A Comparison of Single- and Multi-Objective Programming Approaches to Problems with Multiple Design Objectives

Sibel Yaman and Chin-Hui Lee

***Abstract*— In this paper, we propose and compare single- and multi-objective programming (MOP) approaches to the language model (LM) adaptation that require the optimization of a number of competing objectives. In LM adaptation, an adapted LM is found so that it is as close as possible to two independently trained LMs. The LM adaptation approach developed in this paper is based on reformulating the training objective of a maximum a posteriori (MAP) method as an MOP problem. We extract the individual *at least partially conflicting* objective functions, which yields a problem with four objectives for a bigram LM: The first two objectives are concerned with the best fit to the adaptation data while the remaining two objectives are concerned with the best prior information obtained from a general domain corpus. Solving this problem in an iterative manner such that each objective is optimized one after another with constraints on the rest, we obtain a target LM that is a log-linear interpolation of the component LMs. The LM weights are found such that all the (at least partially conflicting) objectives are optimized simultaneously. We compare the performance of the SOP- and MOP-based solutions. Our experimental results demonstrate that the ICO method achieves a better balance among the design objectives. Furthermore, the ICO method gives an improved system performance.**

I. Introduction

It has been increasingly recognized that realistic problems often involve the consideration of a tradeoff among many design objectives. Consider regularization methods employed to avoid over-fitting, and hence, to improve generalization capabilities of learning machines such as neural networks [1]. Traditionally, regularization is conduced by including an additional term, which penalizes overly high model complexity, in the cost function of a learning algorithm. These regularization methods aim at a tradeoff between accuracy and model complexity, which in most cases do not go hand-in-hand.

Traditional algorithms aim to satisfy multiple objectives by forming a global objective function and solving the resulting problem through the use of classical single-objective programming (SOP) methods. Combining several competing objectives into an overall objective function, such SOP-based approaches promise that the chosen overall objective function is optimized. However, there is no guarantee on the performance of the individual objectives as they are not considered separately. Moreover, one or more objectives tend to dominate the optimization process. It will easily become overwhelming to find an overall objective function that achieves desirable levels for the individual objectives. For these reasons, we articulate that methods of traditional SOP are not enough and take an multi-objective programming (MOP) perspective for solving such problems.

One of the most researched engineering problems from an MOP point of view is regularization. Accuracy versus model complexity trade-off for designing neural networks was studied in [2], [3]. With a similar kind of mind-set, evolutionary MOP of support vector machines (SVMs) was considered in [4] to minimize FA rate, FR rate and the number of support vectors to reduce model complexity. All these previous work indicate that MOP offers a great degree of freedom for obtaining a proper tradeoff among accuracy and model complexity.

To illustrate the use of MOP in a realistic application, we consider the language model (LM) adaptation problem [5], in which a background LM is adapted to an application domain so that the adapted LM is as close as possible to both the background model and the application domain data. Language modeling and adaptation is used in many speech and language processing applications such as speech recognition, machine translation, part-of-speech tagging, parsing, and information retrieval.

Statistical n-grams are the state-of-art language models (LMs) for large vocabulary automatic speech recognition (ASR). A statistical n-gram LM is a representation of an $(n-1)^{st}$ order Markov model in which the probability of the occurrence of a symbol is conditioned upon the occurrence of the preceding $(n-1)$ symbols. Such n-gram models are typically constructed from a large corpus of text based on the co-occurrences of the existing words. In practice, the n-gram LMs are extremely brittle across domains, and even within a domain when the training and the recognition stages involve moderately disjoint time periods.

When the application specific data is of limited amount, a general domain dataset is used to estimate an adequate representation of prior information about the n-gram probabilities, which is called a background LM. The target LM is formed by adapting the background LM to the new application domain by making efficient use of the limited application-specific data. Among the most popular LM adaptation techniques, interpolation-based approaches use the application-specific adaptation data to derive an LM that is merged with the background model. Cache methods exploit self-triggering words inside the application-specific data set to capture short-time shifts in word frequencies, which cannot be captured

S. Yaman and C.-H. Lee are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0250 (e-mail: syaman@ece.gatech.edu;chl@ece.gatech.edu).

by the background model. Constraint specification approaches use the application-specific data set to extract featuresthat the adapted LM is constrained to satisfy. Topic-based approaches use the application-specific data set to extract information about the underlying subject. MAP-based approaches use a prior distribution to exploit how much the n-gram estimates in the specific domain diverge from the background estimates. See [5] for a review of the most popular LM adaptation techniques with pointers to appropriate references.

In this paper, we take an multi-objective programming (MOP) approach to LM adaptation. MOP is concerned with finding the solutions in which a set of objective functions are simultaneously optimized, meaning that it is not possible to improve any objective without degrading some others. Many practical applications such as pattern classification can be posed as MOP problems, e.g., as we did in [6].

The LM adaptation approach developed in this paper is based on reformulating the training objective of the structural MAP (SMAP) method that we proposed in [7] as an MOP problem. We extract the individual *at least partially conflicting* objective functions in the SMAP formulation. For a bigram LM, this yields a problem with four objectives: The first two objectives are concerned with the best fit to the adaptation data while the remaining two objectives are concerned with the best prior information obtained from a general domain corpus. Solving this problem in an iterative manner such that each objective is optimized one after another with constraints on the rest, we obtain a target LM that is a log-linear interpolation of the component LMs. The LM weights are found such that all the (at least partially conflicting) objectives are optimized simultaneously.

The rest of the paper is organized as follows. In Section II, some background information on MOP is provided. In Section III, the iterative constrained optimization technique that we propose to use in LM adaptation is described. In Section IV, an SOP- and an MOP-based approach is described for LM adaptation. In Section **??**, our experimental results are reported. Finally, the conclusion and future work is presented in Section VII.

II. MOP Terminology

Suppose that we are given a set of K competing objectives, $f_k(\theta) \in (0,1)$, $k = 1, ..., K$, each of which is nonlinear function of a set of M decision vectors, $\omega_m \in \Re^n$, $m = 1, ..., M$. The *best compromise solutions*, $\{\hat{\omega}_1, ..., \hat{\omega}_M\}$, are found by MOP, which is formulated as:

$$\hat{\theta} = \{\hat{\omega}_1, ..., \hat{\omega}_M\} = \arg\min_{\theta} \left[f_1(\theta), f_2(\theta), ..., f_K(\theta)\right]. \quad (1)$$

In general, an improvement with regard to one objective causes a deterioration of another. This corresponds to the situation that the objective functions are *at least partially conflicting*, meaning that they are conflicting at least in some regions of the search space. In this paper, we refer to such objective functions as *competing* objectives.

