Learning to approximate unknown misclassification costs when the training sample is biased

Ilkka Autio
Department of Computer Science
P. O. Box 26 (Teollisuuskatu 23)
FIN-00014 University of Helsinki, Finland

10th February 2002

Abstract

Supervised learning is usually associated with problems that have symmetric and known misclassification costs. We present selected ideas of how to handle situations in which these assumptions do not hold.

In this report we had to ignore the fairly unremarkable classifier learning methods of the primary source article [2] in order to focus on the hard issue of cost estimation.

In order to make this presentation reasonably self-contained, we also outline some of the elementary mathematics associated with the problem domain. We have tried to provide enough detail for the readers interested in implementing things. So called "CEO-level" of understanding can be gained by skipping the mathematics and by concentrating on the essay parts and memorizing the key terms. Technical details bordering on the tedious are provided in the footnotes for the benefit of more advanced readers interested in the theoretical issues.

The exercise questions will be "CEO"-level. They can be solved by common sense and will not involve proofs or complicated calculations.

1 The standard setting of supervised learning and regression

In this section, we review the basics of the standard setting of supervised learning. If the reader is already familiar with the basics, then at least the first subsection could be skipped.

1.1 Assumptions about the data

In a standard supervised learning task we have random pairs of the form $(\mathbf{X}, D)$, where $\mathbf{X}$ is a random vector or a collection of random variables and D is a special random variable (or a collection of them). We want to predict unobserved values of D based on observed values of $\mathbf{X}$. Prediction becomes possible if $\mathbf{X}$ and D are statistically dependent on each other and we have an algorithm that can "learn" the dependencies.

The algorithm doing the learning is given a training set S of examples:

$$S = \{(\mathbf{x}_1, d_1), \ldots, (\mathbf{x}_N, d_N)\}.$$

The pairs $(\mathbf{x}_i, d_i)$ are thought of as observed instances of the random pair $(\mathbf{X}, D)$.

It is usually assumed that the pairs are *independent and identically distributed* (i.i.d). Independece means that there are no dependencies between different pairs. Taken literally, the second assumption means that there is a "true" unchanging distribution from which the examples are drawn and that there are no important hidden variables tied to time. It is further assumed that the fully taught system is employed in identical circumstances.

Example 1: If vectors $\mathbf{x}_i$ represent measurable physical and social attributes of adult Finns and d_i represents height, then the i.i.d. assumption makes sense. It is safe to assume that the discovered dependencies stay stable for a few years.

Example 2: Assume the attributes are as above, but in addition to Finns we sample the population of Afghanistan in the training phase. After learning, we intend to predict the height of Finns in Helsinki. The i.i.d. assumption is now unsafe.

1.2 From Bayes minimum risk to practical measures of performance

After the training phase, we have a classifier $f(\mathbf{x})$ which maps observed vectors $\mathbf{x}$ into predictions of the unobserved value d. Mistakes occur when $f(\mathbf{x}) \neq d$. It is usually assumed that all mistakes are equal, but this is not necessarily the case.

Example 3: Consider an automatic luggage screening machine f at a major airport. The vector $\mathbf{x}$ represents the measurements we get when luggage is X-rayed in the machine. Now $f(\mathbf{x}) = 1$ if a bomb is predicted and zero otherwise. If $d = 1$ (a bomb!), then every type $f(\mathbf{x}) = 0$ mistake is serious. False alarms are clearly less serious.

To reflect the relative seriousness of different mistakes, we can associate a different penalty C with each type of mistake. This leads to the concept of Bayes minimum risk classifier f^*:

$$f^*(\mathbf{x}) = \arg\min_k E[C(D,k,\mathbf{x})|\mathbf{X} = \mathbf{x}] = \arg\min_k \sum_d C(d,k,\mathbf{x})P(D = d|\mathbf{X} = \mathbf{x}).$$

In english: The minimum risk classifier predicts the class k that results in the minimum expected penalty. Because $\mathbf{x}$ is known to the classifier, the expectation is taken over conditional class probabilities. $C(d,k,\mathbf{x})$ is the penalty associated with the situation in which the correct answer is d, the predicted answer is k and the observed vector is $\mathbf{x}$. If $d = k$ then the penalty is zero in the standard setting.

