Chapter 1

Multi-Objective Evolutionary Optimization in Statistical Disclosure Control

Rinku Dewri, Indrajit Ray, Indrakshi Ray and Darrell Whitley

Colorado State University, Department of Computer Science,
Fort Collins, CO 80523-1873, USA
{*rinku,indrajit,iray,whitley*}*@cs.colostate.edu*

Statistical disclosure control involves the sanitization of personally identifying information in data sets prior to their dissemination. Such a process typically results in a loss in utility of the data. A data publisher is thus confronted with two requirements to fulfill – ensuring the safety of respondents and maintaining statistical utility in the anonymized data. Existing approaches model the first requirement as a constraint in an optimization framework directed towards maximizing data utility. An immediate consequence of this method is the requirement for exhaustive analysis if the much desired trade-off behavior between privacy levels and data utility has to be explored. In this chapter, we explore an alternative framework based on multi-objective optimization to cater to the data publisher's requirements. We show how the requirement of specifying parameter values of an anonymity model can be eliminated by formulating privacy as an explicit objective to maximize. Further, we discuss the application of evolutionary algorithms as a solution method with a focus on solution representation and operator requirements. We also present empirical results to demonstrate that the approach is a more practical optimization framework for a data publisher.

Contents

1.1. Introduction

Various scientific studies, business processes and legal procedures depend on quality data from large data sources. However, such data sources often contain sensitive personal information, dissemination of which is governed by various privacy requirements. Data in such cases need to be sanitized off personally identifying attributes before it can be shared. Anonymizing data is challenging because re-identifying the values in sanitized attributes is not impossible when other publicly available information or an adversary's background knowledge can be linked with the shared data.

Anonymization of data sets (tables) involve transforming the actual data into a form unrecognizable in terms of the exact values by using *generalization* and *suppression* techniques.[1] Generalization of data is performed by grouping together specific data attribute (*quasi-identifiers*) values into a more general one. An example of this is replacing a specific age by an age range. Quasi-identifier attributes typically correspond to publicly available information such as phone numbers, postal code, age, etc. Data suppression, on the other hand, removes entire tuples making them no longer existent in the data set. Performance of such data modification is gauged by their ability to satisfy one or more privacy constraint, typically enforced by models such as k–anonymity,[2,3] ℓ–diversity[4] and t–closeness.[5] These privacy models prevent the accurate re-identification of an individual or sensitive information pertaining to the individual.

An unavoidable consequence of performing anonymization is a loss in the quality of the data set. Researchers have therefore looked at different methods to obtain an anonymization that can satisfy the privacy constraint with minimal loss of information.

Several algorithms have been proposed to find effective k–anonymization. The μ-argus algorithm is based on the greedy generalization of infrequently occurring combinations of quasi-identifiers and suppresses outliers to meet the k–anonymity requirement.[6] Sweeney's Datafly approach used a heuristic method to generalize the attribute containing the most distinct sequence of values for a provided subset of quasi-identifiers.[2] Sequences occurring less than k times are suppressed. In the same work,

Sweeney proposed a theoretical algorithm that can exhaustively search all potential generalizations to find the one that minimally distorts the data during anonymization. Samarati proposed an algorithm[1] that identifies all generalizations satisfying k–anonymity. Choice of an optimal generalization can then be made based on certain preference information provided by the data recipient. Bayardo and Agrawal proposed a complete search method that iteratively constructs less generalized solutions starting from a completely generalized data set.[7]

A genetic algorithm based formulation to find optimal k–anonymous generalizations was first proposed by Iyengar.[8] Although the method can maintain a good solution quality, it has been criticized for being a slow iterative process. In this context, Lunacek et al. introduced a new crossover operator that can be used with a genetic algorithm for constrained attribute generalization and effectively showed that Iyengar's approach can be made faster.[9]

A major shortcoming in all these approaches is the single-objective treatment of privacy and utility. The focus has always been concentrated on maximizing the utility of an anonymization, given a privacy constraint specified by a pre-defined value(s) for the involved parameter(s). It is important to note that privacy in this framework receives no attention in terms of optimality. Instead, the requirement that a data publisher has strong understanding of the impact of choosing a parameter value in a privacy model is strictly enforced. In order to do so, a data publisher needs to answer questions of the following form.

- How does privacy/utility change with changes in utility/privacy?
- Given a particular choice of privacy and utility levels, is there a solution that can improve one aspect without affecting the other?
- Given a particular choice of privacy and utility levels, is there a solution that can improve one aspect with tolerable change in the other?

Answering these questions using existing techniques will require us to perform an exhaustive enumeration of parameter values in the privacy model to determine what is suitable. Nonetheless, it is imperative that the data publisher understands the implications. There is clearly a trade-off involved. Setting the parameter to a "very low" value impacts the privacy of individuals in the data set. Picking a "very high" value disrupts the inference of any significant statistical information from the anonymized data set. Such conflicting characteristics define the nature of a multi-objective

optimization problem.

In this chapter, we shall explore multi-objective formulations incorporating a data publisher's requirements to maximize data privacy levels, and at the same time, maintain the data utility at a maximum. The chapter will demonstrate how a series of multi-objective optimization problems can be formulated on a given anonymity and data utility model, depending on the requirements of the data publisher. Our approach is significantly different from the assumed norm in the sense that we no longer treat privacy as a constraint in the optimization framework. Rather, privacy is modeled explicitly as an objective to maximize along with the data utility. We shall show how a multi-objective evolutionary algorithm can be employed here to obtain a global picture of the mutual effects of privacy and utility.

The remainder of the chapter is organized as follows. A concise background on multi-objective optimization is presented in Section 1.2. Section 1.3 provides a preliminary background on statistical disclosure control. Section 1.4 provides a description of the multi-objective problems we can formulate on a privacy model. The specifics of the solution methodology as particular to solving the problems using an evolutionary algorithm is also given here. A brief discussion on the interpretation of results is presented in Section 1.5. Finally, Section 1.6 concludes the chapter.

1.2. Multi-objective Optimization

In real world scenarios, often a problem is formulated to cater to several criteria or design objectives and a decision choice to optimize these objectives is sought for. An optimum design problem must then be solved with multiple objectives and constraints taken into consideration. This type of decision making problems fall under the broad category of multi-objective or vector optimization problem. Multi-objective optimization differs from single-objective ones in the cardinality of the optimal set of solutions. Single-objective optimization techniques are used to find the global optima. There is no such concept of a single optimum solution in case of multi-objective optimization. This is due to the fact that a solution that optimizes one of the objectives may not have the desired effect on the others. As a result, it is not always possible to determine an optimum that corresponds in the same way to all the objectives under consideration. Decision making under such situations thus requires some domain expertise to choose from multiple trade-off solutions depending on the feasibility of implementation.

Formally we can state a multi-objective optimization problem (MOOP) in statistical disclosure control (SDC) as follows:

Definition 1.1 (SDC MOOP).
Let $f_1, \ldots, f_M$ denote M objective functions to maximize while performing a modification of a given table PT*. Find a generalized table* RT^* *of* PT *which optimizes the M-dimensional vector function*

$$f(\mathsf{RT}) = [f_1(\mathsf{RT}), f_2(\mathsf{RT}), \ldots, f_M(\mathsf{RT})]$$

where RT *is a generalized version of* PT*.*

The objective functions in this case are either related to the privacy or utility level maintained in an anonymized table. Note that the privacy level can be inferred with respect to different privacy models. Hence the number of objectives can be more than two. In order to find an optimal solution to the SDC MOOP, we must be able to compare anonymizations with respect to all the objectives in hand. However, due to the conflicting nature of the objective functions, a simple objective value comparison between two anonymizations cannot be performed. Most multi-objective algorithms thus use the concept of dominance to compare feasible solutions.

