Probablistic Model for High Precision/Recall Feature Selection

Draft for ICML 2014

January 31, 2014

1 Introduction

Feature selection is the selection of a subset of features F^* from the complete feature space F, for a specific classification/prediction problem, without loosing too much performance on some given measures. The benefit of feature selection include more explainable model, reduced computation time/power, and the avoidance of overfitting.

Existing feature selection algorithms, whose performance show no statistical difference on out benchmark, are mostly ad-hoc. They give various answers to optimisation, but none of them gives the question. In this project, we started from the first principles, proposed a graphical model, and built a new feature selection system upon it.

2 Graphical Model

To begin the derivation, we propose a graphical model of feature selection for binary classification problem, as shown in figure 1. Double-circle nodes represent observed variables, including the data point vector $\vec{x}^d$, selected features f_i^* (where $1 \leq i \leq k$, $f_i \in F$) and the supervised class label y^d. Single-circle nodes represent latent variables, including a selected attribute of data point vector f_i^x, and prediction from a feature y_i^d.

Since jointly optimising this objective is NP-hard, we will build F^* in a greedy manner by choosing the next optimal feature f_k^* given the previous set of optimal features F_{k-1}^* and recursively defining $F_k^* = F_{k-1}^* \cup f_k^*$ with $F_0^* = \emptyset$. The measure of the fitness of a model can either be Expectation or Likelihood.

3 Expectation

The initial objective is defined as:

$$f_k^* = \arg\max_{f_k} E_D \left[P\left(y^d | \vec{x}^d, F_k \right) \right]$$

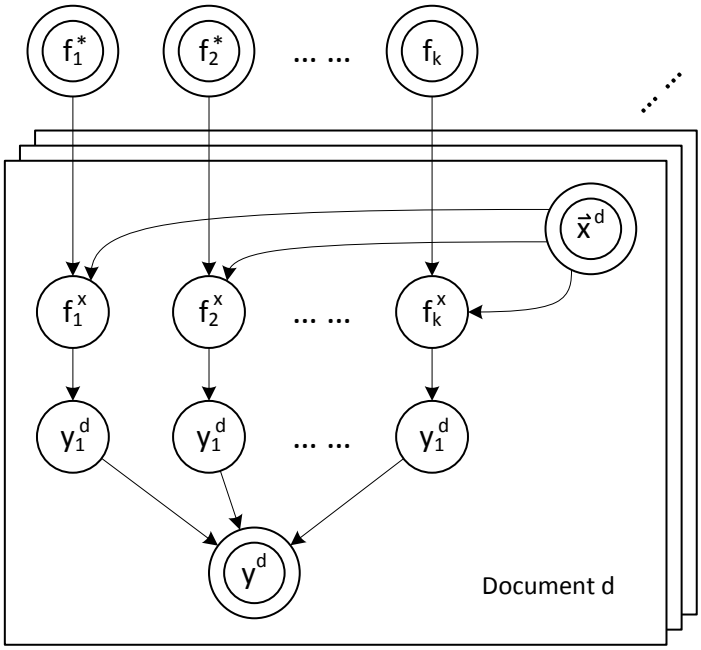

Figure 1: Graphical model. Double-wall circle denotes observed nodes, single-wall circle denotes hidden nodes.

$$
\begin{aligned}
f_k^* &= \arg\max_{f_k} E_D\left[P\left(y^d|\vec{x}^{\,d}, F_k\right)\right] \\
&= \arg\max_{f_k} \frac{1}{|D|}\sum_{d\in D} P\left(y^d|\vec{x}^{\,d}, F_k\right) \\
&= \arg\max_{f_k} \sum_{d\in D} \frac{P\left(y^d, \vec{x}^{\,d}, F_k\right)}{P\left(\vec{x}^{\,d}, F_k\right)} \\
&= \arg\max_{f_k} \sum_{d\in D} \sum_{\{y_i^d\}_{1\le i\le k}} \frac{P\left(y^d, \vec{x}^{\,d}, F_k, \{y_i^d\}_{1\le i\le k}\right)}{P\left(\vec{x}^{\,d}, F_k\right)} \\
&= \arg\max_{f_k} \sum_{d\in D} \sum_{\{y_i^d\}_{1\le i\le k}} \frac{P\left(y^d|\{y_i^d\}_{1\le i\le k}\right)\left(\prod_{i=1}^k P\left(y_i^d|\vec{x}^{\,d}, f_i\right)\right) P\left(\vec{x}^{\,d}\right)\prod_{i=1}^k P\left(f_i\right)}{P\left(\vec{x}^{\,d}, F_k\right)} \\
&= \arg\max_{f_k} \sum_{d\in D} \sum_{\{y_i^d\}_{1\le i\le k}} P\left(y^d|\{y_i^d\}_{1\le i\le k}\right)\prod_{i=1}^k P\left(y_i^d|\vec{x}^{\,d}, f_i\right) \qquad (1)
\end{aligned}
$$

