Data Mining Strategies for CRM Negotiation Prescription Problems

A. Bella, C. Ferri, J. Hernández-Orallo and M.J. Ramírez-Quintana

DSIC, UPV, València, Spain.

{abella,cferri,jorallo,mramirez}@dsic.upv.es

Twenty Third International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems .
IEA/AIE 2010

June 1-4, 2010
Córdoba, Spain

Outline

Introduction

Data Mining

- Problem features (or attributes) have been usually classified as input and output features.
- Input features can be of many kinds: numerical, nominal, structured, etc.
- In supervised learning, a function is learned from inputs to outputs which is eventually applied to new cases to predict an output value given an input.

However, in many application areas some input feature values can be modified or fine-tuned at prediction time.

Introduction

Data Mining

- Problem features (or attributes) have been usually classified as input and output features.
- Input features can be of many kinds: numerical, nominal, structured, etc.
- In supervised learning, a function is learned from inputs to outputs which is eventually applied to new cases to predict an output value given an input.

However, in many application areas some input feature values can be modified or fine-tuned at prediction time.

Example (Motivation: a loan granting model)

Years	Health insurance	...	Loan amount		Granted?
5	Yes	...	50.000		Yes
6	No	...	120.000		No
4	No	...	80.000		Yes
4	No	...	100.000		No

Input features

Output feature

In real scenarios

- The decision of whether the loan must be granted or not might change if one or more of the input feature values can be changed.
- For instance, a loan cannot be granted for 300,000 euros, but it can be granted for 250,000 euros.

Example (Motivation: a loan granting model)

Years	Health insurance	...	Loan amount	Granted?
5	Yes	...	50.000	Yes
6	No	...	120.000	No
4	No	...	80.000	Yes
4	No	...	100.000	No

Input features | Output feature

In real scenarios

- The decision of whether the loan must be granted or not might change if one or more of the input feature values can be changed.
- For instance, a loan cannot be granted for 300,000 euros, but it can be granted for 250,000 euros.

The inverse problem: What is the maximum amount you can grant for this operation?

The only possibility to give a precise answer is to try all the possible input combinations for the loan amount and find out the maximum value which is granted.

Example (Motivation: inverting the loan granting problem)

Years	Health insurance	...	Loan amount	Granted?
4	No	...	10.000	Yes
4	No	...	20.000	Yes
4	No	...	80.000	Yes
4	**No**	**...**	**84.145**	**Yes**
4	**No**	**...**	**84.146**	**No**
4	No	...	100.000	No

This process is inefficient and decisions based on it could be risky: the model will give a negative answer to any amount greater than this maximum.

Example (Motivation: inverting the loan granting problem)

Years	Health insurance	...	Loan amount	Granted?
4	No	...	10.000	Yes
4	No	...	20.000	Yes
4	No	...	80.000	Yes
4	**No**	**...**	**84.145**	**Yes**
4	**No**	**...**	**84.146**	**No**
4	No	...	100.000	No

This process is inefficient and decisions based on it could be risky: the model will give a negative answer to any amount greater than this maximum.

This typical example shows:

- Some input features are crucial in the way that they can be freely modified at prediction time (loan amount).
- The existence of these special attributes makes it quite inefficient to develop a classification/regression model in the classical way (for a new instance hundreds of possible values have to be tried for the loan amount in order to see which combination "loan_amount -granted?" can attain the maximum risk).
- These features are frequently associated to negotiation scenarios.

This paper analyses these special features that we call "negotiable features"

- How we can exploit the relation between these input features and the output to change the problem presentation: a classification problem can be turned into a regression problem over an input feature.
- How to apply these models in real negotiation scenarios where there are several attempts for the output value.

Negotiable Features

Notation

Consider a supervised problem, where

- input attribute domains are denoted by X_i, $i \in \{1, ..., m\}$,
- the output attribute domain is denoted by Y,
- the target (real) function is $f : X_1 \times X_2 \times ... \times X_m \rightarrow Y$,
- labelled instances are $\langle x_1, x_2, \ldots, x_m, y \rangle$, $x_i \in X_i, y \in Y$.

Assumption

- There is a strict total order relation for the output: $\forall y_a, y_b \in Y$, $y_a \prec y_b$ or $y_b \prec y_a$.
  - For numerical outputs, $\prec$ is usually $<$ or $>$.
  - For binary problems, we can set that $NEG \prec POS$.
  - For more than two classes, it can be derived from the cost of each class.
- There is also a total order relation for the input denoted as $\preceq$

Negotiable Features

Notation

Consider a supervised problem, where

- input attribute domains are denoted by X_i, $i \in \{1, ..., m\}$,
- the output attribute domain is denoted by Y,
- the target (real) function is $f : X_1 \times X_2 \times ... \times X_m \rightarrow Y$,
- labelled instances are $\langle x_1, x_2, \ldots, x_m, y \rangle$, $x_i \in X_i, y \in Y$.