In SOP problems, we say that a solution with a smaller objective function value is better than one with a large objective function value. However, in MOP problems, there is no natural ordering in the objective space. For example, let the vector $[f_1, f_2]$ denote the objective function values in a two-objective MOP problem. The vector $[1, 1]$ can be said to be less than $[3, 3]$, but it is not obvious how to compare $[1, 3]$ to $[3, 1]$. Therefore, in MOP problems, there are usually (infinitely) many optimal compromise solutions that form the so-called Pareto optimal set. A decision vector θ^* is *(global) Pareto optimal* if there does not exist another decision vector θ such that $f_k(\theta) \leq f_k(\theta^*),$ for all $k = 1, ..., K$, and $f_p(\theta) < f_p(\theta^*)$ at least for one index p [8]. According to the definition of Pareto optimality, moving from one Pareto optimal solution to another necessitates trading off. Mathematically, every Pareto optimal solution is an equally acceptable solution to the MOP problem.

MOP methods mainly fall into two major categories, in which the original MOP problems are converted into SOP problems either by aggregating the objective functions into an overall objective function or by reformulating the problem with proposer constraints [8]. One of the most common engineering practices to solving MOP problems is the so-called weighting method. It reformulates the original MOP problem as a convex linear combination of the individual objectives with positive weights, $\gamma_k \geq 0$, such that $\sum_k \gamma_k = 1$. The task is, then, to minimize this overall objective function, i.e., to solve

$$\min_{\theta} \sum_{k=1}^{K} \gamma_k f_k(\theta),$$

where the weights, γ_k, reflect the significance of the individual objectives. One of the fundamental limitations of the weighting method is that the feasible objective space is not necessarily convex whereas the Pareto optimal solutions in the non-convex subset of Pareto optimal solutions cannot be found with the weighting method [9]. Another major drawback is that solving the problem with numerous weight vectors will give a limited number of Pareto optimal solutions. It is crucial that these solutions be spread in the objective space as uniformly as possible. The weighting method fails to meet this requirement and generates an irregular discretization of the (convex part of the) Pareto optimal set [8], [9].

III. Iterative Constrained Optimization (ICO)

Consider formulating the multi-objective optimization problem as a set of K single-objective optimization problems in the form

$$\begin{aligned} \min_{\theta} \quad & f_k(\theta) \\ \text{subject to} \quad & f_p(\theta) \leq \overline{f}_p,\ p = 1, ..., K, p \neq k, \end{aligned} \quad (2)$$

for all $k = 1, ..., K$, i.e., the minimization of one objective function, $f_k(\theta)$, with proper constraints on the other $(K-1)$ competing objectives, $f_p(\theta)$, where $\overline{f}_p$'s are the upper bounds for the objective function values to be attained.

Starting with a decision vector $\theta^{(1)}$ and the corresponding objective vector $f^{(1)} = (f_1^{(1)}(\theta), ..., f_K^{(1)}(\theta))$, the goal in ICO is to move into another decision vector $\theta^{(2)}$ yielding an objective function vector $f^{(2)}$ where at least one objective function $f_k, 1 \leq k \leq K$ attains a *considerably improved*

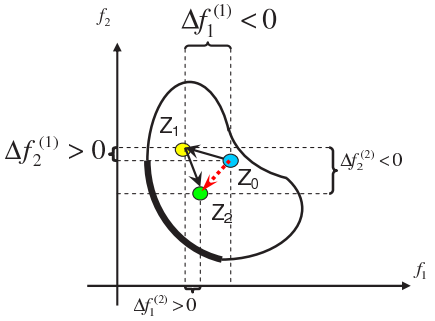

Fig. 1. A close neighborhood of the current compromise solution is searched for a better compromise solution in each iteration. Each intermediate step yields an iterate on a more favorable indifference curve.

value, while others are *possibly slightly degraded*. Consider the illustration in Fig. 1 for a two-objective MOP problem, where point Z_0 is the starting point, and a better compromise solution is being searched for. Because of the conflicting nature of the two objectives, it is possible to achieve a reduction in f_1 whenever f_2 is slightly increased. To compensate for the performance loss in f_2, f_1 should be slightly increased, and the best f_2 for the given f_1 value is searched for. Adjusting the constraints for the objectives by slightly perturbing their most recent values corresponds to setting the constraint bounds as follows:

$$\overline{f} = f + \delta. \tag{3}$$

where f is the vector of the most recent objective function values, and $\delta \in \Re^K$ is a vector of small perturbations added to f.

It is important to note that, in general, it is not necessarily correct that the resulting individual objectives are preferable to the initial ones, especially when there are many competing objectives: What is gained in one iteration can quickly be lost in the subsequent iterations. This is because when one objective is increased, some of the other objectives reduce whereas some may increase. In ICO, a step-size is taken if doing-so reduces a pre-selected cost function.

No single overall performance measure should be taken as a basis for a realistic comparison of the ICO-trained classifiers to more traditional classifiers. For many realistic classification tasks, we typically want to prevent bias towards any of the objectives and desire a symmetry across the many competing objective functions. The ICO method is promising for producing less *outliers* in the objective function space compared to the SOP algorithms with an overall objective function. One way to quantify the degree to which a classifier results in outlier objective values is to compare the upper and lower 5% or 10% percentile averages.

IV. An SOP-Based Approach

In [7], we proposed a structural MAP (SMAP) framework for estimating the n-gram probabilities using a hierarchical structure. The MOP approach that we develop in this paper has its roots in SMAP. For this reason, this section is devoted to an overview of SMAP. The connection between these two approaches is the subject of the next section. For the rest of the discussion in this paper, let the subscript "$\mathbb{A}$" denote the quantities that are estimated from the application-specific data, $\mathbb{A}$; the subscript "$\mathbb{S}$" denote the quantities that are estimated from the general domain data, $\mathbb{S}$; and the subscript "$\mathbb{T}$" denote the quantities that come from the unknown target distribution.

Given an appropriate prior information for the n-gram probabilities, $g(P_{\mathbb{S}}(\omega|h_\omega))$, estimated from $\mathbb{S}$, the target LM probabilities, $P_{\mathbb{T}}(\omega|h_\omega)$, can be found by maximizing the a posteriori probability given the observed text data, W. This corresponds to solving the following problem:

$$P_{\mathbb{T}}(\omega|h_\omega) = \arg\max_P \; P(P_{\mathbb{T}}|W) = \arg\max_P \; P(W|P_{\mathbb{A}}) g^\rho(P_{\mathbb{S}}). \tag{4}$$

where a non-negative number, ρ, is included in (4) to control the contribution from the involved datasets. It is implied by (4) that the greater ρ is, the more the prior information is depended upon, and hence the more the general domain dataset contributes while the more the influence of the application-specific data is discounted. This is useful since, for instance, the prior information might not be reliable and we may be interested in discounting its influence. Or else, the application-specific data may be too limited and we may want to increase the influence of the prior information by tuning ρ to a relatively large value.

