/notes.asp?iWine=2987125	2016	Red	1000 Stories	Carignan	Bourbon Barrel Aged	n/a	USA	California	North Coast	Mendocino County	1
	https://www.cellartracker.com/editnote.asp?iWine=3008228	2016	Red	1000 Stories	Red Blend	Gold Rush Red Bourbon Barrel Aged	n/a	USA	California	n/a	California	1
87.7	https://www.cellartracker.com/notes.asp?iWine=726338	2007	White	12 Mile Trail	Chardonnay	n/a	merryvale	USA	California	Napa Valley	St. Helena	1

TABLE I

ITEM DATASET STRUCTURE

$UserID$	$Review$	Web_Page_URL	$Score$
Dukeies21	Very good	https://www.cellartracker.com/notes.asp?iWine=1903563	97
dssinger	Very pleasant!	https://www.cellartracker.com/notes.asp?iWine=2022206	87
Villa D	Not bad for a twist off cap. Robert Rex's winery does a very good job of offering "clean wines" (little as possible sulfites) which my wife likes because of no after drinking. Still, a very good every day wine.	https://www.cellartracker.com/notes.asp?iWine=2022206	89
ellahazard	Yummy and inexpensive, good gift?Tastes more expensive than it is...	https://www.cellartracker.com/notes.asp?iWine=2022206	
cnr128	Love those Central Coast Syrahs with the peppery berry thing going on...in this case in a Rhone (SMG) blend. Like this quite a bit, especially for the price.	https://www.cellartracker.com/notes.asp?iWine=773039	86

TABLE II

REVIEWS DATASET STRUCTURE

$User\_ID$	$Intake\_qty$	Reviews_qty	Reviews
Dukeies21	459	5	$reviews = \{(\text{"Awful .."}, 75), \ldots, (\text{"this..."}, 89)\}$
dssinger	874	384	$reviews = \{(\text{"Very .."}, 87), \ldots, (\text{"Super ..."}, 76)\}$
Villa D	103	23	$reviews = \{(\text{"The .."}, 75), \ldots, (\text{"At th..."}, 89)\}$
ellahazard	85	58	$reviews = \{(\text{"Fruity .."}, 74), \ldots, (\text{"Tastes ..."}, 84)\}$

TABLE III

USER DATASET STRUCTURE

	User				Wine				Other wines rated				Last wine rated				Score	
x_1	1	0	0	...	1	0	0	...	0.3	0	0.3	...	0	0	0	...	66	$\rightarrow y_1$
x_2	1	0	0	...	0	0	1	...	0.3	0	0.3	...	1	0	0	...	89	$\rightarrow y_2$
x_3	0	1	0	...	0	0	1	...	0	0.5	0.5	...	0	0	0	...	72	$\rightarrow y_3$
x_4	0	1	0	...	0	1	0	...	0	0.5	0.5	...	0	0	0	...	70	$\rightarrow y_3$
x_5	0	0	1	...	1	0	0	...	0.5	0	0.5	...	0	0	0	...	92	$\rightarrow y_3$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_n	0	0	1	...	0	0	1	...	0.5	0	0.5	...	0	0	1	...	68	$\rightarrow y_n$
	$\uparrow$	$\uparrow$	$\uparrow$		$\uparrow$	$\uparrow$	$\uparrow$		$\uparrow$	$\uparrow$	$\uparrow$		$\uparrow$	$\uparrow$	$\uparrow$			
	$Anna$	Bob	$Mary$	...	w^1	w^2	w^3	...	w^1	w^2	w^3	...	w^1	w^2	w^3			

TABLE IV

DATA STRUCTURE