Small data AUC estimation for machine learning methods: Pitfalls and remedies

Tapio Pahikkala

University of Turku, Finland

2017

Binary classification in life sciences

- Two groups, (e.g. sick and healthy patients)
- The positive group stochastically larger than the negative group according to some real-valued function
- Real-valued predictions: predictions for the positive group should be larger than those for the negative group
- Task: perform prediction for new instances, minimize pairwise mistakes
- Area under ROC curve (AUC) criterion

Area under ROC curve (AUC) a.k.a. the Wilcoxon-Mann-Whitney U statistic

$$U = \sum_{i \in S_+} \sum_{j \in S_-} H(f(x_i) - f(x_j)) \qquad AUC = \frac{U}{|S_+||S_-|}$$

$$H(r) = \begin{cases} 1 & \text{if } r > 0 \\ 0.5 & \text{if } r = 0 \\ 0 & \text{if } r < 0 \end{cases}$$

Notation:

f	A real-valued prediction function
S	a set of data
$S_+ \subset S$	the set of positively labelled data in S
$S_- \subset S$	the set of positively labelled data in S
H	Heaviside function

Mann-Whitney U-test a.k.a. Wilcoxon rank sum test

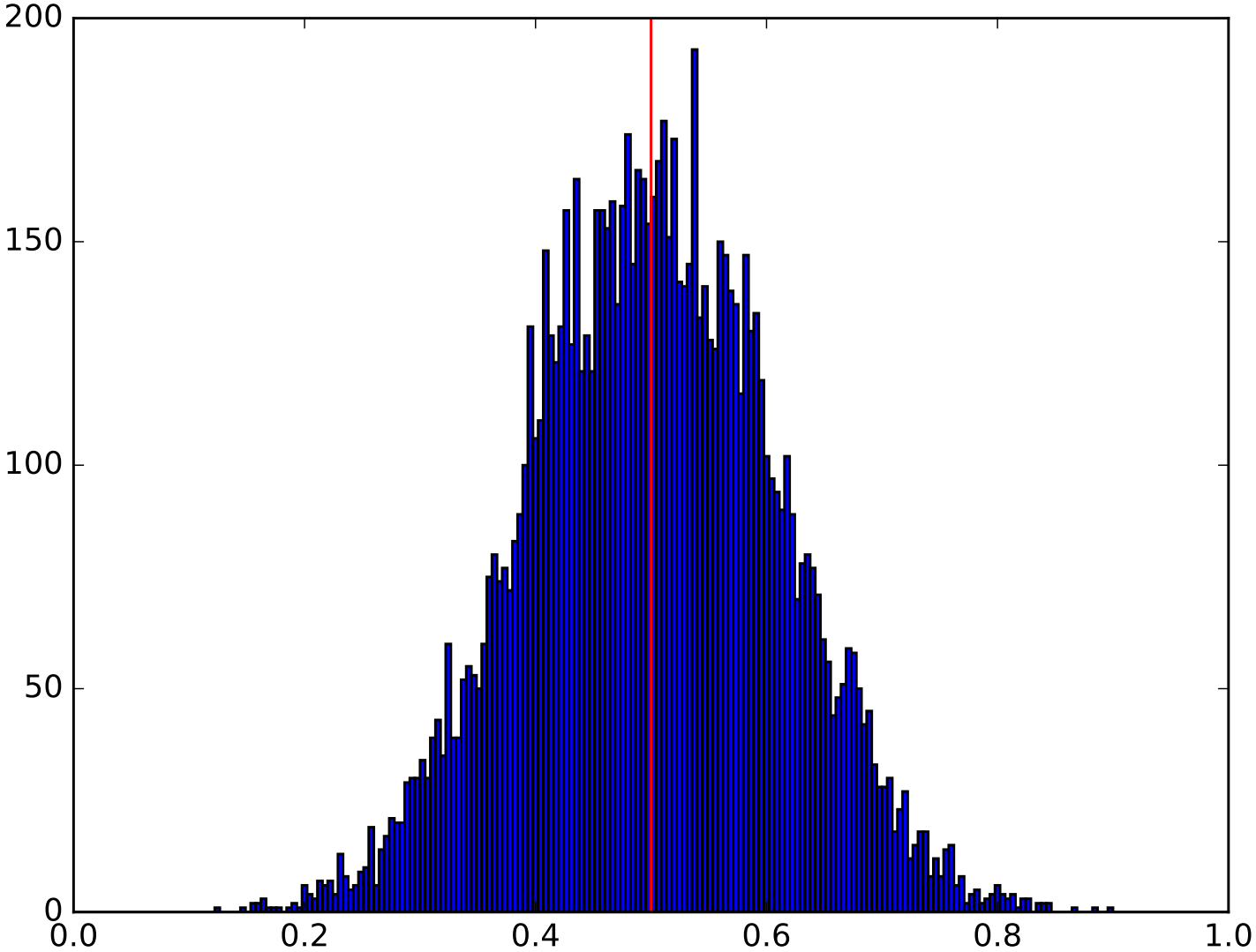
$$U = \sum_{i \in S_+} \sum_{j \in S_-} H(f(x_i) - f(x_j)) \qquad \mathrm{AUC} = \frac{U}{|S_+||S_-|}$$

- Test whether the predictions for the positive class tend to be larger than those for the negative class: $P(f(x_+) > f(x_-)) > P(f(x_-) > f(x_+))$
- The distribution of U-values under the null hypothesis is known
- With significance level, say 0.05, one can check whether U is larger than a certain known critical value
- The level indicates how likely it is to as large value of U as with randomly shuffled class labels (e.g. an implicit permutation test)

Notation:

f	A real-valued prediction function
S	a sample of data
$S_+ \subset S$	the set of positive examples in S
$S_- \subset S$	the set of negative examples in S
H	Heaviside function

15 pos & 15 neg WMW-*U* test AUC distribution



Cross-validation for machine learning methods

- Overfitting: model may fit training data arbitrarily well
- Cross-validation: standard procedure for small sample sizes
- Leave-one-out and tenfold most popular
- Not reliable for AUC-estimation

Reference

A. Airola, T. Pahikkala, W. Waegeman, B. De Baets, T. Salakoski. An experimental comparison of cross-validation techniques for estimating the area under the ROC curve. *Computational Statistics & Data Analysis*, 2011.

Example: Easy way to get perfect looking AUC values with leave-one-out

Balanced support vector machines with weights inversely proportional to class frequencies

- Available in most machine learning libraries
- Leaving out a positive/negative labelled example increases the the positive/negative class weight
- Svm models tend to predict large positive/negative values when positively/negatively labelled training data are held out
- Very easy to get perfect looking AUC values especially when combined with feature engineering and other optimization

Leave-pair-out cross-validation for AUC estimation

- Problem: dependencies between folds break cross-validation
- Solution: fold partition over pairwise preferences, not individual instances
- Leave-pair-out cross-validation
- Almost unbiased, smaller variance than alternatives

Reference

A. Airola, T. Pahikkala, W. Waegeman, B. De Baets, T. Salakoski. An experimental comparison of cross-validation techniques for estimating the area under the ROC curve. *Computational Statistics & Data Analysis*, 2011.