In [7], the Dirichlet density was used to model the prior distribution. Dirichlet density was chosen because it is the conjugate prior density of the multinomial density. Let $\phi_{\mathbb{S}}(\omega|h_\omega)$ denote the hyper-parameters estimated from $\mathbb{S}$. The use of a Dirichlet density yields a hierarchical estimation formula for the hyperparameters as well as a closed-form solution for the n-gram probabilities as

$$P_{\mathbb{T}}(\omega|h_\omega) = \frac{c_{\mathbb{A}}(h_\omega, \omega) + \rho(\phi_{\mathbb{S}}(\omega|h_\omega) - 1)}{c_{\mathbb{A}}(h_\omega) + \sum_{\omega_i}[\rho(\phi_{\mathbb{S}}(\omega_i|h_{\omega_i}) - 1)]}. \tag{5}$$

The hyperparameters for the root nodes, i.e., for $\ell = 1$, are estimated as

$$\phi^1_{\mathbb{S}}(\omega|h_\omega) = 1 + \epsilon c(h^{(1)}_{\omega_i}),$$

where $0 < \epsilon \leq 1$ is a weighting coefficient. Note that when ϵ is small, the unigram observation frequencies are discounted. We obtain a recursive formula for estimating the hyperparameters associated with the node at the ℓ^{th} layer using the hyperparameters at the $(\ell-1)^{st}$ layer in a *top-down* manner as

$$\phi^\ell_{\mathbb{S}}(\omega|h_\omega) = c(h^{(\ell)}_{\omega_i}, \omega_i) + \rho(\phi^{\ell-1}_{\mathbb{S}}(\omega|h_\omega) - 1) + 1,$$

where $\phi^{\ell-1}_{\mathbb{S}}(\omega|h_\omega)$ is the hyperparameter of the parent node.

V. An MOP-Based Approach

In this section, we develop an MOP-based approach to the LM adaptation, which takes its roots in ICO method.

A. Formulation of LM Adaptation as an MOP Problem

Let $P_{\mathbb{A}_n}$ denote an n-gram LM estimated on the application-specific data, $\mathbb{A}$, and $P_{\mathbb{S}_n}$ denote an n-gram LM estimated on the general domain data, $\mathbb{S}$. The KL divergence of the target n-gram distribution, $P_{\mathbb{T}}(\omega|h_\omega)$, from an estimated n-gram model, P_n, (which is either $P_{\mathbb{A}_n}$ or $P_{\mathbb{S}_n}$) is given by

$$D[P_{\mathbb{T}}||P_n] = \sum_h P_{\mathbb{T}}(h) \sum_\omega P_{\mathbb{T}}(\omega|h_\omega) \log \frac{P_{\mathbb{T}}(\omega|h_\omega)}{P_n(\omega|h_\omega)}. \tag{6}$$

Maximizing the likelihood function is equivalent to minimizing the KL divergence of the target model from a model obtained from the data (here $\mathbb{A}$), i.e., to minimizing $D[P_{\mathbb{T}}||P_{\mathbb{A}_n}]$ [10]. In the meantime, the conjugate prior density is the distribution which minimizes the KL divergence of the target posterior model from the prior distribution. Minimizing the KL divergence of the target model from the prior model makes the target model spread out as uniformly as possible without contradicting the given information. Thus, the use of conjugate prior density implicitly minimizes $D[P_{\mathbb{T}}||P_{\mathbb{S}_n}]$. Based on these two results, the LM adaptation problem solved by the SMAP method can be posed as the following MOP problem:

Objective 1: The target n-gram probabilities should be at a minimum distance from the background model. By minimizing the KL divergence of the target model from the background model, given new facts, the new distribution is being chosen which is as hard to discriminate from the well-trained background model as possible.

Objective 2: The target n-gram probabilities should be at a minimum distance from the distribution obtained from limited application-specific data. By minimizing the KL divergence of the target model from a model estimated from the application-specific data, the new distribution is suitable to describe the source generating the application-specific data.

B. At-Least-Partially-Conflicting Objectives in LM Adaptation

The KL divergence $D[P_{\mathbb{T}}||P_{\mathbb{A}_n}]$ (or $D[P_{\mathbb{T}}||P_{\mathbb{S}_n}]$) is minimal (and equal to zero) when the target model, $P_{\mathbb{T}}$, is exactly the same as $P_{\mathbb{A}_n}$ (or $P_{\mathbb{S}_n}$) and any deviation from $P_{\mathbb{A}_n}$ (or $P_{\mathbb{S}_n}$) results in a non-zero KL divergence. Because $P_{\mathbb{T}}$ cannot be exactly the same as $P_{\mathbb{S}_n}$ and $P_{\mathbb{A}_n}$ at the same time, minimizing $D[P_{\mathbb{T}}||P_{\mathbb{S}_n}]$ and minimizing $D[P_{\mathbb{T}}||P_{\mathbb{A}_n}]$ are at least partially conflicting objectives. Since the goal in LM adaptation is to compensate for the insufficiency of the application-specific data set by using the general domain corpus, the best approach to reliably estimating the n-gram probabilities is to establish a compromise between these two KL divergences.

Let i denote an index from the index set $\{\mathbb{A}_1, \mathbb{A}_2, \mathbb{S}_1, \mathbb{S}_2\}$, where, $\mathbb{A}_n$ denotes the quantities of an n-gram model estimated from $\mathbb{A}$ and $\mathbb{S}_n$ denotes the quantities of an n-gram model estimated from $\mathbb{S}$. [1] Based on the results of the previous section, LM adaptation as solved by SMAP can be posed as an MOP problem in two different ways:

A sequential optimization approach

It is possible to find the probabilities of unigrams, bigrams, and so on, one after another. We refer to this approach as *sequential ICO for language modeling and adaptation*. For a bigram model, this means solving the following optimization problems:

$$\begin{aligned} &\min \quad D[P_{\mathbb{T}_1}||P_{\mathbb{A}_1}] \\ &\min \quad D[P_{\mathbb{T}_1}||P_{\mathbb{S}_1}] \end{aligned} \tag{7}$$

and then

$$\begin{aligned} &\min \quad D[P_{\mathbb{T}_2}||P_{\mathbb{A}_2}] \\ &\min \quad D[P_{\mathbb{T}_2}||P_{\mathbb{S}_2}] \end{aligned} \tag{8}$$

An all-in-once optimization approach

It is possible to compute the n-gram probabilities of all orders in only one optimization problem (instead of two). We refer to this approach as *K-objective ICO for language modeling and adaptation.* For a bigram model, this corresponds to solving the following multi-objective problem:

$$\begin{aligned} &\min \quad D[P_{\mathbb{T}_1}||P_{\mathbb{A}_1}] \qquad (9)\\ &\min \quad D[P_{\mathbb{T}_1}||P_{\mathbb{S}_1}] \qquad (10)\\ &\min \quad D[P_{\mathbb{T}_2}||P_{\mathbb{A}_2}] \\ &\min \quad D[P_{\mathbb{T}_2}||P_{\mathbb{S}_2}] \end{aligned}$$

[1]The divergences from unigram models as well as bigram models should be considered since backing-off is used when an unknown n-gram is observed during the recognition (test) stage.