It is not hard to prove that the Bayes minimum risk classifier minimizes the expected penalty (also called the “Bayes risk functional” - for a proof corresponding to the standard setting, see [1]):

$$R(f) = E[C(D, f(\mathbf{X}), \mathbf{X})].$$

If all mistakes are equal, then $f^*(\mathbf{x}) = \arg\max_k P(D = k|\mathbf{X} = \mathbf{x})$ (i.e., the optimum classifier then selects the most probable class as required by common sense).

Proof:

Let $C(d,k,\mathbf{x}) = c > 0$ if $d \neq k$ for all $\mathbf{x}$. If $d = k$, then $C(d,k,\mathbf{x}) = h < c$ for all $\mathbf{x}$. Now:

$$f^*(\mathbf{x}) = \arg\min_k \sum_d C(d,k,\mathbf{x})P(D = d|\mathbf{X} = \mathbf{x})$$

$$= \arg\min_k (C(k,k,\mathbf{x})P(D=k|\mathbf{X}=\mathbf{x}) + \sum_{d\neq k} C(d,k,\mathbf{x})P(D=d|\mathbf{X}=\mathbf{x})$$

$$= \arg\min_k (hP(D=k|\mathbf{X}=\mathbf{x}) + c\sum_{d\neq k} P(D=d|\mathbf{X}=\mathbf{x}))$$

$$= \arg\min_k (hP(D=k|\mathbf{X}=\mathbf{x}) + c(1-P(D=k|\mathbf{X}=\mathbf{x})))$$

$$= \arg\min_k (c + (h-c)P(D=k|\mathbf{X}=\mathbf{x}))$$

$$= \arg\min_k (h-c)P(D=k|\mathbf{X}=\mathbf{x}) = \arg\max_k P(D=k|\mathbf{X}=\mathbf{x}).$$

If mistakes are not equal, we do not necessarily go for the most probable class. Consider the following extreme example:

Example 4:

We have two classes. The costs are: $C(0,0,\mathbf{x}) = 0$, $C(1,1,\mathbf{x}) = 0$, $C(0,1,\mathbf{x}) = 1$ and $C(1,0,\mathbf{x}) = 0$. In this case we see that we should always predict zero regardless of the probabilities.

The important points are:

- If all mistakes are equal, modeling probabilities is everything.
- If the costs wary wildly, then the importance of the costs approaches the importance of the probabilities.

In practice, the probabilities are not known, but are either not explicitly used (e.g., decision trees and support vector machines) or are estimated from the data (e.g., naive Bayes and Bayesian networks).

As the final note on the subject of penalties, we point out that the Bayes optimum criterion is by no means the only possible optimality criterion. It represents the optimum for *risk neutral* behavior. *Risk aversive* decision makers could use minimax instead (always making the decision such that the worst possible penalty is minimized).

Risk neutral behavior is acceptable if we want to maximize profits (or minimize costs) in the long run, but can tolerate serious temporary losses now and then. Risk aversive behavior is reasonable in the case of a small time investor making decisions. Such an investor might not be able to take significant losses at all.

1.3 Regression

Sometimes we need to do function approximation instead of classification. In this setting the random pair $(\mathbf{X}, D)$ consists of a random vector $\mathbf{X}$ of real numbers and D is a real valued random variable.

Example 5: In [2] there is a subproblem of predicting how much money a person will donate to a charity fund. One can try to solve this problem by interpolating based on known donations. One has a set of pairs $(\mathbf{x}_i, d_i)$ in which d_i is a scalar corresponding to money. A solution can be found by fitting a reasonable function (a surface or a plane) to the known pairs and by assuming that the fitted function gives a good estimate of d based on $\mathbf{x}$.

For brevity, we consider just linear functions. Suppose $\mathbf{x}$ is a column vector and $f(\mathbf{x}) = \mathbf{x}^T\mathbf{w} + b$. The point is to estimate $\mathbf{w}$ and b such that $f(\mathbf{x})$ is a good estimate of d. If the known pairs are not approximately on a plane, then the estimate can be arbitrarily bad. This can often be remedied, by doing piecewise approximation, i.e., by trying to find subsets of the pairs that are almost on a plane.

