CHAPTER 1

ELEMENTS OF ITERATIVE TREATMENT LEARNING

1.1 Overview

Much previous and ongoing work has been done on treatment learners. There has also been some work done with ITL. In this section we will give an overview of the main components of ITL.

The tar family of learners were born with [**?**].

1.2 Details

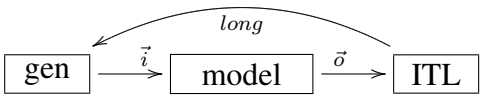
This section goes into the details of ITL.

The software engineering process can be added by the use of models. The model might be a cost/benefit model, like DDP (described in §**??**) or a quality model like COQUALMO CITE. For the purpose of this discussion we assume three operation qualities of models

1. has a well-defined input vector, $\vec{i}$, that can be supplied by the user

2. is a black box, ie its internals can not be modified

3. has a well-defined output vector, $\vec{o}$, that can be accessed by ITL



But what if the model has many uncertain inputs?

1.2.1 Funnel theory

Funnel theory provides a good theortical background for our expectation that in complex domains, there will likely be a small subset of possible decisions that control the behavior of models.

1.2.2 The tar family of learners

ITL utilizes the treatment learner *tar3*. Tar3 is the third generation treatment learner developed by Menzies *et. al.* and is explained in detail in §**??**. Although not done yet, it might be interesting to use other learners like those in §**??** with ITL to see if there are any performance differences.

1.2.3 Extreme sampling

Most discretezition policies deal with discretizing the input attributes. The choice of learning technique is often dictated by the form of the output class. Some learners, M5' or linear regression, can produced theories that predict for continuous classes, while others, Naive Bayes or J4.8, produced theories that predict for discrete classes.

ITL uses extreme sampling to map multiple output parameters to a single discrete output class. First the output vector has to mapped to a single numeric. We chose to use normalized euclidean distance as our mapping function. First each output value is normalized by its maximum value.

$$\vec{v} \xrightarrow{euclidean} [0-1] \xrightarrow{dualclass} 0, 1$$