C. Sequential ICO for Language Model Adaptation

Consider formulating the MOP formulation of the LM adaptation problem as a series of constrained optimization problems as in (7) and (8). In general, $P_{\mathbb{S}}$ is a considerably larger model than $P_{\mathbb{A}}$, and hence one would expect the two objectives to have different scales. The two objectives in each of these problems can be rewritten so that so that we can expect similar "distances". This can be achieved by averaging each KL divergence by the number of n-grams each model has.

To solve the two-objective optimization problems in (7) and (8), two optimization subproblems need to be solved one after another. These two subproblems are

$$\begin{aligned} (\textsc{Problem 1}) \min_{P_{\mathbb{T}_n}(\omega|h_\omega)} \quad & \frac{1}{N_{n_{\mathbb{A}}}} D[P_{\mathbb{T}_n}(\omega|h_\omega)||P_{\mathbb{A}_n}(\omega|h_\omega)] \\ s.t. \quad & \frac{1}{N_{n_{\mathbb{S}}}} D[P_{\mathbb{T}_n}(\omega|h_\omega)||P_{\mathbb{S}_n}(\omega|h_\omega)] \leq d_{\mathbb{S}} \end{aligned}$$

$$\begin{aligned} (\textsc{Problem 2}) \min_{P_{\mathbb{T}_n}(\omega|h_\omega)} \quad & \frac{1}{N_{n_{\mathbb{S}}}} D[P_{\mathbb{T}_n}(\omega|h_\omega)||P_{\mathbb{S}_n}(\omega|h_\omega)] \\ s.t. \quad & \frac{1}{N_{n_{\mathbb{A}}}} D[P_{\mathbb{T}_n}(\omega|h_\omega)||P_{\mathbb{A}_n}(\omega|h_\omega)] \leq d_{\mathbb{A}} \end{aligned}$$

where $n = 1, 2$, and $d_{\mathbb{S}}$ and $d_{\mathbb{A}}$ are the constraint bounds obtained by perturbing the most recent averaged KL divergences. For solving (PROBLEM 1), the constraint is incorporated into the optimization process using a Lagrangian multiplier λ_S. This results in the following Lagrangian function:

$$L(P_{\mathbb{T}_n}, \lambda_{S_n}) = D[P_{\mathbb{T}_n}||P_{\mathbb{A}_n}] + \lambda_{S_n} \cdot D[P_{\mathbb{T}_n}||P_{\mathbb{S}_n}] \tag{11}$$

By equating the gradient of this Lagrangian function with respect to $P_{\mathbb{T}_n}$ to 0 as

$$1 + \log \frac{P_{\mathbb{T}_n}}{P_{\mathbb{A}_n}} \quad + \lambda_{S_n} \cdot \quad 1 + \log \frac{P_{\mathbb{T}_n}}{P_{\mathbb{S}_n}} \quad = 0, \tag{11}$$

we obtain a closed form solution for the probabilities $P_{\mathbb{T}}(\omega|h_\omega)$ as

$$P_{\mathbb{T}}(\omega|h_\omega) = \frac{1}{Z(h_\omega)} [P_{\mathbb{A}}(\omega|h_\omega)]^{\frac{1}{1+\lambda_S}} [P_{\mathbb{S}}(\omega|h_\omega)]^{\frac{\lambda_S}{1+\lambda_S}} \tag{12}$$

where Lagrange multiplier λ_S is the only unknown and $Z(h_\omega)$ is a history-dependent normalization factor.

1) Finding the Component LM Weights:

There is no closed-form solution for λ but it can be found by iterative techniques. For this purpose, the constraint in (PROBLEM 1) can be rewritten as

$$d(\lambda_S) = D^{\lambda_S}[P_{\mathbb{T}}(\omega|h)||P_{\mathbb{S}}(\omega|h)] - d_{\mathbb{S}} = 0 \tag{13}$$

The zeros of this function can be obtained using the Newton's method for nonlinear equations [11] as follows:

$$\lambda_{\mathbb{S},k+1} = \lambda_{\mathbb{S},k} - \frac{d(\lambda_{\mathbb{S},k})}{\frac{\partial d(\lambda_{\mathbb{S},k})}{\partial \lambda_{\mathbb{S},k}}} \tag{14}$$

The derivative of $d(\lambda_{\mathbb{S},k})$ with respect to $\lambda_{\mathbb{S},k}$ turns out to be a function of the target probabilities $P_T^{\lambda_{S,k}}(\omega|h)$ and is computed as:

$$\begin{aligned} &\frac{\partial d^{\lambda_k}}{\partial \lambda_{\mathbb{S},k}} = \textstyle\sum_h P_{\mathbb{T}}(h) \times \\ &\textstyle\sum_\omega \left[1 + \log \frac{P_{\mathbb{T}}(\omega_h)}{P_{\mathbb{S}}(\omega|H)}\right] \cdot \quad \frac{1}{1+\lambda_{\mathbb{S},k}^2} \log \frac{P_{\mathbb{S}}}{P_{\mathbb{A}}} P_T^{\lambda_{\mathbb{S},k}}(\omega|h) \end{aligned} \tag{15}$$

The solution to (PROBLEM 2) is very similar: We just need to switch $P_{\mathbb{A}}$ and $P_{\mathbb{S}}$, and replace $\lambda_{\mathbb{S}}$ with $\lambda_{\mathbb{A}}$.

2) Algorithmic Implementation:

The algorithmic implementation of the sequential ICO for language modeling and adaptation is given in Table I. The algorithm starts with estimating the low-order n-gram probabilities for they are used as the history probabilities, $P_{\mathbb{T}}(h)$, for high-order n-grams. $D[P_{\mathbb{T}}(\omega|h)||P_{\mathbb{A}}(\omega|h)]$ is minimized by refining $\lambda_{S,k}$ so that $d(\lambda_{\mathbb{S},k})$ becomes roughly equal to 0. Once such a $\lambda_{\mathbb{S}}$ is found, it is used to calculate the target n-gram probabilities as in (12). After solving the first subproblem in this manner, the algorithm proceeds with solving the second subproblem. This iterative process stops when there is no progress in neither $D(P_{\mathbb{T}}||P_{\mathbb{A}})$ nor $D(P_{\mathbb{T}}||P_{\mathbb{S}})$. The relation $\lambda_{\mathbb{S}} = 1/\lambda_{\mathbb{A}}$ comes from the fact that to make the initial target probabilities of (PROBLEM 1) the same as those found after solving (PROBLEM 2), we should have $1/\lambda_{\mathbb{S}} = \lambda_{\mathbb{A}}/(1+\lambda_{\mathbb{A}})$.

