

Winner's Report: KDD CUP Breast Cancer Identification

Claudia Perlich, Prem Melville, Yan Liu,
Grzegorz Świrszcz, Richard Lawrence
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598
{perlich,pmelvil,liuya}@us.ibm.com
{swirszcz,ricklavr}@us.ibm.com

Saharon Rosset
Raymond and Beverly Sackler School of
Mathematical Sciences
Tel Aviv University,
Israel
saharon@post.tau.ac.il

ABSTRACT

We describe the ideas and methodologies that we developed in addressing the KDD Cup 2008 on early breast cancer detection, and discuss how they contributed to our success.

1. TASK AND DATA DESCRIPTION

The Siemens KDD Cup 2008 comprised two prediction tasks in breast cancer detection from images. The organizers provided data from 1712 patients for training; of these 118 had cancer. Within the four breast images (two views for each breast per patient) are suspect locations (called *candidates*), that are described by their coordinates and 117 features. No explanation of the features was given to the competition participants. Overall the training set includes 102,294 candidates, 623 of which are positive. A second dataset with similar properties was used as the test set for competition evaluation. The two modeling tasks were:

Task 1: Rank the candidates by the likelihood of being cancerous in decreasing order. The evaluation criterion for this task was the FROC score, which measures how many of the actual *patients* with cancer are identified while limiting the number of *candidate* false alarms to between 0.2 and 0.3 per image. This was meant as a realistic representation when the prediction model is used as an actual decision support tool for radiologists.

Task 2: Suggest a maximal list of test-set patients who are healthy. In this competition, including any patient with cancer in the list will disqualify the entry. This was meant to represent a scenario where the model is used to save the radiologist work by ruling out patients who are *definitely healthy*, and thus the model was required to have *no false negatives*.

Several aspects of the data and the tasks made this competition interesting, including:

- The presence of leakage, whereby patient IDs turned out to carry significant information about a patient's likelihood to be malignant. The issue of leakage in

general is widespread in competitions, and also in real-life efforts. We discuss this competition's example and others in Section 2.

- Unique data properties, including the presence of extreme outliers and the combination of the features with neighborhood-based information from the location of candidates. These properties and our efforts in alleviating and using them, are discussed in Section 3.
- The unique FROC score, which treats patients as positive examples, but candidates as negative examples. This clearly has implications on the way in which models should rank candidates, preferentially combining candidates from different patients over many good candidates from the same patient. We address this in the context of post-processing schemes for model scores in Section 4.

We discuss our final submitted models in Section 5.

2. LEAKAGE IN PATIENT ID

Leakage can be defined as the introduction of predictive information about the target by the data generation, collection, and preparation process. Such information leakage - while potentially highly predictive out-of-sample *within* the study - leads to limited generalization, model applicability, or overestimation of the model performance.

Two of the most common causes for leakage are:

1. Combination of data from multiple sources and/or multiple time points, followed by a failure to completely anonymize the data and hide the different sources.
2. Accidental creation of artificial dependencies and additional information while preparing the data for the competition or proof-of-concept.

This year's KDD Cup data suffered from leakage that was probably due to the first the cause. The patient IDs in the competition data carried significant information towards identifying malignant patients. This is best illustrated through a discretization of the patient ID range in the training data, as demonstrated in Figure 1. The patient IDs are naturally divided into three disjoint bins: between 0 and 20,000 (254 patients; 36% malignant); between 100,000 and 500,000 (414 patients; 1% malignant); and above 4,000,000 (1044 patients, of them 1.7% malignant). We can further observe that all afflicted patients in the last bin (18) have patient IDs in the range 4,000,000 to 4,870,000, and there are only

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGKDD 2008 Las Vegas, NV USA

Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...\$5.00.

3 healthy patients in this range. This gives us a four-bin division of the data with great power to identify malignant patients. This binning and its correlation with the patient’s state generalized perfectly to the test data as well. Our hypothesis is that this leakage reflects the compilation of the competition data from different medical institutions and maybe different equipment, where the identity of the source is reflected in the ID range and is highly informative of the patient’s outcome. For example, one source might be a preventive care institution with only very low base rate of malignant patients and another could be a treatment-oriented institution with much higher cancer prevalence.