Leave-pair-out cross-validation for AUC estimation

$$AUC = \frac{1}{|S_+||S_-|} \sum_{i \in S_+} \sum_{j \in S_-} H\left(f_{S\setminus\{i,j\}}(x_i) - f_{S\setminus\{i,j\}}(x_j)\right)$$

Notation:

S	a sample of data
$S_+ \subset S$	the set of positive examples in S
$S_- \subset S$	the set of negative examples in S
$f_{S\setminus\{i,j\}}$	classifier trained without the i-th and j-th training example
H	Heaviside function

Distribution of LPOCV AUC estimates

- LPOCV AUC estimators are almost unbiased (see prior work)
- Variance may still be an issue
- Shape of the AUC estimate distribution?

Question with the most practical importance

How likely it is to get good looking AUC values from data with no signal e.g. with randomly assigned class labels.

Can we use LPOCV AUC estimates similarly to U-test?

$$U = \sum_{x_i \in S_+} \sum_{x_j \in S_-} H\left(f_{S\setminus\{i,j\}}(x_i) - f_{S\setminus\{i,j\}}(x_j)\right)$$

- The estimates are almost unbiased
- However, the distribution of U-values under the null hypothesis is in general NOT known
- The prediction functions inferred during different LPOCV rounds are different and there can be complex mutual dependencies between them

Abstract definition of a learning algorithm

- Assume that all data are associated with a natural number (e.g. the data are indexed)
- Consider an inducer (machine learning algorithm) as a mapping from a sequence of $\pm$ -labelled data index numbers to a real-valued function of natural numbers
- Feature representations, prior knowledge, etc. are assumed to be a part of the inducer

Definition: Inducer

$$A : \bigcup_{m=1}^{\infty} \{X \times (\pm 1)^m \mid X \subset \mathbb{N}^m\} \to \mathbb{R}^{\mathbb{N}} \qquad A(S) \mapsto f_S$$

Notation

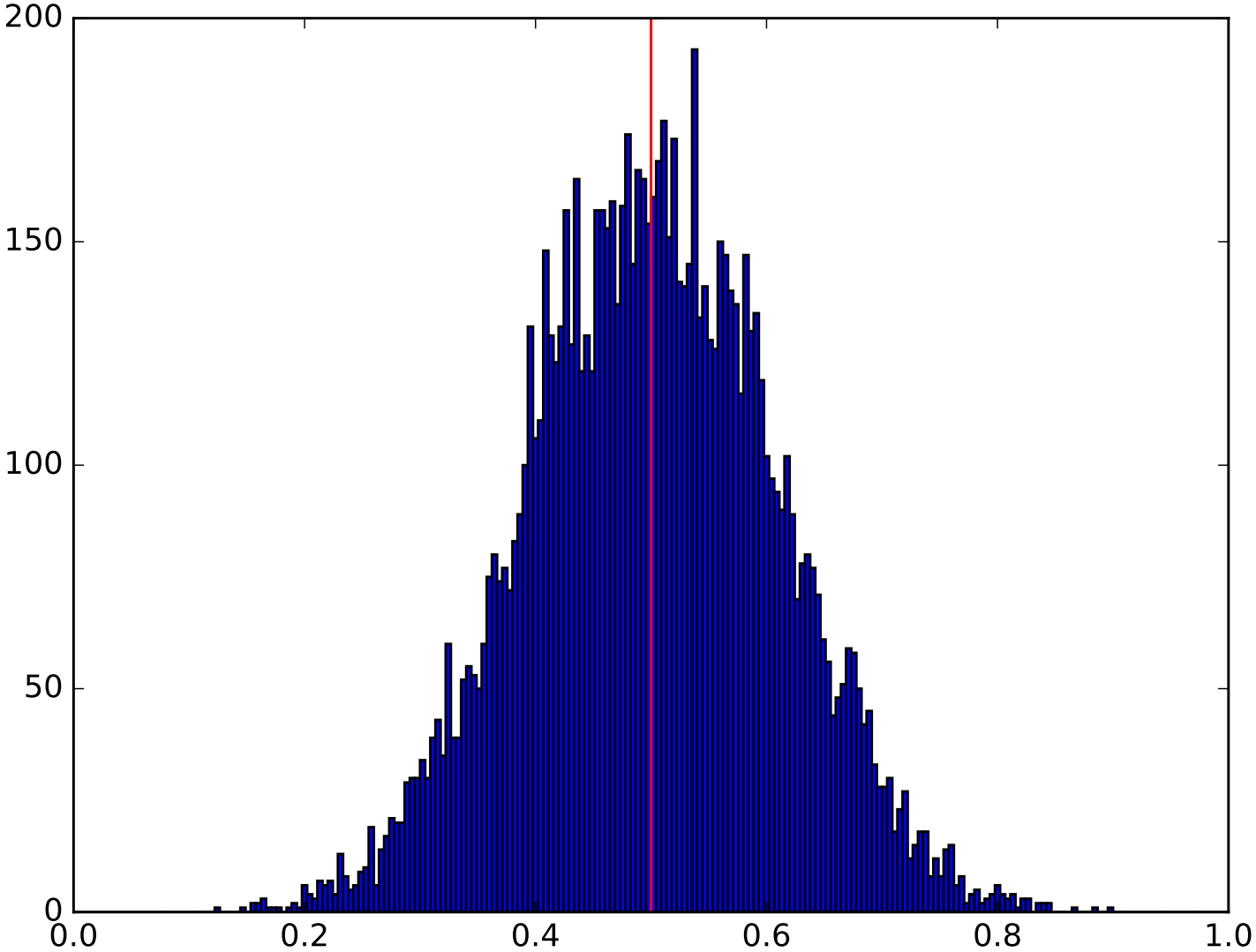
A	Inducer (aka machine learning algorithm)
$\mathbb{R}^{\mathbb{N}}$	the set of all mappings from the index numbers to real values
f_S	model trained with S

Can we use LPOCV AUC estimates similarly to *U*-test?

