Evaluation

- Take standard UCI datasets, produce training/test split

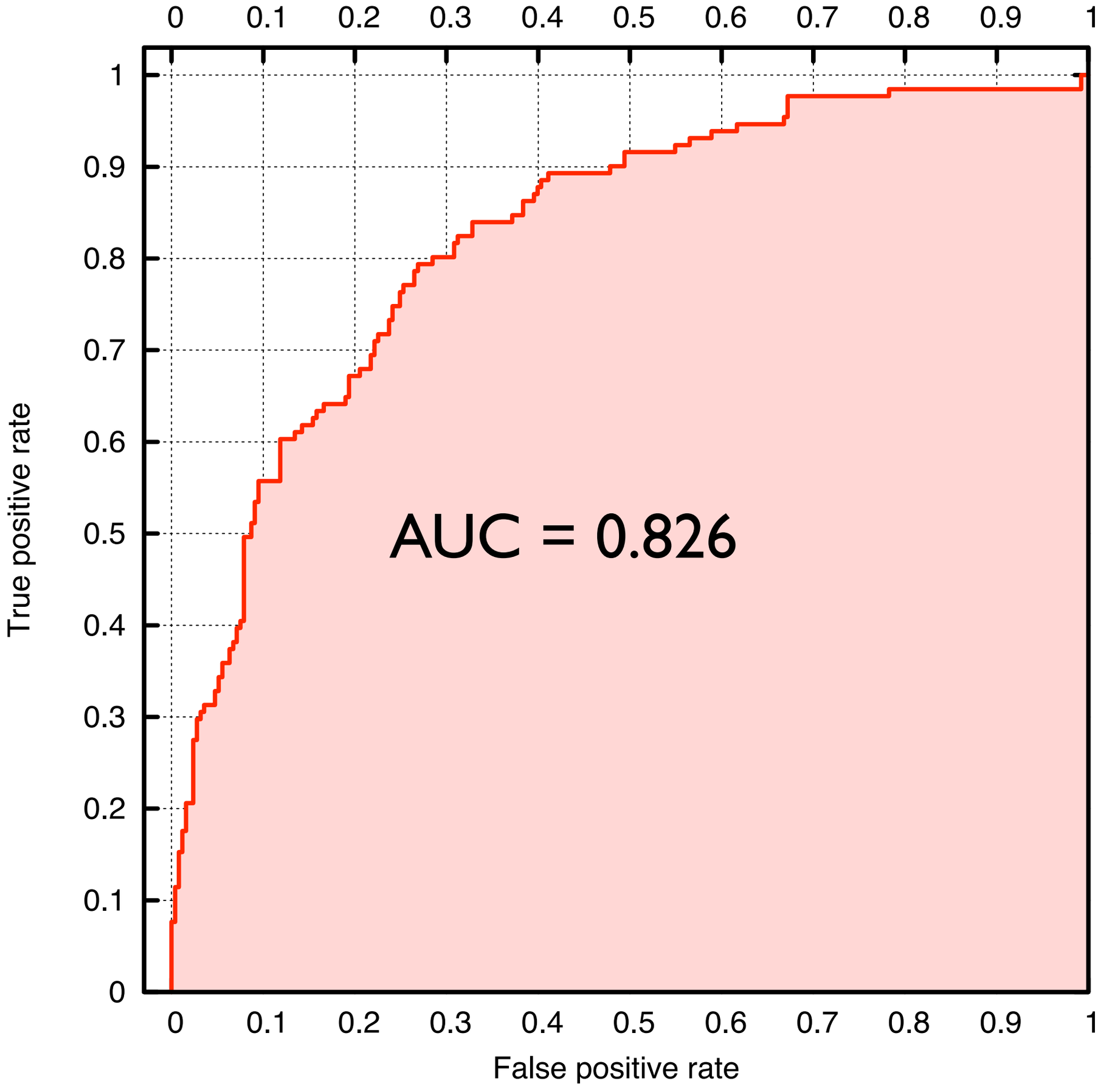
- Bayesian probit regression model with diffuse prior (Albert & Chib, 1993)

- Run Gibbs sampler to produce 1,000 draws from the posterior predictive distribution over the test data, compute marginal posterior means

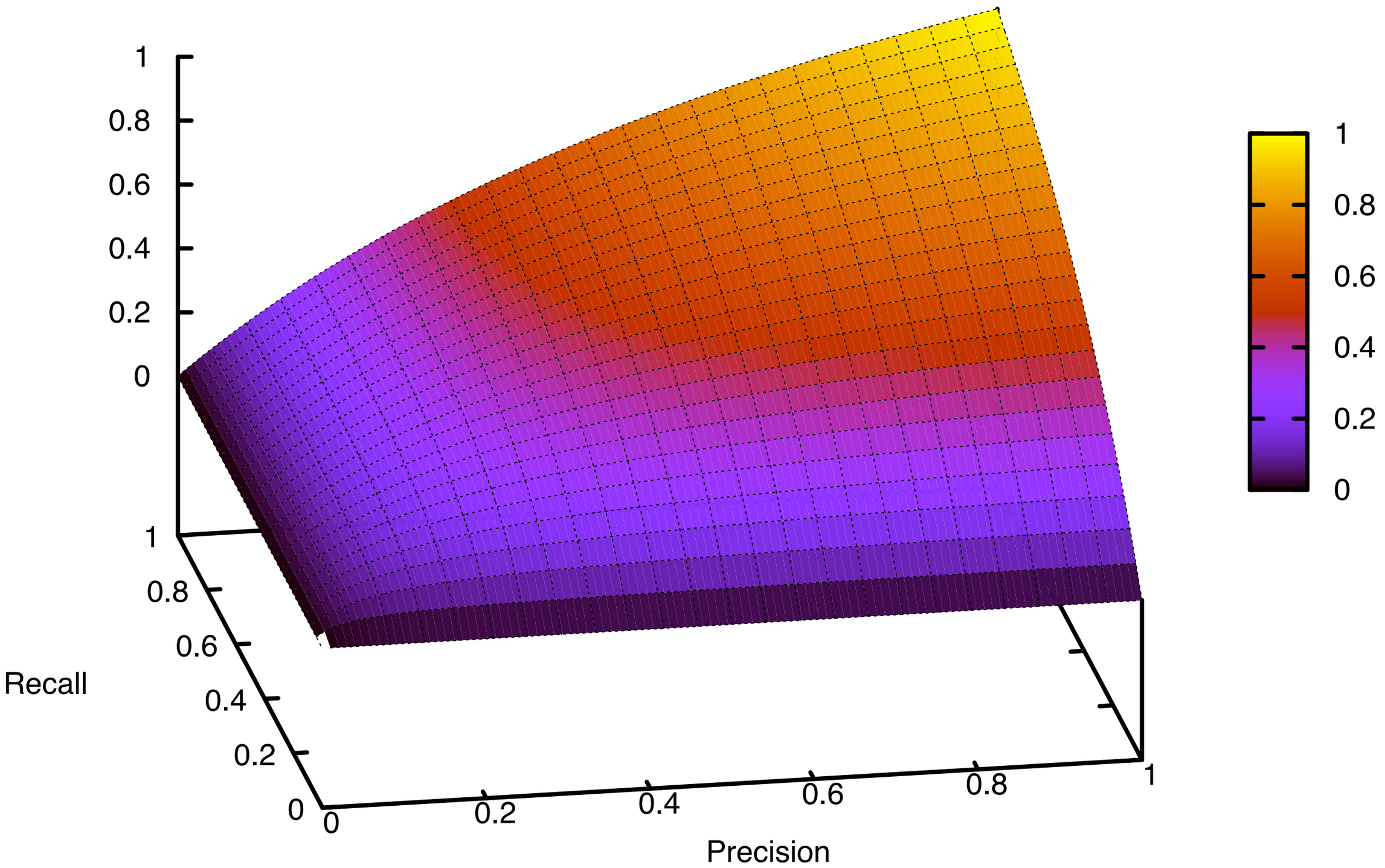
- Use posterior means of $p(y{=}1 \mid x)$ as input to the decoder

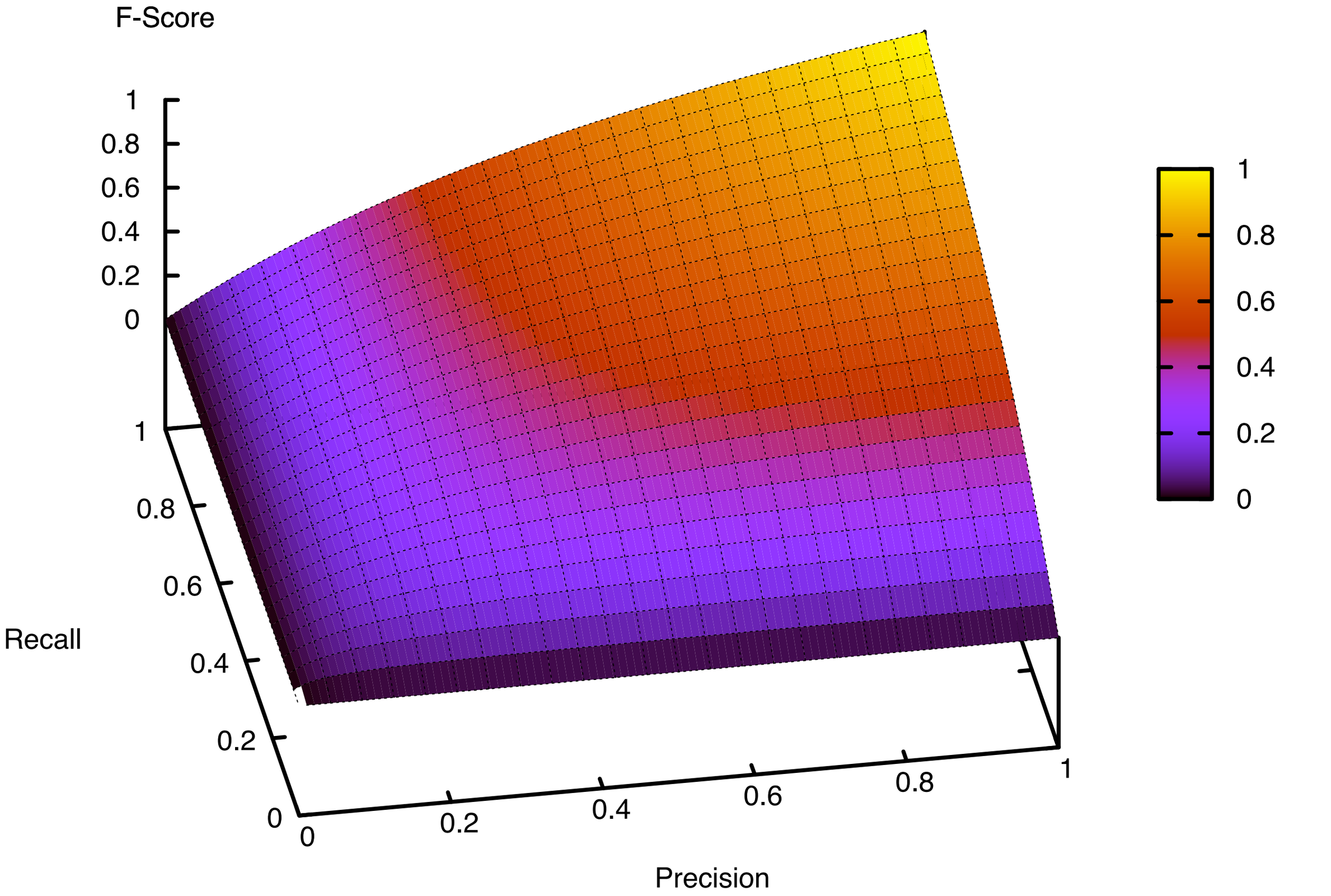
Diabetes

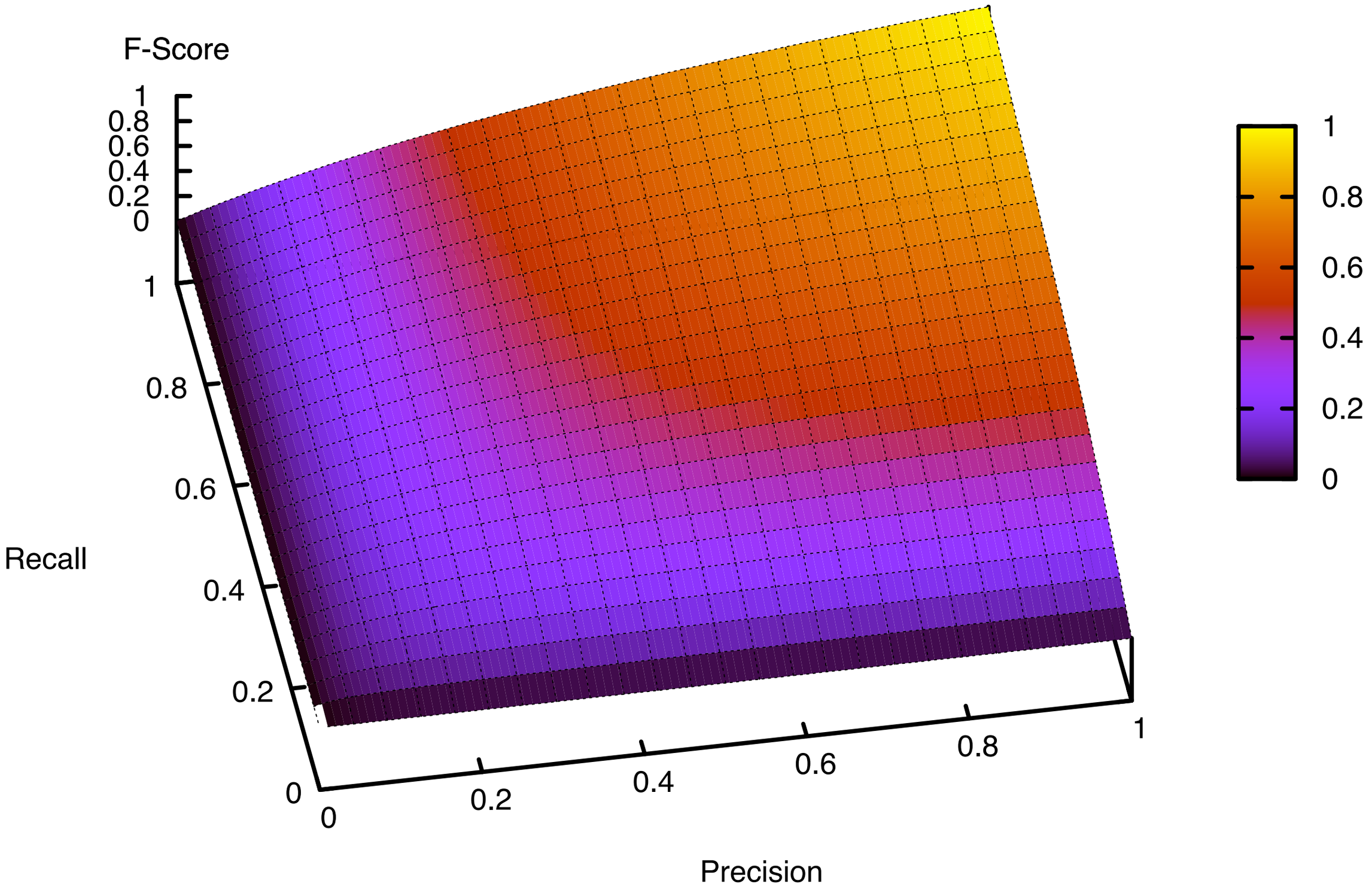
- 384 training instances, 384 test instances
- 8 numeric features (plus intercept)
- 36% positive labels