Definition 1.2 (Dominance and Pareto-optimal set). *Given a table* PT *and M objectives to maximize, a generalized table* RT_1 *of* PT *is said to dominate another generalized table* RT_2 *of* PT *if*

1. $\forall i \in \{1, 2, \ldots, M\}$ $\quad f_i(\mathsf{RT}_1) \geq f_i(\mathsf{RT}_2)$ *and*
2. $\exists j \in \{1, 2, \ldots, M\}$ $\quad f_j(\mathsf{RT}_1) > f_j(\mathsf{RT}_2)$

RT_2 *is then said to be dominated by* RT_1*, denoted by* $\mathsf{RT}_2 \preceq \mathsf{RT}_1$*. If the two conditions do not hold,* RT_1 *and* RT_2 *are said to be non-dominated w.r.t. each other, denoted by the* $\not\preceq$ *symbol. Further, all generalized tables of* PT *which are not dominated by any possible generalized version of* PT *constitutes the Pareto-optimal set.*

In other words, a Pareto-optimal solution is as good as another solution in the Pareto-optimal set and better than other feasible solutions outside the set. The surface generated by these solutions in the objective space is called the *Pareto-front* or *Pareto-surface*. In the context of disclosure control, the Pareto-front for the two objectives – maximize privacy (as given by a parameter, say k) and minimize loss – provides the decision maker

an understanding of the changes in the information loss when k is varied. Consider two anonymized versions RT_1 and RT_2 of a data set, with corresponding k and $loss$ as $(k_1, loss_1)$ and $(k_2, loss_2)$ respectively. Let us assume that $k_1 < k_2$ and $loss_1 = loss_2$. A decision maker using RT_1, and unaware of RT_2, misses on the fact that a higher k value is possible without incurring any increase in the loss. A multi-objective algorithm using the dominance concept can expose this relationship between RT_1 and RT_2, namely $\mathsf{RT}_1 \preceq \mathsf{RT}_2$. As another example, consider the case with $loss_2 - loss_1 = \epsilon > 0$. RT_1 and RT_2 are then non-dominated solutions, meaning that one objective cannot be improved without degrading the other. However, if ϵ is a relatively small quantity acceptable to the decision maker, RT_2 might be preferable over RT_1. Such trade-off characteristics are not visible to the decision maker until a multi-objective analysis is carried out. Thus, the objective of the analysis is to find the Pareto-optimal set from the set of all possible anonymized versions of a given data set.

1.3. Statistical Disclosure Control

Public distribution of personal data is a requirement to facilitate various scientific studies. Statistical organizations collecting information for such purposes often face standard security issues when distributing the data, thereby forcing them to enforce disclosure controls to protect the identity of individuals represented in the collected information. However, the sole purpose of collecting the information would be lost if the controls prohibit any kind of statistical inference being made from the distributed data. Statistical disclosure control thus involves mediating the risk of publicly disseminated information with the statistical utility of the content.

Most models in statistical disclosure control employ data recoding using generalization and suppression schemes in their attempt to hide the information content of a data set in its exact form. A generalization scheme performs a one-way mapping of the data values to a form unrecognizable from the original values or to a form that induces uncertainty in recognizing them. More than often it may not be possible to enforce a chosen level of privacy due to the presence of outliers in the data set. In such a situation, a suppression scheme gets rid of the outliers by removing them from the data set altogether.

The resultant data set from a generalization, coupled with or without a suppression scheme, affects the utility of the data. Statistical inferences suffer as more and more diverse data are recoded to the same value, or

records are deleted by a suppression scheme. One can argue that the privacy requirement ensures the non-inference of any individual information while the utility requirement enforces the inference of accurate aggregate information. A summary statistic relying on accurate individual information therefore deteriorates when stronger privacy is implemented. The harder problem is a quantification of this deterioration, in effect, the information lost in the process of data recoding.

1.3.1. *Preserving privacy*

A data set PT can be visualized as a tabular representation of a multi-set of tuples $r_1, r_2, \ldots, r_{n_{row}}$ where n_{row} is the number of rows in the table. Each tuple (row) r_i comprises of n_{col} values $\langle c_1, c_2, \ldots, c_{n_{col}} \rangle$ where n_{col} is the number of columns in the table. The values in column j correspond to an *attribute* a_j, the domain of which is represented by the ordered set $\Sigma_j = \{\sigma_1, \sigma_2, \ldots, \sigma_{n_j}\}$. The ordering of elements in the set can be implicit by nature of the data. For example, if the attribute is "age", the ordering can be done in increasing order of the values. For categorical data, obtaining an ordering requires the user to explicitly specify a hierarchy on the values. A hierarchy can be imposed based on how the values for the attribute can be grouped together. Fig. 1.1 shows an example hierarchy tree for the attribute "marital status". The leaf nodes in this example constitute the actual values that the attribute can take. The ordering for these values can be assigned based on the order in which the leaf nodes are reached in a preorder traversal of the hierarchy tree. The numbering on the leaf nodes specify this ordering. An internal node in the hierarchy tree specify valid groupings of child nodes.

Given such orderings on the attribute domains, a generalization specifies a valid grouping of the domain values for a particular attribute. Formally, a generalization G_j for an attribute a_j is a partitioning of the set Σ_j into ordered subsets $\langle \Sigma_{j_1}, \Sigma_{j_2}, \ldots, \Sigma_{j_K} \rangle$ which preserve the ordering in Σ_j, i.e. if σ_a appears before σ_b in Σ_j then, for $\sigma_a \in \Sigma_{j_l}$ and $\sigma_b \in \Sigma_{j_m}$, $l \leq m$. Further, every element in Σ_j must appear in exactly one subset. The elements in the subsets maintain the same ordering as in Σ_j. For the age attribute having values in the range of $[10, 90]$, a possible generalization can be $\langle [10, 30], (30, 50], (50, 70], (70, 90] \rangle$. A possible generalization for the marital status attribute can be $\langle$*Not Married, Spouse Absent, Civ-spouse, AF-spouse*$\rangle$. It is important to note that generalizations for categorical data is dependent on how the hierarchy is specified for it. Further, gen-

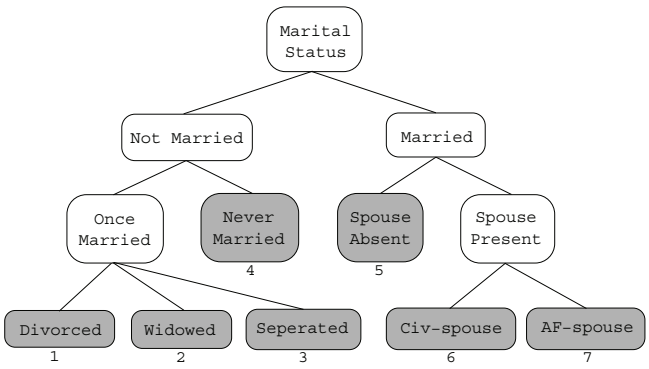

Fig. 1.1. Hierarchy tree for the *marital status* attribute. Numbering on the leaf nodes indicate their ordering in $\Sigma_{marital\ status}$.

eralizations are restricted to only those which respect the hierarchy. The generalization is said to be *constrained* in such a case. For example, the generalization $\langle$*{Divorced, Widowed}, {Separated, Never Married}, Married*$\rangle$ is not valid for marital status since the hierarchy tree specifies that the values *{Divorced, Widowed, Separated}* can only be generalized as *Once Married*, if at all.