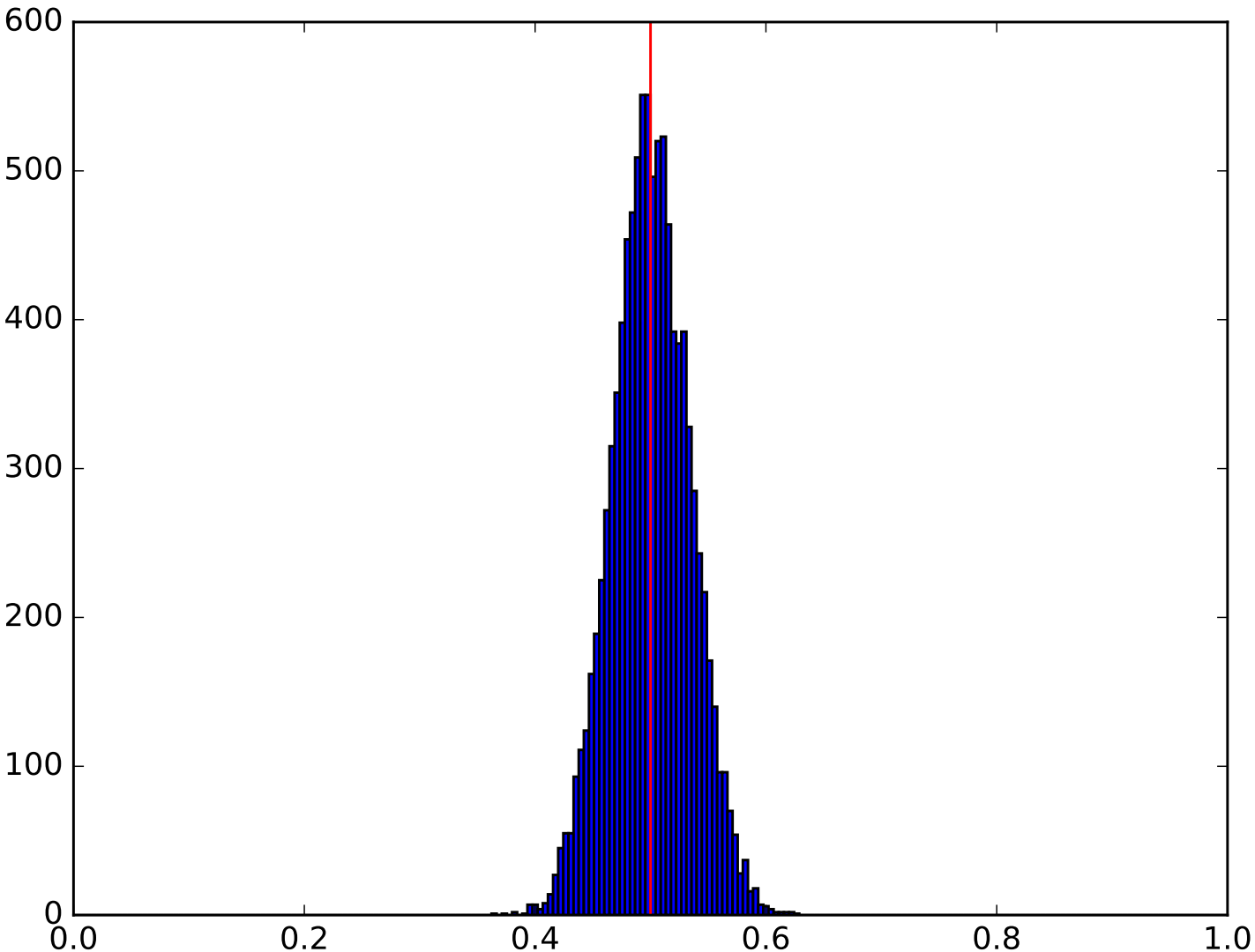
Examples:

- If the learning algorithm is perfectly stable (i.e. leaving out part of the training data does not change the prediction function the algorithm infers), the *U*-value distribution is the same as that of the standard *U* test
- If the inferred prediction functions make random predictions, the *U*-value distribution has considerably low variance
- However, for many learning algorithms, the variance of the *U*-value distribution can be considerably larger than that of the WMW-U test

15 pos & 15 neg WMW-*U* test AUC distribution



15 pos & 15 neg, LPOCV AUC distribution with random predictions



A simple learning algorithm example

Order learner

Consider an algorithm that infers from a training set of $\pm$ -labelled indices whether their order is more ascending or descending and performs prediction accordingly:

- Input a training set

  $$\{(x_1,+1),\ldots,(x_m,+1),(x_{m+1},-1),\ldots,(x_{m+n},-1)\} \in (\mathbb{N}\times\{\pm 1\})^{m+n}$$

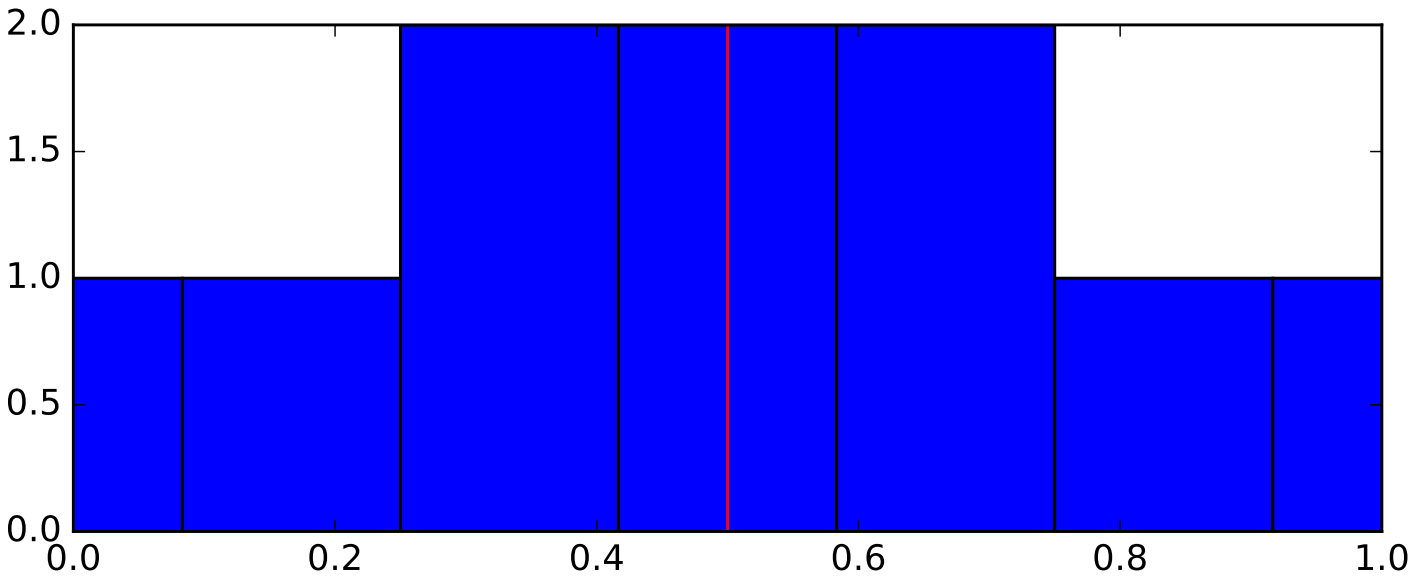
  of m positive and n negative labelled index numbers
- Infer an ascending order (e.g. $f(x)=x$) if $U > mn/2$ on the training set
- Infer a descending order (e.g. $f(x)=-x$) if $U < mn/2$ on the training set
- Infer ties (e.g. $f(x)=0$) if $U = mn/2$ on the training set