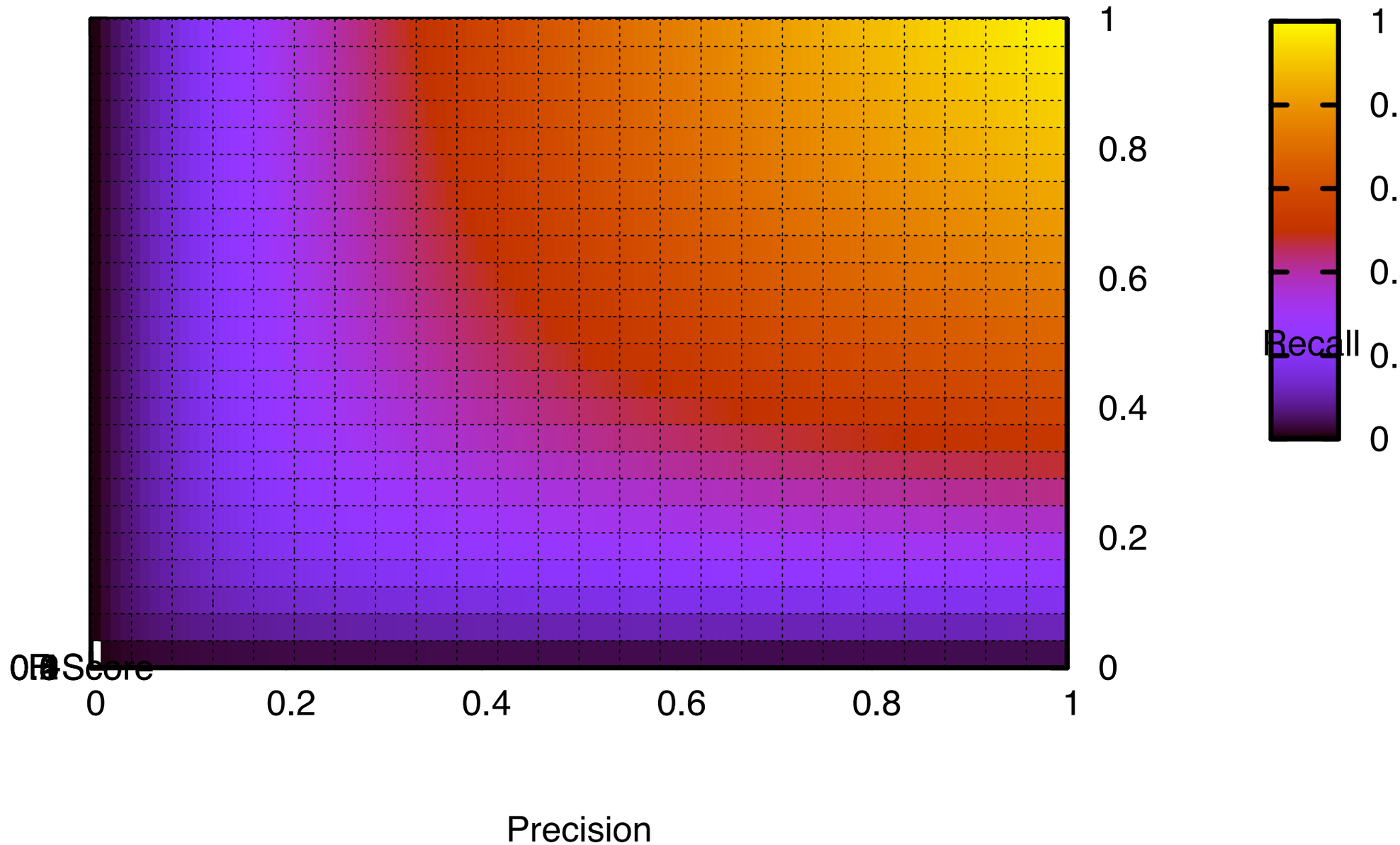

AUC = 0.826
True positive rate
False positive rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

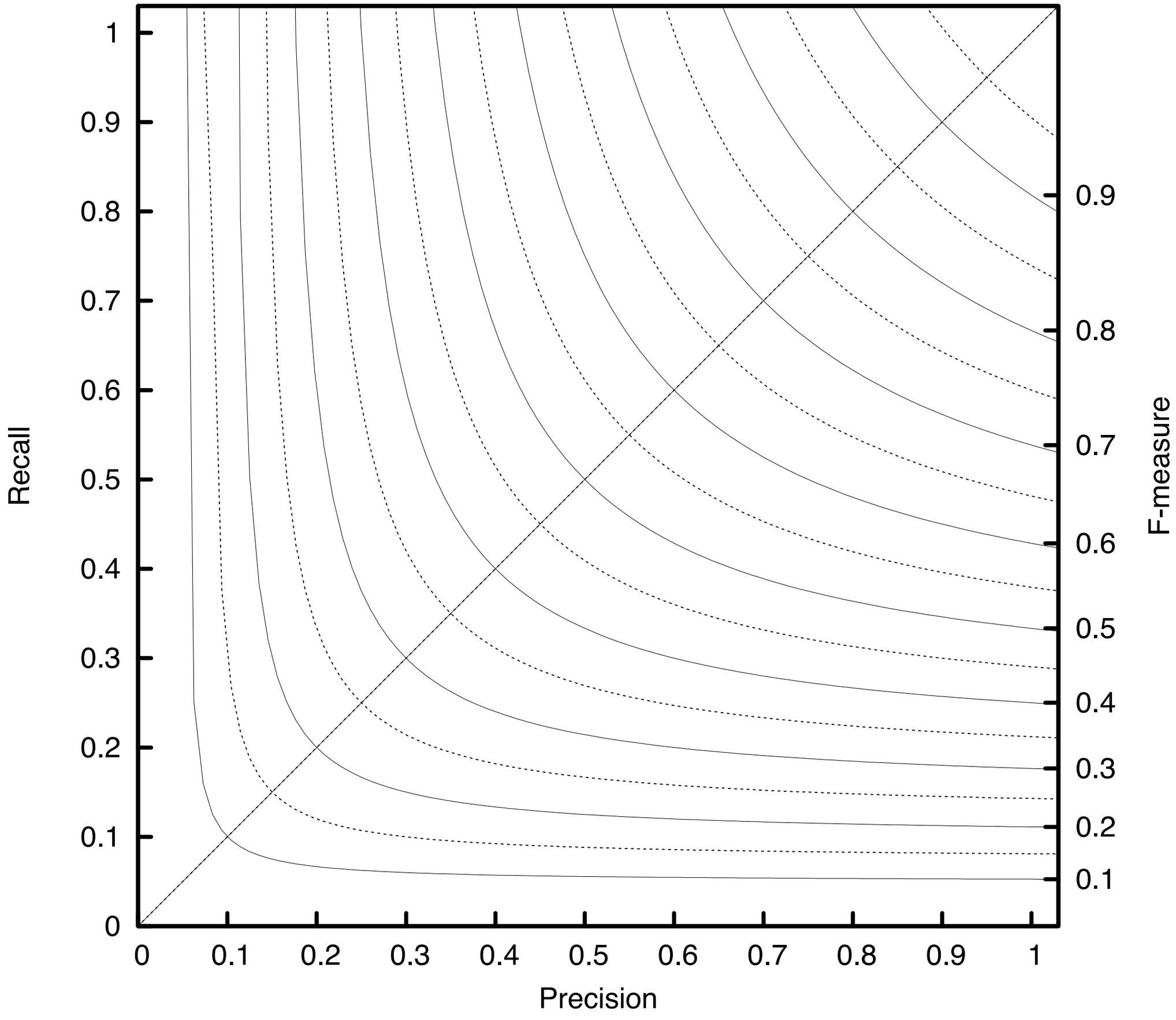
F-Score

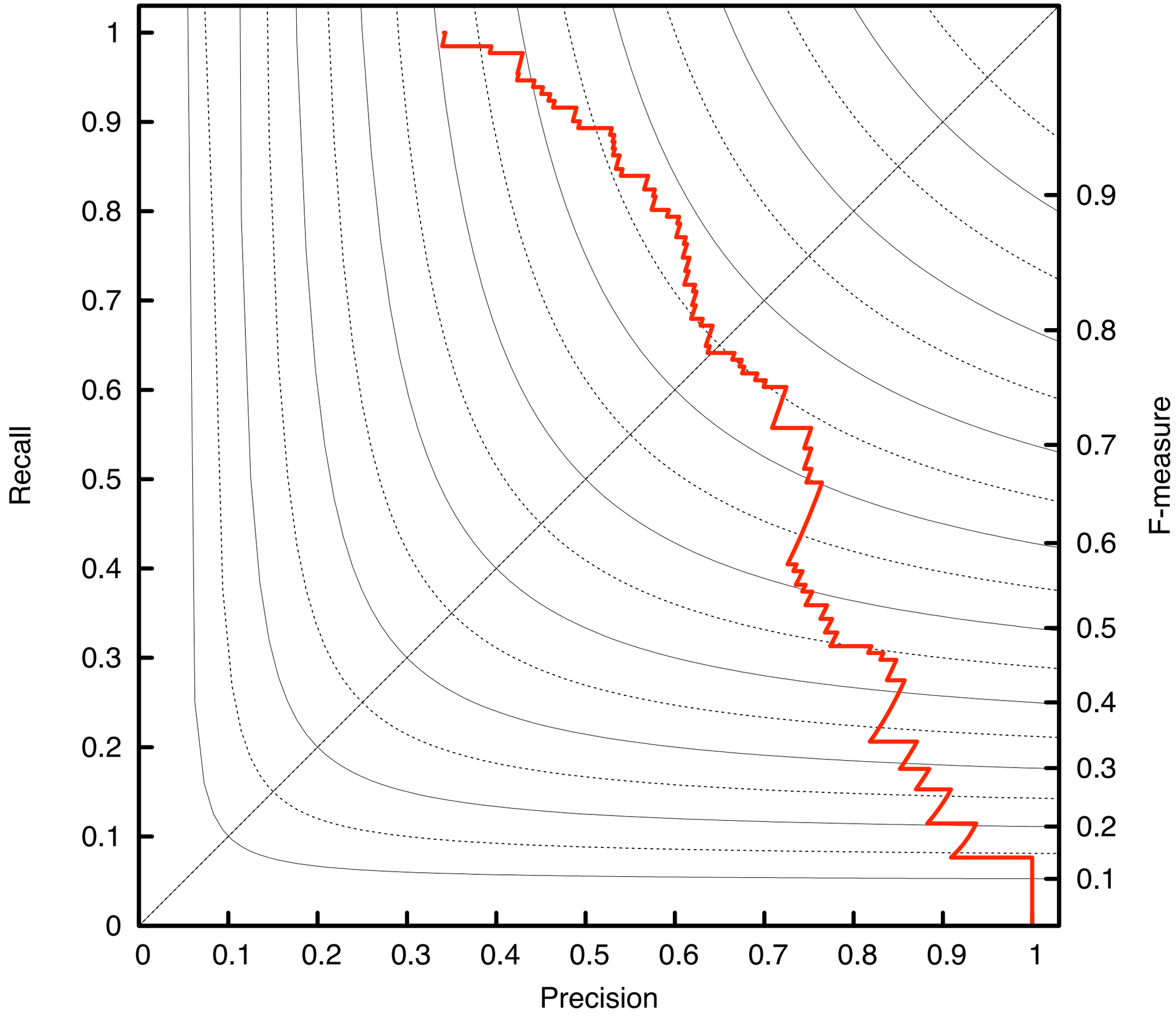


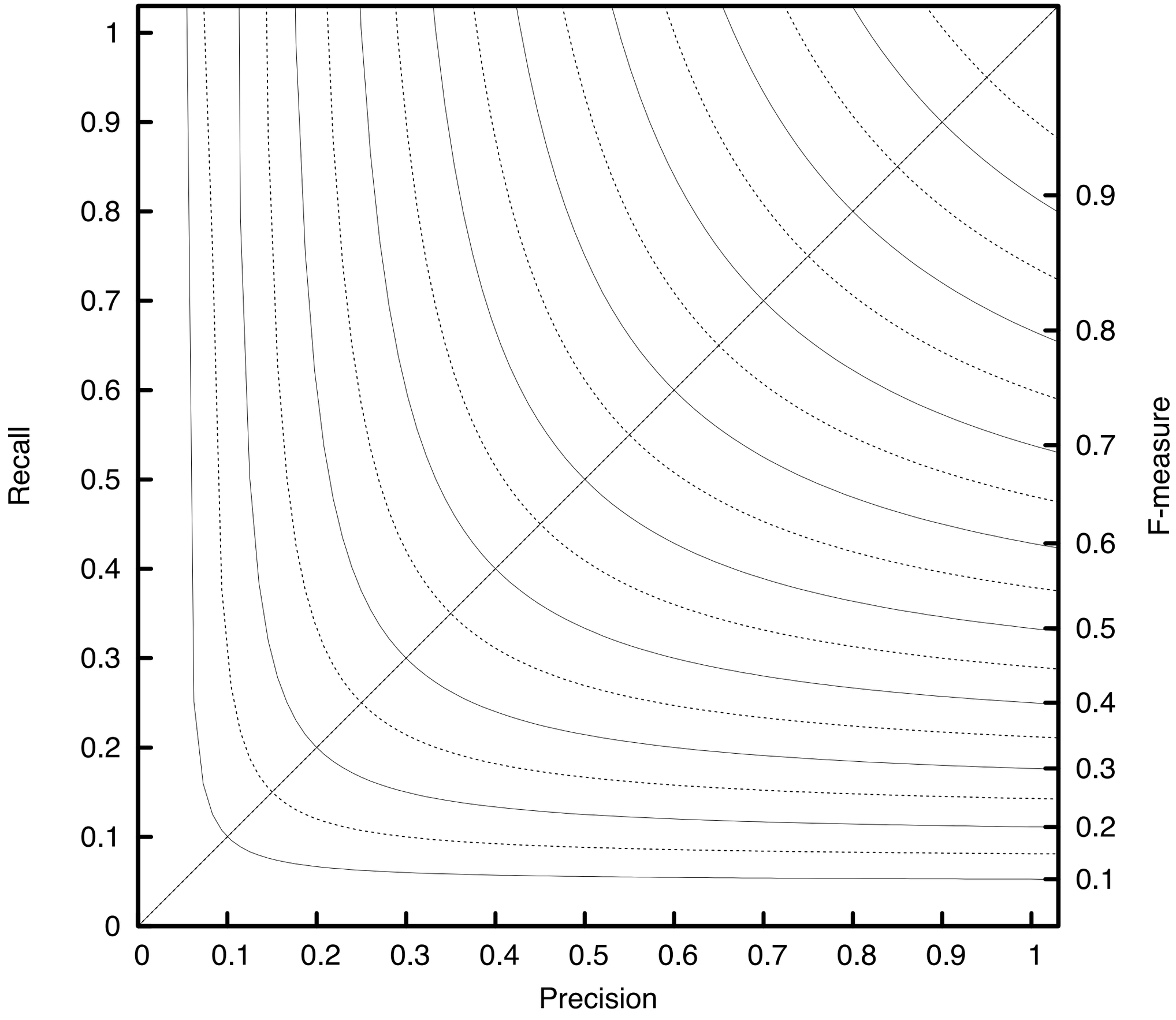

F-Score
1
0.8
0.6
0.4
0.2
0
Recall
0
0.2
0.4
0.6
0.8
1
Precision
1
0.8
0.6
0.4
0.2
0