Assumption

- There is a strict total order relation for the output: $\forall y_a, y_b \in Y$, $y_a \prec y_b$ or $y_b \prec y_a$.
  - For numerical outputs, $\prec$ is usually $<$ or $>$.
  - For binary problems, we can set that $NEG \prec POS$.
  - For more than two classes, it can be derived from the cost of each class.
- There is also a total order relation for the input denoted as $\preceq$

Definition

An input attribute X_i is said to be a negotiable feature (or attribute) for a dataset D iff

1. (adjustability) The values for X_i can be varied at model application time.
2. (sensitivity) Fixing the values of all the other input attributes $X_j \neq X_i$ of the instance, there are two different values for Xi producing different output values for at least n examples from D.
3. (monotonicity) The relation between the input feature X_i and the output Y is either
    - monotonically increasing: for any two values $a, b \in X_i$, if $a \succeq b$ then $f(x_1, x_2, ..., a, ..., x_m) \succeq f(x_1, x_2, ..., b, ..., x_m)$
      or
    - monotonically decreasing: for any two values $a, b \in X_i$, if $a \succeq b$ then $f(x_1, x_2, ..., a, ..., x_m) \preceq f(x_1, x_2, ..., b, ..., x_m)$.

Example (The loan granting problem)

- The age of the customer is not a negotiable feature, since we cannot modify it (condition 1 is violated).
- The bank branch office where the contract can take place is not a negotiable feature since it rarely affects the output (property 2 is violated).
- The loan amount is a negotiable feature and the relation between it and the output attribute granted? is monotonically decreasing.

Inverting Problem Presentation

Definition

Given a supervised problem $f : X_1 \times X_2 \times ... \times X_m \rightarrow Y$, the inversion problem consists in defining the function

$$f^I : X_1 \times X_2 \times ... \times Y \times ... \times X_m \rightarrow X_i$$

where X_i is the negotiable feature.

If X_i is numerical, f^I is a regression problem.

Example (The loan granting problem)

In this problem, f is the function that determines if a loan is granted or no, the negotiable feature X_i is the loan amount and f^I calculates the maximum amount you can grant for this operation.

From the inverted problem to the original problem

For binary problems, it can be done by calculating

$$p(POS|\langle x_1, \ldots, x_{i-1}, a, x_{i+1}, \ldots, x_m \rangle)$$

for any possible value $a \in X_i$.

How to calculate $p(POS|\langle x_1, \ldots, x_{i-1}, a, x_{i+1}, \ldots, x_m \rangle)$

- From the inverted problem, we get a prediction

$$\hat{a} = f^I(x_1, \ldots, x_{i-1}, POS, x_{i+1}, \ldots, x_m)$$

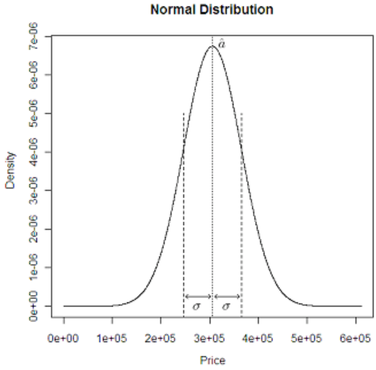
- If we think of $\hat{a}$ as the predicted maximum or minimum for a which makes a change on the class, a reasonable assumption is to give 0.5 probability for this

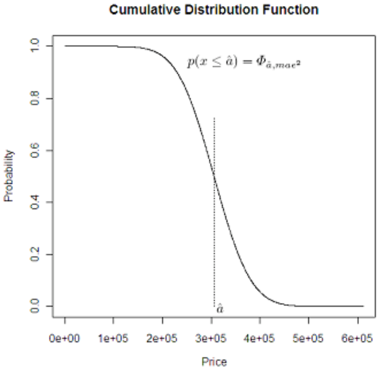
$$p(POS|\langle x_1, \ldots, x_{i-1}, \hat{a} x_{i+1}, \ldots, x_m \rangle) = 0.5$$

- We can assume that the output for f^I follows a normal distribution $N(\hat{a}, mae^2)$.
- Finally, we derive the probability for any possible value a as the cumulative distribution function derived from $N(\hat{a}, mae^2)$, i.e., $\Phi_{\hat{a}, mae^2}$.