D. K-Objective ICO for Language Model Adaptation

We choose to optimize each objective with constraints on the others. We then have four subproblems that will be solved one after another. For instance, one of these subproblems is

$$\begin{array}{ll} \min & D[P||P_{A_2}] \\ \text{subject to} & D[P||P_i] - d_i = 0, i \in \{\mathbb{A}_1, \mathbb{S}_1, \mathbb{S}_2\} \end{array}$$

For convenience, we restate this problem in an equivalent form as

$$\begin{array}{ll} \min & \lambda_{A_2}(D[P||P_{A_2}] - d_{A_2}) \\ \text{subject to} & D[P||P_i] - d_i = 0, i \in \{\mathbb{A}_1, \mathbb{S}_1, \mathbb{S}_2\} \end{array} \quad (16)$$

The reason for incorporating a scaling factor λ_{A_2} and a reference value d_{A_2} for the primary objective function (that is, the objective function of the problem in (16)) will shortly become clear. This problem can be solved by incorporating a Lagrange multiplier for each constraint. The resulting Lagrange function is

$$L(P, \lambda_i) = \sum_i \lambda_i (D[P||P_i] - d_i) \quad (17)$$

where $i \in \{\mathbb{A}_1, \mathbb{A}_2, \mathbb{S}_1, \mathbb{S}_2\}$. The target LM probabilities, $P_{\mathbb{T}}(\omega|h_\omega)$, are such that

$$\frac{\partial L}{\partial P_{\mathbb{T}}(\omega|h_\omega)} = \sum_i \lambda_i \frac{\partial D[P||P_i]}{\partial P_{\mathbb{T}}(\omega|h_\omega)} = \quad (18)$$

$$\sum_i \lambda_i \quad 1 + \log \frac{P_{\mathbb{T}}(\omega|h_\omega)}{P_i(\omega|h_\omega)} \quad = 0$$

Solving this problem, we obtain a log-linear interpolation (LLI) of the component LMs as

$$P_{\mathbb{T}}(\omega|h_\omega) = \frac{1}{Z(h_\omega)} \prod_i P_i(\omega|h_\omega)^{\lambda_i} \quad (19)$$

where $Z(h_\omega)$ is a history-dependent normalization factor. Note that the same form of $P_{\mathbb{T}}(\omega|h_\omega)$ is obtained irrespective of the subproblem being solved. (Remember that there are four subproblems to be solved for a bigram target LM.) This is the reason for the inclusion of λ_{A_2} and d_{A_2} in (16).

1) Algorithmic Implementation:

The algorithmic implementation of the proposed ICO method for LM adaptation is given in Table 1. The algorithm starts with some initial LM weights λ. These LM weights are refined in a manner that the subproblems in (16) are solved one after another.

In order to set some initial constraint bounds for each objective, all the KL divergences are evaluated. For the primary objective, the resulting KL divergence is discounted and set as the constraint bound while for the other objectives, the KL divergences are inflated and set as the constraint bounds.

The LM weights are found so that (i) the deviation of the primary objective from the reference value is minimized, and (ii) there is no deviation of the objectives in the constraints from the constraint bounds.

2) Finding the Component LM Weights:

There are no closed-form solutions for the Lagrange multipliers, λ_i, but they can be found by iterative techniques. Given the constraint bounds d_i, λ_i's should be such that the derivative of the Lagrange function with respect to λ_i vanish (by KKT optimality conditions), i.e., $\frac{\partial L}{\partial \lambda_i} = D[P||P_i] - d_i = 0, \forall i$.

Let $\mathcal{D}(\lambda_i)$ denote the deviation of a KL divergence from the respective constraint bound, i.e., $\mathcal{D}(\lambda_i) = D[P||P_i] - d_i$. It is a function of λ_i since $P(\omega|h_\omega)$ used for calculating $D[P||P_i]$ is obtained by (12). The zeros of this function can be obtained using the Newton's method for nonlinear equations as $\lambda_{i+1} = \lambda_i - \mathcal{D}(\lambda_i)/\mathcal{D}'(\lambda_i)$ where

$$\mathcal{D}'(\lambda_i) = \frac{\partial \mathcal{D}}{\partial \lambda_i} = \frac{\partial \mathcal{D}}{\partial P_T} \cdot \frac{\partial P_T}{\partial \lambda_i} \quad (20)$$

$$= \sum_{(\omega, h_\omega)} P_{\mathbb{T}}(h_\omega)(1 + \log \frac{P_{\mathbb{T}}(\omega|h_\omega)}{P_i(\omega|h_\omega)}) \cdot \left[(1 - \frac{\lambda_i}{(\sum_j \lambda_j)^2})(\log P_i(\omega|h_\omega)) P_{\mathbb{T}}(\omega|h_\omega) \right] \quad (21)$$

3) Adjusting the Constraint Bounds:

If it was possible to know the KL divergences that a desirable target LM would yield, this problem could be relatively straightforward. However, since we initially are not given these KL divergences, we choose to adjust the goals in an iterative manner.

The constraint bounds are obtained by perturbing the most recent KL divergences. The reference value for the primary objective is obtained by decreasing the most recent value (for instance, by multiplying by 0.9). The constraint bounds for the objectives in the constraints are obtained by slightly increasing their most recent values (for instance, by multiplying by 1.1). This is to ensure that the KL divergence of the target model from the primary objective is reduced while the others are tolerated to increase. The algorithmic implementation of the proposed ICO method for LM adaptation is given in Table II.

4) Finding and Validating the Step-Size:

The two objectives in the LM adaptation application interact in a complicated manner. It is hard to analyze which objective has what kind of impact on the speech recognition performance, For this reason, the perplexity is used as the utility function to judge the preferability of the new step found with the Armijo rule. That is, with the ICO method for the LM adaptation application, the step-size found by Armijo rule is validated and the LM weights are updated *only* in those cases which do not result in degradation in terms of perplexity.

VI. EXPERIMENTAL RESULTS

In this section, we report our experimental results comparing the two proposed LM adaptation approaches with log-linear interpolation and SMAP.

A. Experimental Results with SMAP

In our experiments we used the Wall Street Journal (WSJ0) dataset, which was designed to provide a wealth of general-purpose speech data with large vocabularies [12]. It has a set of over 1.6 million standardized sentences for LM training collected from 1987 until 1989. The test domain data is composed of WSJ newswire stories collected in November 1992. Unfortunately, n-gram LMs are extremely brittle even within domain when training and recognition involve moderately disjoint time periods, yet there is no LM adaptation set provided within the WSJ0 dataset.