Given the generalizations $G_1, G_2, \ldots, G_{n_{col}}$, the data set PT can be transformed to the *anonymized* data set RT by replacing each value v_{ij} at row i and column j in PT by $G_j(v_{ij})$ where $G_j(v_{ij})$ gives the index of the subset to which v_{ij} belongs to in the generalization G_j. For the example generalization of the age attribute shown earlier, an age value of say 35 would thus be recoded by the range $(30, 50]$, or in other words, $G_{age}(35) = 2$. Note that, if a particular generalization G_j is equal to the domain of values Σ_j, say $G_{age} = \langle[10, 90]\rangle$ or $G_{marital\ status} = \langle$*Marital Status*$\rangle$, all values of the corresponding attribute will be transformed to the same subset index 1, in which case all information in that attribute is lost and the *cell is suppressed.*

Once the transformation is complete, equivalent tuples may appear in RT. Two tuples are equivalent if the recoded values for the quasi-identifiers are the same in both. Thus, the adopted generalization scheme maps the original values (which may be different) corresponding to a column in the two tuples to the same anonymized form. For example, both age values 35 and 45 will be mapped to $(30, 50]$ by the generalization shown earlier. Equivalent tuples can then be grouped together into equivalence classes. In other words, an equivalence class groups all tuples in PT that got trans-

formed to the same tuple in RT under some generalization.

Although a generalization transforms the exact content of some or all data values, it is not impossible to re-establish the identity of a represented individual from the anonymized data set. Sweeney demonstrated this by using a *linking attack* on an anonymized medical insurance data set[3] of the Massachusetts state employees. The data set had been distributed for the purpose of research. Linking attacks can be launched by using information from other publicly available data sources to directly match shared attributes between the two data sources and arrive at a re-identification. For example, the governor of Massachusetts at that time, William Weld, was a state employee and hence his medical records were in the data set. Although the name was not revealed in the data, other pieces of information such as postal code, date of birth, gender etc. of the represented individuals were present. Governor Weld lived in Cambridge, Massachusetts. Sweeney managed to purchase a voter's registration list (containing information such as name, postal code, date of birth, gender etc.) for Cambridge. Sweeney observed from the voter's list that six people had Weld's particular birth date, three of which were men, and Governor Weld was the only one in his postal code, thereby revealing the exact way to identify his records on the medical data.

Matching shared attributes between different data sources can be made ambiguous by altering the released information to map to more number of individuals represented in the data set. In other words, the larger the size of the equivalence classes induced by a generalization, the more uncertainty there will be in identifying an individual. The k–anonymity model is thus proposed.[2,3]

Definition 1.3 (k–Anonymity problem). *Given a data set* PT*, find a set of generalizations for the quasi-identifiers in* PT *such that the equivalence classes induced by anonymizing* PT *using the generalizations are all of size at least* k*.*

The problem can also be explained as obtaining the generalizations under which every tuple in RT is same as at least $k-1$ other tuples. A linking attack in this case can at best identify the equivalence class of an individual, but cannot certify which one of the k records in the equivalence class corresponds to the individual.

k–anonymity is conceptually a simple privacy model but has certain drawbacks when other forms of attacks are taken into consideration. To

demonstrate such attacks, the set of attributes is first divided into *sensitive* and *non-sensitive* classes. A sensitive attribute is one whose value must not be revealed (or get revealed) for any tuple in the data set. All other attributes are considered non-sensitive. Non-sensitive attributes usually constitute the quasi-identifiers. A *homogeneity attack* can result in unwanted disclosure of the sensitive attribute value for an individual if the equivalence classes in a k–anonymization have little or no diversity in the sensitive attribute values. An attacker, after successfully determining the equivalence class to which an individual's record belongs, knows the possible sensitive attribute values for the individual. If all sensitive attribute values in that equivalence class are the same, the attacker establishes an exact identification. Even for the case when the values are not all same, the attacker can apply existing *background knowledge* on the individual to eliminate possibilities. Such attacks exploit the non-existence of diversity in the sensitive attribute values in an equivalence class. Thus, the ℓ–diversity model enforces a diversity property on the classes.[4]

Let a_s be a sensitive attribute in a data set with the domain of values $\Sigma_s = \{\sigma_1, \sigma_2, \ldots, \sigma_{n_s}\}$. Further, let $Q_1, \ldots, Q_p$ be the equivalence classes induced by a generalization. If $c(\sigma)_j$, where $\sigma \in \Sigma_s$, denotes the count of the number of tuples with the sensitive attribute value σ in Q_j, then the ℓ–diversity problem can be stated as follows.

Definition 1.4 (ℓ–Diversity problem). *Given a data set* PT*, find a set of generalizations for the quasi-identifiers in* PT *such that for each equivalence class induced by anonymizing* PT *using the generalizations, the relation*

$$\frac{c(\sigma)_j}{|Q_j|} \leq \frac{1}{\ell} \tag{1.1}$$

holds for all $\sigma \in \Sigma_s$ *and* $j = 1, \ldots, p$.

In other words, the ℓ–diversity property guarantees that a sensitive attribute value cannot be associated with a particular tuple with a probability more than $1/\ell$. The higher the value of ℓ, the better is the privacy.

However, the ℓ–diversity model ignores the distribution of the sensitive attribute values. If it so happens that certain sensitive attribute values are not well represented in the original data set then achieving ℓ–diversity may not only become difficult but can also be impossible. The difficulty arises because there may not be many diverse values of the sensitive attribute that can help maintain the diversity property for a high value of ℓ. For example,

if a sensitive attribute can take only 2 possible values, say tested *positive* or *negative* in a viral test, then one can never find a 3–diverse anonymization. Further, two different generalizations resulting in anonymizations with the same ℓ can be very different in terms of the statistical information they portray to an attacker. For example, if an equivalence class has equal number of records with positive and negative values – a 2–diverse class – then an attacker can infer that an individual tested positive (or negative) with 50% probability. On the other hand, consider the case when 90% records have positive and 10% records have negative in the sensitive attribute. This equivalence class is also 2–diverse, but with the difference that an attacker now knows that 90% of the individuals represented in the class tested positive. Such *skewness attacks* demonstrate that even if two equivalence classes are similarly diverse, the privacy risk present in them can be very different. Another attack possible on an ℓ–diverse equivalence class is the *similarity attack*. Similarity attacks exploit semantic relationships present between the sensitive attribute values of an equivalence class to deduce important information about an individual. For example, if an individual's record belongs to an equivalence class with the sensitive attribute (say disease) values "gastric ulcer", "gastritis" and "stomach cancer" then the attacker at least knows that the individual has some stomach related problem. The t–closeness model is thus proposed to avoid such forms of attacks.[5] The t–closeness model limits what an attacker can learn from an equivalence class on top of what it already knows from the entire anonymized data set.