Example

A normal distribution with centre at $\hat{a} = 305,677.9$ and standard deviation $\sigma = 59,209.06$ (Left) and its associated cumulative distribution function (Right).





Negotiation using Negotiable Feature Models

A real scenario: the problem of retailing

Real data from an estate agent's, which sells flats and houses.

- Conventional attributes describing the property (location, ...).
- A special attribute which is the negotiable feature, price (π).

The simplest negotiation scenario

- One seller and one buyer who both negotiate for one product (flat).
- The buyer is interested in one specific product that he/she will buy if its price is under a maximum price.
- The seller has a "minimum price" (π_{min}) such that any $\pi > \pi_{min}$ is profitable but any $\pi < \pi_{min}$ is not acceptable.
- Finally, the profit obtained by the product is $Profit(\pi) = \pi - \pi_{min}$.

Negotiation using Negotiable Feature Models

A real scenario: the problem of retailing

Real data from an estate agent's, which sells flats and houses.

- Conventional attributes describing the property (location, . . .).
- A special attribute which is the negotiable feature, price (π).

The simplest negotiation scenario

- One seller and one buyer who both negotiate for one product (flat).
- The buyer is interested in one specific product that he/she will buy if its price is under a maximum price.
- The seller has a "minimum price" (π_{min}) such that any $\pi > \pi_{min}$ is profitable but any $\pi < \pi_{min}$ is not acceptable.
- Finally, the profit obtained by the product is $Profit(\pi) = \pi - \pi_{min}$.

The goal of the seller

To sell the product at the maximum possible price (π_{max}) which is defined as the value such that

- $\pi_{max} > \pi_{min}$
- $f(x_1, \ldots, x_{i-1}, \pi_{max}, x_{i+1}, \ldots, x_m) = POS$
- $f(x_1, \ldots, x_{i-1}, \pi_{max} + \epsilon, x_{i+1}, \ldots, x_m) = NEG, \forall \epsilon > 0.$

This goal is more precisely to maximise the expected profit

$$E(Profit(\pi)) = \hat{p}(POS|\pi) \cdot Profit(\pi)$$

where $\hat{p}$ is the estimated probability given by the negotiable feature model and $\hat{p}(POS|\pi) = \hat{p}(POS|\langle x_1, \ldots, x_{i-1}, \pi, x_{i+1}, \ldots, x_m \rangle)$.

The goal of the seller

To sell the product at the maximum possible price (π_{max}) which is defined as the value such that

- $\pi_{max} > \pi_{min}$
- $f(x_1, \ldots, x_{i-1}, \pi_{max}, x_{i+1}, \ldots, x_m) = POS$
- $f(x_1, \ldots, x_{i-1}, \pi_{max} + \epsilon, x_{i+1}, \ldots, x_m) = NEG, \forall \epsilon > 0$.

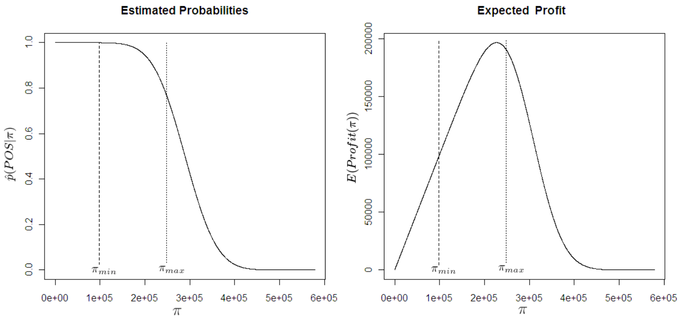
This goal is more precisely to maximise the expected profit

$$E(Profit(\pi)) = \hat{p}(POS|\pi) \cdot Profit(\pi)$$

where $\hat{p}$ is the estimated probability given by the negotiable feature model and $\hat{p}(POS|\pi) = \hat{p}(POS|\langle x_1, \ldots, x_{i-1}, \pi, x_{i+1}, \ldots, x_m \rangle)$.

Example

An estimated probability curve (Left) and its associated expected profit curve (Right).



Negotiation with one bid

To use the model and the expected profit formula to choose the price that has to be offered to the customer.

Negotiation with several bids

To use the model in order to set a sequence of bids to get the maximum overall expected profit.