Here, we rewrote the expectation of a binary event as its probability, factorized the conditional probability in the joint probability divided by the marginal, marginalised over $\{y_i^d\}_{1\le i\le k}$, factorized joint probability in conditional and prior following the graphical model and exploited d-separation to remove irrelevant conditions and to cancel terms in the equation.

Now we can optimise our initial objective aiming three goals of classification tasks: Precision, Recall, and General n-out-of-k.

3.1 Precision

In order to select the subset of features providing high precision binary classifier we need the agreement of all features predictors y_i^d to predict true. That is, when y^d is true we need all of the predictors y_i^d equals to true, and if y^d were false, just one of the predictors y_i^d would have to be false. Then the probability $P\left(y^d|\{y_i^d\}_{1\leq i\leq k}\right)$ can be expressed as follows:

$$P\left(y^d|\{y_i^d\}_{1\leq i\leq k}\right) = I\left[y^d = \bigwedge_{i=1}^{k} y_i^d\right] = \begin{cases} y^d = 1 \text{ when } \{y_i^d = 1\}_{1\leq i\leq k} \\ y^d = 0 \text{ when } \{y_i^d\}_{1\leq i\leq k, \exists y_j^d \| y_j^d = 0} \end{cases} \tag{2}$$

Then, we combined equations (1) and (2), separated each term according to the actual label value y^d and used the probability sum rule to rewire the second term as follows:

$$\begin{aligned} f_k^* &= \arg\max_{f_k} \sum_{d\in D} I\left[y^d = \bigwedge_{i=1}^{k} y_i^d\right] \prod_{i=1}^{k} P\left(y_i^d|\vec{x}^{\,d}, f_i\right) \\ &= \arg\max_{f_k} \sum_{d\in D} \begin{cases} y^d = 1 : \prod_{i=1}^{k} P\left(y_i^d = 1|\vec{x}^{\,d}, f_i\right) \\ y^d = 0 : \sum_{\substack{\{y_i^d\}_{1\leq i\leq k}, \\ \exists y_i^d \| y_i^d = 0}} \prod_{i=1}^{k} P\left(y_i^d|\vec{x}^{\,d}, f_i\right) \end{cases} \\ &= \arg\max_{f_k} \sum_{d\in D} \begin{cases} y^d = 1 : \prod_{i=1}^{k} P\left(y_i^d = 1|\vec{x}^{\,d}, f_i\right) \\ y^d = 0 : 1 - \prod_{i=1}^{k} P\left(y_i^d = 1|\vec{x}^{\,d}, f_i\right) \end{cases} \\ &= \arg\max_{f_k} \sum_{d\in D} \begin{cases} y^d = 1 : \prod_{i=1}^{k} P\left(y^d = 1|\vec{x}_i^{\,d}, f_i\right) \\ y^d = 0 : 1 - \prod_{i=1}^{k} P\left(y^d = 1|\vec{x}_i^{\,d}, f_i\right) \end{cases} \end{aligned} \tag{3}$$

From (3) we can intuitively describe how this equation is related to the precision metric. In the confusion matrix, Precision is calculated dividing the numbers of true positives by the number of all examples predicted as positive (true positives and false positives). When y^d is true, (3) gives higher score to feature that have higher probability to predict true, encouraging true positives. Meanwhile, if y^d is false, (3) gives lower score to features that have higher probability to predict true, penalising false positives.