Definition 1.5 (t–Closeness problem). *Given a data set* PT*, find a set of generalizations for the quasi-identifiers in* PT *such that for each equivalence class induced by anonymizing* PT *using the generalizations, the difference in distribution of a sensitive attribute in a class and the distribution of the attribute in the whole data set is no more than a threshold* t*.*

The question that comes to mind at this point is whether a privacy model can be applied to any data set given a particular value for its parameter. For k–anonymity, a data set is made anonymous by suppressing all tuples that belong to equivalence classes of size less than k. Similar suppression methods can be used to enforce the ℓ–diversity or t–closeness property. The case without suppression can be modeled into the scenario with suppression by assigning an infinite loss when suppression is performed.[7] However, it should be noted that the presence of outliers will always force the requirement for suppression, in which case the loss measurement will al-

ways become infinite. Furthermore, even though suppression is not allowed, such an approach enforces a privacy property by suppressing outliers. If all the data points in the data set have to stay in the anonymized data set as well, the desired privacy properties cannot be ascertained even after adopting such modeling.

As pointed out earlier, a fundamental outcome of applying these disclosure control mechanisms is a degradation in the quality of the data set for statistical studies. It is not trivial how decisions on a parameter of a privacy model affects the information content of a data set. In the next section, we shall see how the information lost as a result of a particular anonymization technique can be quantified to measure data utility.

1.3.2. *Estimating information loss*

An optimization algorithm requires a numeric representation of the information loss associated with a particular generalization. A quantified loss value enables the optimization algorithm to compare two generalizations for their relative effectiveness. Loss (cost) metrics assign some notion of penalty to each tuple whose data values get generalized or suppressed, thereby reflecting the total information lost in the anonymization process. However, cost quantification is a relatively harder and sparsely researched area. One should understand that the notion of information loss can vary from application to application. Hence, most of the proposed cost metrics are not rigorous enough to capture the data utility as would be perceived by a data publisher. They are rather estimates to compare different privacy models under a common test bed.

Early notion of utility is based on the number of generalization steps one has to take to achieve a given privacy requirement.[1] Such a method assumes that attribute domains can be progressively generalized and a partial order can be imposed on the domain of all generalizations for an attribute. For instance, postal codes can be generalized by dropping a digit from right to left at each generalization step. Postal addresses can be generalized to the street, then to the city, to the county, to the state, and so on. Given that such orderings can be imposed, a distance can be computed between two different generalizations for an attribute. The result is a distance vector with an entry for each attribute. Maximal utility is then said to be achieved if one can find generalizations for the attributes that satisfy the privacy requirement and results in a non-dominated distance vector computed from the origin (the case of no generalization). A non-dominated

distance vector is one whose distance values are not higher than in another vector in all the attributes. In other words, utility is considered to be most when generalizations recode data values only to the extent necessary to achieve the privacy property.

Numerical estimations of information loss started with the *general loss metric* proposed by Iyengar.[8] The general loss metric computes a normalized information loss for each data value in an anonymized data set. The assumption here is that information in every column is potentially important and hence a flexible scheme to compute the loss for both numeric and categorical data is required.

Consider the data value v_{ij} at row i and column j in the data set PT. The general loss metric assigns a penalty to this data value based on the extent to which it gets generalized during anonymization. Let $g_{ij} = G_j(v_{ij})$ be the index of the subset to which v_{ij} belongs to in the generalization G_j, i.e. $v_{ij} \in \Sigma_{j_{g_{ij}}}$. The penalty for information loss associated with v_{ij} is then given as follows:

$$loss(v_{ij}) = \frac{|\Sigma_{j_{g_{ij}}}| - 1}{|\Sigma_j| - 1} \tag{1.2}$$

For categorical data, the loss for a cell is proportional to the number of leaf nodes rooted at an internal node (the generalized node) of the hierarchy tree. The loss attains a maximum value of one when the cell is suppressed ($G_j = \langle \Sigma_j \rangle$), or in other words, when the root of the tree is the generalized node. Subtracting one ensures that a non-generalized value incurs zero loss since the cardinality of the subset to which it belongs would be one. The *generalization loss* is then obtained as the total loss over all the data values in the data set.

$$GL = \sum_{i=1}^{n_{row}} \sum_{j=1}^{n_{col}} loss(v_{ij}) \tag{1.3}$$

Further, when a row is suppressed, all cells in the row are suppressed irrespective of the generalization. Each cell thereby incurs a loss of one. Let n_{sup} be the number of rows to be suppressed in the data set. The *suppression loss* for the data set is then given as,

$$SL = n_{col} \times n_{sup} \tag{1.4}$$

A widely used cost metric, called the *discernibility metric*, assigns a penalty to each tuple based on the number of tuples in the transformed data set that are indistinguishable from it.[7] A tuple belonging to an equivalence

class of size j is assigned a penalty of j. A suppressed tuple is assigned a penalty equal to the number of tuples in the data set. The idea behind using the size of the equivalence class as a measure of data utility is to penalize generalizations that result in equivalence classes bigger than what is required to enforce a given privacy requirement. A variant of this is to use the normalized average equivalence class size.[10]

Data utility is often measured in conjunction with privacy in an attempt to combine both objectives into a single metric. A metric of such nature favors generalizations that result in maximum gain in the information entropy for each unit of anonymity loss resulting from the generalization. Methods employing such a metric progressively increase the amount of generalization, called a *bottom up generalization* approach,[11] or decrease it, called a *top down specialization* approach,[12] with the objective of maximizing the metric without violating the anonymity requirement. Another metric, called *usefulness*, measures utility as the average diversity in the tuples belonging to an equivalence class.[13] This measurement is similar to the general loss metric, with differences being in the treatment of interval based attribute domains. For such domains, the loss is assigned as the normalized distance between the maximum and minimum values of the attribute in the equivalence class. Further, a complementary metric, called *protection*, uses the inverse of the tuple diversities as a measure of the privacy factor. By doing so, the two metrics inherently exhibit a reciprocal relationship, useful when a data publisher wants to modulate the anonymization process towards one objective or the other.

Researchers also argue that utility metrics should not only capture the information loss caused by the anonymization but also account for the importance of the different attributes. For example, given a disease analysis data set, an age attribute may be considered more critical than a zip code attribute. Generalizations that are able to maintain the age attribute more accurately should thus be favored. Based on this, the *weighted normalized certainty penalty* metric uses a weighted sum of the loss measurements in different attributes of the data set.[14] The loss measurement is similar as in the general loss metric and the usefulness metric. Introduction of such preference characteristics indicates that the measurement of utility can be a very subjective matter after all.

We want to re-emphasize that the notion of utility is still not understood well, probably much because of its subjective nature. Nonetheless, multi-objective formulations do not assume any inherent property in the cost metric used. The cost metric is only used in its functional form to evaluate

the effectiveness of a generalization in maintaining the utility factor. The metric may be different for different purposes, however, the objective of the analysis remains the same.

1.4. Evolutionary Optimization

Any disclosure control mechanism has to cater to two primary objectives – maintaining a high privacy level and facilitating statistical inquiries by reducing the information loss. One should understand that both of these objectives are rather subjective in nature. The privacy level required in a shared data source is dictated by the sensitivity of the personal information contained in the source. This is often determined by the concern displayed by the individuals represented in the source about the extraction of certain pieces of information in its exact form. The requirement of data utility is determined by the nature of the analysis to be performed on the data set. It is therefore possible what evaluates as unusable for one statistical study is sufficient in another context. Such subjectivity compels a data publisher to reevaluate the control mechanisms under the light of the expressed relevance of the two objectives. Furthermore, the two objectives are conflicting in nature, meaning that a control mechanism cannot attain both at the same time. Thus, even if both objectives are specified as being equally important, a chosen control will always exhibit a bias towards one or the other. Given that such bias could be unavoidable, the data publisher must make a best attempt in understanding the deviations brought forth in the objectives by the selection of one control over another. This not only postpones the subjectivity of the two objectives to a post-analysis stage, but also aids the decision making process with a comprehensive overview of the effects of incremental modification in the expressed subjective relevance.

