Measuring Data Dependencies in Large Databases

Gregory Piatetsky-Shapiro, Christopher J. Matheus
GTE Laboratories, MS 45
40 Sylvan Road, Waltham MA 02154
{gps0,matheus}@gte.com

Abstract

Data dependencies play an important role in analyzing and explaining the data. In this paper, we look at dependencies between discrete values and analyze several dependency measures. We examine a special case of binary fields and show how to efficiently use SQL interface for analyzing dependencies in large databases.

1 Introduction

Analysis of data dependencies is an important and active area of research. A number of methods have been developed in database theory for determining functional dependencies (Mannila and Raiha 1987), (Siegel 1988), where the value of one field *certainly and precisely* determines the value of second field.

There are many more approximate dependencies, where the value of one field determines the value of another field with some uncertainty or imprecision. Knowing such dependencies is helpful for understanding domain structure, relating discovered patterns, data summarization (Piatetsky-Shapiro and Matheus 1991), and improving learning of decision trees (Almoaullim and Dietterich 1991).

Several methods have recently been developed for discovery of dependency networks. A method for determining dependencies in numerical data using the Tetrad differences is given in (Glymour et al 1987, Spirtes et al 1993). Methods for analyzing the dependency networks and determining the directionality of links and equivalence of different networks are presented in (Geiger et al 1990). Pearl (1992) presents a comprehensive approach to inferring causal models. For discrete-valued data, there is a Cooper and Herskovitz (1991) Bayesian algorithm for deriving a dependency network. A problem with these approaches is that they rely on assumptions on data distribution, such as normality and acyclicity of the dependency graph. Not all of these methods provide a readily available quantitative measure of dependency strength.

In this paper we deal with discrete-valued fields. We propose a direct and quantitative measure of how much the knowledge of field X helps to predict the value of field Y. This measure does not depend on data distribution assumptions, and measures dependency in each direction separately. The measure, called a *probabilistic dependency*, or *pdep(X, Y)*, is a natural [1] generalization of functional dependency. A normalized version of $pdep(X, Y)$ is equivalent to Goodman and Kruskal (1954) measure of association τ (tau). The *pdep* measure can be efficiently computed, since it takes no more time than sorting value pairs of X and Y.

We analyze the behaviour of *pdep(X, Y)* and τ under randomization and prove surprisingly simple formulas for their expected values. In particular, if N is the number of records and d_X is the number of distinct values

[1] So natural, that it was rediscovered several times. A measure similar to *pdep* was proposed by Russell (1986) under the name of *partial determination*. We proposed *pdep* measure in (Piatetsky-Shapiro and Matheus, 1991). Independently, desJardins (1992, p. 70) proposed a measure called *uniformity* (as a variation of Russell's measure), which turns out to be identical to *pdep* measure.

of X, then $E[\tau(X,Y)] = (d_X - 1)/(N-1)$. This formula has significant implications for work on automatic derivation of dependency networks in data, since it measures the bias in favor of fields with more distinct values. It also has potential applications for the analysis of decision tree accuracy and pruning measures.

Note that the dependence of Y on X does not necessarily indicate *causality*. For example, we found a data dependency **discharge diagnosis** → **admission diagnosis**, even though the causal dependency is in the other direction. However, the dependency information in combination with domain knowledge of time (or other) order between variables helps in understanding domain structure (e.g. the discharge diagnosis is a refinement of the admission diagnosis).

Measures like χ^2 can test for independence between the discrete field values and provide a significance level for the independence hypothesis. The *pdep* measure does two things that χ^2 does not: 1) it indicates the direction of the dependence, and 2) it directly measures how much the knowledge of one field helps in predicting the other field. Our experiments indicate that χ^2 significance levels are similar to *pdep* significance levels obtained using the randomization approach. Thus, it is possible to use χ^2 to determine the presence of dependence and use *pdep* to determine the direction and strength of dependence.

This paper is limited to measures of dependency between *nominal* (discrete and unordered) values, such as marital status or insurance type. Various regression techniques exist for dealing with dependency between continuous field values, such as height or weight. An intermediate case is that of ordinal fields, which are discrete yet ordered (e.g. number of children or level of education). Statistical measures of association between ordinal values include γ, proposed by Goodman and Kruskal (1954, 1979) and Kendall's tau (Agresti 1984). Those measures, however, are symmetric and cannot be used for determining the direction of dependency.

2 A Probabilistic Dependency Measure

In the rest of this paper we will assume that data is represented as a table with N rows and two fields, X and Y (there may be many other fields, but we examine two fields at a time).