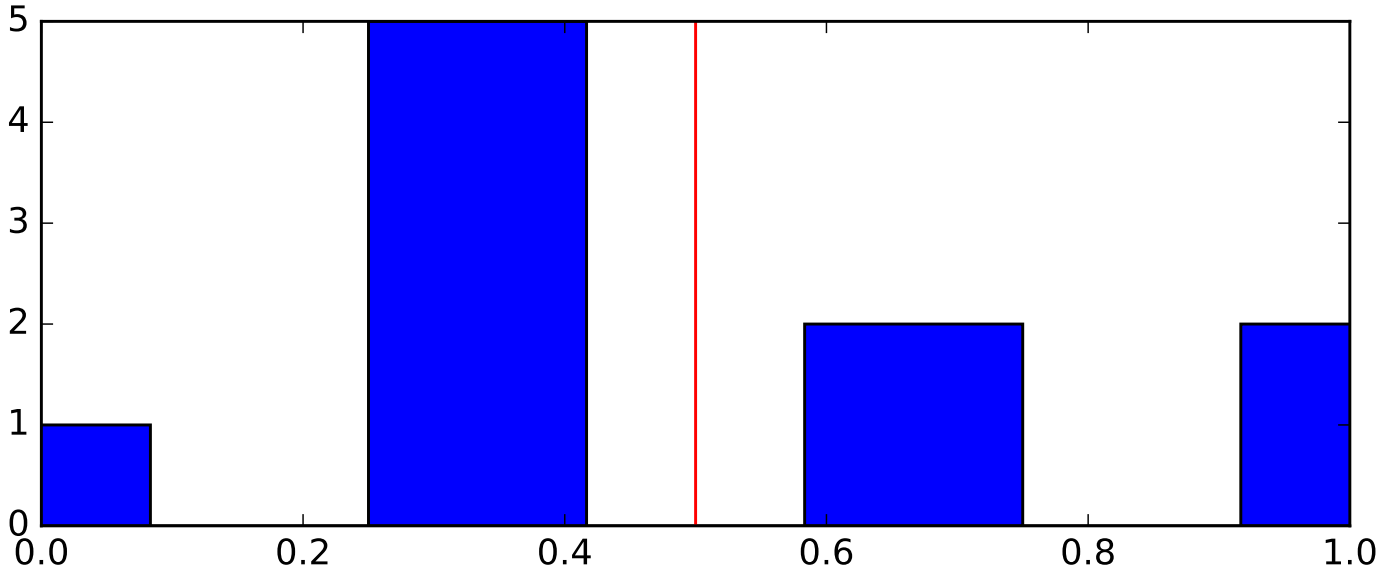
Order learner with 5 data points

WMW-U and LPOCV-U values for all possible assignments of two negative and three positive labels for five data points:

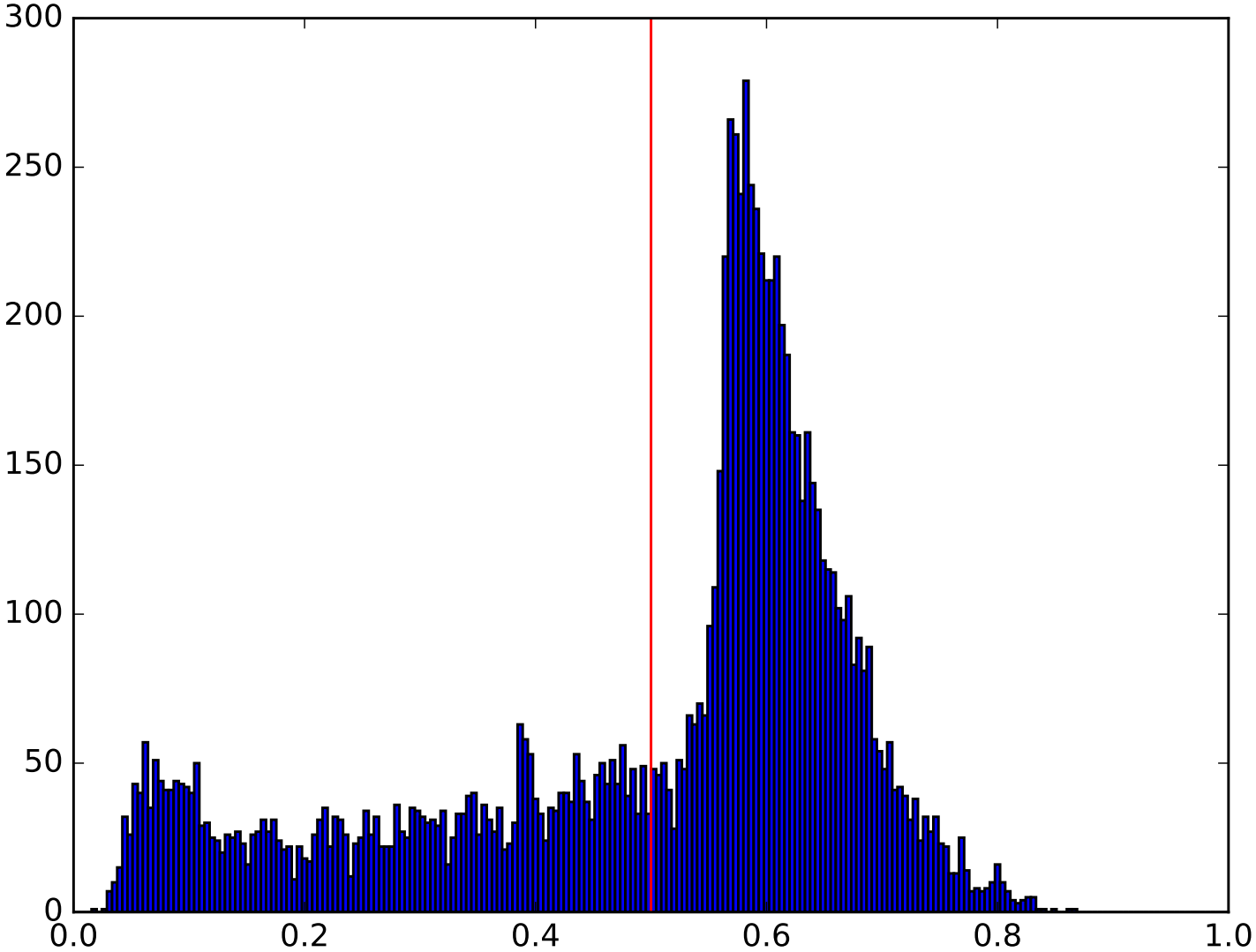
$x_1 = 1$	$x_2 = 2$	$x_3 = 3$	$x_4 = 4$	$x_5 = 5$	WMW-U	LPOCV-U
-1	-1	1	1	1	6	6
-1	1	-1	1	1	5	4
-1	1	1	-1	1	4	2
-1	1	1	1	-1	3	0
1	-1	-1	1	1	4	2
1	-1	1	-1	1	3	2
1	-1	1	1	-1	2	2
1	1	-1	-1	1	2	2
1	1	-1	1	-1	1	4
1	1	1	-1	-1	0	6
					$\mu = 3$ $\sigma^2 = 3$	$\mu = 3$ $\sigma^2 = 3.4$