Classical approaches to solve multi-objective problems are mostly based on transformations of the problem into single objective instances. A typical method is the assignment of weights to objectives and then performing a single objective optimization of the scalarized objective function values. Other methods require the specification of preference orderings on the objective functions. Noticeably, these methods can only return a single solution from a single run and hence trade-off analysis shall require multiple runs to be performed. Recent advances in population-based evolutionary optimization have proved to be particularly effective in overcoming this bottleneck. The artificial intelligence community has seen a major flow of algorithms using dominance as the decisive factor for natural selection. These algorithms

can not only identify non-dominated solutions, but also eliminate the need for repeated analysis by using population based approaches.

In the next few sections, we shall consider the k–anonymity model with the general loss metric as a working platform to describe the problem formulations. Nevertheless, formulations for other privacy models and utility metrics are not very dissimilar. We shall also see how specifics related to the application of an evolutionary algorithm to the problems are resolved.

1.4.1. *Multi-objective analysis*

A multi-objective analysis is not intended to provide the data publisher a "best" value for the parameter(s) involved in an anonymization technique. Rather, the methodology is meant to understand the implications of choosing a particular value for the parameter(s) in terms of the resulting privacy and the data utility. Hence, we shall often find that one or more solutions returned by the optimization process are trivially not acceptable either in terms of privacy or utility, or in some cases both. It is not our objective to consider such solutions as degenerate and prohibit them from appearing in the solution set. For example, an extreme solution will correspond to a situation where every tuple in the data set belongs to its own equivalence class, thereby resulting in no privacy and maximum utility. Another extremity is the case where all tuples are grouped together in a single equivalence class resulting in maximum privacy but no utility. One cannot deny the fact that in the case of privacy versus utility, both of these are possible solutions. The multi-objective optimization does not incorporate the required domain knowledge to identify these extremities (or other such solutions) as being impractical. Only the data publisher has the requisite knowledge to make such identification and disregard such solutions. This is often a post-optimization process.

Furthermore, the multi-objective analysis is not a direct means of protecting privacy. The problem formulations we present here shall show how privacy can be modeled as an objective to maximize rather than being treated as a constraint. By doing so, we can provide clues to the data publisher as to what levels of privacy can be obtained (for example by using k–anonymity) for a given amount of information loss. This trade-off analysis can be used by the data publisher to finally decide what privacy can it offer. More specifically, it offers the data publisher a method to understand the effects of setting a parameter in a privacy model on the utility of the data.

1.4.1.1. *In the absence of suppression*

The presence of outliers in a data set makes it difficult to find a suitable value of k when suppression of data is not allowed. In this formulation, we strictly adhere to the requirement that no tuple in the data set can be deleted. Intuitively, such a strict requirement makes the k–anonymity problem insensible to solve for a given k as the optimization algorithm will be forced to overly generalize the data set in its effort to ensure k–anonymity. The outliers usually belong to very small equivalence classes and the only way to merge them into a bigger one is by having more generalization. This results in more information loss which is often not acceptable to an user.

Although solving the k–anonymity problem is not possible in terms of its strict definition, it is worth noting that a generalization can still affect the distribution of the equivalence classes even when suppression is not allowed. An equivalence class E_k in this description groups all tuples that are similar to exactly $k-1$ other tuples in the anonymized data set. An ideal generalization would then maintain an acceptable level of loss by keeping the number of rows in smaller equivalence classes (small k) relatively lower than in the bigger equivalence classes. Although this does not guarantee complete k–anonymity, the issue of privacy breach can be solved to a limited extent by reducing the probability that a randomly chosen row would belong to a small equivalence class.

With this motivation we define the *weighted-k–anonymity* multi-objective problem to find generalizations that produce a high weighted-k value and low generalization loss. Each equivalence class E_k defines a k value, $k \leq n_{row}$, for its member tuples – every tuple in the equivalence class is same as exactly $k-1$ other tuples in the same class.

Note that this notion of an equivalence class is different from the one stated in the k–anonymity problem (Def. 1.3). Two rows in the original data set belong to the same equivalence class in the k–anonymity definition if the generalization transforms them into the same tuple. In this formulation, two rows belong to the same equivalence class E_i if a generalization makes them i–anonymous.

The weighted-k for a particular generalization inducing the equivalence classes $E_1, E_2, \ldots, E_{n_{row}}$ on the anonymized data set is then obtained as follows:

$$k_{weighted} = \frac{\sum_{i=1}^{n_{row}} (i \cdot |E_i|)}{\sum_{i=1}^{n_{row}} |E_i|} \tag{1.5}$$

The problems introduced by the presence of outliers can also be ad-

dressed by using the concept of local recoding.[15] A local recoding scheme produces a k–anonymization by using an individual generalization function (instead of a global one) for each tuple in the data set. This is a more powerful scheme compared to having a single generalization function since outliers can be easily suppressed without the drawbacks of an over generalization, hence data utility can be maintained. The weighted-k–anonymity based generalization is orthogonal to this concept in certain ways. Local recoding explores the domain of generalization functions and uses multiple points in this domain to recode different subsets of the data set differently. This puts outliers in their own subset(s), thereby making it easy to enforce a given minimum equivalence class size (k). Weighted-k–anonymity, on the other hand, works with a single generalization function and instead of trying to enforce a fixed minimum equivalence class size, flexibly creates equivalence classes of different sizes with no minimum size constraint. The outliers then must lie on smaller equivalence classes in order to maximize data utility. The common criteria in both the methods is that the outliers gets treated differently than the rest of the data set.

In most cases, not all equivalence classes with all possible k values will be generated. The weighted-k value provides a sufficiently good estimate of the distribution of the equivalence classes. A high weighted-k value implies that the size of the equivalence classes with higher k is relatively more than the size of the lower k ones. The multi-objective problem is then formulated as finding the generalizations that maximize the weighted-k and minimize the generalization loss.

1.4.1.2. *With pre-specified suppression tolerance*

In this problem, we enable suppression and allow the user to specify an acceptable fraction, denoted by η, of the maximum suppression loss possible ($n_{row} \cdot n_{col}$). Such an approach imposes a hard limit on the number of suppressions allowed.[7] However, by allowing the user to specify a suppression loss limit independent of k, the optimization procedure can be made to explore the trade-off properties of k and generalization loss within the constraint of the imposed suppression loss limitation.

When suppression is allowed within an user specified limit, all tuples belonging to the equivalence classes $E_1, \ldots, E_d$ can be suppressed, such that d satisfies the relation

$$\sum_{i=1}^{d}(|E_i| \cdot n_{col}) \leq \eta \cdot n_{row} \cdot n_{col} < \sum_{i=1}^{d+1}(|E_i| \cdot n_{col}) \tag{1.6}$$

Satisfying the relationship results in the suppression of tuples beginning from the ones which has the least number of duplicates (smaller equivalence class) and ending when further suppression will violate the specified tolerance. Thus, the k value induced by the generalization is equal to $d+1$, which also satisfies the suppression loss constraint. We can now define our optimization problem as finding the generalizations that maximize d and minimize the generalization loss. The problem can also be viewed as the maximization of k and minimization of GL satisfying the constraint $SL \leq \eta \cdot n_{row} \cdot n_{col}$.