First, we simplify the representation of f as follows:

$$f(\mathbf{x}) = \mathbf{x}^T\mathbf{w} + b = \begin{pmatrix} \mathbf{x}^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{w} \\ b \end{pmatrix}.$$

Next, we set:

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^T & 1 \\ \vdots & \vdots \\ \mathbf{x}_N^T & 1 \end{pmatrix}, \mathbf{a} = \begin{pmatrix} \mathbf{w} \\ b \end{pmatrix}, \mathbf{d} = \begin{pmatrix} d_1 \\ \vdots \\ d_N \end{pmatrix}.$$

For a perfect fit to known data, $\mathbf{a}$ should satisfy:

$$\mathbf{X}\mathbf{a} = \mathbf{d}.$$

Because $\mathbf{X}$ is usually not a square matrix (cannot be inverted), the system usually does not have an exact solution. The "best possible fit", or "least squares", solution is:

$$\mathbf{a}^* = \arg\min_{\mathbf{a}} ||\mathbf{X}\mathbf{a} - \mathbf{d}||^2.$$

Assuming that the square matrix $\mathbf{X}^T\mathbf{X}$ can be inverted, the solution is:

$$\mathbf{a}^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{d}.$$

If inversion is still impossible or numerically unstable, there exists a solution based on *Moore-Penrose -pseudoinverse* of $\mathbf{X}$ (see [1]). The inversion solution can then be seen as a special case of the pseudoinverse.

Proof (when $\mathbf{X}^T\mathbf{X}$ is invertible):

We minimize the expression by differentiation w.r.t. $\mathbf{a}$. The zeros of the derivative must be local minima, because the expression cannot have local maxima (and in this case local minima are also global minima).

$$\begin{aligned} ||\mathbf{Xa} - \mathbf{d}||^2 &= (\mathbf{Xa} - \mathbf{d})^T(\mathbf{Xa} - \mathbf{d}) \\ &= \ldots \\ &= \mathbf{a}^T\mathbf{X}^T\mathbf{Xa} - 2\mathbf{d}^T\mathbf{Xa} + \mathbf{d}^T\mathbf{d}. \end{aligned}$$

We then differentiate the parts:

$$\frac{d}{d\mathbf{a}}\mathbf{d}^T\mathbf{d} = 0.$$

$$\frac{d}{d\mathbf{a}}\mathbf{d}^T\mathbf{Xa} = \mathbf{X}^T\mathbf{d}.$$

And the tedious one:

$$\frac{d}{d\mathbf{a}}\mathbf{a}^T\mathbf{X}^T\mathbf{Xa} = 2\mathbf{X}^T\mathbf{Xa}.$$

The last one follows from:

$$\frac{\partial}{\partial a_k}\mathbf{a}^T\mathbf{X}^T\mathbf{Xa} = \frac{\partial}{\partial a_k}||\mathbf{Xa}||^2 = \frac{\partial}{\partial a_k}\sum_i(\sum_j X_{ij}a_j)^2$$

$$= \sum_i 2(\sum_j X_{ij}a_j)\frac{\partial}{\partial a_k}\sum_j X_{ij}a_j = \sum_i 2(\sum_j X_{ij}a_j)X_{ik} = 2\mathbf{a}^T\mathbf{X}^T\mathbf{X}_{\bullet k},$$

which is the k:th column of the row vector $2\mathbf{a}^T\mathbf{X}^T\mathbf{X}$. Hence, that row contains the partial derivatives. We transpose the row, because we want column vectors. Because the derivative has just a single zero point, the corresponding local minimum must be the global minimum as well.

Finally, we set the derivative equal to zero:

$$2\mathbf{X}^T\mathbf{Xa} - 2\mathbf{X}^T\mathbf{d} = 0,$$

from which we get:

$$\mathbf{a} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{d}.$$

The matrix $\mathbf{X}^T\mathbf{X}$ has close ties to the estimate of the covariance matrix of the data. Assuming that the data has been normalized, $E[\mathbf{x}] = \mathbf{0}$, we have $Cov(x) = E[\mathbf{x}\mathbf{x}^T]$. The practical estimate of $Cov(\mathbf{x})$ is the sample average:

$$\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^T = \frac{1}{N}\mathbf{X}^T\mathbf{X}.$$

Hence, the reader should not be confused if the least squares solution is sometimes formulated on the basis of $Cov(\mathbf{x})$.

The least squares fitting of a general linear model is often called "Ordinary Least Squares" (OLS) in the statistics literature. The model fitting process is called "regression" for historical reasons.