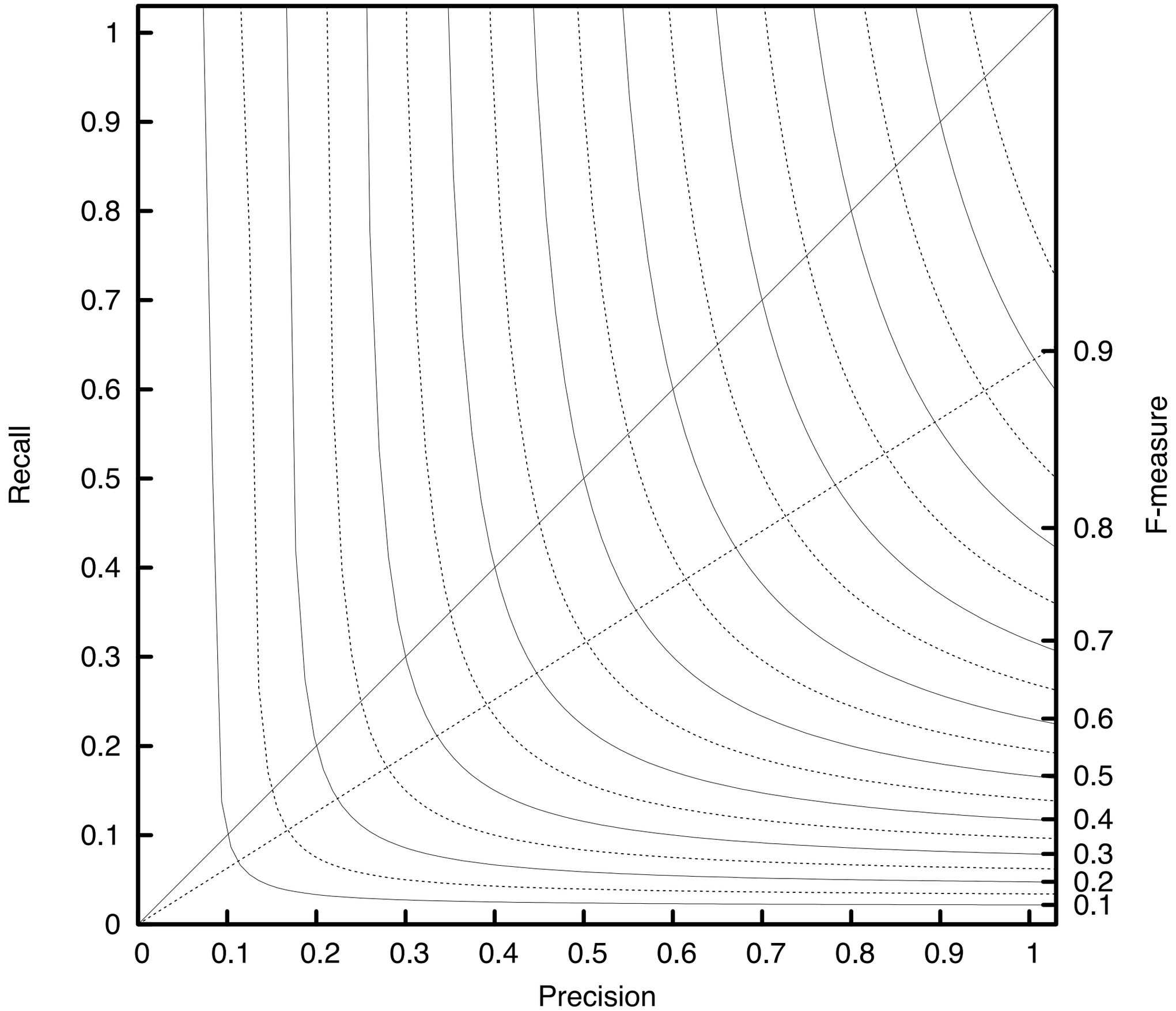

F-Score
1
0.8
0.6
0.4
0.2
0
Recall
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
Precision

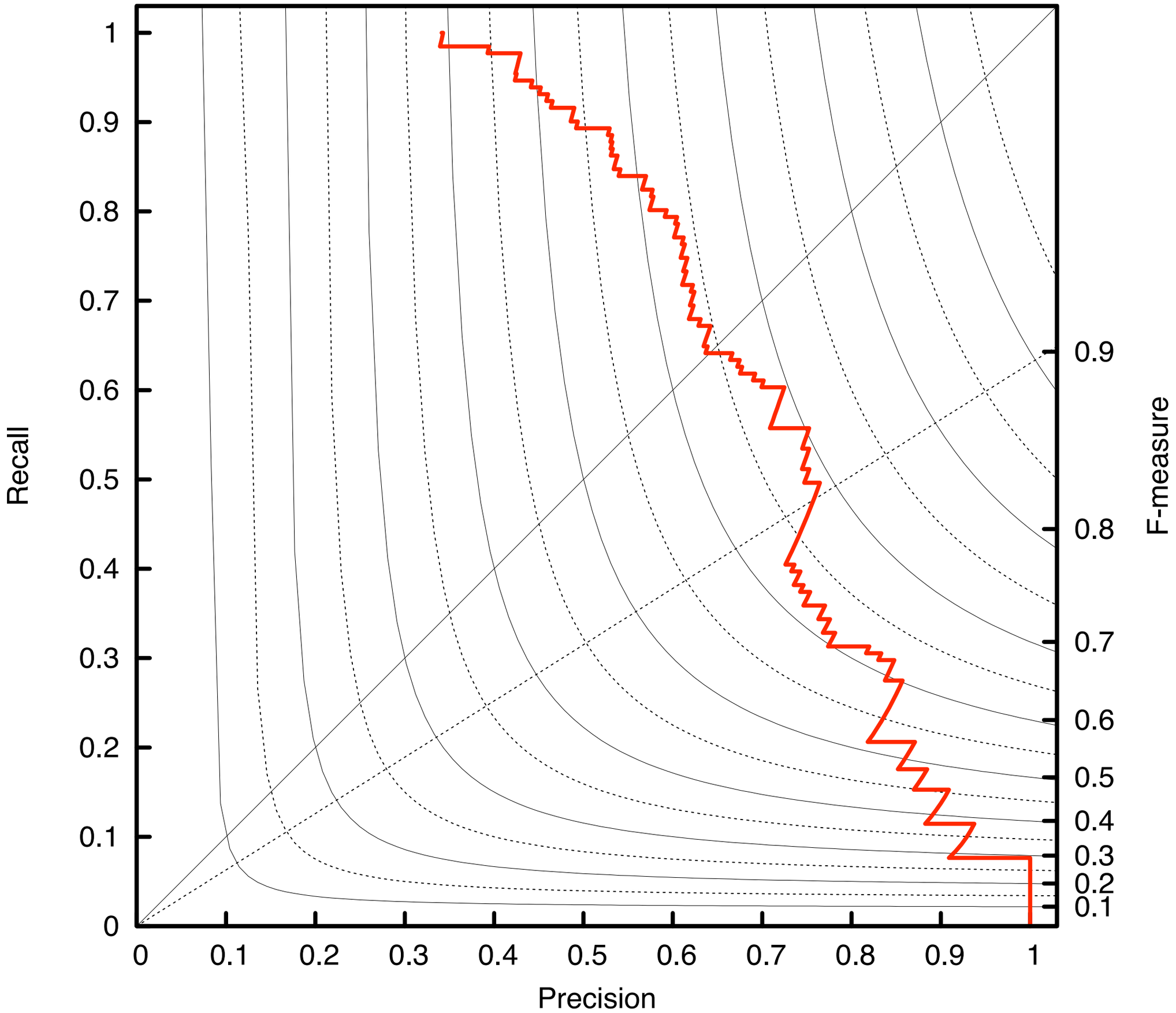

1
0.8
0.6
0.4
0.2
0
Recall
1
0.8
0.6
0.4
0.2
0
FScore
0
0.2
0.4
0.6
0.8
1
Precision