1.4.1.3. *For comprehensive overview*

The third problem is formulated as an extension of the second one where the user does not provide a maximum limit on the suppression loss. The challenge here is the computation of k, GL and SL for a generalization without having a baseline to start with. Since the three quantities are dependent on each other for their computation, it is important that we have some base k value to proceed. The weighted-k value is adopted at this point. Although not very precise, the weighted-k value provides a good estimate of the distribution of the equivalence classes. If a very high weighted-k value is obtained for a generalization then the number of tuples with low k's is sufficiently small, in which case we can suppress them. If the weighted-k value is low then most of the tuples belong to equivalence classes with low k. In this case, a higher amount of suppression is required to achieve an acceptable k for the anonymized data set. Also, high weighted-k generally implies a high generalization loss. Such trade-off characteristics are the point of analysis in this problem.

To start with, a particular generalization's weighted-k value is first computed. Thereafter, all tuples belonging to an equivalence class of $k < k_{weighted}$ are suppressed, enabling the computation of SL. This makes the k for the anonymized data set equal to at least $k_{weighted}$. The generalization loss GL is then computed from the remaining data set. The multi-objective problem is defined as finding the generalizations that maximize $k_{weighted}$ and minimize the generalization and suppression losses.

1.4.2. *Solution encoding*

Before applying an evolutionary algorithm to obtain solutions to the fore mentioned problems, a viable representation of the generalizations has to be designed for the algorithm to work with. Consider the numeric attribute

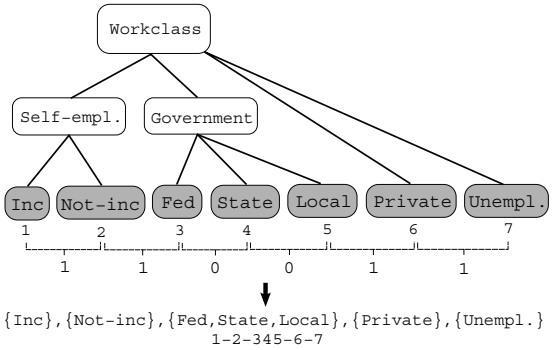

Fig. 1.2. Example generalization encoding for the *workclass* constrained attribute. i^{th} bit is 0 if i^{th} and $(i+1)^{th}$ intervals are combined, otherwise 1.

age with values in the domain $[10, 90]$. Since this domain can have infinite values, the first task is to granularize the domain into a finite number of intervals. For example, a granularity level of 5 shall discretize the domain to $\{[10, 15], (15, 20], \ldots, (85, 90]\}$. Note that this is not the generalization used to anonymize the dataset. The discretized domain can then be numbered as $1 : [10, 15], 2 : (15, 20], \ldots, 16 : (85, 90]$. The discretized domain still maintains the same ordering as in the continuous domain. A binary string of 15 bits can now be used to represent all possible generalizations for the attribute. The i^{th} bit in this string is 0 if the i^{th} and $(i+1)^{th}$ intervals are supposed to be combined, otherwise 1. For attributes with a small domain size and a defined ordering of the values, the granularization step can be skipped. For categorical data, a similar encoding can be obtained once an ordering on the domain values is imposed as discussed in Section 1.3. Fig. 1.2 shows an example generalization encoding for a "workclass" attribute. The individual encodings for each attribute are concatenated to create the overall encoding for the generalizations for all attributes.

1.4.3. *Non-dominated Sorting Genetic Algorithm-II*

The Non-dominated Sorting Genetic Algorithm-II (NSGA-II)[16] is a popular evolutionary algorithm to perform multi-objective optimization. It employs the concept of dominance, as discussed in Section 1.2, to find Pareto-optimal solutions in the space of all possible generalizations of a given data set. Note that NSGA-II is just an algorithm of choice in this study. Other evolutionary algorithms do exist that can help perform a similar analysis.[17] However, the primary objective in this chapter is not to focus

on performance analysis of multi-objective optimization but to demonstrate the usage of pre-existing techniques in the artificial intelligence community to resolve the problems in disclosure control. The extensive usage of NSGA-II in solving real world problems has motivated us to choose NSGA-II over others. Having said so, these algorithms are typically very generic. Any application of the algorithm thus requires appropriate problem and operator representations. We have already discussed how a solution in SDC can be represented for NSGA-II. Other components are discussed in subsequent sections.

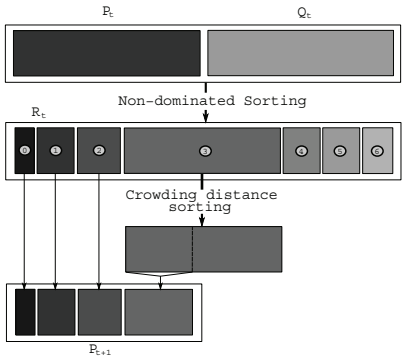

Fig. 1.3. One generation of NSGA-II.

The algorithm starts with a population P_0 of N random generalizations. A generation index $t = 0, 1, \ldots, Gen_{MAX}$ keeps track of the number of iterations of the algorithm. Each trial generalization is used to create the anonymized dataset and the corresponding values of the quantities to be optimized are calculated. Each generation of NSGA-II then proceeds as follows. An offspring population Q_t is first created from the parent population P_t by applying the usual genetic operations of selection, crossover and mutation.[18] This is done by first forming a mating pool of best solutions, next recombining solutions from this pool to generate offspring, and finally mutating the offspring to get a new solution. For constrained attributes, a special crossover operator is used as discussed in the next subsection. The offspring population also gets evaluated. The parent and offspring populations are then combined to form a population $R_t = P_t \cup Q_t$ of size $2N$. A non-dominated sorting is applied to R_t to rank each solution based on the number of solutions that dominate it. Rank 1 solutions are all non-dominated solutions in the population. A rank r solution is only

dominated by solutions of lower ranks.

The population P_{t+1} is generated by selecting N solutions from R_t. The preference of a solution is decided based on its rank; lower the rank, higher the preference. By combining the parent and offspring populations, and selecting from them using a non-dominance ranking, NSGA-II implements an elite-preservation strategy where the best solutions obtained are always passed on to the next generation. However, since not all solutions from R_t can be accommodated in P_{t+1}, a choice is likely to be made when the number of solutions of the currently considered rank is more than the remaining positions in P_{t+1}. Instead of making an arbitrary choice, NSGA-II uses an explicit diversity-preservation mechanism. The mechanism, based on a *crowding distance metric*,[16] gives more preference to a solution with a lesser density of solutions surrounding it, thereby enforcing diversity in the population. The NSGA-II crowding distance metric for a solution is the sum of the average side-lengths of the cuboid generated by its neighboring solutions. Fig. 1.3 depicts a single generation of the algorithm. For a problem with M objectives, the overall complexity of NSGA-II is $O(MN^2)$.

1.4.4. *Crossover for constrained attributes*

The usual single point crossover operator in a genetic algorithm randomly chooses a crossover point. It then creates an offspring by combining the bit string before the crossover point from one parent and the bit string after the crossover point from the other. As shown in Fig. 1.4 (left), such an operation can result in an invalid generalization for constrained attributes. Iyengar proposed modifying such invalid generalizations to the nearest valid generalization.[8] However, finding the nearest valid generalization can be time consuming, besides destroying the properties on which the crossover operator is based on. In this regard, Lunacek et al. proposed a special crossover operator that always create valid offspring for constrained attributes.[9] Instead of randomly choosing a crossover point, their operator forces the crossover point to be chosen at a location where the bit value is one for both parents. By doing so, both parts (before and after the crossover point) of both parents can be guaranteed to be valid generalizations individually, which can then be combined without destroying the hierarchy requirement. Fig. 1.4 (right) shows an instance of this operator.