2 The probit sample selection rule

In the later sections we introduce a procedure with which we can estimate unknown misclassification costs. This procedure uses a simple linear binary classifier called *the probit sample selection rule.*

Suppose we would like to use a general linear model in estimating probabilities:

$$P(Y = 1|\mathbf{w}) = \mathbf{w}^T\mathbf{c}.$$

The problem is that $\mathbf{w}^T\mathbf{c}$ is not bounded and probabilities have the requirement $P(Y = 1|\mathbf{w}) \in [0, 1]$. Straightforward linear models will not do, meaning that we must fudge them a little:

$$P(Y = 1|\mathbf{w}) = F(\mathbf{w}^T\mathbf{c}).$$

The function F must limit the output of $\mathbf{w}^T\mathbf{c}$ to the interval $[0, 1]$. We choose F to be the cumulative distribution function of the standard normal distribution $N(0, 1)$ [1]. This choice is called the *probit model.*

[1] Recall that the cumulative distribution function of a probability distribution tells the probability of the random variable having a value smaller than the argument. Example: $F(0) = 0.5$ for the standard normal distribution, because the normally distributed variable is smaller than 0 with the probability of 0.5.

The problem of estimating $\mathbf{c}$ is different from the problem of least squares estimation of the previous section. The presence of F introduces nonlinearities in the model and our model answers are qualitatively different. For a subset of the data, we have $Y = 1$. The Y is a binary attribute of the data. The important thing is that Y does not have to be an explicit attribute. We could, for example, define $Y = 0$ for rows lacking certain attribute values. Such a Y could be called an *observation indicator*.

We now suppose that Y is an observation indicator. Additionally, we suppose there is a linear noisy observation rule: $Y = 1$ iff $\mathbf{w}^T\mathbf{c} + U > 0$, where the noise term U is $N(0,1)$. We observe that $P(Y = 1|\mathbf{w}) = P(\mathbf{w}^T\mathbf{c} + U > 0|\mathbf{w})$. Now $\mathbf{w}^T\mathbf{c} + U > 0$ holds iff $U > -\mathbf{w}^T\mathbf{c}$. Because U is $N(0,1)$ we have $P(U > -\mathbf{w}^T\mathbf{c}) = P(U < \mathbf{w}^T\mathbf{c}) = F(\mathbf{w}^T\mathbf{c})$. Hence, if the observation rule is true, then the probit model is also literally true (it is not just a model).

Next, we would like to have a working probit probability model for the conditional probability of observing an example given the collection of attributes $\mathbf{w}$ of the example. The model has the vector of parameters $\mathbf{c}$ that should be estimated. The estimation can be based on the *maximum likelihood principle.*

The maximum likelihood principle is quite simple. We write out a parametric expression for the probability of the data and then try to maximize that probability by adjusting the parameters. The justification is that it is likely that our data is typical - we adjust the parameters to make our data more probable. In practice, we often maximize the logarithm of the probability instead (the calculations tend to get simpler).

In our application, the log-likelihood of the data is:

$$\log \prod_i P(Y = y_i|\mathbf{w}) = \sum_i \log P(Y = y_i|\mathbf{w}) = \sum_{i:y_i=1} \log P(Y = 1|\mathbf{w}) + \sum_{i:y_i=0} \log P(Y = 0|\mathbf{w})$$

$$= \sum_{i:y_i=1} \log F(\mathbf{w}^T\mathbf{c}) + \sum_{i:y_i=0} \log(1-F(\mathbf{w}^T\mathbf{c})) = \sum_i y_i \log F(\mathbf{w}^T\mathbf{c}) + \sum_i (1-y_i) \log(1-F(\mathbf{w}^T\mathbf{c})).$$

The log-likelihood function is concave (every local maximum is also a global maximum), which means that the $\mathbf{c}$ maximizing the expression can be solved by hill-climbing (e.g., Newton-Rhapson).

The important points are:

- If there is a linear rule that approximates the selection rule well, we can find that approximation from the data.
- The probabilities were necessary in the rule estimation process, as the process was based on the maximum likelihood principle.

- The selection problem is a classification problem. For now, we chose the probit-rule, because of certain theoretical properties we need later.

3 Unknown non-constant costs under a biased setting

In the previous sections we had misclassification costs that were entirely determined by the actual class and the predicted class. It is also possible to have a scenario in which the feature vector $\mathbf{x}$ affects the costs directly, i.e., $\mathbf{x}$ affects the costs $C(i, j, \mathbf{x})$. Below, we give an example resembiling the problem setting in [2] except for some simplifications.

Example 6:

Let us first transform the Bayes risk minimization criterion into an equivalent benefit maximization criterion. This can be done by assuming that benefits have negative costs (or by changing the min operation into a max operation).