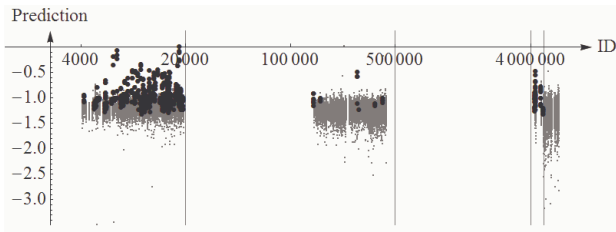


Figure 1: Distribution of malignant (black) and benign (gray) candidates depending on Patient ID on the X-axis in log scale. The Y-axis is the score of a linear SVM model on the 117 features. The vertical lines show the boundaries of the identified ID bins.

While it is clear that such leakage does not represent a useful pattern for real application, we consider its discovery and analysis an integral and important part of successful data analysis. Furthermore, we suggest that the problem in this dataset may actually run deeper and that all participants unknowingly benefited somewhat from a leakage problem and all reported performances are likely to be inflated.

If indeed the predictiveness of the identifiers was caused by the combination of data from different sources, there may be additional implicit leakages due to differences in data collection settings (e.g., machine calibration). This would still be present even if the patient IDs had been removed. We test this hypothesis with the following experiment: If such a leakage exists (say the average grayscale is slightly different), it should be possible to predict the data source (i.e., one of the four identifier bins) from negative candidates only. We cannot include positives because we already know that the cancer prevalence is correlated with the bins. Our analysis shows that both group 1 (ID below 20000) and group 4 (ID above 4,870,000) are easily identified by a logistic model from the 117 provided features with AUCs of 0.865 and 0.75 respectively.

Given this result we feel confident to conclude that any reasonable model can infer the patient group to some extent from the 117 variables and thereby implicitly the cancer prevalence in that patient population. So all models built on this data set are likely to overestimate the true predictive performance of cancer detection when applied to an entirely different population.

More generally, experience has shown that leakages play a central role in many modeling competitions, including KDD-Cup 2007 [4], where the organizers’ preparation of the data for one task exposed some information about the response for the other task, and KDD-Cup 2000 [1], where internal

testing patterns that were left in the data by the organizers supplied a significant boost to those who were able to identify them.

Exploratory data analysis seems to have become something of a lost art in the KDD community. In proper exploratory analysis, the modeler carefully and exhaustively examines the data with little preconception about what it contains, and allows patterns and phenomena to present themselves, only then analyzing them and questioning their origin and validity. A careful exploratory analysis of the data for this competition would most likely have identified this leakage. We hope that our discovery of this leakage can serve as a reminder of the value of open-minded exploratory analysis.

3. MODELING APPROACHES

3.1 Incorporating IDs

Given the obvious predictive value of the ID information we incorporated it as a categorical variable for the classification models with 4 possible bin numbers {1,2,3,4}. We also explored building 4 separate models, but this did not yield better results, presumably because for some of them the number of training points is rather small.

3.2 Classification

In order to investigate the generalization performance of different methods, we created a stratified 50% training and test split by patient. We ensured that exactly half of the positive patients were assigned to each split.

We explored the use of various learning algorithms for the underlying candidate classification problem including Neural Networks, Logistic regression and various SVM. Ultimately, linear models (logistic regression or linear SVMs) yielded the most promising results, and in this section we discuss various directions we explored for improving results with SVMs. The summary of all results are presented in Table 1. For all approaches described below we used SVMs as implemented in the *SVMPerf* package [2].

Down-sampling: In order to deal with the great imbalance between the positive and negative class, we experimented with down-sampling negatives. We found that maintaining all positives, and using only 25% of the negative candidates improved the performance of linear SVMs from a FROC of 0.0842 to 0.0893. As such, we used this down-sampled training set for further exploration.