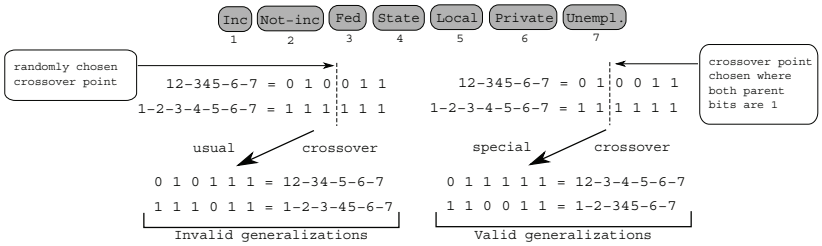

Fig. 1.4. Usual single point crossover (left) and special crossover for constrained attributes (right). Usual operator may generate invalid offspring generalizations. Special operator always creates valid offspring generalizations.

1.4.5. *Population initialization*

In order to be able to use the special crossover operator, the validity of the parent solutions must be guaranteed. This implies that the initial random population that NSGA-II starts with must contain trial solutions with valid generalizations for the constrained attributes. For a given hierarchy tree, the following algorithm can generate valid generalizations for the constrained attributes in the initial population.

Starting from the root node, a node randomly decides if it would allow its subtrees to be distinguishable. If it decides not to then all nodes in its subtrees are assigned the same identifier. Otherwise the root of each subtree receives an unique identifier. The decision is then translated to the root nodes of its subtrees and the process is repeated recursively. Once all leaf nodes are assigned an identifier, two adjacent leaf nodes in the imposed ordering are combined only if they have the same identifier. Since a parent node always make the decision if child nodes will be combined or not, all generalizations so produced will always be valid.

1.5. Some Empirical Results

We show here some results obtained by applying the NSGA-II algorithm to a standard "adult census" test data set[a]. The data was extracted from a census bureau database and has been extensively used in studies related to k–anonymization. All rows with missing values are removed from the dataset to finally have a total of 30162 rows. The attributes "age", "education", "race", "gender" and "salary class" are kept unconstrained, while the attributes "workclass", "marital status", "occupation" and "native coun-

[a]ftp://ftp.ics.uci.edu/pub/machine-learning-databases/adult/

try" are constrained by defining a hierarchy tree on them. The remaining attributes in the dataset are ignored.

For NSGA-II, we set the population size as 200. The maximum number of iterations is set as 250. A single point crossover is used for unconstrained attributes while Lunacek et al.'s crossover operator is used for constrained attributes. Also, mutation is only performed on the unconstrained attributes. The remaining parameters of the algorithm are set as follow: crossover rate = 0.9, mutation rate = 0.1 with binary tournament selection. We ran the algorithm with different initial populations but did not notice any significant difference in the solutions obtained. The results here are from one such run.

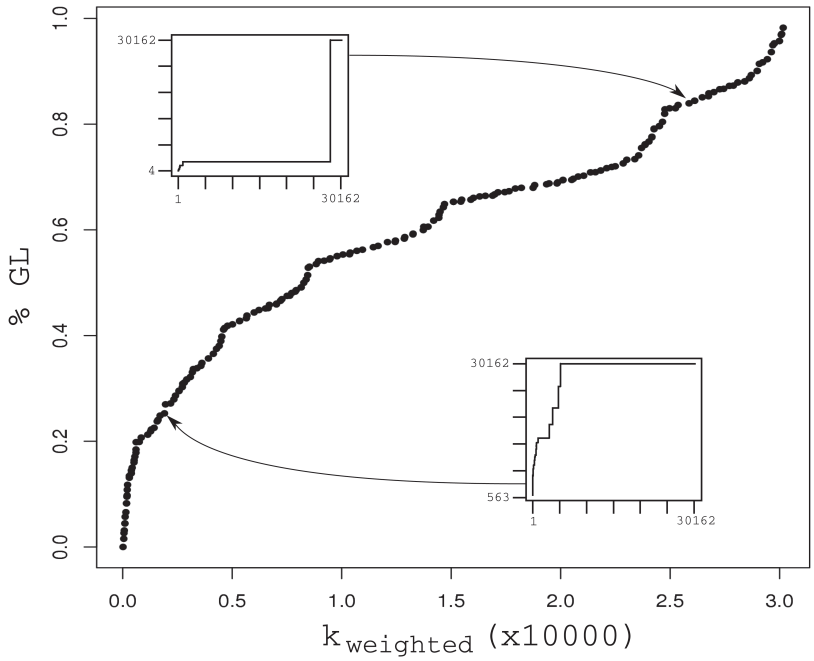

Fig. 1.5. Solutions found by NSGA-II when suppression is not allowed. Each point corresponds to a generalization with a particular value of $k_{weighted}$ and GL, thereby demonstrating the trade-off behavior. Generalization loss increases as bigger equivalence classes are generated. Inset figures show cumulative distribution of $|E_k|$ as k increases. Distribution of equivalence class sizes can be very different for two solutions.

Fig. 1.5 shows the different trade-off solutions obtained by NSGA-II when no suppression is allowed on the data set. A point in the plot corresponds to a solution that induces a particular distribution of equivalence class sizes (k values) on the anonymized data set. As expected, the generalization loss increases as the distribution of equivalence classes gets more inclined towards higher k values. In the absence of suppression, a single k value is often hard to enforce for all tuples in the data set. Thus, a solution here results in different k values for different tuples. A higher weighted-k value signifies that most tuples have a high k value associated with them, in which case, the generalization loss is higher. A solution with low weighted-k value results in a generalization with low k values for its tuples.

The inset figures in the plot depict the cumulative distribution of the number of tuples belonging to equivalence classes (y-axis) with different k values (x-axis). Note that the distributions for the two example solutions are not complementary in nature. For the solution with lower generalization loss, the distribution has a continuously increasing trend, implying that equivalence classes of different k values exist for the solution. The other solution shows an abrupt increase signifying that the tuples either belong to equivalence classes with very small k or ones with very large k. The sought balance in the distribution can therefore exist with an acceptable level of generalization loss.

Fig. 1.6 shows the trade-off between k and $loss = GL + SL$ when a maximum of 10% suppression loss is allowed. The top-leftmost plot shows all the solutions obtained for the problem. Each subsequent plot (follow arrows) is a magnification of the steepest part in the previous plot. Each plot shows the presence of locally flat regions where a substantial increase in the k value does not have a comparatively high increase in the *loss*. These regions can be of interest to a data publisher since it allows one to provide higher levels of data privacy without compromising much on the information content.

Interestingly, the trend of the solutions is similar in each plot. The existence of such repeated characteristics on the non-dominated front suggests that a data publisher's choice of a specific k, no matter how big or small, can have avenues for improvement, specially when the choice falls in the locally flat regions. A choice of k made on the rising parts of the front is seemingly not a good choice since the publisher would then be paying a high cost in degraded data quality without getting much improvement on the privacy factor. The rational decision choice in such a case would be to lower the value of k to a flat region of the front.