TABLE I

SEQUENTIAL ICO ALGORITHM FOR LM ADAPTATION

I. Set $P_{\mathbb{T}}(h) = \frac{1}{N_1}$, where N_1 is the number of unigrams.
II. For n=1,2,...(i.e., unigrams, bigrams,...)
Repeat until no progress in $D(\cdot)$'s:
$\lambda_0 = 0.5$.
//Solve (PROBLEM 1):
ii. until $d(\lambda_{\mathbb{S},k}) \approx 0$.
Compute $\lambda_{\mathbb{S},k}$ from Equation (17).
iii. $\lambda_{\mathbb{S}} = \lambda_{\mathbb{S},k}$
iv. Compute $P_{\mathbb{T}}(\omega|h_\omega)$ from Equation (15).
v. $\lambda_{\mathbb{A}} = 1 \setminus \lambda_{\mathbb{S}}$.
//Solve (PROBLEM 2)
vi. until $d(\lambda_{\mathbb{A},k}) \approx 0$.
Compute $\lambda_{\mathbb{A},k}$ from Equation (17).
iii. $\lambda_{\mathbb{A}} = \lambda_{\mathbb{A},k}$
iv. Compute $P_{\mathbb{T}}(\omega|h_\omega)$ from Equation (15).
v. $\lambda_{\mathbb{S}} = 1 \setminus \lambda_{\mathbb{A}}$.
//New history probabilities for higher-order n-grams:
$P_{\mathbb{T}}(h_\omega) = P_{\mathbb{T}}(\omega|h_\omega)$.

TABLE II

K-OBJECTIVE ICO ALGORITHM FOR LM ADAPTATION

I. Start with some initial LM weights, $\lambda_i, i \in \{A_1, S_1, A_2, S_2\}$.
II. Repeat
For each $i \in \{A_2, S_2, A_1, S_1\}$ (in this order):
i. Find the initial constraint bounds:
$d_k^0 = 0.9D[P_T||P_i]$, if $k = i$,
$d_k^0 = 1.1D[P_T||P_i]$, if $k \neq i$
ii. We want to solve:
$\min \lambda_i(D[P||P_i] - d_i^0)$
subject to $D[P||P_k] - d_k^0 = 0, k \neq i$
iii. Compute $\lambda_k, k \in \{A_2, S_2, A_1, S_1\}$ so that
$\mathcal{D}(\lambda_i)$ is minimized and $\mathcal{D}(\lambda_k), k \neq i$ is 0.
iv. Evaluate P_T for these values of λ_ks.

Therefore, in [7], we constructed an artificial adaptation set to illustrate the use of the proposed LM adaptation framework. For doing so, we separated some of the sentences in the original text material as an artificial application-specific set. To make sure that this portion of the dataset is more relevant to the application than the rest, we selected those sentences which are rich in terms of the test set n-grams. After doing so, we trained an application-specific LM using this separated portion, which constituted 10% of the available text data. The remaining 90% was used to train a background model. Both models were trained simply using the maximum likelihood principle [13]. In [7], we demonstrated that this separation of adaptation material from the training data was appropriate to show the use of an LM adaptation technique.

1) Performance of SMAP LM Adaptation:

We performed experiments to find the effect of different parameters, ρ and ϵ, on the perplexity and WER.

1. Effect of ρ

First, we explored the effect of the forgetting factor, ρ. It is a parameter serving two purposes: First, it controls how much information each node inherits from its parent node. Second, it controls how much the prior information contributes in SMAP probability calculation.

The perplexity and WER are plotted as functions of ρ for different values of ϵ in Fig. 2 and Fig. 3, respectively. Fig. 2 shows that the perplexity first reduced as ρ increased, reaching its minimum at 47.65 at $\rho = 0.01$ for $\epsilon = 0.01$, and increased thereafter. Upon a comparison of Fig. 2 and Fig. 3, we observe that the perplexity and WER in general change in parallel except that the perplexity was higher when $\rho = 0.1$ than when $\rho = 10^{-6}$, yet the WER was lower when $\rho = 0.1$ than when $\rho = 10^{-6}$.

This behavior suggests, first, that when the contributions from the parent nodes were very small or very large, i.e., ρ was at extremes, the resulting LMs were not so right. When ρ was large, the node-specific information is dominated by the parent-specific information. When ρ was small, the nodes did not inherit significant information from parent nodes. Secondly, with an appropriate selection of ρ, the prior information helped improve the performance by contributing in the n-gram probability estimation.

2. Effect of ϵ

Secondly, we explored the effect of changing ϵ by fixing ρ. The parameter ϵ has an influence on the estimation of hyperparameters at the root nodes, which are then propagated to all other tree nodes. The perplexity and WER are plotted as functions of ϵ for different values of ρ in Fig. 4 and Fig. 5, respectively. Note that Fig. 4 and Fig. 5 are just other ways of looking at Fig. 2 and Fig. 3.

We observe that the perplexity slightly reduced as ϵ increased except for the case when $\rho = 0.1$. The same result held true for the change of WER as well. When ϵ is very small, the n-gram counts in the root nodes are excessively smoothed. This means that even when the frequencies of two unigrams differed by orders of magnitude, the corresponding hyperparameters were very close.

4. SMAP Adaptation with a Relevant and an Irrelevant Adaptation Dataset

Our final experimental study with SMAP concerned about the performance of the SMAP LM adaptation framework. For this reason, in addition to the relevant adaptation set,$\mathbb{A}_r$, we constructed an irrelevant set, called an irrelevant adaptation set and denoted as $\mathbb{A}_i$, which is expected *not-to-help* the recognition. Our experimental

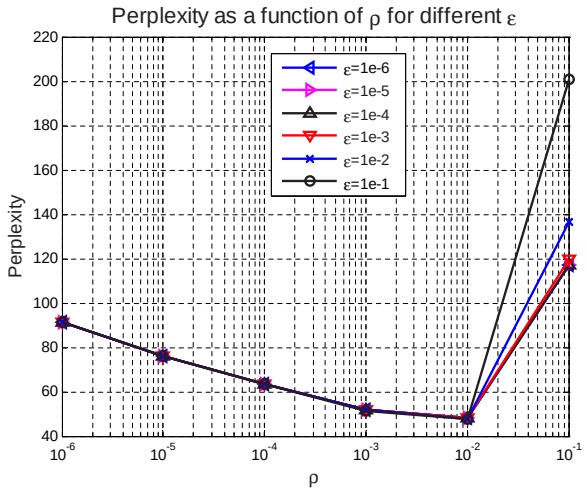

Fig. 2. The perplexity when ρ is changed for fixed ϵ.

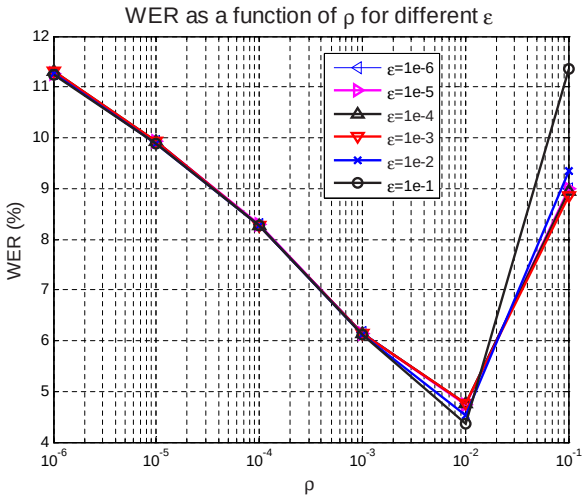

Fig. 3. The WER when ρ is changed for fixed ϵ.

findings are plotted in Fig. 6. The curves marked with "□" and "○" were obtained when $\mathbb{A}_i$ and $\mathbb{A}_r$ were used, respectively.