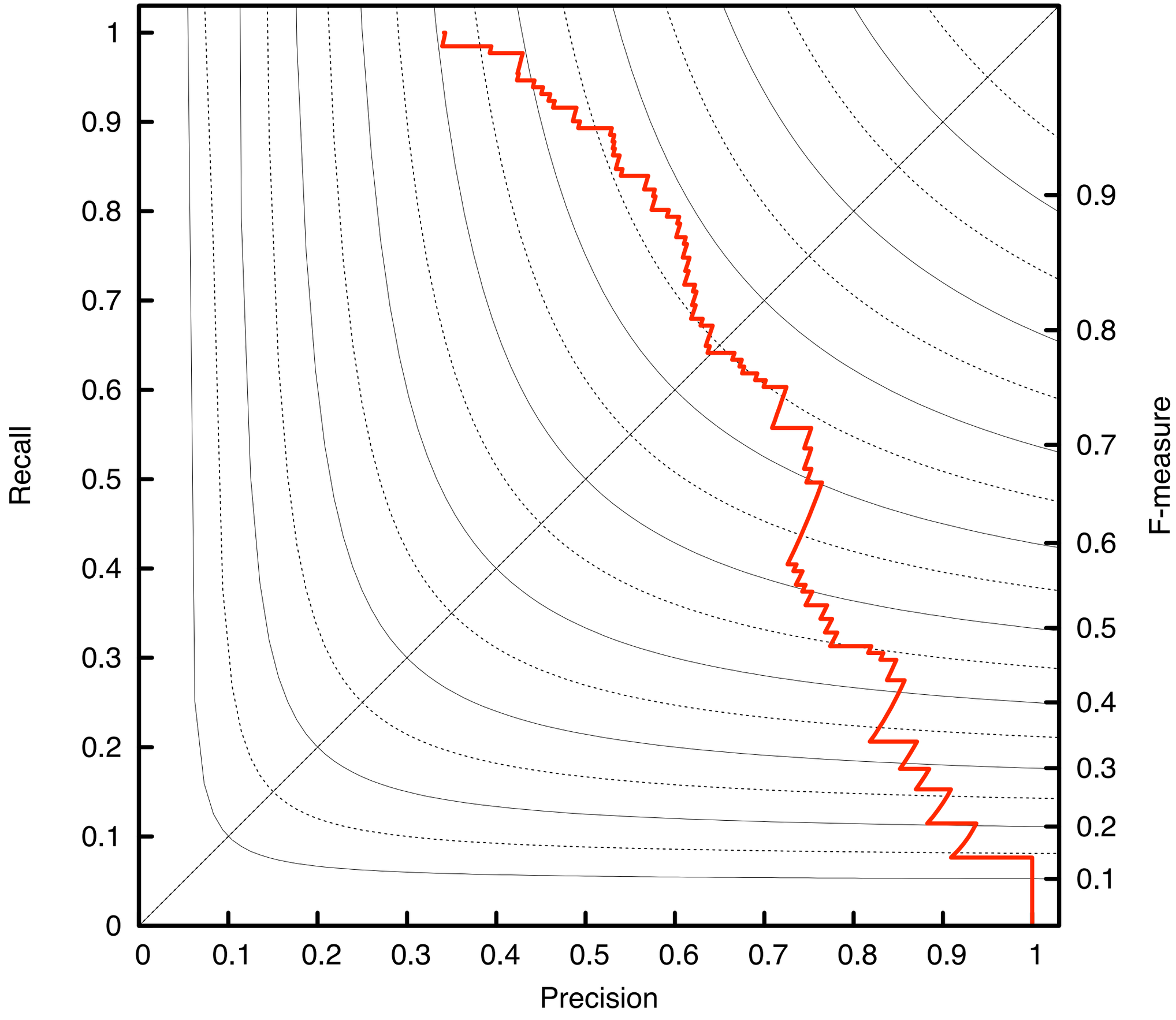

Recall
Precision
F-measure
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

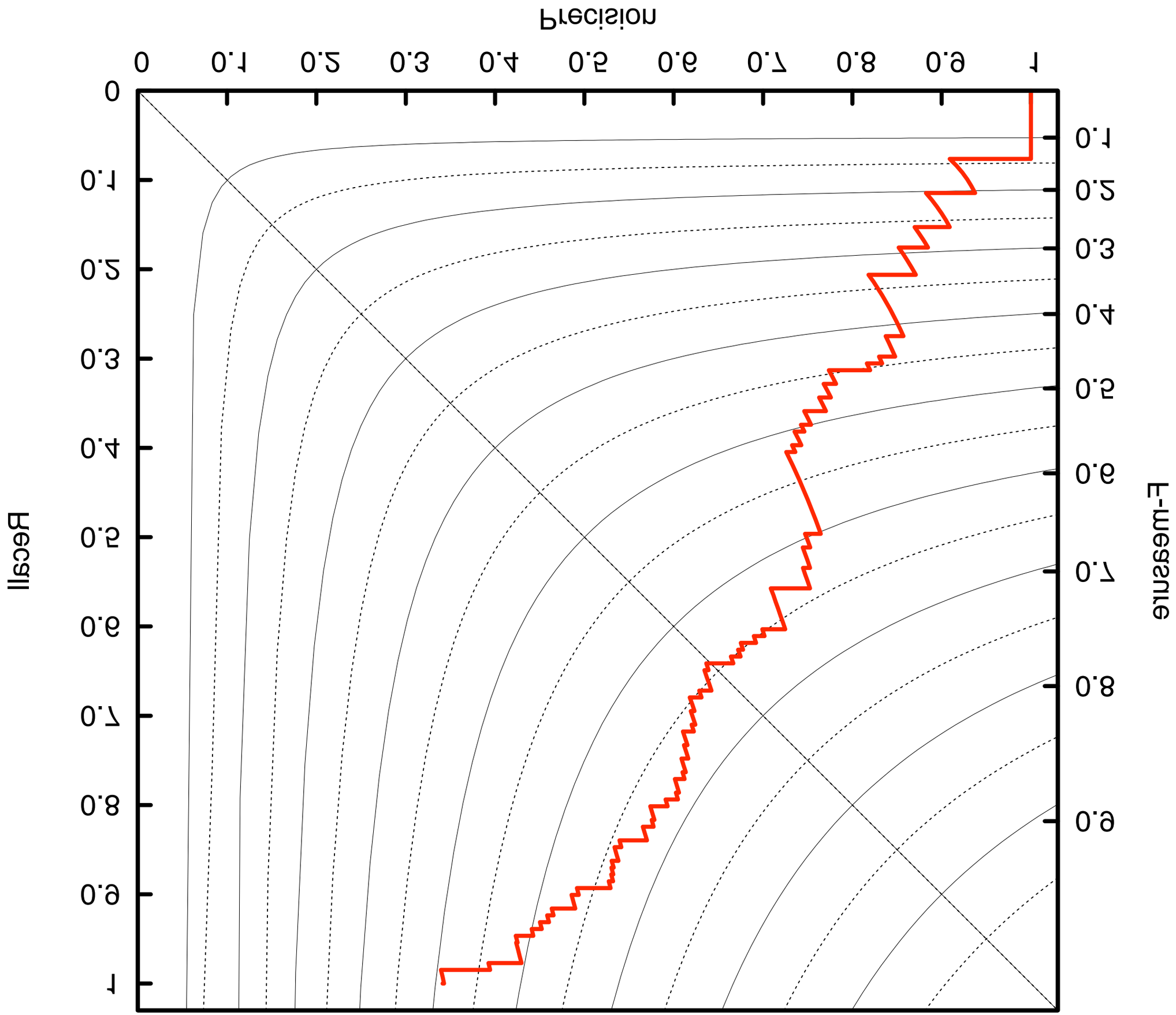

Recall
Precision
F-measure
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

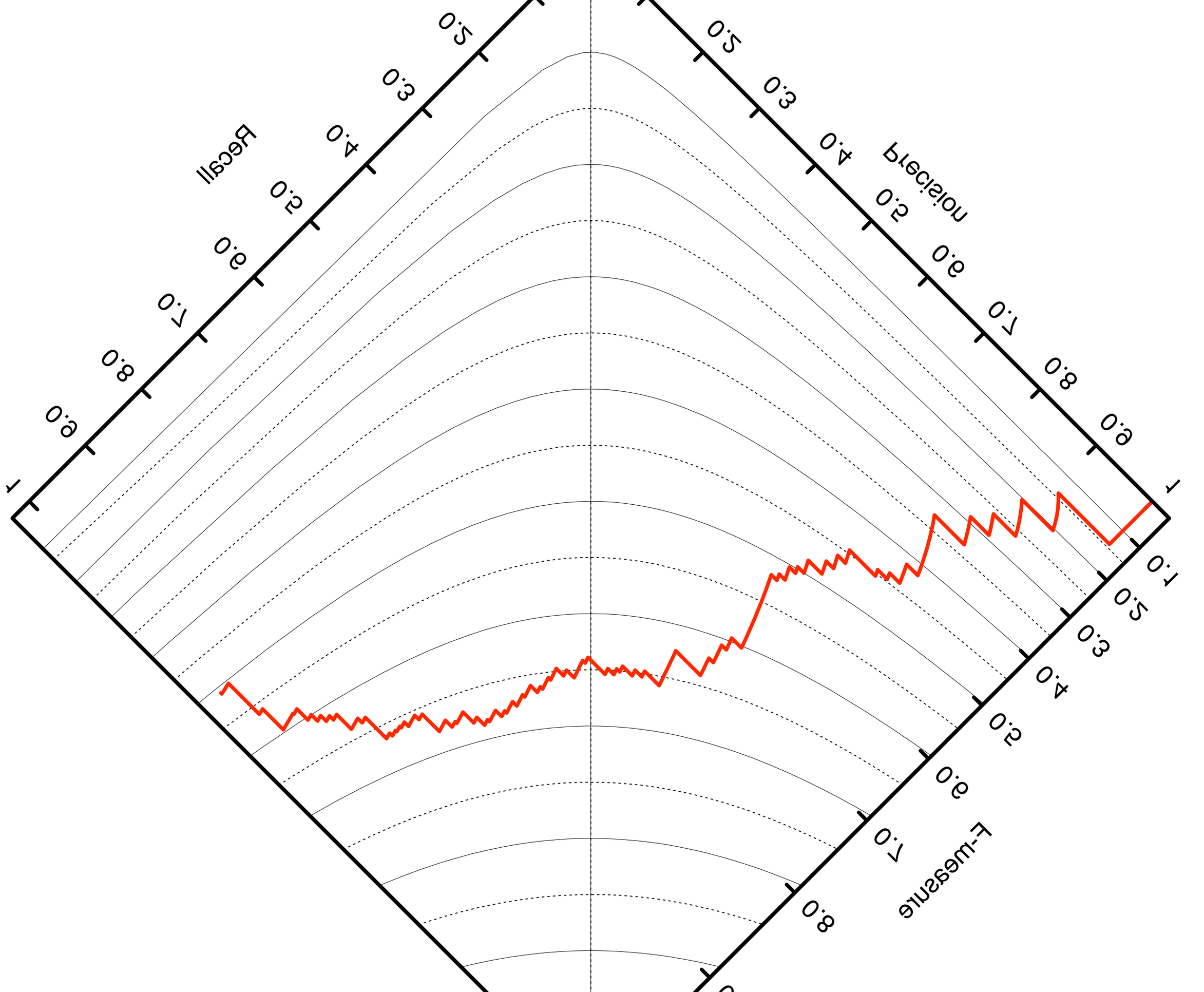

Recall
Precision
F-measure
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Recall
Precision
F-measure
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Recall
Precision
F-measure
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Recall
Precision
F-measure
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Precision
Recall
F-measure
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Precision
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Recall
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
F-measure
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

empirical

Precision

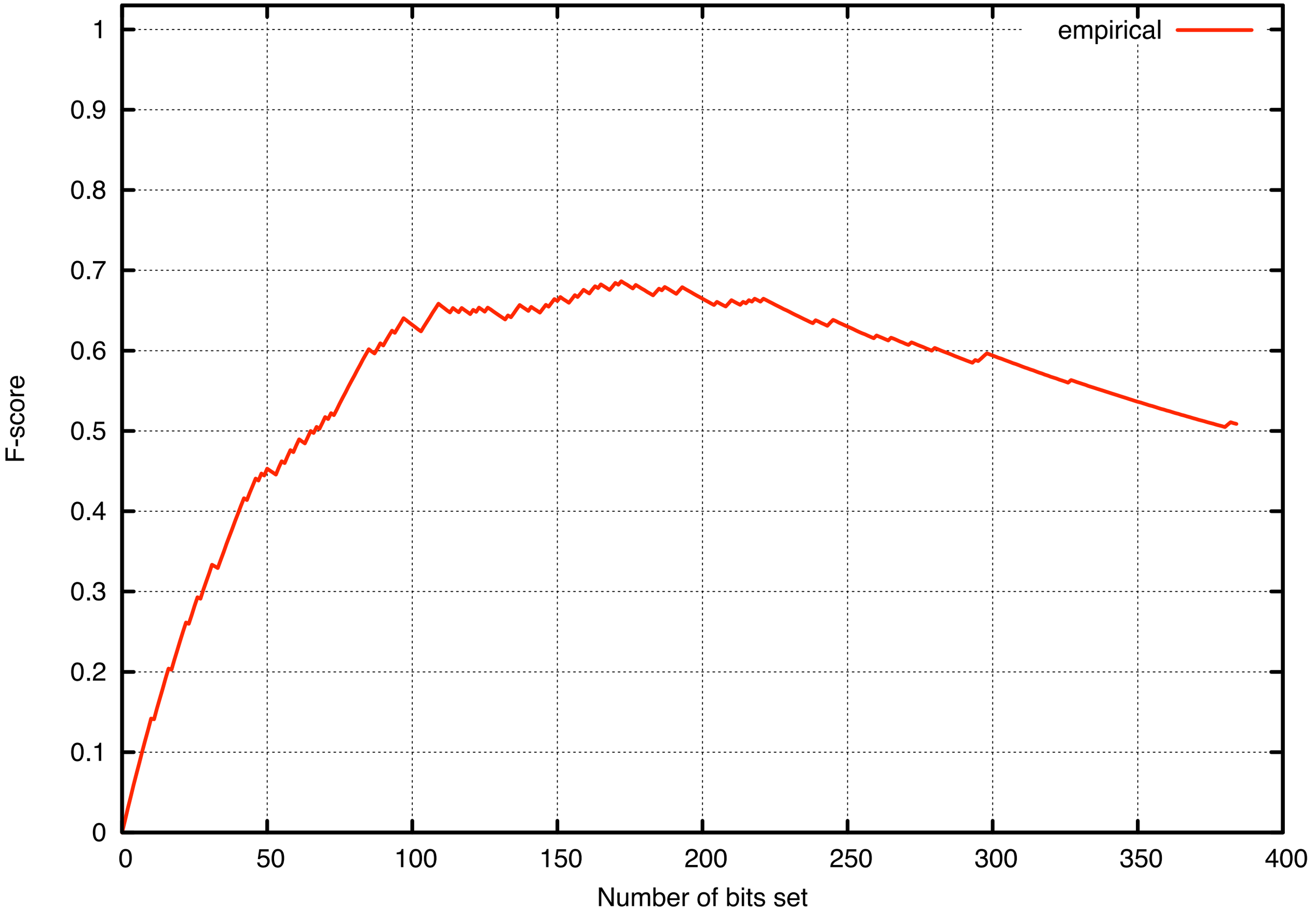
Recall

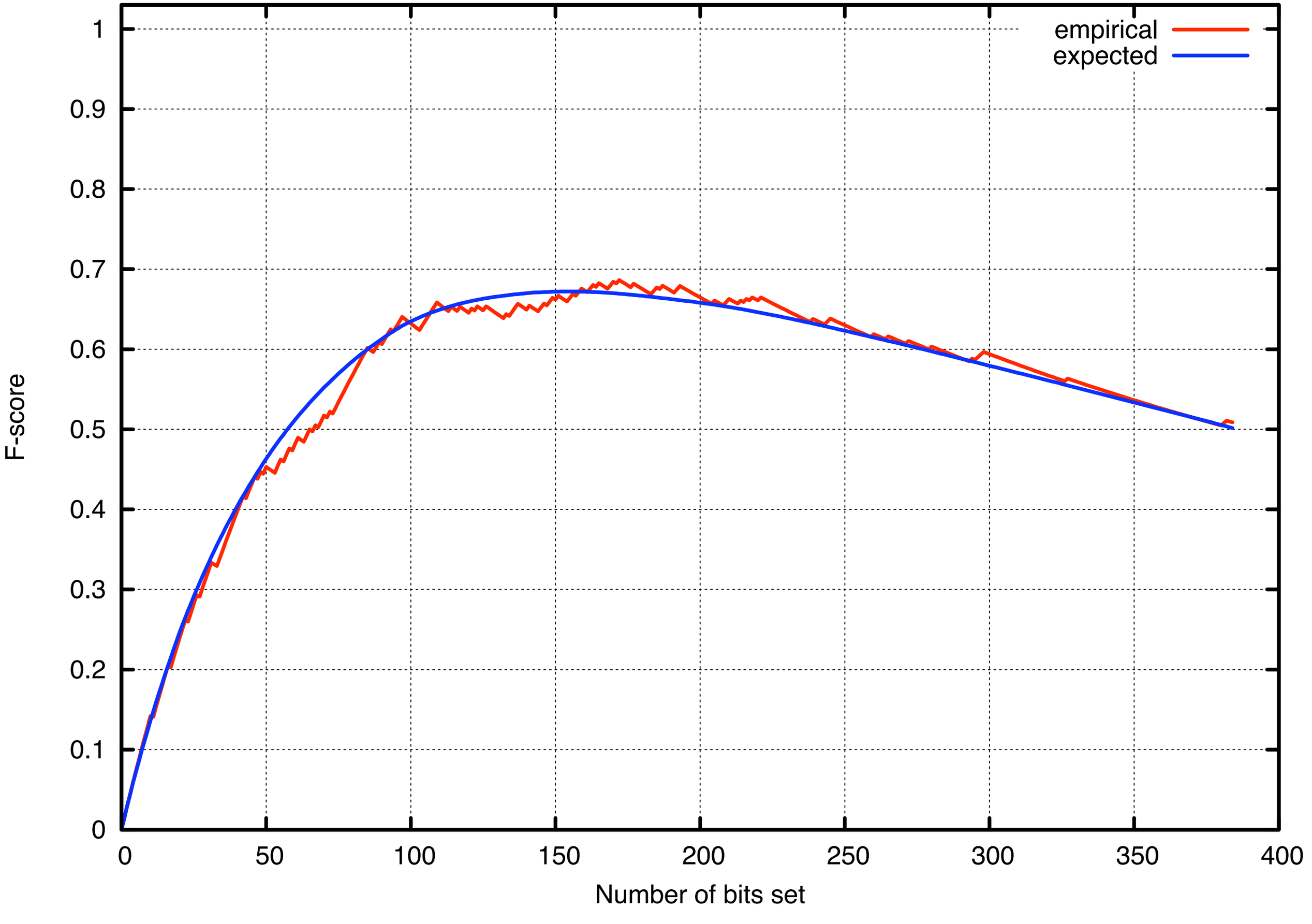
F-score

Number of bits set

0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

0 50 100 150 200 250 300 350 400

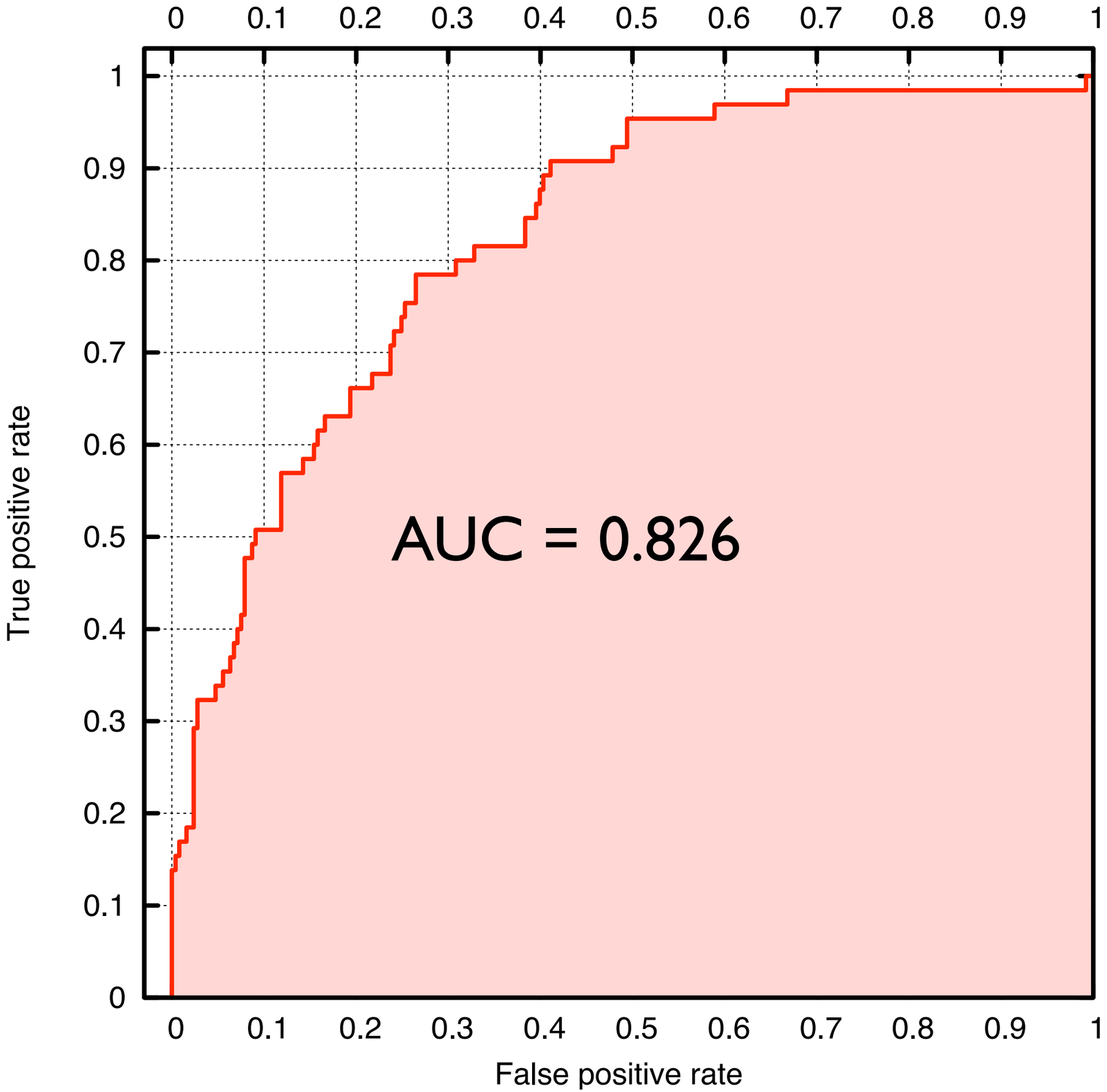

empirical
F-score
Number of bits set
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
50
100
150
200
250
300
350
400