We want to define a dependency measure $dep(X,Y)$ with several desirable features. We want $dep(X,Y)$ to be in the interval [0, 1]. If there is a functional dependency between X and Y, $dep(X,Y)$ should be 1. If the dependency is less than functional, e.g. some amount of random noise is added to X or Y, we want $dep(X,Y)$ to decrease as the amount of noise increases. When X, Y are two independent random variables, we want $dep(X,Y)$ to be close to zero. Finally, to measure the direction of dependency $dep(X,Y)$ should be asymmetric and not always equal to $dep(Y,X)$. With these criteria in mind, we define the dependency measure as follows.

> Definition 1. Given two rows R_1, R_2, randomly selected from data table, the *probabilistic dependency* from X to Y, denoted $pdep(X,Y)$ is the conditional probability that $R_1.Y = R_2.Y$, given that $R_1.X = R_2.X$. Formally,
>
> $$pdep(X,Y) = p(R_1.Y = R_2.Y \mid R_1.X = R_2.X)$$

We note that $pdep(X,Y)$ approaches (and becomes equal to) 1 when the dependency approaches (and becomes) a functional one.

To derive the formula for $pdep(X,Y)$, we first examine the *self-dependency measure*[2] $pdep(Y)$, which is the probability that Y values will be equal in two randomly selected rows. Without loss of generality, let Y take values $1,\ldots,M$, with frequencies $y_1,\ldots,y_M$. The probability that both randomly selected rows will

[2] denoted $pdep1(Y)$ in Piatetsky-Shapiro (1992)

have $Y = j$ is $p(Y = j) \times p(Y = j) = y_j^2/N^2$, and $pdep(Y)$ is the sum of these probabilities over all j, i.e.

$$pdep(Y) = \sum_{j=1}^{m} p(Y = j)^2 = \sum_{j=1}^{m} \frac{y_j^2}{N^2} \tag{1}$$

Let X take values $1, \ldots, K$ with frequencies $x_1, \ldots, x_K$, and let n_{ij} be the number of records with $X = i$, $Y = j$. Assume that the first row R_1 had $X = i$. Selection of the second row R_2 is limited to the subset of size x_i where $X = i$. The probability that two rows randomly chosen from that subset will have the same Y value is equal to $pdep(Y|X = i)$ for that subset, which is

$$pdep(Y|X = i) = \sum_{j=1}^{M} p(Y = j|X = i)^2 = \sum_{j=1}^{M} \frac{n_{ij}^2}{x_i^2} \tag{2}$$

Since the probability of choosing row R_1 with value $X = i$ is x_i/N, we can compute $pdep(X, Y)$ as the weighted sum of $pdep(Y|X = i)$, i.e.

$$pdep(X, Y) = \sum_{i=1}^{K} p(X = i) pdep(Y|X = i) = \sum_{i=1}^{K} \frac{x_i}{N} \sum_{j=1}^{M} \frac{n_{ij}^2}{x_i^2} = \frac{1}{N} \sum_{i=1}^{K} \sum_{j=1}^{M} \frac{n_{ij}^2}{x_i} \tag{3}$$

The following table 1 shows a sample data set and a corresponding frequency table.

Table 1: An example dataset

```
        X  Y                      Y = 1   2 | x_i
Data    ------      Frequency     --+-------+--
File     1  1      Table      X= 1| 2   0 | 2
         1  1                    2| 1   1 | 2
         2  1                    3| 0   1 | 1
         2  2                   --+-------+--
         3  2                  y_j| 3   2 | 5
```

Here we have $pdep(Y) = 9/25+4/25 = 0.52$, and $pdep(X, Y) = 0.8$, while $pdep(X) = 4/25+4/25+1/25 = 0.36$, and $pdep(Y, X) = 0.533$.

By itself, however, the measure $pdep(X, Y)$ is insufficient. If, for example, almost all values of Y are the same, then any field will be a good predictor of Y. Thus we need to compare $pdep(X, Y)$ with $pdep(Y)$ to determine the relative significance of $pdep(X, Y)$. The relationship between these measures is given in theorem 1.

Theorem 1.

$$pdep(X, Y) - pdep(Y) = \frac{1}{N^2} \sum_{h=1}^{K-1} \sum_{i=h+1}^{K} \sum_{j=1}^{M} \frac{(x_h n_{ij} - x_i n_{hj})^2}{x_h x_i} \tag{4}$$

Proof: By rearranging sums (details omitted for lack of space).

This theorem implies

Corollary 1:

$$pdep(X,Y) \geq pdep(Y)$$

It also implies that $pdep(X,Y) = pdep(Y)$ only in a rare case when $n_{ij}/n_{hj} = x_i/x_h$ for all h, i, j, which implies that $pdep(Y|X = i) = pdep(X,Y)$ for all i.