3.2 Recall case

In order to select the subset of features providing high recall binary classifier we need at least one y_i^d to predict true. Therefore, we need a disjunction operation between the predictors y_i^d. That is, when y^d is true, we need at least one predictor y_i^d to equals to true, and if y^d were false, all of the predictors y_i^d would have to be false. Then the probability $P\left(y^d|\{y_i^d\}_{1\leq i\leq k}\right)$ can be expressed as follows:

$$P\left(y^d|\{y_i^d\}_{1\leq i\leq k}\right) = I\left[y^d = \bigvee_{i=1}^{k} y_i^d\right] = \begin{cases} y^d = 1 \text{ when } \{y_i^d\}_{1\leq i\leq k, \exists y_j^d \| y_j^d = 1} \\ y^d = 0 \text{ when } \{y_i^d = 0\}_{1\leq i\leq k} \end{cases} \tag{4}$$

Here, we combined equations (1) and (4), separated each term according to the actual label value y^d and used the probability sum rule to rewrite the second term as follows:

$$
\begin{aligned}
f_k^* &= \arg\max_{f_k} \sum_{d \in D} I\left[y^d = \bigvee_{i=1}^{k} y_i^d\right] \prod_{i=1}^{k} P\left(y_i^d | \vec{x}^{\,d}, f_i\right) \\
&= \arg\max_{f_k} \sum_{d \in D} \begin{cases} y^d = 0 : \prod_{i=1}^{k} P\left(y_i^d = 0 | \vec{x}^{\,d}, f_i\right) \\ y^d = 1 : \sum_{\substack{\{y_i^d\}_{1 \le i \le k}, \\ \exists y_i^d \| y_i^d = 1}} \prod_{i=1}^{k} P\left(y_i^d | \vec{x}^{\,d}, f_i\right) \end{cases} \\
&= \arg\max_{f_k} \sum_{d \in D} \begin{cases} y^d = 0 : \prod_{i=1}^{k} P\left(y_i^d = 0 | \vec{x}^{\,d}, f_i\right) \\ y^d = 1 : 1 - \prod_{i=1}^{k} P\left(y_i^d = 0 | \vec{x}^{\,d}, f_i\right) \end{cases} \\
&= \arg\max_{f_k} \sum_{d \in D} \begin{cases} y^d = 0 : \prod_{i=1}^{k} P\left(y^d = 0 | \vec{x}_i^{\,d}, f_i\right) \\ y^d = 1 : 1 - \prod_{i=1}^{k} P\left(y^d = 0 | \vec{x}_i^{\,d}, f_i\right) \end{cases}
\end{aligned}
\tag{5}
$$

From (5) we can intuitively describe how this equation is related to the precision metric. In the confusion matrix, Recall is calculated dividing the numbers of actual true positives by the number of all true positives (true positives and false negatives). When y^d is false, (5) gives higher score to feature that have higher probability to predict false, encouraging true negatives. Meanwhile, if y^d is true, (5) gives lower score to features that have higher probability to predict false, penalizing false negatives.

3.3 General n-out-of-k

In order to balance Expectation and Recall, we use a voting notation. When y^d is true, then at lease n out of k selected predictors y_i^d are true. However, if y^d is false, then the number of predictors y_i^d that predict true is strictly less than n. Then the probability $P\left(y^d | \{y_i^d\}_{1 \le i \le k}\right)$ can be expressed as follows:

$$
P\left(y^d | \{y_i^d\}_{1 \le i \le k}\right) = I\left[\sum_{i=1}^{k} y_i^d \ge n\right] \tag{6}
$$

Here, we combined equations (1) and (6), separated the result according to the prediction of y_i^d and used the probability sum rule to rewrite the second term as follows:

$$
\begin{aligned}
f_k^* &= \arg\max_{f_k} \sum_{d\in D} \sum_{\{\vec{y}^d \| \sum_{i=1}^k y_i^d \geq n\}} \prod_{i=1}^{k} P\left(y_i^d | f_i, \vec{x}^d\right) \\
&= \arg\max_{f_k} \sum_{d\in D} \sum_{\{\vec{y}^d \| \sum_{i=1}^k y_i^d \geq n\}} P\left(y_1^d | f_1, \vec{x}^d\right) \prod_{i=2}^{k} P\left(y_i^d | f_i, \vec{x}^d\right) \\
&= \arg\max_{f_k} \sum_{d\in D} \left[P\left(y_1^d = 1 | f_1, \vec{x}^d\right) + P\left(y_1^d = 0 | f_1, \vec{x}^d\right)\right] \sum_{\{\vec{y}^d \| \sum_{i=1}^k y_i^d \geq n\}} \prod_{i=2}^{k} P\left(y_i^d | f_i, \vec{x}^d\right) \\
&= \arg\max_{f_k} \sum_{d\in D} \Bigg[P\left(y_1^d = 1 | f_1, \vec{x}^d\right) \sum_{\{\vec{y}^d \| \sum_{i=2}^k y_i^d \geq n-1\}} \prod_{i=2}^{k} P\left(y_i^d | f_i, \vec{x}^d\right) \\
&\qquad + P\left(y_1^d = 0 | f_1, \vec{x}^d\right) \sum_{\{\vec{y}^d \| \sum_{i=2}^k y_i^d \geq n\}} \prod_{i=2}^{k} P\left(y_i^d | f_i, \vec{x}^d\right) \Bigg] \\
&\approx \arg\max_{f_k} \sum_{d\in D} \Bigg[P\left(y^d = 1 | f_1, \vec{x}^d\right) \underbrace{\sum_{\{\vec{y}^d \| \sum_{i=2}^k y_i^d \geq n-1\}} \prod_{i=2}^{k} P\left(y_i^d | f_i, \vec{x}^d\right)}_{recursion} \\
&\qquad + P\left(y^d = 0 | f_1, \vec{x}^d\right) \underbrace{\sum_{\{\vec{y}^d \| \sum_{i=2}^k y_i^d \geq n\}} \prod_{i=2}^{k} P\left(y_i^d | f_i, \vec{x}^d\right)}_{recursion} \Bigg]
\end{aligned}
\tag{7}
$$