Order learner with 5 data points

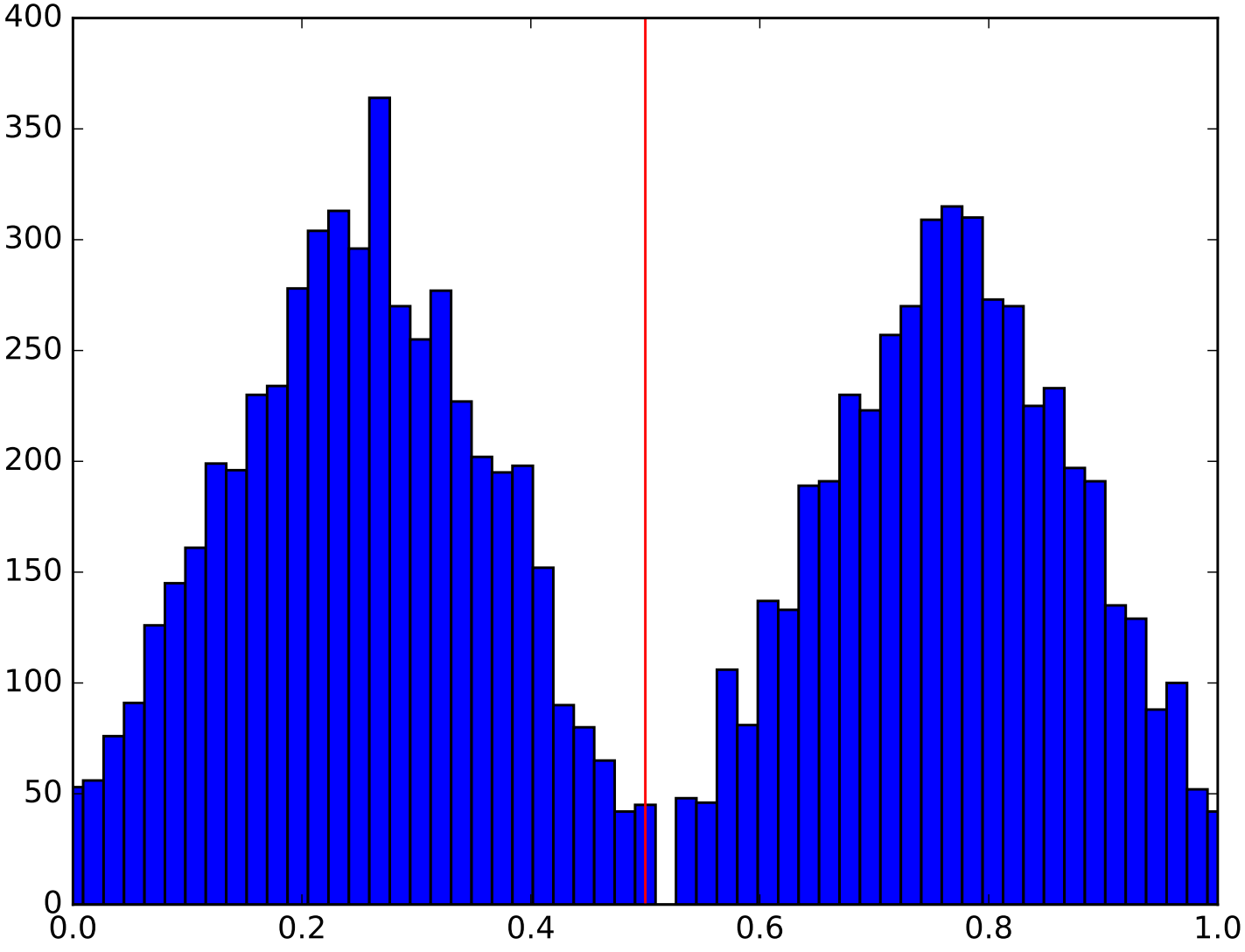




15 pos & 15 neg, Order learner, LPOCV AUC distribution



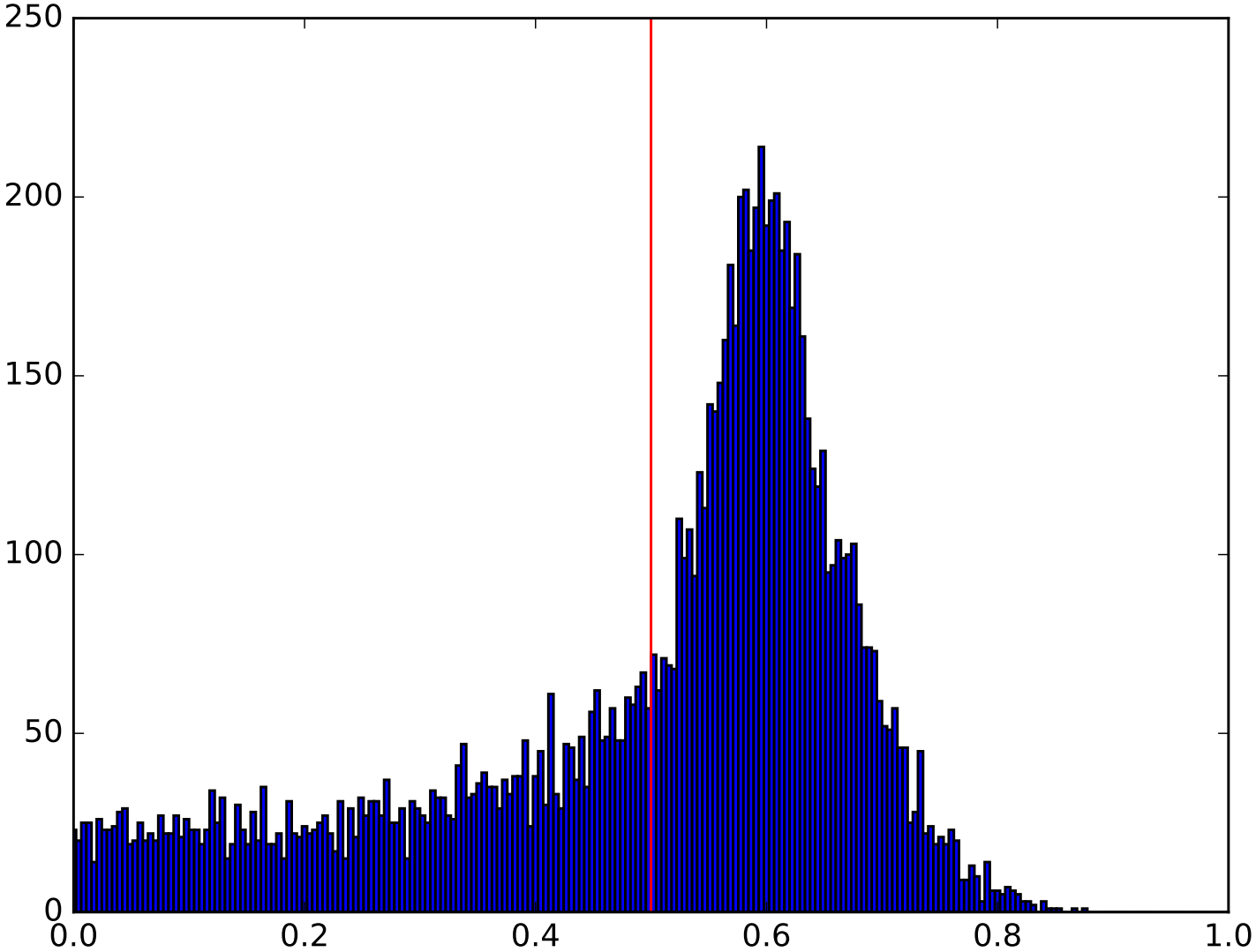
28 pos & 2 neg, Order learner, LPOCV AUC distribution