To account for the relationship between $pdep(X,Y)$ and $pdep(Y)$ we normalize *pdep* using a standard statistical technique called *proportional reduction in variation* (Agresti 1990). The resulting measure is the Goodman and Kruskal τ (tau) measure of association:

$$\tau(X,Y) = \frac{pdep(X,Y) - pdep(Y)}{1 - pdep(Y)} \tag{5}$$

The τ measure is always between 0 and 1. If $\tau(A,B) > \tau(B,A)$, we infer that $A \rightarrow B$ dependency is stronger, and vice versa. For data in Table 1, $\tau(Y,X) = 0.271$, while $\tau(X,Y) = 0.583$. We conclude that the dependency $X \rightarrow Y$ is stronger than $Y \rightarrow X$.

We can understand these measures in the following way. Suppose we are given an item drawn from the same distribution as the data file, and we need to guess its Y. One strategy is to make guesses randomly according to the marginal distribution of Y, i.e. guess value $Y = j$ with probability y_j. Then the probability for correct guess is $pdep(Y)$. If we also know that item has $X = a$, we can improve our guess using conditional probabilities of Y, given that $X = a$. Then our probability for success, averaged over all values of X, is $pdep(X,Y)$, and $\tau(X,Y)$ is the relative increase in our probability of successfully guessing Y, given X.

A difficulty with *pdep* and τ is determining how significant are their values. In our experience with analysis of dependencies for fields in customer and insurance databases, τ values are rarely above 0.05, even for strong dependencies that are extremely significant as measured by χ^2. This reflects the diffuse phenomena under study: the target field is not completely determined by any single predictor. However, knowledge of weak predictors is important, since a combination of several weak predictors may give a strong predictor. When the studied population is large, and the target field is important, even weak improvements in predictive abilities are very valuable.

We have used the randomization testing approach (Jensen 1991) to analyze the significance of *pdep* values. Consider a file with fields X and Y, and let $pdep(X,Y) = p_0$. The idea of randomization is to randomly permute Y values while keeping X values in place. We can estimate the probability of $pdep(X,Y) \geq p_0$ as the percentage of permutations where $pdep(X,Y) \geq p_0$, assuming that all permutations of Y values are equally likely.

2.1 Expected value of *pdep*

We have analyzed the expected value of *pdep* measure under randomization and derived the following formula:

Theorem 2 (Piatetsky-Rotem-Shapiro)
Given N records of fields X and Y, where X has $d_X = K$ distinct values,

$$E[pdep(X,Y)] = pdep(Y) + \frac{K-1}{N-1}(1 - pdep(Y)) \tag{6}$$

The proof of this theorem is in the appendix.

The interesting implication of this formula is that for a fixed distribution of Y values, $E[pdep(X,Y)]$ depends only on the number of distinct X values and not on their relative frequency.

Theorem 2 gives us the expected value of τ under randomization:

Corollary 2.1

$$E[\tau(X,Y)] = \frac{E[pdep(X,Y)] - pdep(Y)}{1 - pdep(Y)} = \frac{K-1}{N-1} \tag{7}$$

So, if $d_X > d_Z$, then for any field Y, X is expected to be a better predictor of Y than Z. Tau values that are higher than the expected value indicate additional relationship between the fields. This formula is especially important when the number of distinct field values is close to the number of records.

We can further refine our analysis of dependency by observing that $pdep(X,Y)$ (and τ) will indicate a significant dependency for large values of d_X, even for a random permutation of all Y values (which destroys any intrinsic relationship between X and Y). We can compensate for this effect by introducing a new measure μ, which normalizes $pdep(X,Y)$ with respect to $E[pdep(X,Y)]$ instead of $pdep(Y)$:

$$\mu(X,Y) = \frac{pdep(X,Y) - E[pdep(X,Y)]}{1 - E[pdep(X,Y)]} = 1 - \frac{1 - pdep(X,Y)}{1 - pdep(Y)} \frac{N-1}{N-K} \tag{8}$$

Since $E[pdep(X,Y)] \geq pdep(Y)$, we have $\mu(X,Y) \leq \tau(X,Y)$. When N/K increases, $E[pdep(X,Y)]$ asymptotically decreases to $pdep(Y)$, and $\mu(X,Y)$ asymptotically increases to $\tau(X,Y)$. The additional advantage of μ is that it increases, like χ^2, if the data set size is doubled, whereas *pdep* and tau do not change.

We used the randomization approach to compute the exact significance values for $pdep(X,Y)$ (see Appendix). However, for large datasets the exact computation is too computationally expensive. Instead, we use χ^2 statistic for measuring the significance of dependency. Our experiments indicate that randomization and χ^2 significance levels are quite close.

Finally, we note that the two-field dependency analysis of Y on X can be straightforwardly generalized into a multi-field dependency analysis of Y on several fields, $X_1, X_2, \ldots$ by replacing conditional probabilities $p(Y = b|X = a)$ with $p(Y = b|X_1 = 1, X_2 = 2, \ldots)$ (see also Goodman and Kruskal, 1979).