In this example, we operate a charity fund and want to attract donations. Potential donors are mailed an invitation to donate. Mailing invitations costs us money. We have a database of previous donors and non-donors with which we can attempt to identify likely donors and predict how much they donate. For each database entry, we have a feature vector $\mathbf{x}$. Additionally, we have the previous donation size as an attribute for the previous donors. In case of the non-donors, this attribute is missing.

If we predict that a person will not donate, then we just ignore him and save the mailing costs. For the moment, we assume that ignoring a person does not cost us anything, i.e., we avoid the question of how much could have been gained had we not ignored him. The assumption is valid as long as we are only interested in raw money - things could get different if we wanted to penalize "lost opportunities".

We have two classes. Persons of class 1 donate money if asked to. Mailing an invitation costs 1 unit of money. The benefits are as follows [2]:

	class = 0	class = 1
predict 0	0	0
predict 1	-1	f(x)-1

[2] If you want to conceptualize this as risk minimization, just reverse the signs of the costs

From the table above we see that the benefit $-C(1, 1, \mathbf{x})$ is a function of $\mathbf{x}$ and not a simple constant. The problem can be broken into two subproblems:

- Making good predictions.
- Learning to approximate f.

The problem of making good predictions is well understood if the costs are known. If one can estimate $P(D = 1|\mathbf{X} = \mathbf{x})$ well for each $\mathbf{x}$, then one can plug the estimates and the costs directly into the Bayes minimum risk classifier to obtain a prediction. This procedure is sometimes called *direct cost minimization* in data mining literature [2].

The problem of approximating f is *not* well understood in this context. One can guess that f could be estimated by regression. What might not be obvious, is that ordinary regression would be biased in our example. In the training stage, donation values are only available in the case of previous donors. Further, the regression problem must be solved first, meaning that we cannot use a bayes classifier at this stage.

The regression problem is solved, if for every new $\mathbf{x}$, we can calculate an approximation of $f(\mathbf{x})$ regardless of the class of $\mathbf{x}$ or the potential predictions. The bias results from the fact that the regression has to be based on a non-random subpopulation of the previous donors. For example, the donations in that subpopulation could be lower than in the subpopulation that would have donated but was ignored earlier. Further, removing the bias from the regression problem allows us to speculate how much a person would have donated had he not turned down the offer[3].

The biased approximation problem of the type just described can be solved under certain circumstances. The solution is called the *Heckman 2-step procedure* that was first introduced in [3]. Heckman was later awarded the Nobel prize in economics for inventing the procedure. In the next section, we try to explain the basics of the procedure. The reader should not worry too much about the details - if the procedure had been obvious and could be easily explained, it probably would not have earned Heckman the Nobel prize.

[3]The last question might seem insane at first: If a person turns down an offer, does it not mean that he decided to donate precisely zero units of money ? Yes and no. The decision to refuse an offer does mean a donation of zero units from the viewpoint of an outside observer. Yet the decision to donate is not necessarily governed by the same rule that decides the size of the donation. Consider the hypothetical case of a husband and a wife: The husband decides if the offer is accepted or rejected, the wife decides the amount, and the decisions are made semi-independently without prior discussion. It is now clearly acceptable to ask how much would have been donated had the offer been accepted.

The main point is that we can assume there to be two rules at work. The first rule provides the values of interest (e.g., the donation amounts). The second rule is the sample selection or the observation rule. The second rule determines whether the value of interest is observed or not.

4 The original Heckman procedure

Suppose there are two linear equations, one for the regression and one for the sample selection rule:

$$y = \mathbf{x}^T \mathbf{b} + U_1,$$

$$z^* = \mathbf{w}^T \mathbf{c} + U_2.$$

The first equation is the regression equation whose parameters we would like to figure out. The second equation is the probit model specifying the sample selection process. Values of y are observed iff $z^* > 0$. We can define a dummy variable z such that $z = 1$ iff $z^* > 0$ (and zero otherwise). [4] We have distinquished $\mathbf{x}$ from $\mathbf{w}$ as they do not have to be the same sets of variables. In practice they might be partially overlapping sets of variables.