For instance, the overall expected profit of a sequence of three bids is:

$E(Profit(\langle\pi_1,\pi_2,\pi_3\rangle)) = \hat{p}(POS|\pi_1)\cdot Profit(\pi_1) + (1-\hat{p}(POS|\pi_1))\cdot\hat{p}(POS|\pi_2)\cdot Profit(\pi_2) + (1-\hat{p}(POS|\pi_1))\cdot(1-\hat{p}(POS|\pi_2))\cdot\hat{p}(POS|\pi_3)\cdot Profit(\pi_3)$,

where $\pi_1 > \pi_2 > \pi_3 \geq \pi_{min}$.

Negotiation with one bid

To use the model and the expected profit formula to choose the price that has to be offered to the customer.

Negotiation with several bids

To use the model in order to set a sequence of bids to get the maximum overall expected profit.

For instance, the overall expected profit of a sequence of three bids is:

$E(Profit(\langle \pi_1, \pi_2, \pi_3 \rangle)) = \hat{p}(POS|\pi_1) \cdot Profit(\pi_1) + (1 - \hat{p}(POS|\pi_1)) \cdot \hat{p}(POS|\pi_2) \cdot Profit(\pi_2) + (1 - \hat{p}(POS|\pi_1)) \cdot (1 - \hat{p}(POS|\pi_2)) \cdot \hat{p}(POS|\pi_3) \cdot Profit(\pi_3)$,

where $\pi_1 > \pi_2 > \pi_3 \geq \pi_{min}$.

Negotiation Strategies

Baseline method

To increase a percentage α the minimum price (obtained from the training set).

- 1 bid: To increase the minimum price by α.
- N bids: The value of α will increase or decrease by $\alpha/(N+1)$ in each bid.

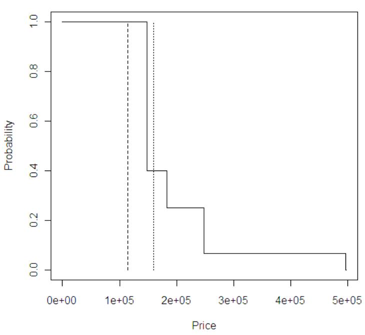
Example

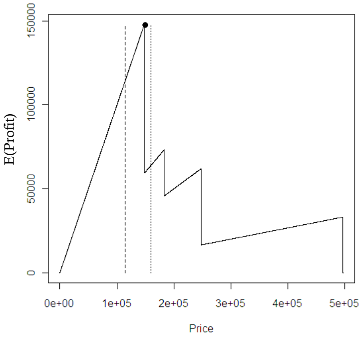
If we obtain that the best α is 0.4

- For 1 bid, the best profit will be obtained increasing in 40% the minimum price.
- For 3 bids, the three values of α would be 50%, 40% and 30%.

Maximum Expected Profit (MEP) strategy (1 bid)

This strategy chooses the price that maximises the value of the expected profit. $\pi_{MEP} = argmax_{\pi}(E(Profit(\pi)))$.

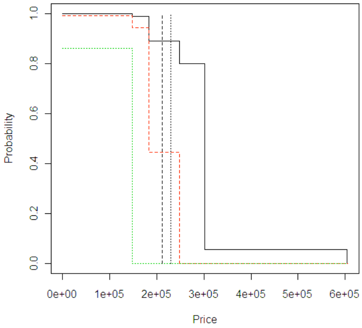


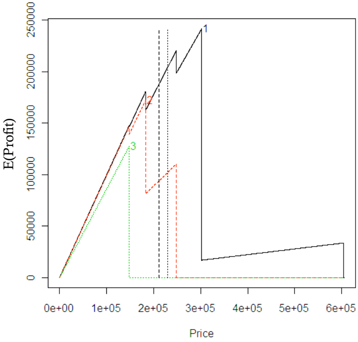


Best Local Expected Profit (BLEP) strategy (N bids).

Consists in applying the MEP strategy iteratively

- The first offer is the MEP.
- If it is not accepted, the estimated probabilities curve is normalised.
- The next offer will be calculated by applying the MEP strategy to the normalised probabilities.

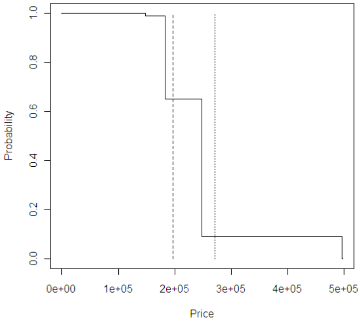


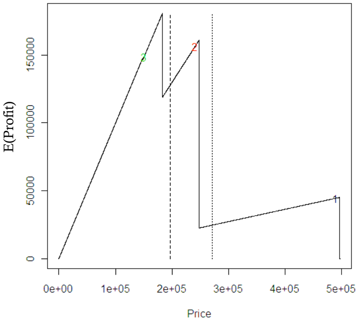


Maximum Global Optimisation (MGO) strategy (N bids).