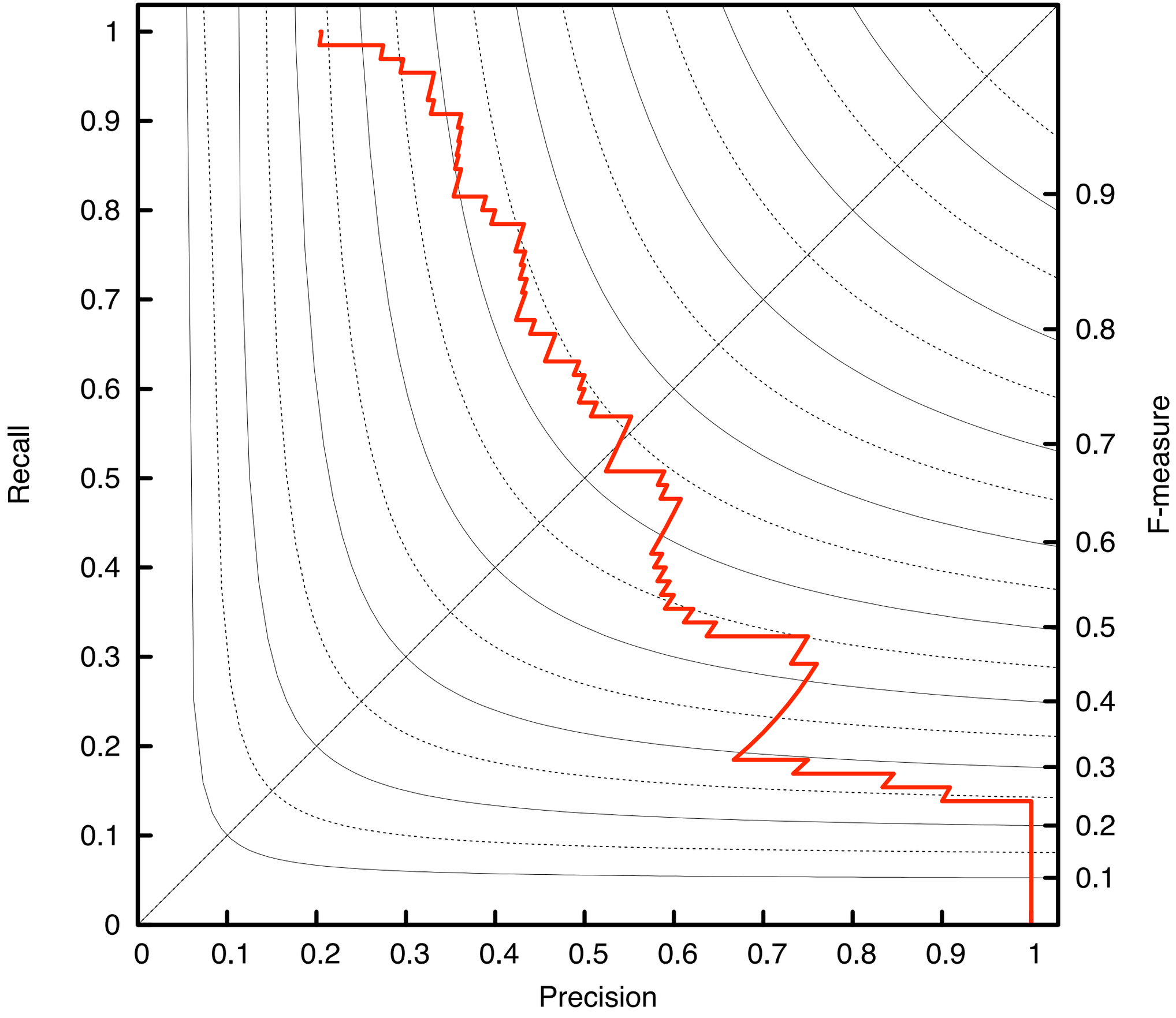

empirical
expected
F-score
Number of bits set
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
50
100
150
200
250
300
350
400

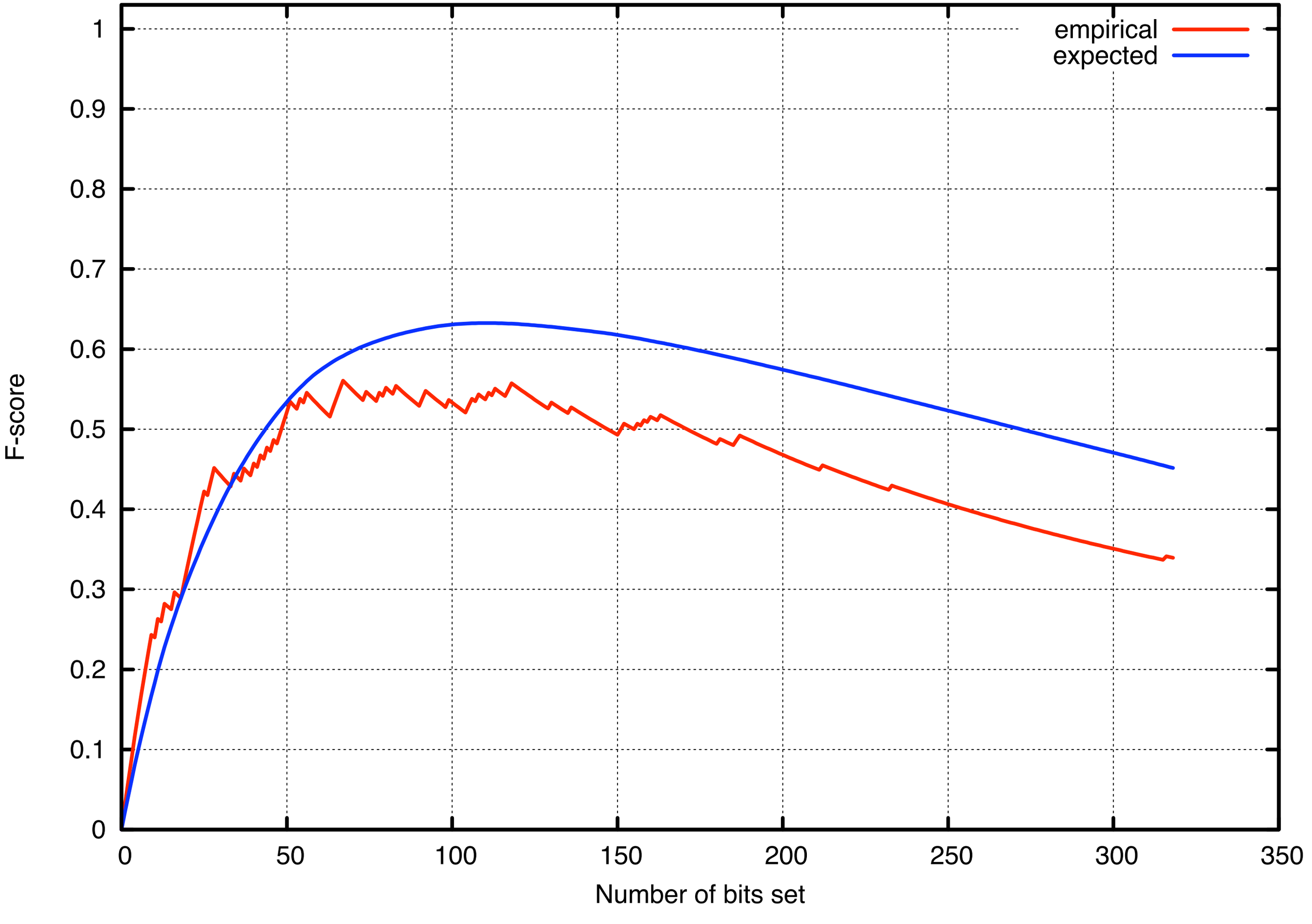
Decoding strategy	F-score
xval threshold	0.617
threshold 0.5	0.624
MEU	0.664
oracle threshold	0.686

Diabetes, modified

- 384 training instances, 318 test instances
- 8 numeric features (plus intercept)
- 36% positive labels in training data
- 20% positive labels in test data


0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
True positive rate
False positive rate
AUC = 0.826

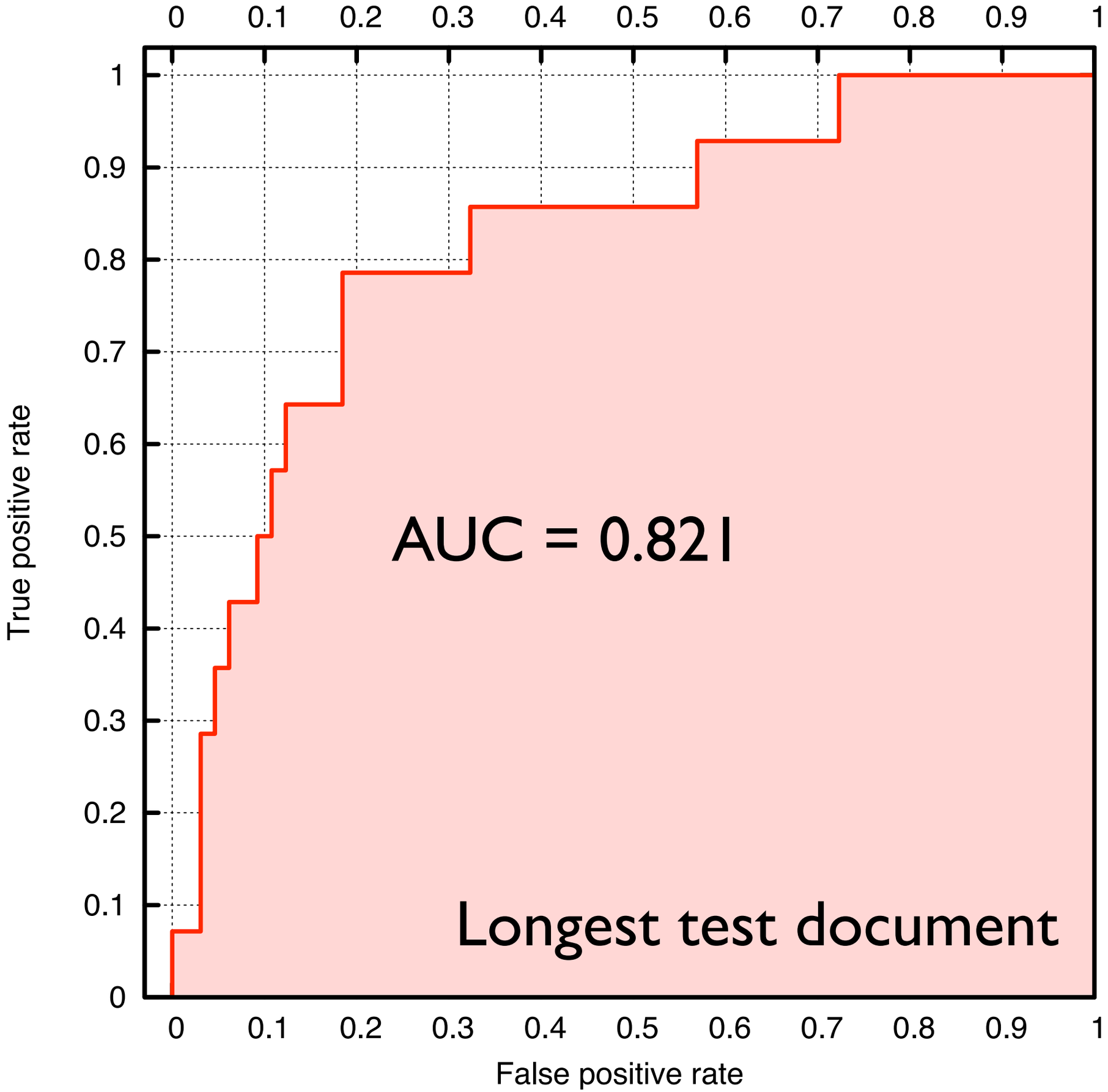

Recall
Precision
F-measure
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

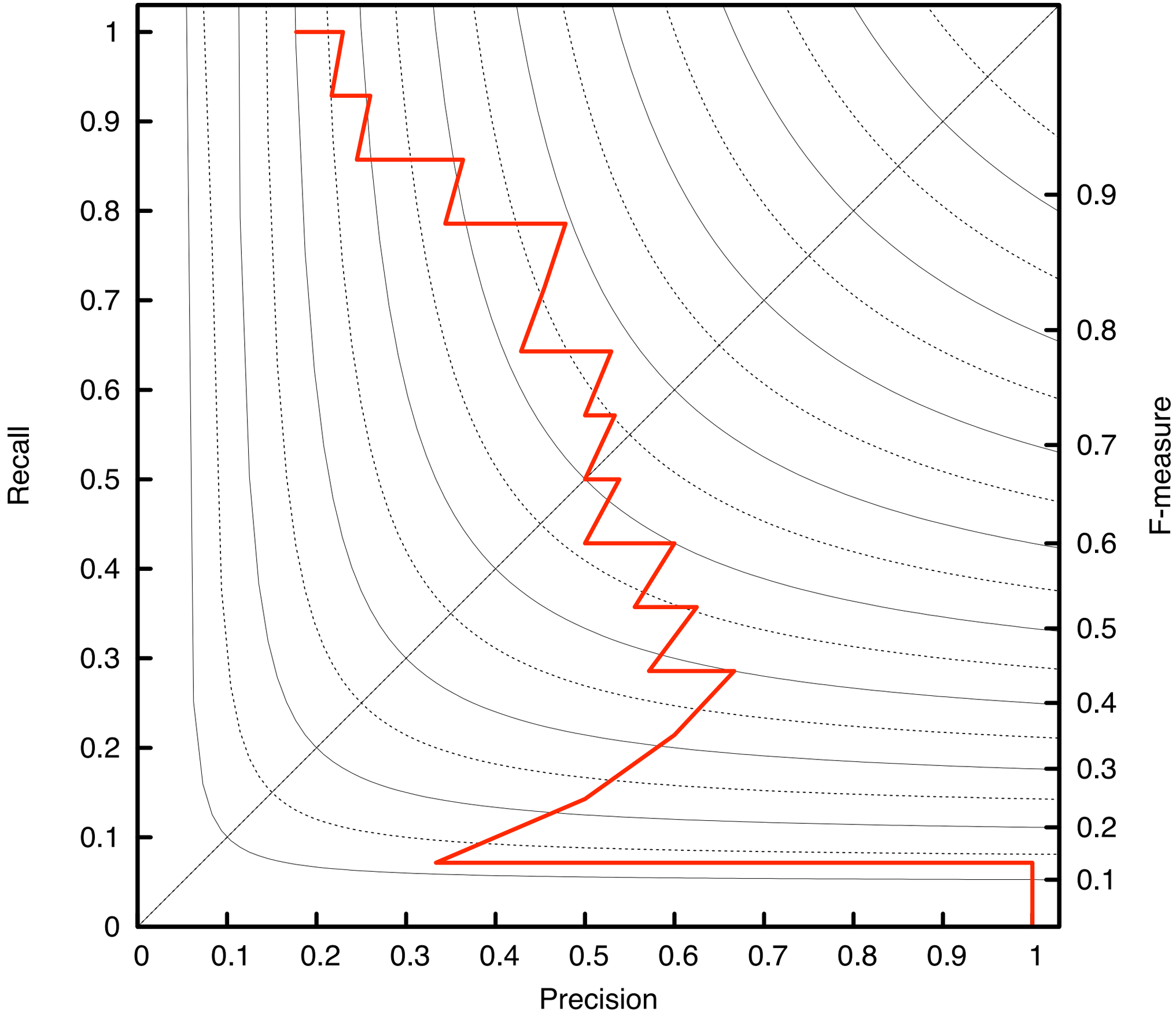

empirical
expected
F-score
Number of bits set
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
50
100
150
200
250
300
350