3 Binary Fields

An important and frequent special case is that of binary fields. Assuming that there is a significant dependency between X and Y, as measured by standard tests for 2x2 tables, we want to know the direction of the dependency. When both X and Y have only two values, then we cannot use τ or μ measures to determine the direction of dependency, since $\tau(X,Y) = \tau(Y,X)$.

For binary fields, the value (w.l.g. denoted as "t" for *true*) which indicates the presence of a desirable feature is generally more useful than the other value (denoted as "f" for *false*). So we need to compare the rules $(X = t) \rightarrow (Y = t)$ versus $(Y = t) \rightarrow (X = t)$. Let ff, ft, tf, tt, denote the counts of X=f, Y=f; X=f, Y=t; X=t, Y=f; and X=t, Y=t, respectively.

Then $(X = t) \rightarrow (Y = t)$ is true for $\frac{|X=t \,\&\, Y=t|}{|X=t|} = \frac{tt}{tt+tf}$ of cases, while $(Y = t) \rightarrow (X = t)$ is true for $\frac{tt}{tt+ft}$ of cases. Hence the rule $(X = t) \rightarrow (Y = t)$ is stronger if $tf < ft$, and the inverse rule is stronger if $tf > ft$.

The following table shows the statistics for a sample of customer data for fields CHILD-0-5, CHILD-0-17, and FEMALE-35-44, which have a value "Y" if the household has such a person, and "N" otherwise.

In the first case, the rule Child 0-17 $\rightarrow$ Child 0-5 is correct $\frac{238}{238+524} = 31\%$ of the time, while the rule Child 0-5 $\rightarrow$ Child 0-17 is correct 100%, so the second rule is obviously better. In the second case, rule Child 0-17 $\rightarrow$ Female 35-44 has correctness $\frac{327}{327+435} = 43\%$, while the inverse rule has correctness $\frac{327}{327+560} = 37\%$. Thus, the first rule is preferred.

		Child 0-5			Female 35-44	
		N	Y		N	Y
Child 0-17	N	5363	0	N	4803	560
	Y	524	238	Y	435	327

4 Using SQL interface to compute Dependencies

Some databases are so large that they cannot be analyzed in memory. Such databases can be analyzed by using a DBMS query interface to retrieve data and to perform at least some of the computation. We limit our discussion to SQL interfaces, since SQL, with all its limitations, is the de-facto standard query language. Using SQL for data access makes analysis tools more portable to other DBMS.

Here we examine the use of SQL queries on a file *File1* to perform the dependency analysis. In the simplest case of both X and Y being discrete, we can extract the necessary statistics for computing *pdep* with the query

```
select X, Y, count(*) from File1 group by X, Y
```

A more interesting case arises when we need to discretize numeric data fields. Consider a field Y which contains toll-call revenue. While Y has many distinct values, we are interested in whether $Y = 0$ (no usage); $0 < Y \leq 3$ (low usage); $3 < Y \leq 10$ (medium usage); or $Y > 10$ (high usage). Such discretization may need to be done frequently, and for different sets of ranges.

To get the necessary statistics for computing dependency of discretized Y upon some X we have to issue four standard SQL queries:[3]

```
select X, count(*) from File1 where Y = 0 group by X;
select X, count(*) from File1 where Y > 0 and Y <= 3 group by X;
select X, count(*) from File1 where Y > 3 and Y <= 10 group by X;
select X, count(*) from File1 where Y > 10 group by X;
```

If X is discretized into 5 buckets and Y is discretized into 3, then $5 \times 3 = 15$ queries would be needed to compute the dependency. Fortunately, several popular DBMS, such as Oracle^{tm} or Raima^{tm}, have SQL extensions that allow us to compute the necessary statistics with just one query.

4.1 Dynamic Discretization using Raima

Dynamic discretization is relatively straightforward using Raima, which has a conditional column function **if(condition1, expr1, expr2)**, meaning if **condition1** is true, then **expr1** else **expr2**. Expressions **expr1, expr2** may in turn contain conditional columns. The necessary code for the above example is:

```
select X, if(Y=0, 0, if(Y<=3, 2, if(Y<=10,5,10))), count(*) from File1
group by 1, 2;
```

In general, the code for discretization is specified as one long conditional column in a straightforward manner.

[3]The last query can be avoided if we have previously computed `select X, count(*) from File1 group by X;`

4.2 Dynamic Discretization using Oracle

The discretization is more complex with Oracle SQL which does not have the conditional column feature. Instead, we will use the functions LEAST and GREATEST to achieve the same effect. We will also rely on the fact that most numerical fields typically have a limited number of digits after the period. For our databases, the only non-integer values are the dollar amounts which 2 digits after the period. That means that if $Y > 0$, then $100 * Y \geq 1$.