Kernel selection: We compared the performance of SVMs using different kernels. In particular, we tested linear SVMs, RBF kernels and polynomial kernels of degree 2 and 3. We found that linear kernels performed the best, with a FROC of 0.0885. Linear SVMs have the added advantage of being extremely fast compared to the other approaches. The RBF kernels not only took the longest time to run, but also had a dismal performance of 0.0229. Given these results, we adopted linear SVMs for all the experiments below.

Loss function: Most work in the use of SVMs has focused on minimizing the error rate or zero-one loss function. In recent work, Joachims [2] presented efficient ways to train SVMs to maximize alternative multivariate performance measures, such as the area under the ROC curve (AUC). Given that the evaluation metric for Task 1 is re-

Approach	FROC
Logistic Regression	0.0877
Logistic Regression no outliers	0.0858
Linear SVM	0.0842
Down-sampled Linear SVM	0.0885
Polynomial kernels (d=2)	0.0803
Polynomial kernels (d=3)	0.0774
RBF Kernels	0.0229
Maximizing AUC	0.0893
Maximizing Precision at k	0.0869
Maximizing Recall at k	0.0865
Bagging linear SVM (c=20)	0.0900
Bagging linear SVM (c=500)	0.0873
Constrained Logistic Regression	0.0793

Table 1: Comparing FROC of different approaches.

lated to AUC, we trained an SVM to maximize AUC. We also compared maximizing Precision and Recall at k , which is the Precision/Recall of a classifier that predicts exactly k instances as positive. In particular, given p positive instances in the training set, we used $k = p/2$ and $k = 2p$ for Precision and Recall respectively. Since AUC is most closely related to the FROC metric, we find that maximizing it performs the best, improving the FROC (of minimizing error rate) from 0.0885 to 0.0893. In subsequent experiments, we explore improving on the use of these AUC-maximizing SVMs.

Regularization: The regularization parameter in linear SVMs controls the trade-off between fitting the training data and maximizing the margin. In order to explore the sensitivity of our classifier to this parameter, we tested various values. We observed the best result of 0.0905 using the regularization parameter, $c = 500$. In the following experiments, we use both this setting, as well as $c = 20$, which is the default we used for all experiments above.

Bagging: As observed by Valentini and Dietterich [5], bagging applied to SVMs can significantly improve classification accuracy over using single SVMs. Since bagging is a variance-reduction technique, they propose applying bagging to SVMs with low bias and high variance. In particular, for linear SVMs, they show that decreasing bias by increasing the regularization parameter and then applying bagging is very effective. Although Valentini and Dietterich’s results are for maximizing classification accuracy, we test the effectiveness of bagging in our setting for maximizing FROC. We applied 10 iterations of bagging linear SVMs with the regularization parameter, c , set to 20 and 500. We observed that with $c = 20$, bagging does improve the FROC from 0.0893 to 0.0900. However, for $c = 500$, the FROC drops from 0.0905 to 0.0873. We will see later, in Section 4, that bagging the lower-bias SVMs with $c = 500$ does in fact perform better with appropriate post-processing of model scores.

3.3 Outlier Treatment

The majority of the 117 features exhibit distributions with heavy tails and at times significantly skewed. Given that linear models seem to perform well on this domain, we considered the treatment of outliers as a potential way to improve model estimation and avoid extreme predictions caused by

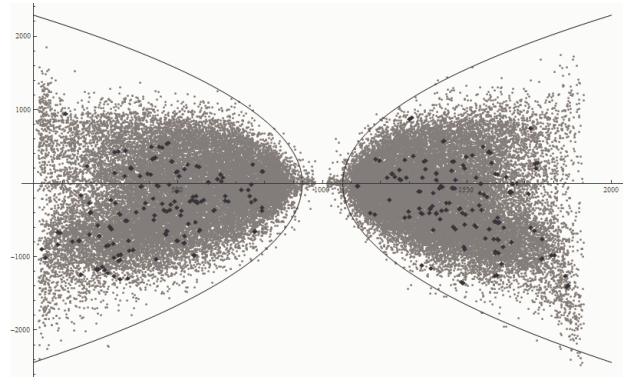


Figure 2: Distribution of malignant (black diamond) and benign (gray dot) within the picture, CC perspective.