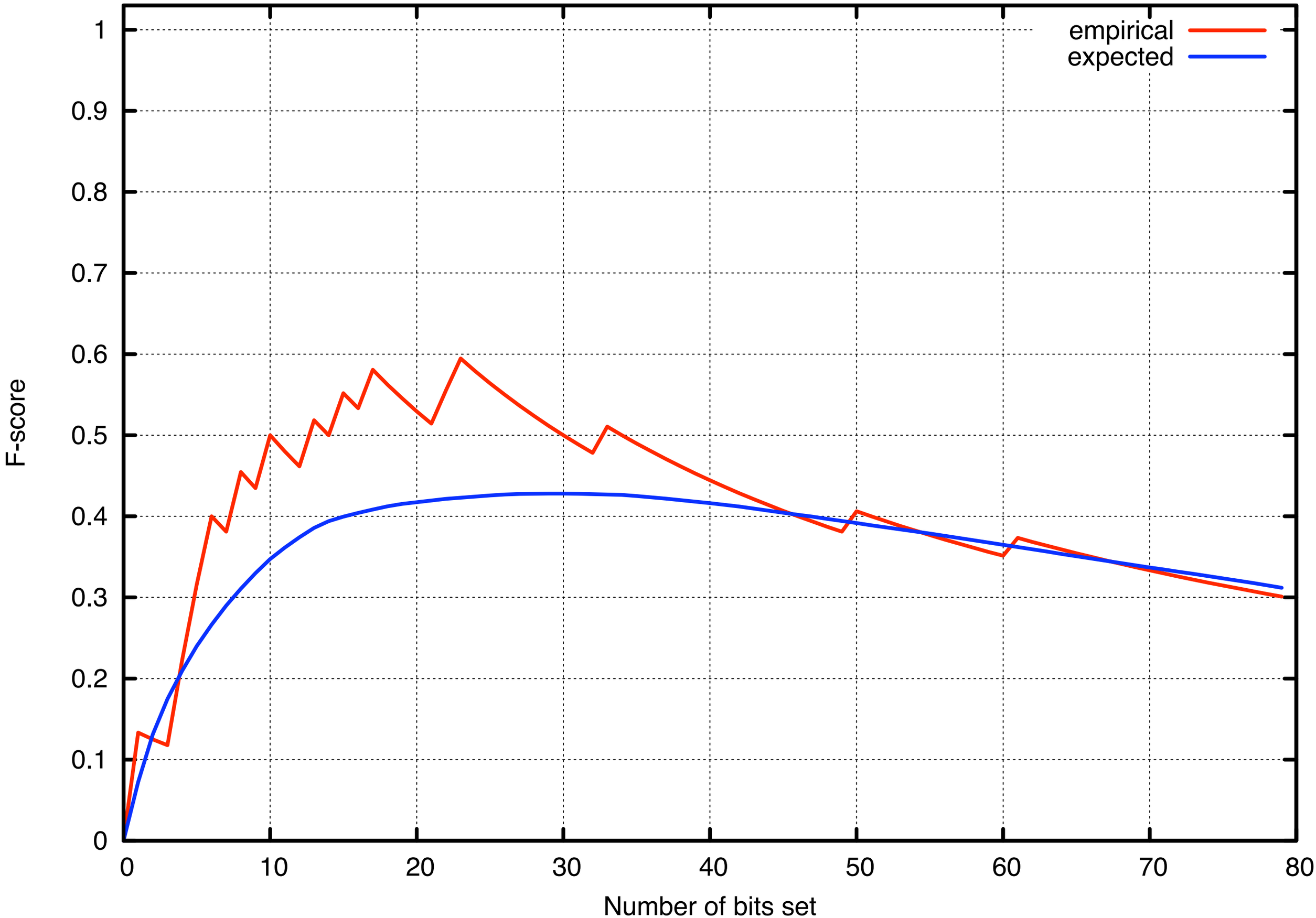
Decoding strategy	F-score
xval threshold	0.464
threshold 0.5	0.516
MEU	0.545
oracle threshold	0.561

Broadcast News Summarization

- Dataset from Maskey & Hirschberg 2005
- 3535 training instances, 197 documents
- 408 test instances, 19 documents (median length 14), per-document decoding
- 29 numeric features (plus intercept)
- 25% positive labels


0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
True positive rate
False positive rate
AUC = 0.821
Longest test document

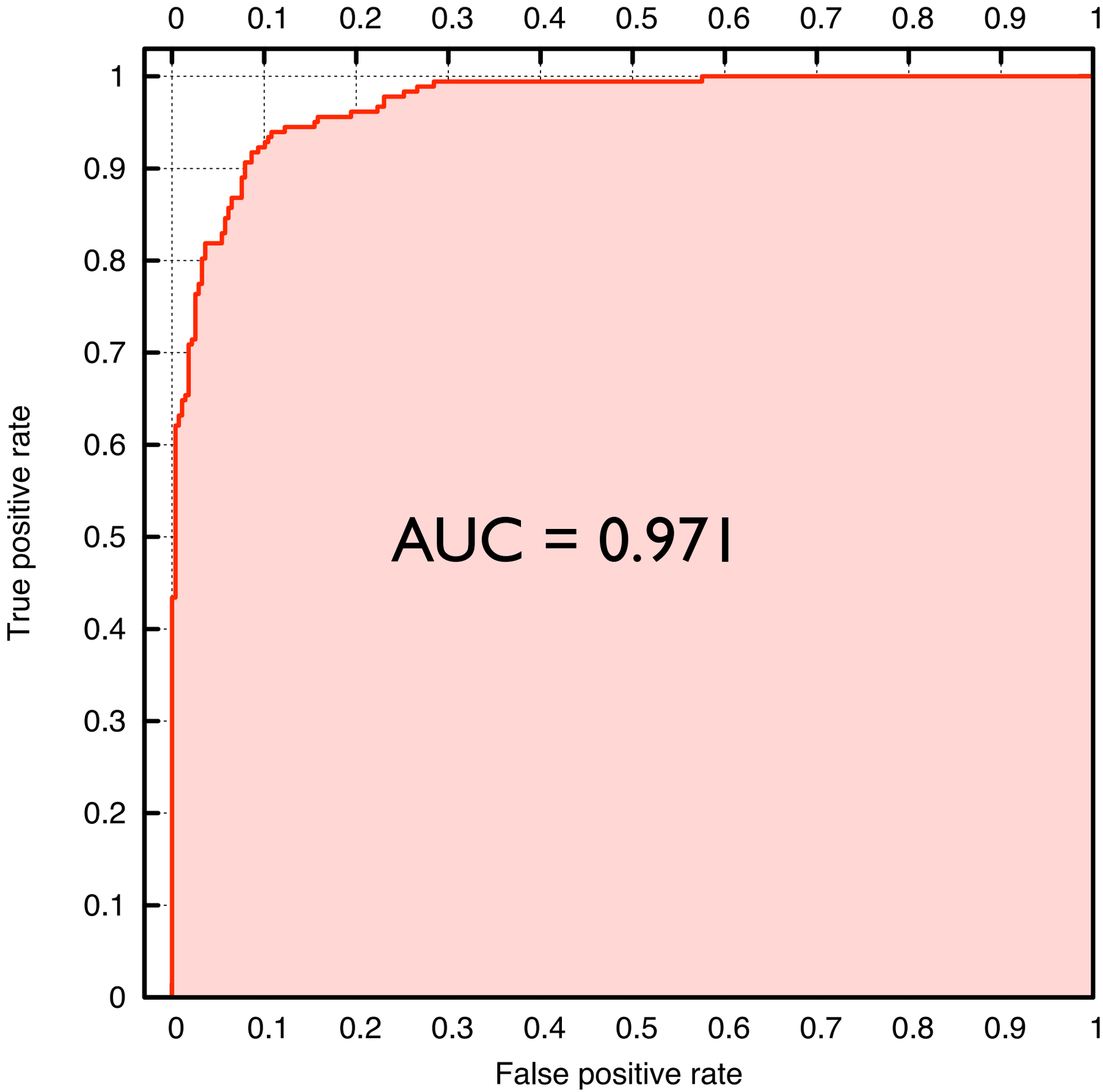

Recall
Precision
F-measure
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

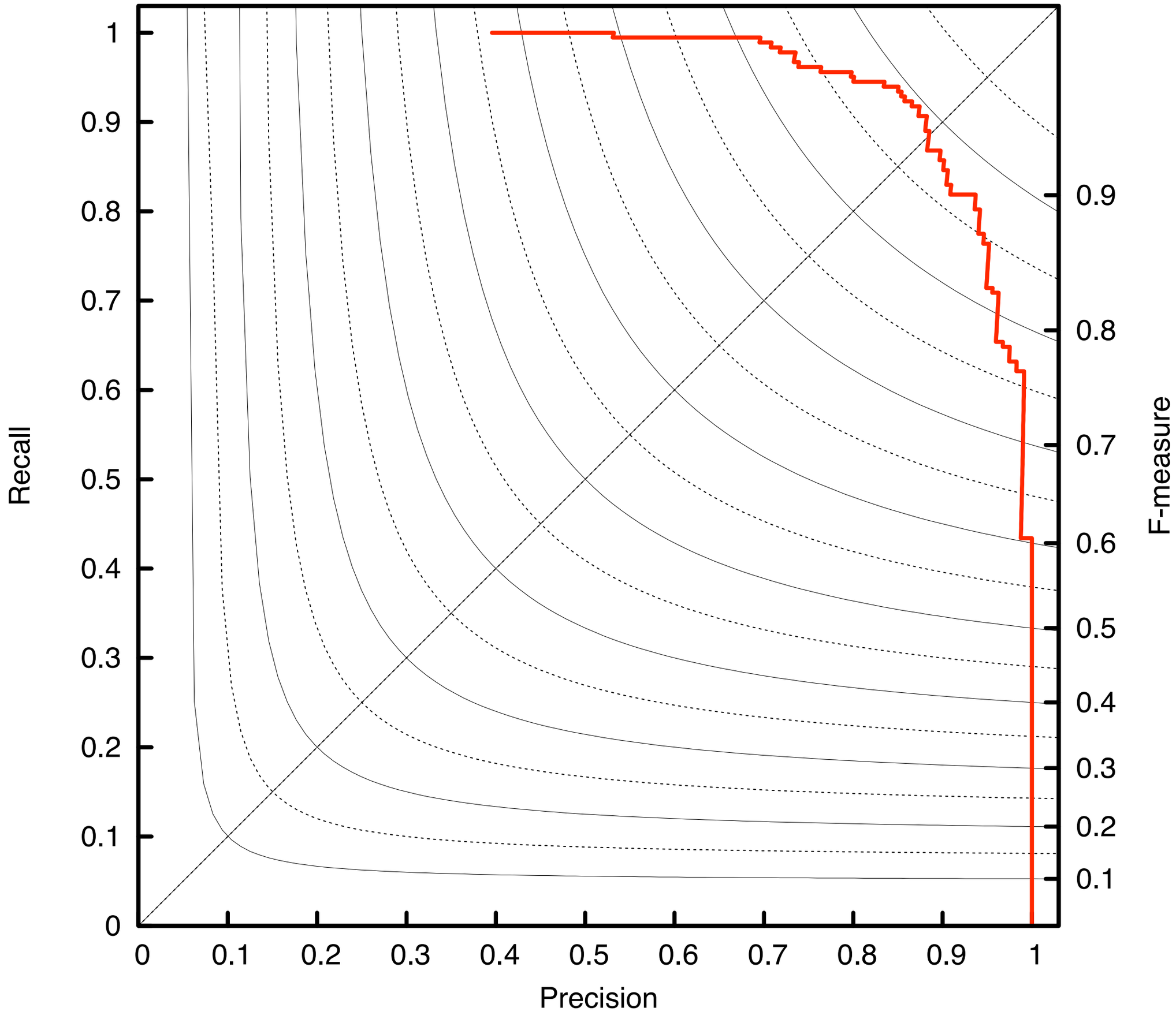

empirical
expected
F-score
Number of bits set
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
10
20
30
40
50
60
70
80