We can formulate the discretization problem as follows. Given a field Y, a set of non-overlapping ranges $range_i = [a_i\ lop\ Y\ lop\ b_i]$, where lop is $<$ or $\leq$, and a value v_i for each range, we want to specify a function DIS(Y) that will return v_i if Y is in $range_i$, and zero otherwise. Then, the statistics for dependency between X and discretized Y can be computed by the query

```
select X, DIS(Y), count(*) from File1
group by X, DIS(Y)
```

The case where X is also discretized is handled by replacing X in the above query with DIS(X).

Consider the expression LEAST(1, 100*GREATEST(Y, 0)). It is 1 if $Y > 0$ and zero if $Y \leq 0$. Below we show how to specify similar indicator functions *ind(range)* for each comparison Y op a (we include $Y = a$ as a simplification of range $a \leq Y \leq a$). The M constant below is 10^d, where d is the number of Y digits after the period.

- ind($Y = a$) is `1 - LEAST(1, ABS(`$M * Y$`))`
- ind($Y > a$) is `LEAST(1, `M`*GREATEST(0,`$Y - a$`))`
- ind($Y \geq a$) is `LEAST(1, `M`*GREATEST(0,`$Y - a + 1/M$`))`
- ind($Y < a$) is `LEAST(1, `M`*GREATEST(0,`$a - Y$`))`
- ind($Y \leq a$) is `LEAST(1, `M`*GREATEST(0,`$a - Y + 1/M$`))`

If a range contains two comparisons, e.g. $a < Y \leq b$, then $\text{ind}(a < Y \leq b) = \text{ind}(a < Y) \times \text{ind}(Y \leq b)$. Finally, we get the dynamic discretization function by

$$DIS(Y) = \sum_i v_i \times ind(range_i) \tag{9}$$

If one of the v_i is zero, as is the case for the first range in our example, then the corresponding term may be omitted. The complete Oracle SQL query to discretize Y for the above example is

```
select X,     2*LEAST(1, 100*GREATEST(0,Y-0))*LEAST(1, 100*GREATEST(0,3-Y)) +
              5*LEAST(1, 100*GREATEST(0,Y-3+0.01))*LEAST(1, 100*GREATEST(0,10-Y)) +
              10*LEAST(1, 100*GREATEST(0,Y-10+0.01)) Y, count(*)
from File1
group by X,   2*LEAST(1, 100*GREATEST(0,Y-0))*LEAST(1, 100*GREATEST(0,3-Y)) +
              5*LEAST(1, 100*GREATEST(0,Y-3+0.01))*LEAST(1, 100*GREATEST(0,10-Y)) +
              10*LEAST(1, 100*GREATEST(0,Y-10+0.01))
```

5 Summary

Knowing data dependencies is important for understanding the domain and has many applications in analyzing, refining, and presenting information about the domain.

In this paper, we have discussed ways to measure the strength of approximate, or probabilistic, dependencies between nominal values and showed how to determine the significance of a dependency value using the randomization approach. We proved formulas for the expected values of *pdep* and Goodman-Kruskal τ under data randomization.

We also described how to efficiently use SQL interface for analyzing dependencies in large databases.

Acknowledgments. We thank Philip Chan, Greg Cooper, Marie desJardins, Greg Duncan, and Gail Gill for their insightful comments and useful suggestions. We are grateful to Shri Goyal for his encouragement and support.

6 References

Agresti, A. (1984). *Analysis of Ordinal Categorical Data.* New York: John Wiley.

Agresti, A. (1990). *Categorical Data Analysis*, New York: John Wiley.

Almoaullim, H., and Dietterich, T. (1991). Learning with Many Irrelevant Features, In Proceedings of AAAI-91, 547–552.

Cooper, G., and Herskovits, E. (1991). A Bayesian Method for the Induction of Probabilistic Networks from Data. Stanford Knowledge Systems Laboratory Report KSL-91-02.

desJardins, M. E. (1992). PAGODA: A Model for Autonomous Learning in Probabilistic Domains, Ph.D. Thesis, Dept. of Computer Science, University of Berkeley.

Geiger, D., Paz, A., and Pearl, J. (1990). Learning Causal Trees from Dependence Information. *Proceedings of AAAI-90*, 770–776, AAAI Press.

Glymour, C., Scheines, R., Spirtes, P., and Kelly, K. (1987). *Discovering Causal Structure.* Orlando, Fla.: Academic Press.

Goodman, L. A., and Kruskal, W. H. (1954). Measures of Association for Cross Classification, *J. of American Statistical Association*, 49, 732–764.

Goodman, L. A., and Kruskal, W. H. (1979). *Measures of Association for Cross Classifications.* Springer-Verlag, New York.