To obtain the N offers that maximise the expected profit

$$\pi_{MGO} = argmax_{\langle \pi_1,\ldots,\pi_N \rangle}(\hat{p}(POS|\pi_1) \cdot Profit(\pi_1) + (1-\hat{p}(POS|\pi_1)) \cdot \hat{p}(POS|\pi_2) \cdot Profit(\pi_2) + \ldots + (1-\hat{p}(POS|\pi_1)) \cdot \ldots \cdot (1-\hat{p}(POS|\pi_{N-1})) \cdot \hat{p}(POS|\pi_N) \cdot Profit(\pi_N)).$$



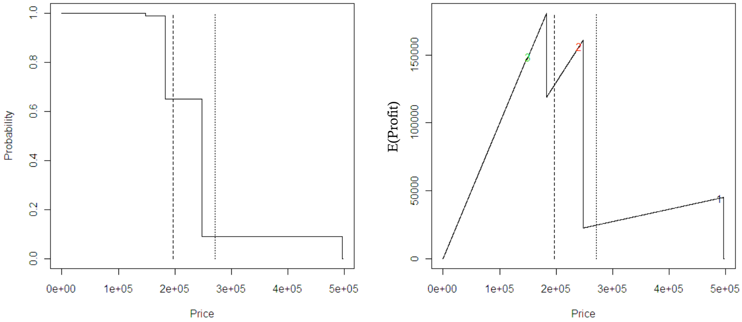


One option is just using a Montecarlo approach.

Maximum Global Optimisation (MGO) strategy (N bids).

To obtain the N offers that maximise the expected profit

$$\pi_{MGO} = argmax_{\langle \pi_1, \ldots, \pi_N \rangle} (\hat{p}(POS|\pi_1) \cdot Profit(\pi_1) + (1 - \hat{p}(POS|\pi_1)) \cdot \hat{p}(POS|\pi_2) \cdot Profit(\pi_2) + \ldots + (1 - \hat{p}(POS|\pi_1)) \cdot \ldots \cdot (1 - \hat{p}(POS|\pi_{N-1})) \cdot \hat{p}(POS|\pi_N) \cdot Profit(\pi_N)).$$



One option is just using a Montecarlo approach.

Experimental Results

Experimental Setting

- 2.800 properties (flats and houses) from an estate agents.
  - 10% training = 280 flats.
  - 90% test = 2.520 flats.
- The owners price is considered the Maximum price.
- We have used a *J48* decision tree (with Laplace correction and without pruning).
- The "inversion problem" solution has been implemented with the *LinearRegression* and *M5P* regression techniques.

Method	Sold flats	Sold price	Profit
All flats sold at π_{min}	2,520	356,959,593	0
All flats sold at π_{max}	2,520	712,580,216	355,620,623
1 bid			
Baseline (80%)	1,411	200,662,464	89,183,317
MEP (*J48*)	1,360	302,676,700	129,628,471
MEP (*LinearRegression*)	1,777	354,973,300	159,580,109
MEP (*M5P*)	1,783	358,504,700	161,736,313
3 bids			
Baseline (100%, 80%, 60%)	1,588	264,698,467	124,483,288
BLEP (*J48*)	1,940	382,921,400	173,381,116
BLEP (*LinearRegression*)	2,056	400,953,200	174,832,025
BLEP (*M5P*)	2,063	404,009,700	176,874,221
MGO (*J48*)	1,733	390,529,770	176,020,611
MGO (*LinearRegression*)	1,918	475,461,200	232,600,223
MGO (*M5P*)	1,906	476,171,900	234,259,509

Conclusions

- We have analysed the relation between negotiable features and problem presentation.
  - By using the monotonic dependency we can change of problem presentation to do the inversion problem.
  - The original problem is indirectly solved by estimating probabilities using a normal distribution.
- We have applied our approach to real negotiation problems.
- We have developed several negotiation strategies and we have seen how they behave for one or more bids, showing that our approach highly improves the results of the classical baseline method.

Future Work

- More complex scenarios
  - More than one customer and/or more than one product.
  - More than one negotiable feature at a time.
- More complex negotiation strategies and situations can be explored (e.g. the customer can counter-offer,...).