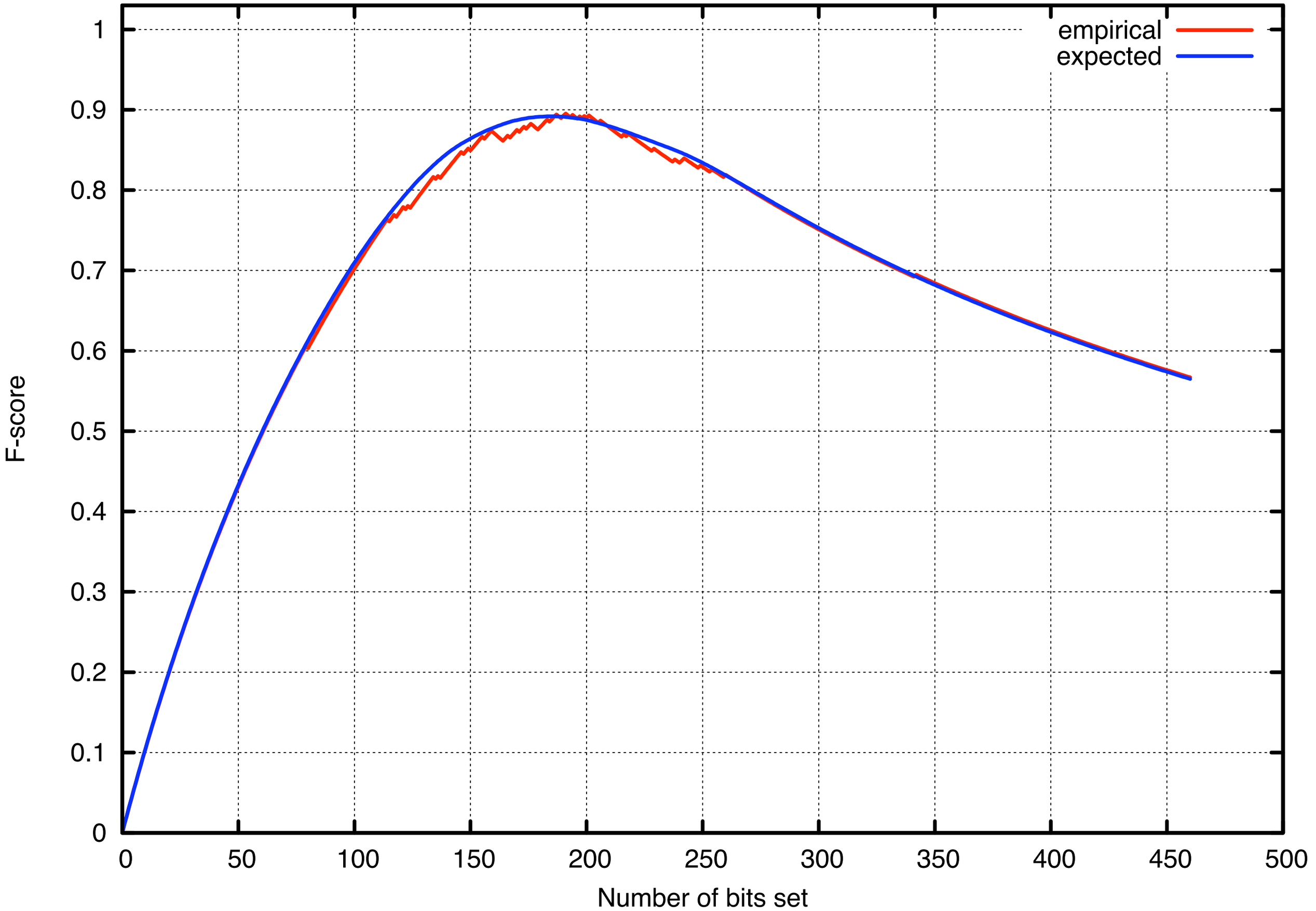
Decoding strategy	mean F-score
xval threshold	0.618
threshold 0.5	0.352
Maskey/Hirschberg	0.544
MEU	0.599
oracle threshold	0.724

Spam

- 4141 training instances
- 460 test instances
- 57 numeric features (plus intercept)
- 39% positive labels


0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
True positive rate
False positive rate
AUC = 0.971

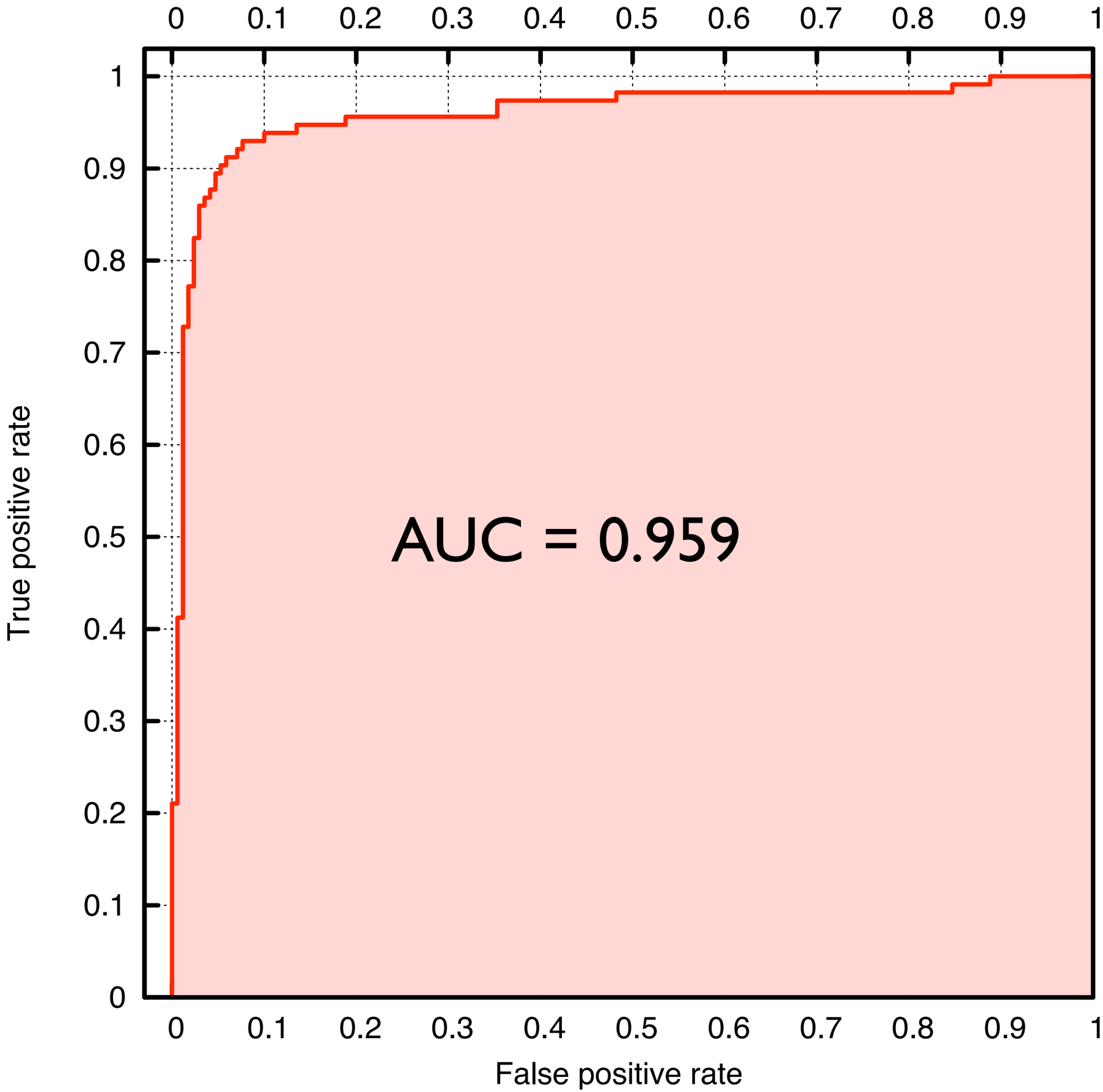

Recall
Precision
F-measure
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

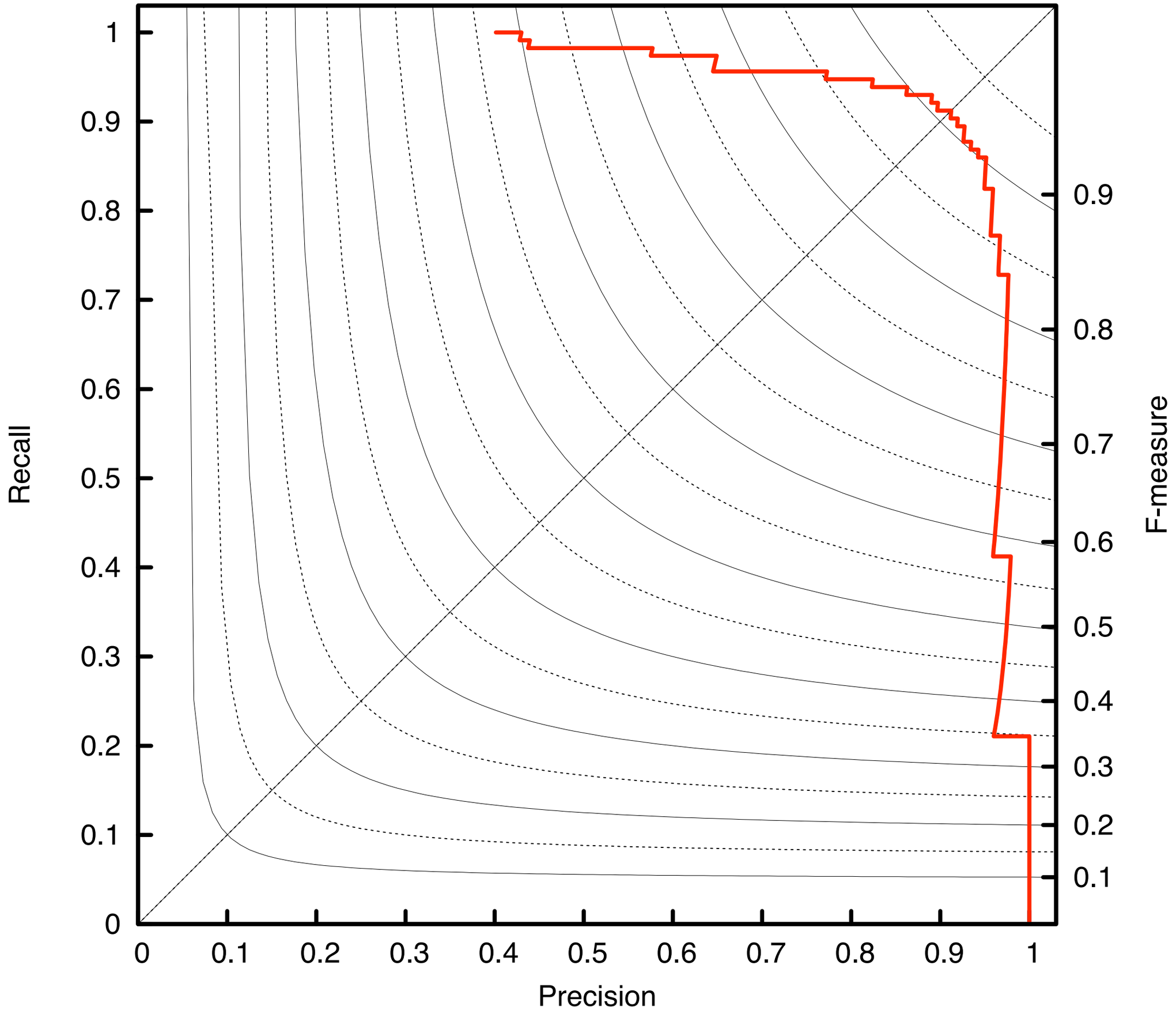

empirical
expected
F-score
Number of bits set
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
50
100
150
200
250
300
350
400
450
500

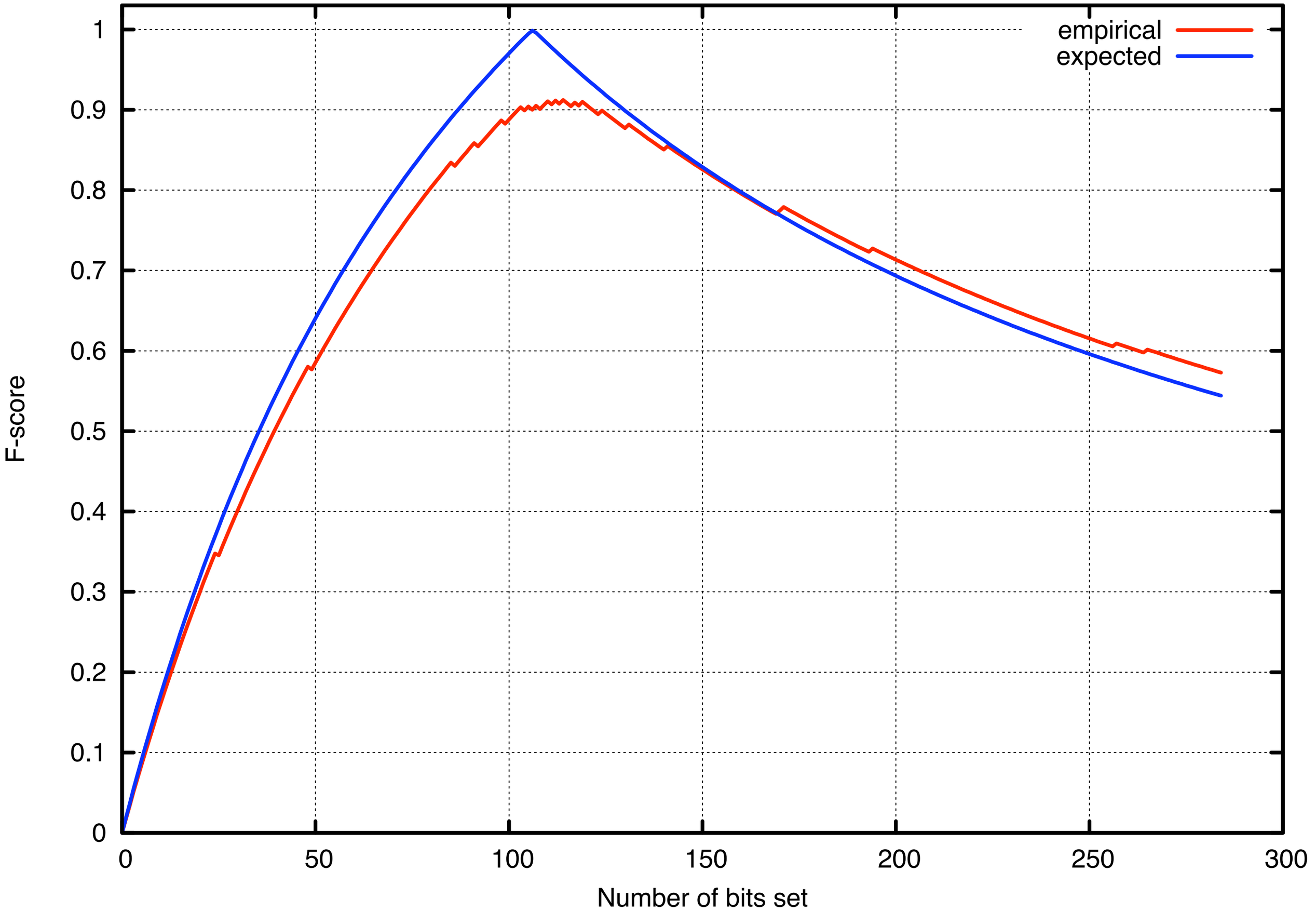
Decoding strategy	F-score
threshold 0.5	0.880
MEU	0.892
oracle threshold	0.895

Breast Cancer

- 285 training instances, 284 test instances
- 30 numeric features (plus intercept)
- 37% positive labels


AUC = 0.959
True positive rate
False positive rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