Jensen, D. (1991). Knowledge Discovery Through Induction with Randomization Testing, in G. Piatetsky-Shapiro, ed., *Proceedings of AAAI-91 Knowledge Discovery in Databases Workshop*, 148–159, Anaheim, CA.

Mannila, H. and Raiha, K.-J. (1987). Dependency inference. *Proceedings of the Thirteenth International Conference on Very Large Data Bases (VLDB'87)*, 155–158.

Pearl, J. (1992). Probabilistic Reasoning in Intelligent Systems: Networks of plausible inference, 2nd edition. San Mateo, Calif.: Morgan Kaufmann.

Piatetsky-Shapiro, G. and Matheus, C. (1991). Knowledge Discovery Workbench: An Exploratory Environment for Discovery in Business Databases, *Proceedings of AAAI-91 Knowledge Discovery in Databases Workshop*, 11–24, Anaheim, CA.

Piatetsky-Shapiro, G. (1992). Probabilistic Data Dependencies, in J. Zytkow, ed., Proceedings of Machine Discovery Workshop, Aberdeen, Scotland.

Piatetsky-Shapiro, G. and Matheus, C. (1992). Knowledge Discovery Workbench for Discovery in Business Databases, In special issue of *Int. J. of Intelligent Systems* on Knowledge Discovery in Databases and KnowledgeBases, 7(7) 675–686.

Russell, S. J. (1986). Analogical and Inductive Reasoning, Ph. D. Thesis, Dept. of Computer Science, Stanford University, chapter 4.

Siegel, M. (1988). Automatic rule derivation for semantic query optimization, *Proceedings Expert Database Systems Conference*, 371–385.

Spirtes, P., Glymour, C., and Scheines, R. (1993). *Causality, Prediction, and Search.* New York: Springer-Verlag.

A Analysis of the PDEP measure under Randomization

We use the randomization testing approach (Jensen 1991) for analyzing the *pdep* measure. First, we prove a formula for the expected value of pdep under randomization, and then we discuss ways of measuring the significance of *pdep* value.

Consider a file with only two fields X and Y, and let $pdep(X,Y) = p_0$. The idea of randomization is to randomly permute Y values while keeping X values in place. We can estimate the probability of $pdep(X,Y) \geq p_0$ as the percentage of permutations where $pdep(X,Y) \geq p_0$, assuming that all permutations of Y values are equally likely.

A.1 Expected Value of PDEP under Randomization

Without loss of generality we can assume that X takes values from 1 to K, and Y takes values from 1 to M. Let N be the number of records. Then we have the following theorem, proposed by Gregory Piatetsky-Shapiro and proved by Doron Rotem.

Theorem 2 (Piatetsky-Rotem-Shapiro): The expected value of $pdep(X,Y)$ under randomization is

$$E[pdep(X,Y)] = pdep(Y)\frac{N-K}{N-1} + \frac{K-1}{N-1} = pdep(Y) + \frac{K-1}{N-1}(1 - pdep(Y)) \qquad (10)$$

Proof: For convenience, we reformulate this problem in terms of balls and cells. We will call a set of tuples with equal y-values a *y-set*. We have M y-sets, each consisting of y_j balls. Partitioning of the record according to K distinct X values is equivalent to partitioning the N cells into K compartments, each of x_i cells. From the definition of *pdep* it is clear that

$$E[pdep(X,Y)] = \sum_{i=1}^{K}\sum_{j=1}^{My} E[pdep(X=i, Y=j)] \qquad (11)$$

First, we will compute $E[pdep(X=i, Y=j)]$ by looking at the problem of placing y_j balls among N cells, so that r balls fall into a compartment of size x_i, contributing $r^2/(Nx_i)$ to *pdep*.

Let $\binom{n}{i} = \frac{n!}{(n-i)!i!}$ denote the number of ways to select i items from n.

For a given y-set with y_j balls, the probability that r balls fall into a compartment of size x_i is

$$\frac{\binom{y_j}{r}\binom{N-y_j}{x_i-r}}{\binom{N}{x_i}} = \frac{\binom{x_i}{r}\binom{N-x_i}{y_j-r}}{\binom{N}{y_j}} \tag{12}$$

Then $E[pdep(X=i,Y=j)]$ is the sum of the above equation from $r=1$ to $r=x_i$ ($r=0$ contributes zero to *pdep*), which is

$$E[pdep(X=i,Y=j)] = \frac{1}{N\binom{N}{y_j}}\sum_{r=1}^{x_i}\binom{x_i}{r}\binom{N-x_i}{y_j-r}\frac{r^2}{x_j} = \frac{1}{N\binom{N}{y_j}}\sum_{r=1}^{x_i}\binom{x_i-1}{r-1}\binom{N-x_i}{y_j-r}r \tag{13}$$

We reduce this sum to a closed form using the following combinatorial lemmas.