Based on [3], we can derive:

$$E[Y|\mathbf{x}, selection] = E[Y|\mathbf{x}] + E[U_1|selection].$$

Regression on the data for which y is known would deliver us $E[Y|\mathbf{x}, selection]$ instead of $E[Y|\mathbf{x}]$. The reason is that knowing y always implies that the selection rule was also satisfied. Therefore, $E[Y|\mathbf{x}, selection]$ might be a bad estimate of y for those $\mathbf{x}$ not satisfying the selection rule. We would like to know the $E[Y|\mathbf{x}]$ untainted by the selection process.

It can be shown that:

$$E[Y|\mathbf{x}, selection] = E[Y|\mathbf{x}, U_2 > -\mathbf{w}^T\mathbf{c}]$$

$$= \mathbf{x}^T\mathbf{b} + E[U_1|U_2 > -\mathbf{w}^T]$$

[4] Some additional details for the advanced reader: The noise term U_1 is $N(0, s)$ (the variance s is unknown at this point), and the noise term U_2 is $N(0, 1)$ as per the probit model. Because the noise can be correlated, we define $corr(U_1, U_2) = r$ (the coefficient r is unknown at this point).

$$= \mathbf{x}^T\mathbf{b} + rsd(\mathbf{w}) = \begin{pmatrix} \mathbf{x}^T & d(\mathbf{w}) \end{pmatrix} \begin{pmatrix} \mathbf{b} \\ rs \end{pmatrix}.$$

On the last line $d(\mathbf{w}) = \frac{f(\mathbf{w}^T\mathbf{c})}{F(\mathbf{w}^T\mathbf{c})}$ [5]. The big idea is that if one knew $d(\mathbf{w})$ for all training examples, then d could become an additional independent variable in a regression. The new regression problem would have $\mathbf{x}$ and d as independent variables and we could simultaneously estimate the parameters $\mathbf{b}$, r and s [6]. Having $\mathbf{b}$ would immediately give us $E[Y|\mathbf{x}] = \mathbf{x}^T\mathbf{b}$.

The Heckman 2-stage procedure is as follows:

1. Find the maximum likelihood estimate of the $\mathbf{c}$ in the probit model. Then augment each data row i with the additional attribute $d_i = d(\mathbf{w}_i) = \frac{f(\mathbf{w}_i^T\mathbf{c})}{F(\mathbf{w}_i^T\mathbf{c})}$. The probit analysis is performed on all available data.

2. Solve the augmented regression problem. The regression analysis is performed on the augmented data of the subset in which y was observed.

At this point, we should emphasize that we allowed some simplifications. For example, the second stage should not be done with ordinary least squares method. See [3] for details. There is also the problem of the method using a lot of assumptions.

5 Conclusions: About heuristics inspired by the Heckman procedure

Instead of using $d(\mathbf{w}) = \frac{f(\mathbf{w}^T\mathbf{c})}{F(\mathbf{w}^T\mathbf{c})}$ as an additional attribute in the final regression, one could experiment with other attributes based on the selection rule evaluation $\mathbf{w}^T\mathbf{c}$. Such choices might not have direct theoretical justifications, but might still approximate the Heckman procedure in some sense.

The most obvious alternative is the direct probit estimate $P(Y = 1|\mathbf{w}) = F(\mathbf{w}^T\mathbf{c})$. Instead of using the direct probit estimate, one could go a little further and experiment with non-probit estimates as well. In [2], the authors seem to get good results without probit models.

Speaking of the heuristics in [2], there is one additional point that should be emphasized: In the problem setting of that article (which is almost identical to

[5] In this context, F is the cumulative distribution function of $N(0,1)$ and f is the corresponding probability density function.

[6] r and s would have to be separated afterwards by a special procedure

Example 6 we presented earlier), the Heckman observation problem coincides with the main classification problem. Hence, the Heckman-inspired correction heuristic can use the class probabilities. In the general case, the problems do not necessarily coincide. The observation problem is always binary, whereas the main classification problem might have more than two classes. Further, the observation boundaries need not coincide with class boundaries (e.g., some examples of a class K could be observable and some might not).

References

[1] RNI Lecture notes on pattern recognition (in Finnish). Available directly from the Rolf Nevanlinna Institute. [7]

[2] Bianca Zadrozny and Charles Elkan. Learning and Making Decisions When Costs and Probabilities are Both Unknown. Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining (KDD01).

[3] James J. Heckman. Sample selection bias as a specification error. Econometrica, vol. 47: 153-161, 1979.

[7] The notes are vastly better than any book on pattern recognition that we have seen. Additionally, the notes can be downloaded for free.