In (7), the result is recursively defined. This recursion can be turned into dynamic programming, as shown in figure 2, reducing time complexity from $O\left(2^k\right)$ to $O\left(nk\right)$.

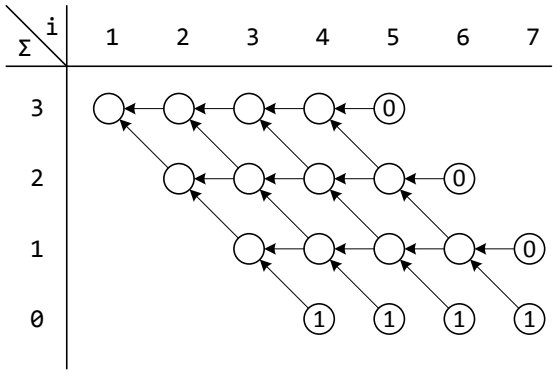

Figure 2: Dynamic programming table to turn the recursion in (7) to dynamic programming.

4 Likelihood

$$
\begin{aligned}
f_k^* &\equiv \arg\max_{f_k} log \prod_{d \in D} P\left(y^d | \vec{x}^{\,d}, F_k\right) \\
&= \arg\max_{f_k} \sum_{d \in D} log \left(\frac{P\left(y^d, \vec{x}^{\,d}, F_k\right)}{P\left(\vec{x}^{\,d}, F_k\right)} \right) \\
&= \arg\max_{f_k} \sum_{d \in D} log \sum_{\{y_i^d\}_{1 \le i \le k}} \frac{P\left(y^d, \vec{x}^{\,d}, F_k, \{y_i^d\}_{1 \le i \le k}\right)}{P\left(\vec{x}^{\,d}, F_k\right)} \\
&= \arg\max_{f_k} \sum_{d \in D} log \sum_{\{y_i^d\}_{1 \le i \le k}} \frac{P\left(y^d | \{y_i^d\}_{1 \le i \le k}\right) \prod_{i=1}^{k} P\left(y_i^d | \vec{x}^{\,d}, f_i\right) P\left(\vec{x}^{\,d}\right) \prod_{i=1}^{k} P\left(f_i\right)}{P\left(\vec{x}^{\,d}, F_k\right)} \\
&= \arg\max_{f_k} \sum_{d \in D} log \sum_{\{y_i^d\}_{1 \le i \le k}} P\left(y^d | \{y_i^d\}_{1 \le i \le k}\right) \prod_{i=1}^{k} P\left(y_i^d | \vec{x}^{\,d}, f_i\right)
\end{aligned}
$$

Here, we rewrote the likelihood of a binary event as its probability, factorized the conditional probability in the joint probability divided by the marginal, marginalized over $\{y_i^d\}_{1 \le i \le k}$, factorized joint probability in conditional and prior following the graphical model and exploited d-separation to remove irrelevant conditions and to cancel terms in the equation.

Since $\sum_{d \in D} log \sum_{\{y_i^d\}_{1 \le i \le k}} P\left(y^d | \{y_i^d\}_{1 \le i \le k}\right) \prod_{i=1}^{k} P\left(y_i^d | \vec{x}^{\,d}, f_i\right)$ is exactly the same as the result in the expectation case, and the logarithm is a monotonous function, the choice of likelihood instead of expectation has no further affect on the algorithm.