Lemma 2.1:

$$\sum_{i=0}^{a}\binom{a}{i}\binom{n-a}{b-i} = \binom{n}{b} \tag{14}$$

Proof: Consider the task of placing b balls into n cells. Assume that the first a cells are marked as special. The number of placements of b balls into n cells, so that exactly i of them will fall into special cells is $\binom{a}{i}\binom{n-a}{b-i}$. Summing this for i from 0 to a gives the sum on the left. This sum is also equal to the total number of placements of b balls into n cells (regardless of the partition), which is $\binom{n}{b}$. Q.E.D.

Corollary 2.2

$$\sum_{i=0}^{a} i\binom{a}{i}\binom{n-a}{b-i} = a\binom{n-1}{b-1} \tag{15}$$

Proof: By reducing to Lemma 2.1.

Corollary 2.3

$$\sum_{i=0}^{a}(i+1)\binom{a}{i}\binom{n-a}{b-i} = \binom{n}{b} + a\binom{n-1}{b-1} \tag{16}$$

Proof: By summing the previous two equations.

We can now reexamine the sum in earlier equation 13 after replacing r with $r' = r-1$,

$$\sum_{r=1}^{x_i} r\binom{x_i-1}{r-1}\binom{N-x_i}{y_j-r} = \sum_{r'=0}^{x_i-1}(r'+1)\binom{x_i-1}{r'}\binom{(N-1)-(x_i-1)}{(y_j-1)-r'} \tag{17}$$

We see that Corollary 2.3 applies, with $a = x_i-1$, $b = y_j-1$, and $n = N-1$. Hence $E[pdep(X=i,Y=j)]$ (from equation 13) is equal to

$$\frac{1}{N\binom{N}{y_j}}\left[\binom{N-1}{y_j-1} + (x_i-1)\binom{N-2}{y_j-2}\right] = \frac{y_j}{N^2}\left(1+\frac{(x_i-1)(y_j-1)}{N-1}\right) \tag{18}$$

Finally, to compute $E[pdep(X,Y=j)]$ we sum the above over all x_i, obtaining (since $\sum x_i = N$)

$$E[pdep(X,Y=j)] = \sum_{i=1}^{K}\frac{y_j}{N^2}\left(1+\frac{(x_i-1)(y_j-1)}{N-1}\right) = \frac{y_j}{N^2}\left(K+\frac{(N-K)(y_j-1)}{N-1}\right)$$

$$= \frac{y_j^2}{N^2}\frac{N-K}{N-1} + y_j\frac{K-1}{N(N-1)} \tag{19}$$

As we can see there is no dependence of this expression on x_i, but only on N, y_j, and K. Finally, the value of $E[pdep(X,Y)]$ is obtained by summing the above equation over all y-sets:

$$E[pdep(X,Y)] = \sum_{j=1}^{M}\left(\frac{y_j^2}{N^2}\frac{N-K}{N-1} + y_i\frac{K-1}{N(N-1)}\right) \tag{20}$$

Since $\sum_{j=1}^{M} y_j = N$ and $\sum_{j=1}^{M} y_j^2/N^2 = pdep(Y)$, the above simplifies to

$$E[pdep(X,Y)] = pdep(Y)\frac{N-K}{N-1} + \frac{K-1}{N-1} = pdep(Y) + \frac{K-1}{N-1}(1-pdep(Y)) \tag{21}$$

End of Proof.

A.2 Determining significance of PDEP

Table 2: Sample Data partitioned by different X values.

	X	Y	
	1	1	pdep(X,Y) = 0.800
Data	1	1	
File			
	2	1	pdep(Y) = 0.52
	2	2	N = 5, K = 3
	3	2	

For our sample data, we have dependency $X \rightarrow Y$ with $pdep(X,Y) = 0.800$. How significant is this? Let us consider permutations of record numbers and their Y values, while keeping X values in the same place. Each permutation is partitioned into three parts: records with $X = 1$, $X = 2$, and $X = 3$. Let us denote a particular permutation by listing its Y values, with a vertical bar to denote the partition. The above table would be denoted as **[1,1 | 1,2 | 2]**, where the first section **1,1** represents Y values for $X = 1$, the second part **1,2** is Y values for $X = 2$, and the last part **2** is Y value for $X = 3$.

For the purpose of measuring *pdep*, the order of Y values within a partition is irrelevant. Consider a permutation that leads to a partition **[v1,v2 | v3,v4 | v5]**. Any permutation of values in the first (or second) part will lead to a partition with the same *pdep*. Partitions are differentiated by the count of values with $Y = 1$ and $Y = 2$ in each part, called the partition *signature*. Since the number of values in each part is 2, 2, and 1, respectively, each signature will appear at least 2!2!1! = 4 times among the 5!=120 possible permutations.