Several observations can be drawn from Fig. 6. First of all, using $\mathbb{A}_r$ reduced the WER considerably. SMAP LM adaptation framework was able to reduce WER to 4.37% using the entire $\mathbb{A}_r$, which was relatively 15.5% better than the WER obtained with $\mathbb{A}_r$. On the other hand, performing SMAP adaptation with $\mathbb{A}_i$ proved itself useless. This is because not only the ML modeling yielded better WER results than LM adaptation but also using more of $\mathbb{A}_i$ did not yield any improvement.

B. Experimental Results with ICO

The application-specific model had about 474K of bigrams while the background model had about 1,56 millions of bigrams. Since the computation of the KL divergence is demanding, at this stage we performed experiments using the bigram LMs. By performing LM adaptation, more than 1,6 millions of bigram probabilities are to be estimated.

1) Performance Evaluation of Sequential ICO for Language Modeling and Adaptation:

The average KL divergences of the target model from two independently models, one trained on the general domain data and the other trained on the application-specific data, are being minimized.

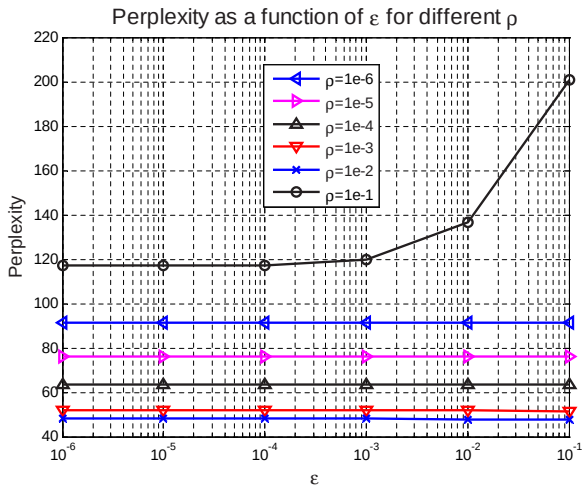

Fig. 4. The perplexity is slightly changed with ϵ for reasonable values of ρ.

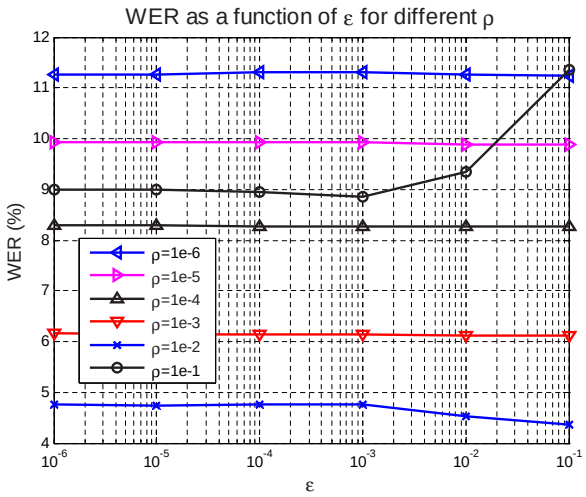

Fig. 5. The WER is slightly changed with ϵ for reasonable values of ρ.

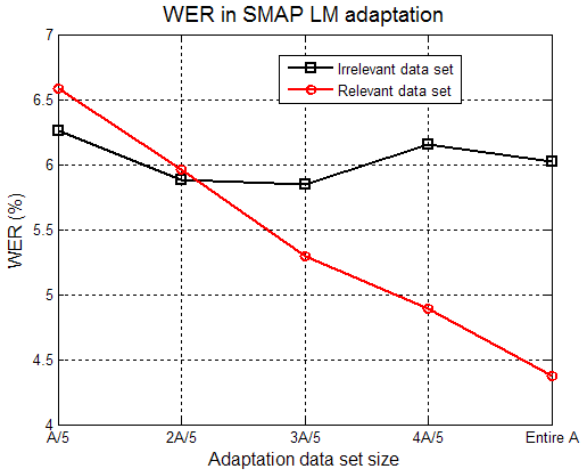

Fig. 6. The performance of SMAP LM adaptation when relevant adaptation data is made available.

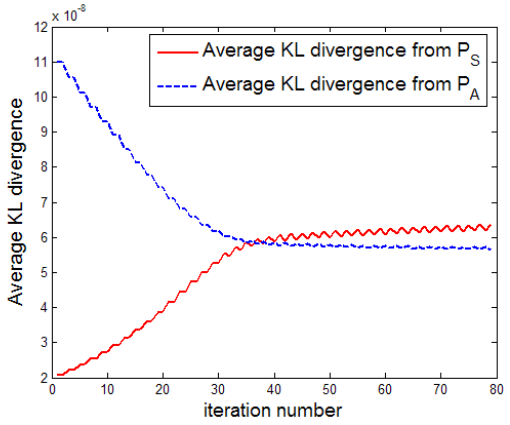

Fig. 7. The average KL divergences of the target model to the background model $P_{\mathbb{S}}$ and to the application-specific model $P_{\mathbb{A}}$ move towards a balance.

TABLE III

COMPARISON OF LM ADAPTATION METHODS OF INTEREST

	Bigram	
Model	PP	WER (%)
LLI	80.06	7.06
SMAP	80.53	7.19
MOP	79.18	6.91

The goal in ICO is then to reach a balance among these two distance measures. Our experimental result on the change of the average KL divergences is shown in Fig. 6. The constraints were obtained so that the most recent objective functions were increased by 1%. As shown in Fig. 7, the average KL divergences moved towards a better balance. Moreover, each objective followed a zigzag pattern throughout the iterative process. This is because the two objectives attempt to modify the target model probabilities to be close to their respective model.

2) Performance Evaluation of K-objective ICO for Language Modeling and Adaptation:

We then performed experiments on model perplexity and ASR word error rate (WER) to compare the performance of the proposed LM adaptation framework with the SMAP method and Klakow's LLI model. Our experimental results are reported in Table III. In our ASR experiments, we used the same design as in [7]. The SMAP model was trained with $\rho = 0.0001$ for the propagation of hyper-parameters, $\epsilon = 0.01$, and $\rho = 0.1$.