extreme x values on the submission set. We calculated an upper and lower bound for each of the 117 features based on the inter-quartile range. We define Q_k^{25} as the 25th and Q_k^{75} as the 75th percentile of feature k . We replaced each value $x_k < Q_k^{25} - 3 * (Q_k^{75} - Q_k^{25})$ with the lower bound $Q_k^{25} - 3 * (Q_k^{75} - Q_k^{25})$ and $x_k > Q_k^{75} + 3 * (Q_k^{75} - Q_k^{25})$ with the upper bound $Q_k^{75} + 3 * (Q_k^{75} - Q_k^{25})$. This adjustment affected a non-negligible percentage of candidates and typically hurt the model performance significantly. We concluded that there is relevant information in the extreme values.

3.4 Additional Features

Candidate Location: We considered that the location of a candidate relative to nipple or breast boundary might be indicative of malignant candidates. Figure 2 shows the normalized and aligned locations of all candidates in the CC pictures (the MLO picture looks comparable). It turned out that even sophisticated features derived from the location – like high-degree polynomials, functions of the distance from the deduced boundaries etc. yielded no improvement. This indicates that either there are no “high risk” areas in the breast, or that the 117 original features already contain the relevant location information of the candidate.

Number of candidates per patient: Another direction that leads to additional informative features is the number of candidates belonging to a patient. Once again, all the attempts to construct features using this quantity lead either to deterioration or non-substantial improvement of the performance on the test data.

3.5 Modeling Neighborhood Dependence

As suggested in one of the hints provided by organizers, a cancerous lesion should be visible in both views (MLO and CC) of a breast (although in extremely rare cases some lesions may only be visible in one view). Therefore it would be useful to examine whether we can formulate meaningful constraints based on this observation and exploit correlations in classification decisions for the candidates from the same region of a breast.

Constraint Formulation: The data provided contains the coordinates of each candidate as well as those of the nipple in

the images. There are many meaningful ways to formulate the constraints. We choose a simple approach based on location adjacency, i.e. we compute the Euclidean distance from the candidates to the nipple in both views for each breast, select the pairs of candidates from different views with distance difference less than a threshold (we set the threshold as 20 in our experiments, resulting in 29139 constraints) and make constraints that the selected pairs of examples $(x_{i,\text{MLO}}, x_{i,\text{CC}})$ should have the same predicted labels, i.e. $f(x_{i,\text{MLO}}) = f(x_{i,\text{CC}})$.

Pairwise Constraint Kernel Logistic Regression: We use pairwise kernel logistic regression, which is able to plug in additional pairwise constraints together with labeled data to model the decision boundary directly [6]. Suppose we have a set of training examples $(x_1, y_1), \dots, (x_m, y_m)$ and a set of pairwise constraints $(x_{11}, x_{12}, +) \dots (x_{n1}, x_{n2}, -)$ constructed from both labeled and unlabeled data, where “+” means that the example pair (x_{i1}, x_{i2}) belongs to the same class and “-” means different classes. In our setting, we only consider the positive constraints, i.e. the pair of examples belong to the same class. To make the optimization problem feasible to solve, we define a convex loss function via the logit loss as follows:

$$\begin{aligned} \mathcal{O}(f) = & \frac{1}{m} \sum_{i=1}^m \log(1 + e^{-y_i f(x_i)}) + \\ & \frac{\mu}{n} \sum_{i=1}^n \log(1 + e^{f(x_{i1}) - f(x_{i2})}) + \lambda \Omega(\|f\|_{\mathcal{H}}), \end{aligned} \quad (1)$$

where the first term is the loss on labeled training examples, the second is the loss associated with the difference between the predicted labels of the example pairs and third term is the regularizer. The pairwise constraint coefficient μ is set to 1. For simplicity, we define f as a linear classifier, i.e. $f(x) = w^T x$. Since the optimization function is convex, a gradient search algorithm can guarantee the finding of the global optimum. It is easy to derive the parameter estimation method using the interior-reflective Newton method, and we omit the detailed discussion. The constrained logistic regression unfortunately did not compare favorably to other competing methods (see Table 1).