Consider partition **[1,1 | 1,2 | 2]**. There are $\binom{3}{2}$ =3 ways to choose 2 records with $Y = 1$ in the first part, $\binom{2}{1}\binom{1}{1}$ = 2 ways to choose records with $Y = 1$ and $Y = 2$ in the second part, and one way to choose the remaining record in the third part. Thus, there are 3x2=6 choices that lead to signature **[1,1 | 1,2 | 2]**. For comparison, signature **[2,2 | 1,1 | 2]** can be chosen in 3 ways.

Regardless of the signature, each choice can be permuted in 2!2!1! = 4 ways to get a pdep-equivalent partition. These results are summarized in tables 3A and 3B.

Table 3A. Pdep and prob. of all signatures

-y-signature---	choices	probability	pdep
[1,1 | 2,2 | 1]	3	3x4/120 = 0.1	1.0
[2,2 | 1,1 | 1]	3	3x4/120 = 0.1	1.0
[1,1 | 1,2 | 2]	6	6x4/120 = 0.2	0.8
[1,2 | 1,1 | 2]	6	6x4/120 = 0.2	0.8
[1,2 | 1,2 | 1]	12	12x4/120 = 0.4	0.6

Table 3B. Pdep probability summary

pdep	probability
1	0.2
0.8	0.4
0.6	0.4

Here the probability of $pdep(X,Y) \geq 0.8$ for a random permutation of Y values is $p(pdep = 1) + p(pdep = 0.8)$ = 0.2+0.4 = 0.6.

We can also use table 3B to check Theorem 2. From table we get $E[pdep(X,Y)] = 1 \times 0.2 + 0.8 \times 0.4 + 0.6 \times 0.4 = 0.76$. For this data $pdep(Y) = 0.52$, $K = 3$, and $N = 5$, and Theorem 2 gives $E[pdep(X,Y)] = 0.52 + \frac{3-1}{5-1}(1 - 0.52) = 0.76$, same answer!

Let us consider a general case where X has values $1, 2, \ldots K$, and Y has values $1, 2, \ldots M$. Let x_i be the count of $X = i$, y_j be the count of $Y = j$, n_{ij} be the count of $X = i$, $Y = j$, and N be the total number of records. Let $C(N, n_1, n_2, \ldots, n_k) = N!/(n_1!n_2! \ldots n_k!)$ be the number of ways to distribute (without remainder) N items into K bins, so that bin i will have n_i items.

The probability of a particular combination is obtained by the following reasoning. Let us group together records with the same value of X. This will produce K bins, one bin for each value. First bin has n_{11} records with $Y = 1$, second has n_{12}, etc. There are $C(y_1, n_{11}, n_{21}, \ldots, n_{K1})$ ways to put y_1 records with $Y = 1$ into those bins. Similarly, there are $C(y_2, n_{12}, n_{22}, \ldots, n_{K2})$ ways to place y_2 records with $Y = 2$. After all records have been placed, records in each bin can be independently permuted. This will produce $x_1!x_2! \ldots x_K!$ permutations. The product of all those factors should be divided by $N!$, the total number of permutations. Hence the total probability of a particular configuration $[n_{ij} \ldots]$ is

$$prob([n_{ij} \ldots]) = \frac{\prod_{j=1}^{M} C(y_j, n_{1j}, \ldots, n_{Kj}) \prod_{i=1}^{K} x_i!}{N!} = \frac{\prod_{j=1}^{M} C(y_j, n_{1j}, \ldots, n_{Kj})}{C(N, x_1, \ldots, x_K)} \tag{22}$$

It is possible to enumerate all the partition signatures and construct a complete distribution of *pdep* values and their probabilities. However, the number of different signatures grows very fast and it is impractical to enumerate them for a large number of partitions. Instead, we can use χ^2 statistic to measure the significance of the dependency.

To compare χ^2 and *pdep* significance levels, we computed them for datasets obtained by repeating N times the data in Table 1. The following table summarizes the results, which indicate that *pdep* and χ^2 are quite close.

Table 5. Sig(*pdep*) obtained by randomization vs significance of χ^2.

Significance	N=2	3	4	6	8	16
sig(*pdep*)	0.934	0.984	0.9973	0.99995	$1 - 1.5 \times 10^{-6}$	$1 - 7.8 \times 10^{-13}$
sig(χ^2)	0.946	0.987	0.9971	0.99985	$1 - 8.6 \times 10^{-6}$	$1 - 7.4 \times 10^{-11}$

Note that since *pdep* and τ values do not change, when the data set is doubled (which increases significance), *pdep* or τ values cannot be used by themselves to measure the significance of dependency and should be used only together with N.