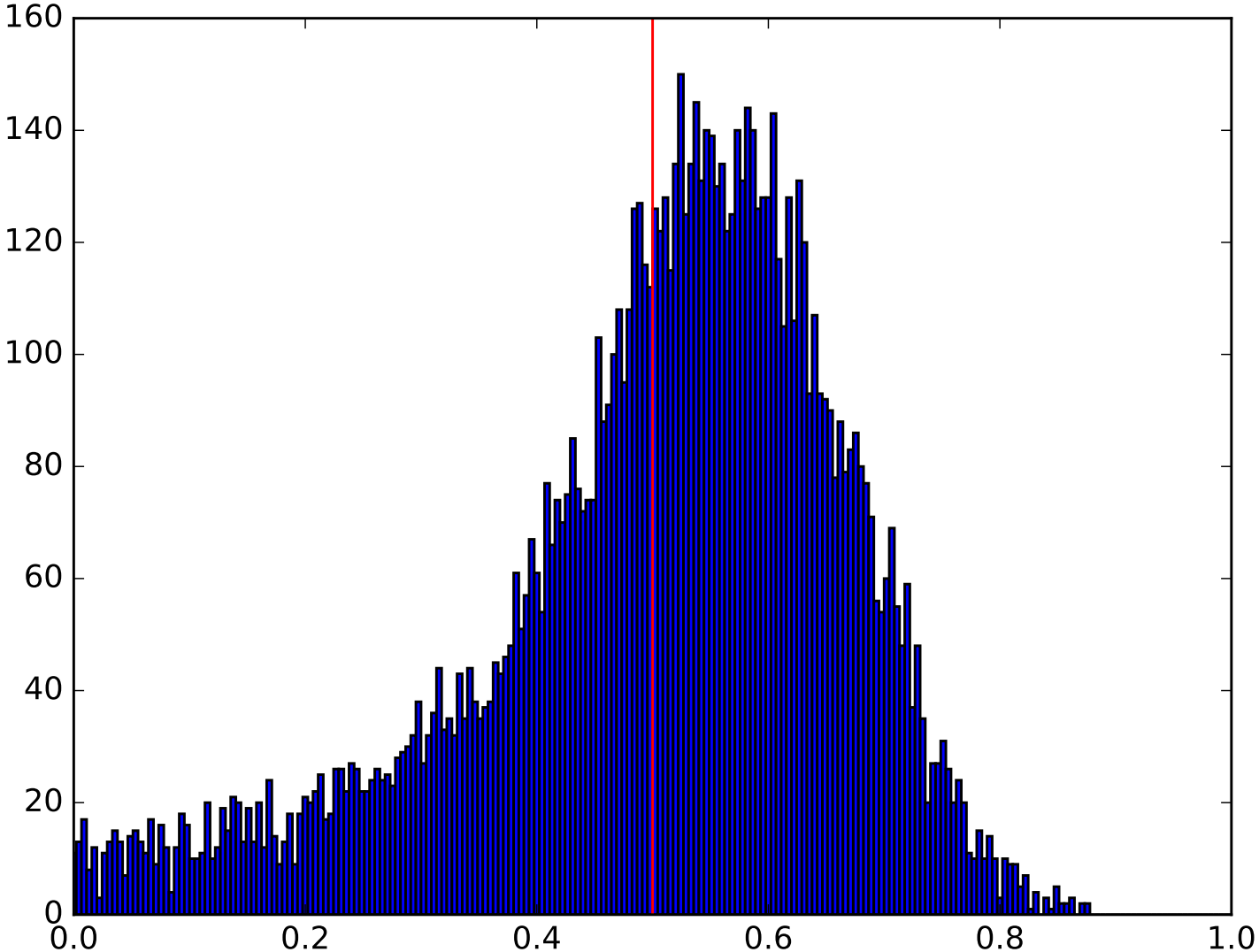
Simulation study

- Data set of 30 points drawn from multi-variate normal distribution
- 15 points is assigned a positive and 15 a negative class label
- Class labels are randomly assigned
- The random assignment repeated 10000 times to estimate the AUC distribution
- Support vector machine as a learning algorithm