As shown in Table III, the MOP-based approach is superior to SMAP by 3.8% in terms of relative reduction in WER in ASR experiments. This is because MOP leaves more flexibility in finding the n-gram estimates while SMAP attempts to merge the conflicting goals a priori in an overall function. In the meantime, in comparison to Klakow's LLI, the MOP solution performs relatively 2.1 % better in terms of WER. Although both had the same form of the solution, the distinction was in the way the LM weights (i.e., the Lagrange multipliers) were estimated. Although the improvements do not seem significant, the major gains in solving the LM adaptation problem with MOP are twofold: (i) Upon observing the behavior of each objective, we have full freedom to tune the system into different operating points to meet different requirements. (ii) Meantime, by observing each objective, we can easily avoid extremes, i.e., the cases that the target LM is too dependent on the application specific data or on the general domain data.

VII. CONCLUSION AND FUTURE RESEARCH

In this paper, we described a multi-objective programming based method, called iterative constrained optimization (ICO), where we formulated the original multi-objective programming problem as an iterative process of the optimization of individual objectives with proper constraints on the remaining competing objectives.

In this work, we considered language model (LM) adaptation, where a background LM is adapted to an application domain so that the adapted LM is as close as possible to both the background model and the application domain data. For this, we first considered an SOP-based approach. We, then, formulated the original problem as an MOP problem and solved it using the ICO method. Finally, we compared the performance of the SOP- and MOP-based solutions for each of the applications. Our experimental results demonstrated that the ICO method achieves a better balance among the design objectives. Furthermore, the ICO method gave an improved system performance.

We believe ICO is well suited for many problems in a wide range of applications. We will further this line of research in several theoretically rich directions. One of the first ones is about the automatic means to infer the constraint bounds. We have observed in our experiments that different settings for the constraint bounds directly translate into different end results, as our intuition also suggests. Based on our experience, we foresee that the constraint bounds can be set by analyzing the sensitivity of the problem on the changes of the individual objectives.

REFERENCES

[1] T. M. Mitchell, *Machine Learning*, McGraw-Hill, 1997.
[2] G. P. Liu and V. Kadirkamanathan, "Learning with Multi Objective Criteria," in *Proc. of IEE Conference on Artificial Neural Networks*, 1995, pp. 53–58.
[3] A. P. Braga, R. H. C. Takahashi, M. A. Costa, and R. A. Teixeira, *Multi-Objective Machine Learning*, chapter Multi-Objective Algorithms for Neural Networks Learning, pp. 151–171, Springer, Berlin Heidelberg, 2006.
[4] T. Suttorp and C. Igel, *Multi-Objective Machine Learning*, chapter Multi-objective optimization of support vector machines, pp. 199–220, Springer, Berlin Heidelberg, 2006.
[5] J. R. Bellegarda, "Exploiting latent semantic information in statistical language modeling," *In Proceedings of IEEE*, vol. 88, pp. 1279–1296, 2000.
[6] S. Yaman and C.-H. Lee, "An Iterative Constrained Optimization Approach to Classifier Design," in *Proc. of the IEEE Conference on Acoustics, Speech, and Signal Processing*, Toulouse, France, 2006.
[7] S. Yaman, J.-T. Chien, and C.-H. Lee, "Structural Bayesian Language Modeling and Adaptation," Antwerp, Belgium, 2007.
[8] K. Miettinen, *Nonlinear Multiobjective Optimization*, Springer, 1999.
[9] I. Das and J. Dennis, "Closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems," Tech. Rep. 96–36, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1996.
[10] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2007.
[11] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, 1999.
[12] D. B. Paul and J. M. Baker, "The design for the wall street journal based CSR corpus," in *the Proc. of Internation Conference of Spoken Language Processing*, Banff, Alberta, Canada, September 1992.
[13] C. D. Manning and H. Schtze, *Foundations of Statistical Natural Language Processing*, The MIT Press, 1999.

Sibel Yaman is a Ph.D. candidate in School of Electrical and Computer Engineering at Georgia Institute of Technology, Atlanta, GA, where she earned her M.S. degree in 2004. She received her B.S. degree in electrical and electronic engineering from Bilkent University, Ankara, Turkey in 2002. She is a recipient of the Microsoft Research Graduate Fellowship for 2006-2007. She has been selected as a Best Student Paper Award finalist in ICASSP 2006. Her research interests include automatic language identification, multi-objective optimization techniques for pattern classification, and language modeling.

Chin-Hui Lee is a professor at School of Electrical and Computer Engineering, Georgia Institute of Technology in Atlanta, Georgia. Dr. Lee received the B.S. degree in Electrical Engineering from National Taiwan University, Taipei, Taiwan in 1973, the M.S. degree in Engineering and Applied Science from Yale University, New Haven, CT., in 1977, and the Ph.D. degree in Electrical Engineering with a minor in Statistics from University of Washington, Seattle, WA., in 1981.

After graduation in 1981, Dr. Lee joined Verbex Corporation, Bedford, MA., and was involved in research on connected word recognition. In 1984, he became affiliated with Digital Sound Corporation, Santa Barbara, where he engaged in research and product development in speech coding, speech synthesis, speech recognition and signal processing for the development of the DSC-2000 Voice Server. Between 1986 and 2001, he was with Bell Laboratories, Murray Hill, New Jersey, where he became a Distinguished Member of Technical Staff and Director of the Dialogue Systems Research Department. His research interests include multimedia communication, multimedia signal and information processing, speech and speaker recognition, speech and language modeling, spoken dialogue processing, adaptive and discriminative learning, biometric authentication, information retrieval, and bioinformatics. His research scope is reflected in ”Automatic Speech and Speaker Recognition: Advanced Topics”, published by the Kluwer Academic Publishers in 1996. From August 2001 to August 2002 he was a visiting professor at School of Computing, The National University of Singapore. In September 2002, he joined the Faculty of School of Electrical and Computer Engineering at Georgia Institute of Technology.

Prof. Lee has participated actively in professional societies. He is a member of the IEEE Signal Processing Society, Communication Society, Computer Society, and the International Speech Communication Association. In 1991-1995, he was an associate editor for the IEEE Transactions on Signal Processing and Transactions on Speech and Audio Processing. During the same period, he served as a member of the ARPA Spoken Language Coordination Committee. In 1995-1998 he was a member of the Speech Processing Technical Committee of the IEEE Signal Processing Society (SPS), and later became the chairman of the Speech TC from 1997 to 1998. In 1996, he helped promote the SPS Multimedia Signal Processing (MMSP) Technical Committee in which he is a founding member.

Dr. Lee is a Fellow of the IEEE, and has published more than 300 papers and 25 patents on the subject of automatic speech and speaker recognition and multimedia information Processing. He received the SPS Senior Award in 1994 and the SPS Best Paper Award in 1997 and 1999, respectively. In 1997, he was awarded the prestigious Bell Labs President’s Gold Award for his contributions to the Lucent Speech Processing Solutions product. In 2006, he received the IEEE signal Processing Society’s Technical Achievement Award. Dr. Lee is also a frequent invited speaker in international communities. In 2000, he was named one of the six Distinguished Lecturers by the IEEE Signal Processing Society. In 2007, he was named one of the two inaugural Distinguished Lecturers for the International Speech Communication Association.