4. POSTPROCESSING

The FROC evaluation metric implies that given prediction scores from a model (at the candidate level), one might be wise to reorder them so that a larger variety of patients are represented at the top of the list. In what follows we offer some heuristic implementations of this idea and a theoretical result on one of them.

4.1 Heuristic Approach

Task 1 uses the predictions from the classification model to order the candidates and computes the FROC metric from this ordering. Classification algorithms focused on maximizing accuracy may not perform well at ranking instances. As observed in Section 3 attempting to maximize AUC improves ranking of candidates, which in turn improves FROC. However, improving AUC is not guaranteed to improve FROC. This is because AUC measures the area under the curve of true positive rate versus false positive rate, whereas the Y-axis of a FROC curve is the true positive rate at a patient level. As such, a higher true positive

Table 2: Comparing FROC before and after post-processing model scores.

Approach	Raw scores	Post-processed
Bagging linear SVMs ($c=20$)	0.0900	0.0930
Bagging linear SVMs ($c=500$)	0.0873	0.0959

rate at a candidate-level does not improve FROC unless the positive candidates are from different patients. For instance, it is better to have 2 correctly identified candidates from different patients, instead of 5 correctly identified candidates from the same. So it is best to re-order candidates based on model scores so as to ensure we have many different patients up front.

In order to do this, we create a pool of the top n candidates, as ordered by our model. We then select the candidates with the highest scores for each patient in this pool, and move these to the top of our list. We repeat this process iteratively with the remaining candidates in our pool until we have exhausted all candidates.

We only do this for the top n candidates, since the FROC metric is based only on the area under the curve for a small range of false alarm rates at the beginning of the curve. We leave the ordering of the remaining candidates untouched. The only parameter this post-processing procedure requires is the choice of n for the number of top-ranked candidates we want to re-order. We can select this parameter based on an estimate of the maximum number of candidates that have to be classified as positive before we hit the upper bound of the false alarm rate used in the FROC metric. For the specific FROC metric used to evaluate Task 1, it can be shown that the optimum value for n is the *number of positive candidates* + $1.2 \times$ *number of patients*.

Since the true number of positive candidates in the test set is not known, we estimate this from the positive rate in the training set. For our train-test split we used $n = 1500$, and report the results before and after post-processing in Table 2. Clearly, this approach of re-ordering model scores has a significant impact on the resulting FROC, with no additional modeling required. Post-processing the scores from the low-bias, AUC-maximizing SVMs provided at times amazing improvements with FROC up to 0.0959. Even though our best result so far was for bagging low-bias AUC-maximizing SVMs with this post-processing, we found in further 10-fold cross-validation studies that the performance of this approach had high variance and was not as reliable as more conservative models.

4.2 Theoretical Approach

Assume we have a perfect estimate for the probability p_i , that each candidate is malignant, and that malignancy of each candidate is independent of any other candidates (given their probability). Assume further that we have a patient j , denoted by her ID for which we have included k candidates at the top of our list. Denote these candidates’ probabilities by p_1, \dots, p_k . Once we include all of these, the probability that none of them is malignant (and therefore we have not yet identified this patient as malignant) is $PN(j) = \prod_{i=1 \dots k} (1 - p_i)$. Given another candidate for this patient with probability of malignancy p_{k+1} , we can add

the candidate to the list, and find a new malignant patient with probability $P * p_{k+1}$ or add another false alarm with probability $1 - p_{k+1}$. We can then investigate the expected contribution to FROC and other metrics that measure patient detection versus candidate false alarms (in particular the corresponding AUC version) from applying this logic. The following algorithm and proposition make the optimal policy in terms of AUC and FROC explicit.

ALGORITHM 1 (POSTPROCESSING). *Input: N the number of candidates, n the number of patients, the sequence S of pairs $\{ID_i, p_i\}_{i=1}^N$. Output: $Z = \{z_i\}_{i=1}^N$.*

1. Set $PN(j) = 1$, $j = 1, \dots, n$.
2. Sort S according to the value of p_i in a descending order (higher p_i 's go first)
3. Append $\{-1, -1\}$ at the end of S (for technical reasons, it is assumed here that all $ID_i > 0$)
4. For $i = 1$ to N
 - (a) Set $y_i = PN(ID_i) \times \frac{p_i}{1-p_i}$
(if $PN(ID_i) = 0$ and $p_i = 1$ set $y_i = 0$).
 - (b) $PN(ID_i) = PN(ID_i) \times (1 - p_i)$.