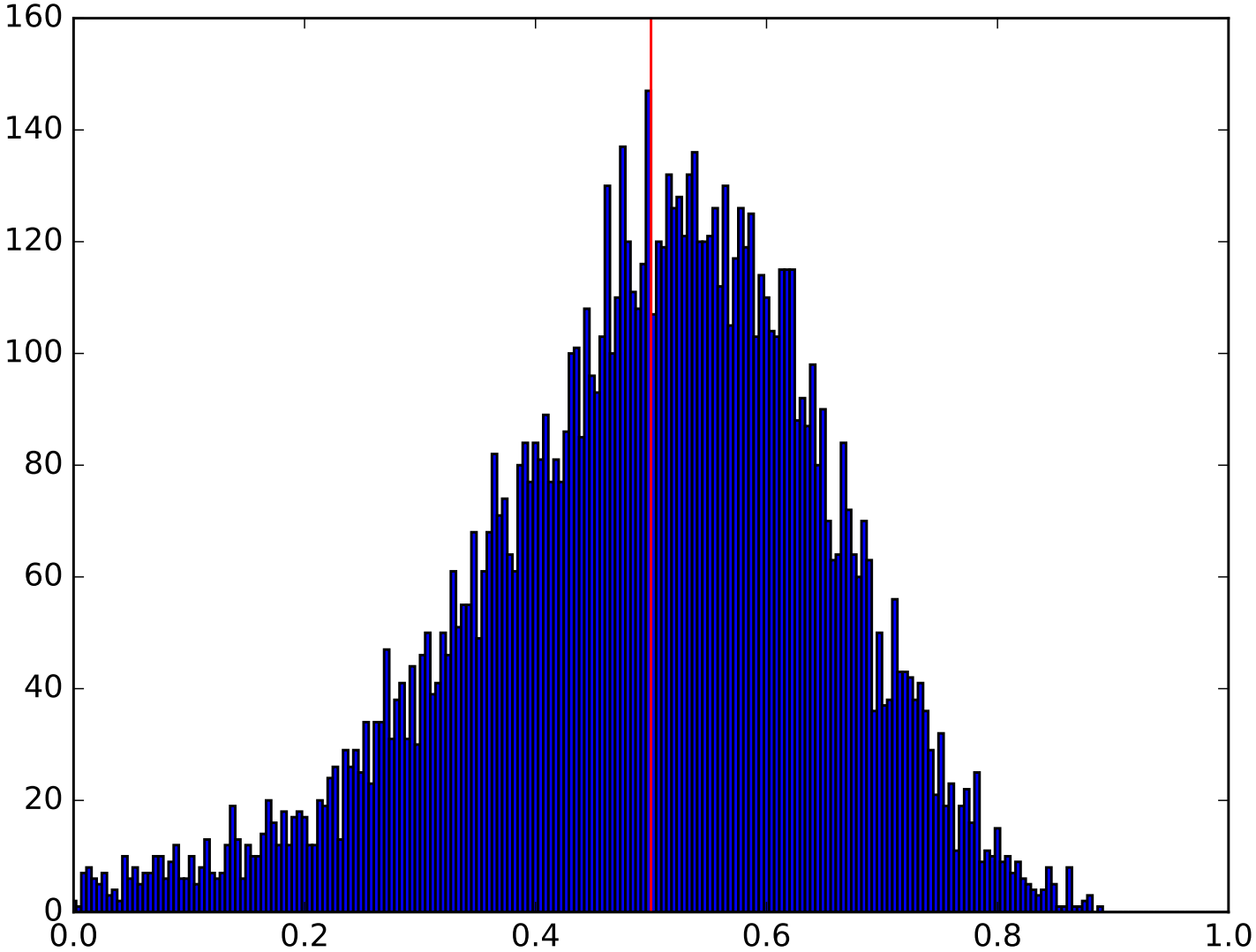
Support vector machine, LPOCV AUC distribution, 1 feature