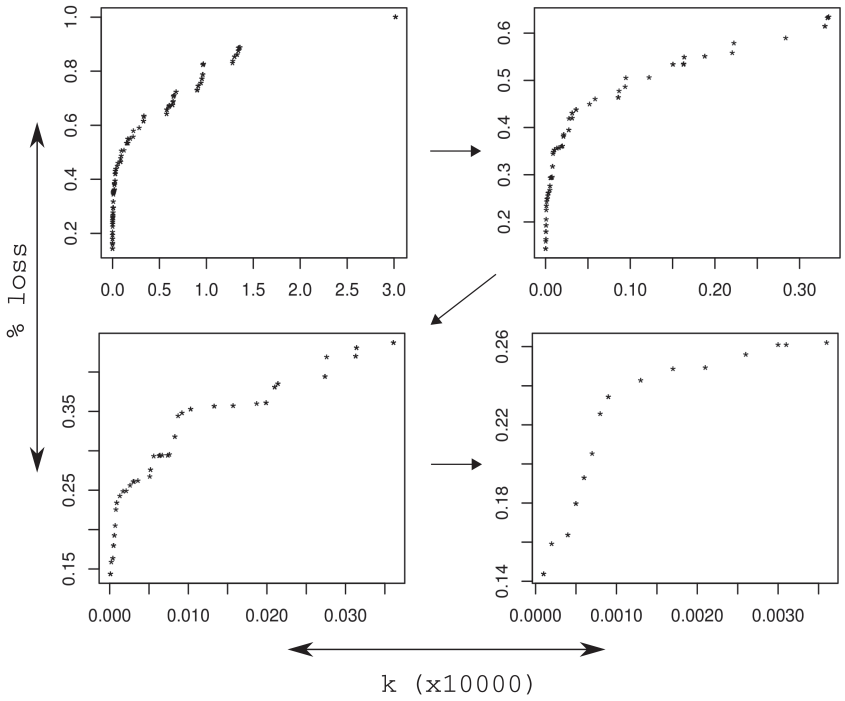

Fig. 1.6. Solutions found by NSGA-II with suppression tolerance $\eta = 10\%$. Top-leftmost plot shows all obtained solutions. Each point corresponds to a generalization with a particular value of k and *loss*, and with a maximum of 10% suppression loss. Each subsequent plot (follow arrows) is a magnification of a region of the previous plot. Solution set depicts flat regions in the non-dominated front, implying that one objective can be substantially improved with negligible changes in the other. Such trends are visible in all parts of the front.

1.6. Summary

We presented the method of multi-objective analysis to demonstrate that the choice of a parameter in a privacy model can be made in a much informed manner rather than arbitrarily. The multi-objective problems are formulated to cater to differing requirements of a decision maker, primarily focused on the maximization of the privacy parameter and minimization of the losses. For generalizations without suppression, an unique value for the parameter may not be available. However, generalizations may be possible that provide a higher level of privacy for a higher fraction of the dataset without compromising much on its information content. When suppression

is allowed up to a hard limit, the nature of the non-dominated solution set can provide invaluable information on whether an anonymization exists to improve a particular value of the model parameter without much degradation in quality of the data. First-level explorations in this context can begin with gaining an overall understanding of the trade-off characteristics in the search space. The formulations presented in this chapter also address the data publisher's dilemma. They provide a methodology to analyze the problem of data anonymization in manners that appeal to the actual entity that disseminates the data. We believe that such an analysis not only reinstates the data publisher's confidence in its choice of a particular privacy model parameter, but also identifies ways of examining if the level of privacy requested by a human subject is achievable within the acceptable limits of perturbing data quality.

Future work in this direction can start with examination of the framework with other models of privacy preservation. Hybrid models catering to different forms of attacks are also required. Work on this can begin with an exploration on what trade-offs are generated when looking for the existence of two or more privacy properties simultaneously.

Acknowledgment

This work was partially supported by the United States Air Force Office of Scientific Research under contracts FA9550-07-1-0042 and FA9550-07-1-0403. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing official policies of the U.S. Air Force or other federal government agencies.

References

1. P. Samarati, Protecting Respondents' Identities in Microdata Release, *IEEE Transactions on Knowledge and Data Engineering*. **13**(6), 1010–1027, (2001).
2. L. Sweeney, Achieving k–Anonymity Privacy Protection Using Generalization and Suppression, *International Journal on Uncertainity, Fuzziness and Knowledge-based Systems*. **10**(5), 571–588, (2002).
3. L. Sweeney, k–Anonymity: A Model for Protecting Privacy, *International Journal on Uncertainity, Fuzziness and Knowledge-based Systems*. **10**(5), 557–570, (2002).
4. A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. ℓ–Diversity: Privacy Beyond k–Anonymity. In *Proceedings of the 22nd International Conference on Data Engineering*, p. 24, Atlanta, GA, USA, (2006).

5. N. Li, T. Li, and S. Venkatasubramanian. t–Closeness: Privacy Beyond k–Anonymity and ℓ–Diversity. In *Proceedings of the 23rd International Conference on Data Engineering*, pp. 106–115, Istanbul, Turkey, (2007).
6. A. Hundepool and L. Willenborg. Mu and Tau Argus: Software for Statistical Disclosure Control. In *Proceedings of the Third International Seminar on Statistical Confidentiality*, (1996).
7. R. J. Bayardo and R. Agrawal. Data Privacy Through Optimal k-Anonymization. In *Proceedings of the 21st International Conference on Data Engineering*, pp. 217–228, Tokyo, Japan, (2005).
8. V. S. Iyengar. Transforming Data to Satisfy Privacy Constraints. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 279–288, Alberta, Canada, (2002).
9. M. Lunacek, D. Whitley, and I. Ray. A Crossover Operator for the k-Anonymity Problem. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, pp. 1713–1720, Seattle, Washington, USA, (2006).
10. K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Mondrian Multidimensional K-Anonymity. In *Proceedings of the 22nd International Conference on Data Engineering*, p. 25, Atlanta, GA, USA, (2006).
11. K. Wang, P. Yu, and S. Chakraborty. Bottom-Up Generalization: A Data Mining Solution to Privacy Protection. In *Proceedings of the 4th IEEE International Conference on Data Mining*, pp. 249–256, Brighton, UK, (2004).
12. B. C. M. Fung, K. Wang, and P. S. Yu. Top-Down Specialization for Information and Privacy Preservation. In *Proceedings of the 21st International Conference on Data Engineering*, pp. 205–216, Tokyo, Japan, (2005).
13. G. Loukides and J. Shao. Capturing Data Usefulness and Privacy Protection in K-Anonymisation. In *Proceedings of the 2007 ACM Symposium on Applied Computing*, pp. 370–374, Seoul, Korea, (2007).
14. J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. Fu. Utility-Based Anonymization Using Local Recodings. In *Proceedings of the 12th Annual SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–790, Philadelphia, PA, USA, (2006).
15. A. Takemura. Local Recoding by Maximum Weight Matching for Disclosure Control of Microdata Sets. CIRJE F-Series CIRJE-F-40, CIRJE, Faculty of Economics, University of Tokyo, (1999).
16. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II, *IEEE Transactions on Evolutionary Computation.* **6**(2), 182–197, (2002).
17. K. Deb, *Multi-objective Optimization Using Evolutionary Algorithms.* (John Wiley & Sons Inc., 2001).
18. D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning.* (Addison-Wesley, 1989).