In words, this algorithm replaces the p_i 's with y_i 's, which are adjusted to take into account the patient-level evaluation used by FROC. We can show that this ordering is optimal in terms of FROC if the p_i 's are indeed the true probability of malignancy.

THEOREM 1. *The expected value of FROC (and its generalization to AUC) for the sequence Y obtained using Algorithm 1 is higher than for any other reassignment of values of p_i 's.*

For the above theorem to hold, p_i 's must be true probabilities of malignancy for each candidate, which are clearly not what our models generate. Some modeling approaches, like SVMs, do not even generate probabilities that can be interpreted as probabilities. In the case of SVMs, Platt correction [3] is a common approach to alleviate this problem. We thus applied this post-processing approach to three different models:

- Logistic regression raw predictions. These are expected to be somewhat overfitted and therefore not good as probability estimates. Since the emphasis of the algorithm is on the largest p_i 's we modified them simply by capping them, leading to:
- Logistic regression predictions, capped at different thresholds (e.g., 0.5)
- SVMs with Platt correction

Disappointingly, Algorithm 1 did not lead to a significant improvement in holdout FROC on any of these models, implying that our algorithm, while theoretically attractive, has little practical value when dealing with probability estimates instead of true probabilities. We did observe that the AUC improved initially (below 0.05) but not in the relevant area of 0.2-0.3. Attempts to improve the calibration of our probabilities using non-parametric approaches resulted in performances similar to the heuristic post-processing.

5. SUBMISSIONS AND RESULTS

Task 1: We initially submitted a logistic model that included in addition to the 117 provided features (no feature selection, outlier treatment or other features) only the categorical bin feature with the 4 possible values 1,2,3,4. We applied the heuristic-based post processing and this model scored a FROC of 0.0929 on the submission set. For our final submission we used a bagged linear SVMs with the ID features, zero-one loss, $c = 20$ and the heuristic post processing. This approach scored the winning result of 0.0933.

Task 2: While the ID leakage had some influence on our Task 1 model, it was clearly essential for our Task 2 submission. It also explains the huge gap between our submission and the next best result. We investigated different models with and without ID features after ranking increasingly by the maximum candidate score of each patient. Models without ID typically produce the first false positive patient within about 200 patients of the training set (interestingly they are the positive patients from bin 3 just at the boundary between the two bins as seen in Figure 1). Conversely, we were fairly confident that bin 4 alone (more than 1000 on the training and 970 on the submission set) had no positive patients at all. Models with ID features rank all patients of bin 4 first and next some patients from bin 2. The first false negatives occur typically around 1100. On this task - the logistic model did perform better than the SVM models. We finally submitted the 1020 first ranked patients from a logistic model that included the ID features in addition to the original 117 provided features.

6. REFERENCES

- [1] A. Inger, N. Vatnik, S. Rosset, and E. Neumann. KDD-CUP 2000: Question 1 winner's report. *SIGKDD Explorations*, 2000.
- [2] T. Joachims. A support vector method for multivariate performance measures. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 377-384, New York, NY, USA, 2005. ACM.
- [3] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*. MIT Press, 1998.
- [4] S. Rosset, C. Perlich, and Y. Liu. Making the most of your data: Kdd cup 2007 "how many ratings" winner's report. *SIGKDD Explorations*, 2007.
- [5] G. Valentini and T. G. Dietterich. Low bias bagged support vector machines. In *Proceedings of 20th International Conference on Machine Learning (ICML-2003)*, pages 752-759, Washington, DC, 2003.
- [6] R. Yan, J. Zhang, J. Yang, and A. Hauptmann. A discriminative learning framework with pairwise constraints for video object classification. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'04)*, 2004.