Support vector machine, LPOCV AUC distribution, 2 features

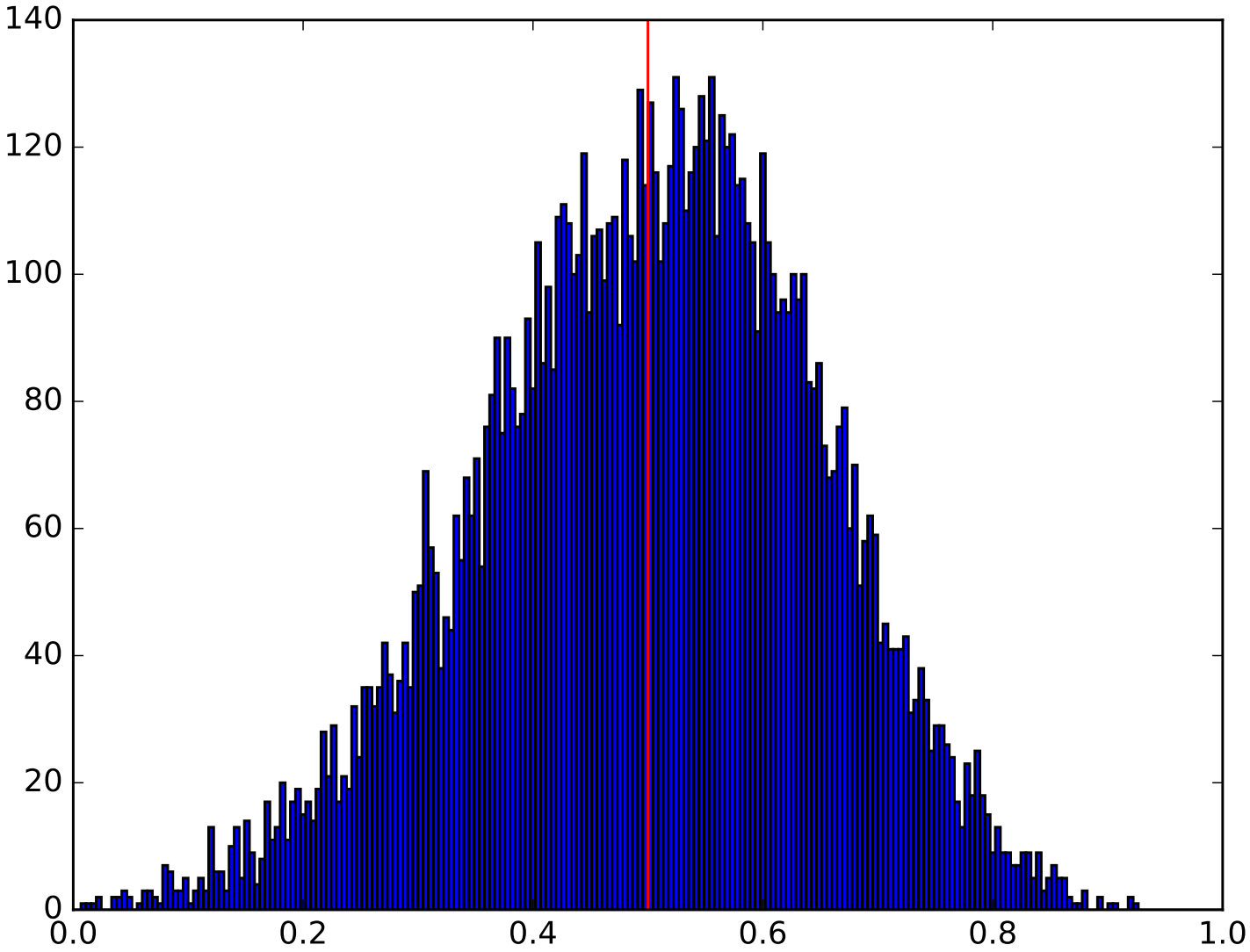


Support vector machine, LPOCV AUC distribution, 4 features

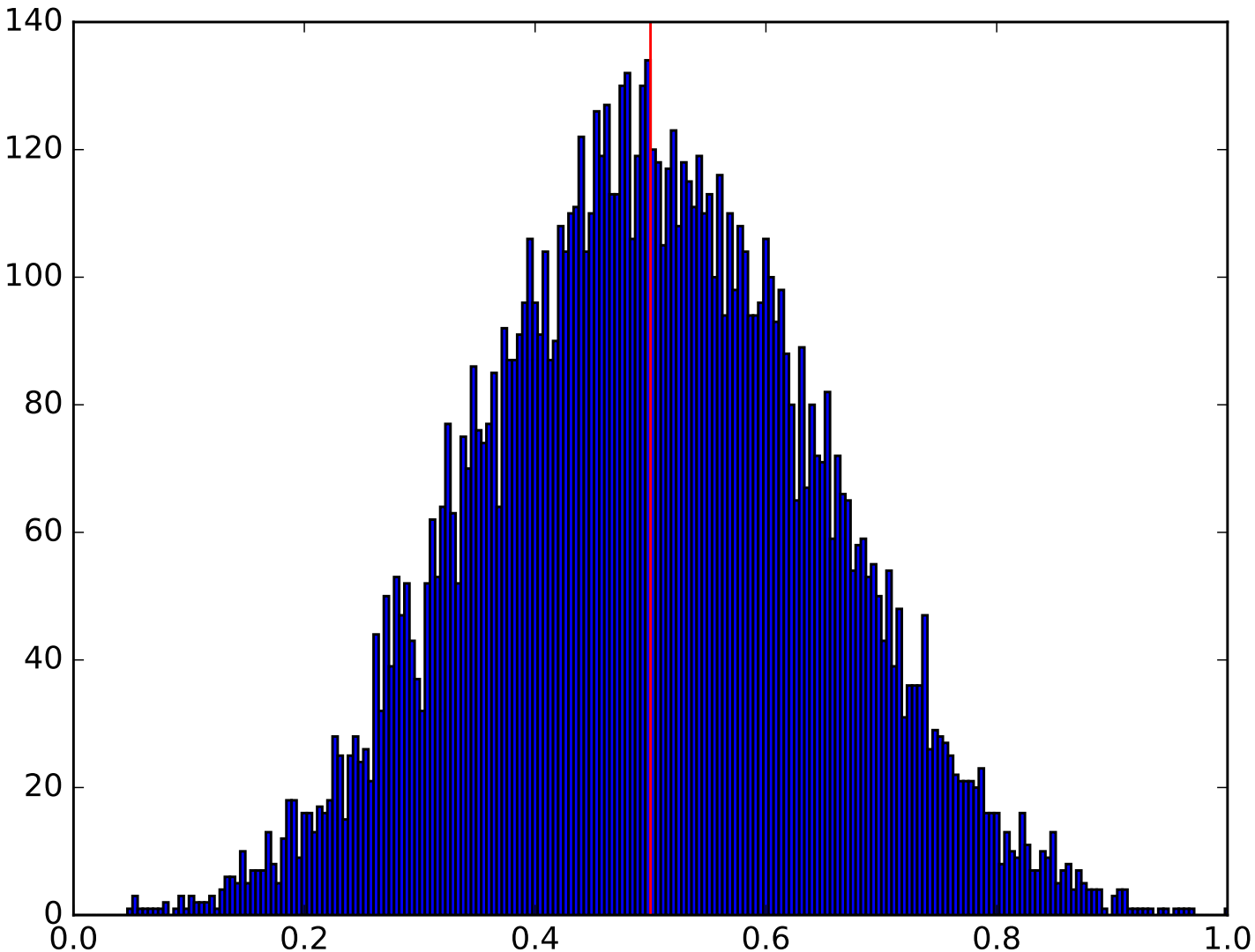


Support vector machine, LPOCV AUC distribution, 8 features

h



Support vector machine, LPOCV AUC distribution, 16 features



How to perform WMW-type of U-test with LPOCV

Brute force approach with a direct permutation test:

- Sample a set of random permutations of the labels, say 10000
- Estimate the U-value distribution of the LPOCV with the learning algorithm on the data set under consideration
- Determine the critical U-values corresponding to the desired significance level from the estimated distribution

U-test unconditional to inducer?

Recall:

Definition: Inducer

$$A : \bigcup_{m=1}^{\infty} \{X \times (\pm 1)^m \mid X \subset \mathbb{N}^m\} \to \mathbb{R}^{\mathbb{N}} \qquad A(S) \mapsto f_S$$

Observation

If we fix the numbers m and n of the positively and negatively labelled data, we observe that LPOCV divides the set of all possible inducers into a finite set of equivalence classes corresponding to the number of possible AUC value distributions.

U-test unconditional to inducer?

Definition: LPOCV mapping

Let

$$L_{m,n} : \mathscr{A} \times \mathscr{C}_{m,n} \to [0,1] \qquad (A, C) \mapsto L_{m,n}(A, C)$$

denote a function that maps an inducer $A \in \mathscr{A}$ and a labelling $C \in \mathscr{C}_{m,n}$ to the corresponding LPOCV AUC estimate $L_{m,n}(A, C)$.

Notation

$\mathscr{A}$	The set of all possible inducers
$\mathscr{C}_{m,n}$	The set of all possible labellings of $m+n$ points into m positive and n negative points e.g. $\lvert\mathscr{C}_{m,n}\rvert = \binom{m+n}{m}$

U-test unconditional to inducer?

U-test unconditional to inducer

Given a significance level, say $\alpha = 0.05$, and the numbers m and n, the two groups determined by a labelling C can be significantly separated by an inducer A if

$$L_{m,n}(A, C) > \beta_{\alpha,m,n}$$

where $\beta_{\alpha,m,n} \in (0.5, 1]$ is a critical value of AUC.

Question: what are the critical values $\beta_{\alpha,m,n} \in (0.5, 1]$?

Open problem

Given a significance level, say $\alpha = 0.05$, and the numbers m and n, what is the minimum AUC value $\beta_{\alpha,m,n} \in (0.5, 1]$ such that

$$\max_{A \in \mathscr{A}} \left(\frac{|\{C \in \mathscr{C}_{m,n} \mid L_{m,n}(A, C) > \beta_{\alpha,m,n}\}|}{|\{C \in \mathscr{C}_{m,n}\}|} \right) < \alpha$$

Solvable with combinatorial optimization?

RLScore

RLScore: Regularized Least-Squares based algorithm library.

- Includes fast leave-pair-out cross-validation algorithm implementations

Available at

`http://staff.cs.utu.fi/~aatapa/software/RLScore/`

Reference

Tapio Pahikkala and Antti Airola. Rlscore: Regularized least-squares learners. *Journal of Machine Learning Research*, 17(221):1–5, 2016.