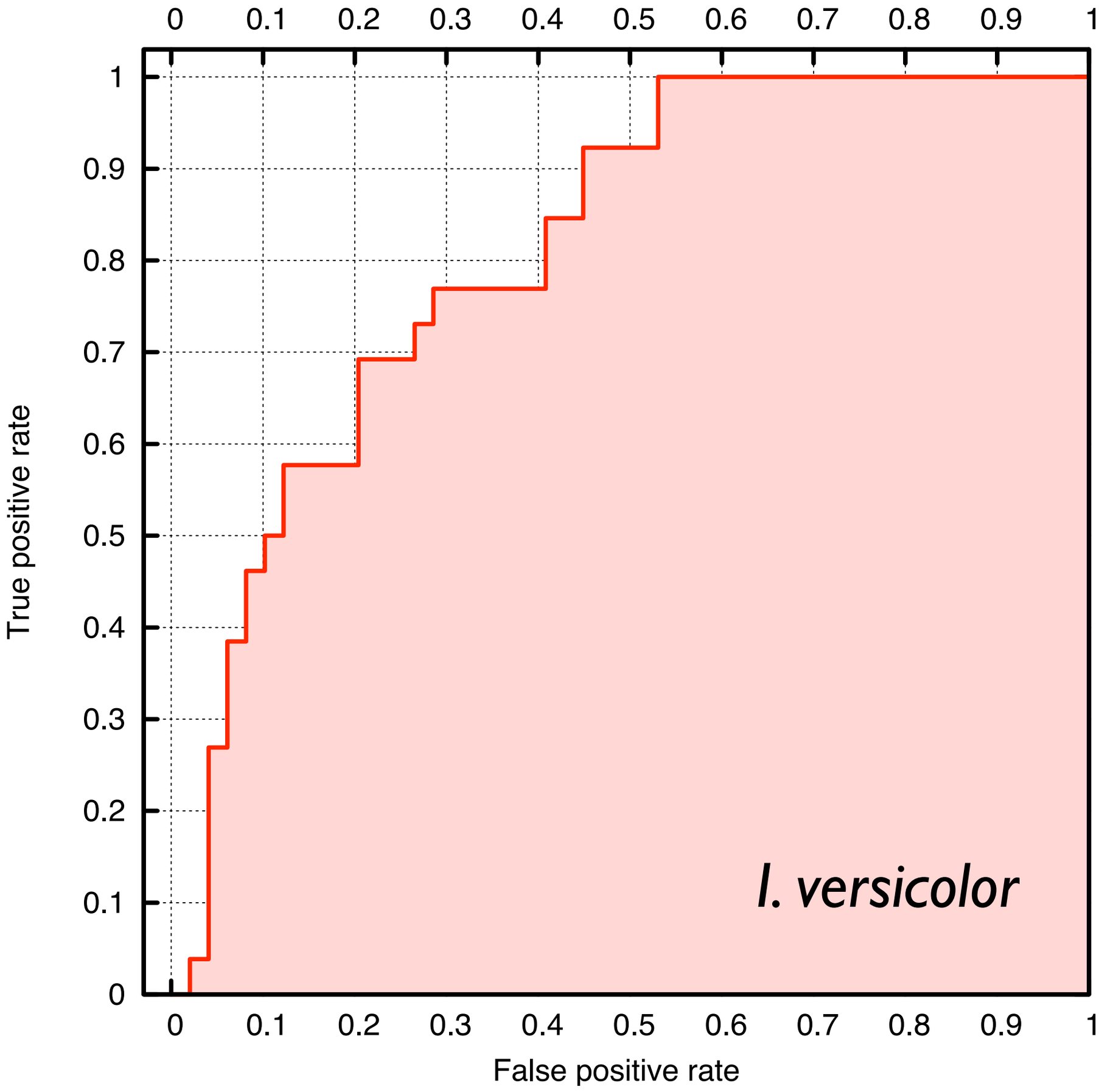

Recall
Precision
F-measure
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

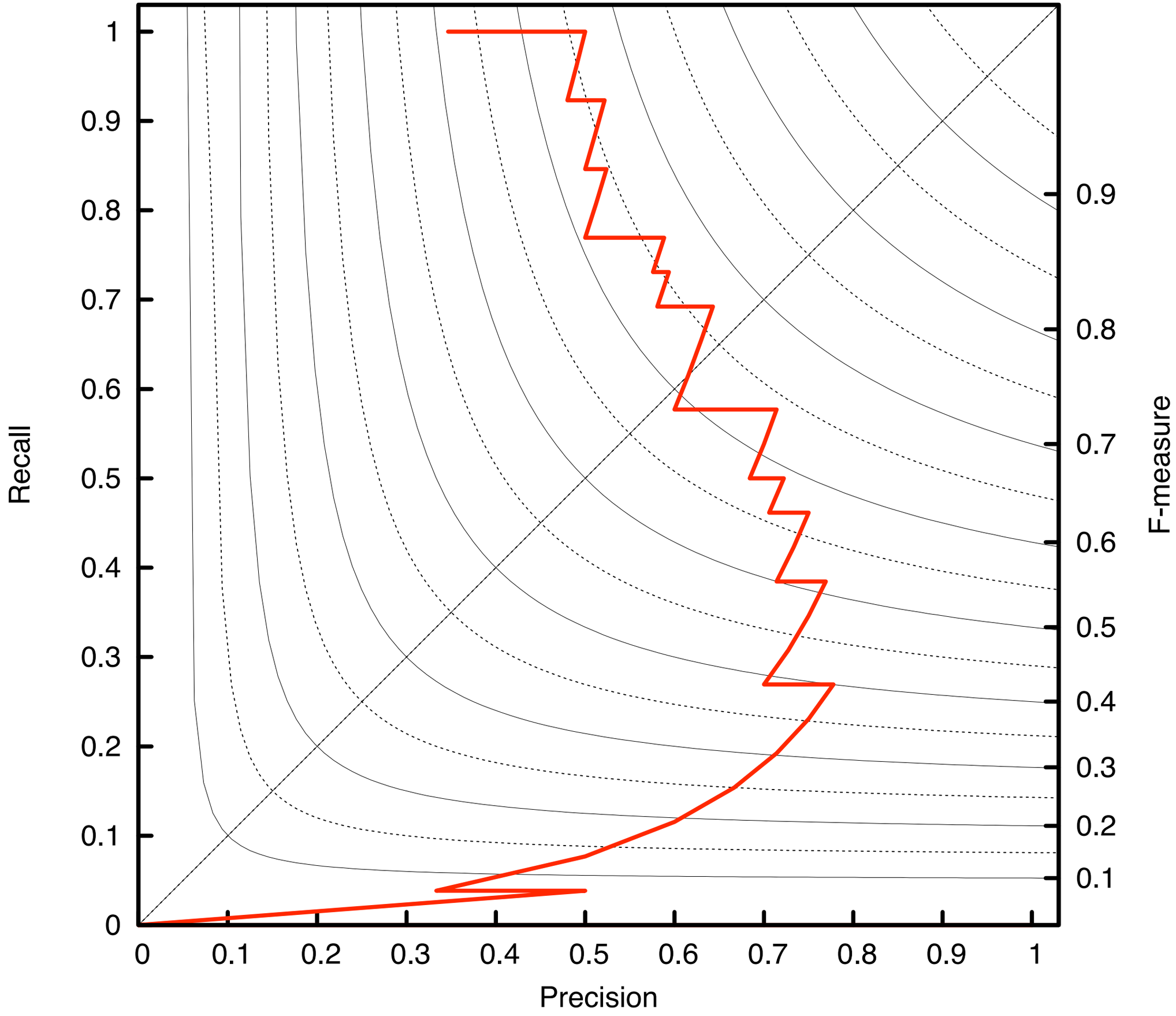

empirical
expected
F-score
Number of bits set
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
50
100
150
200
250
300

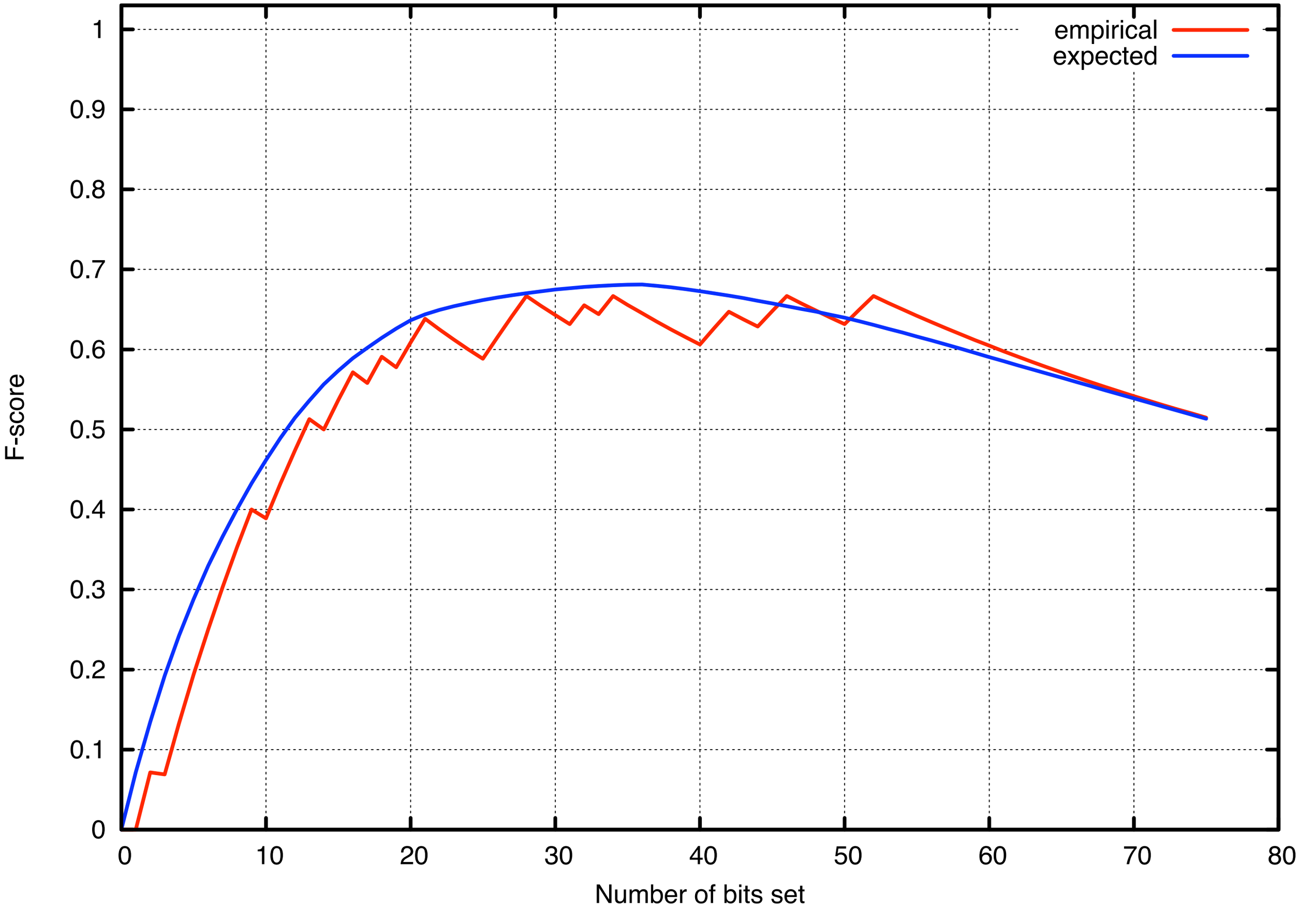
Decoding strategy	F-score
threshold 0.5	0.900
MEU	0.900
oracle threshold	0.912

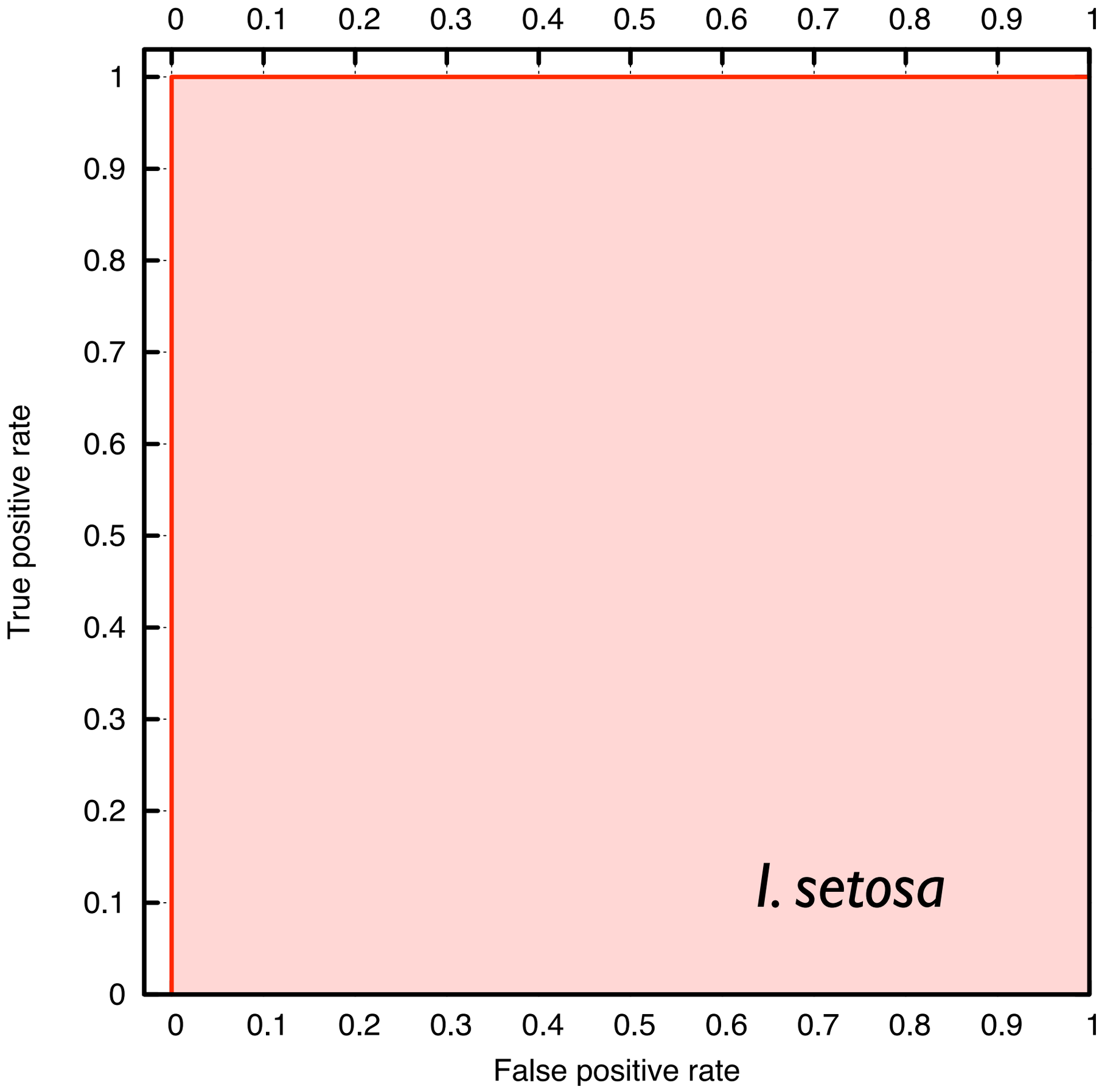
Iris

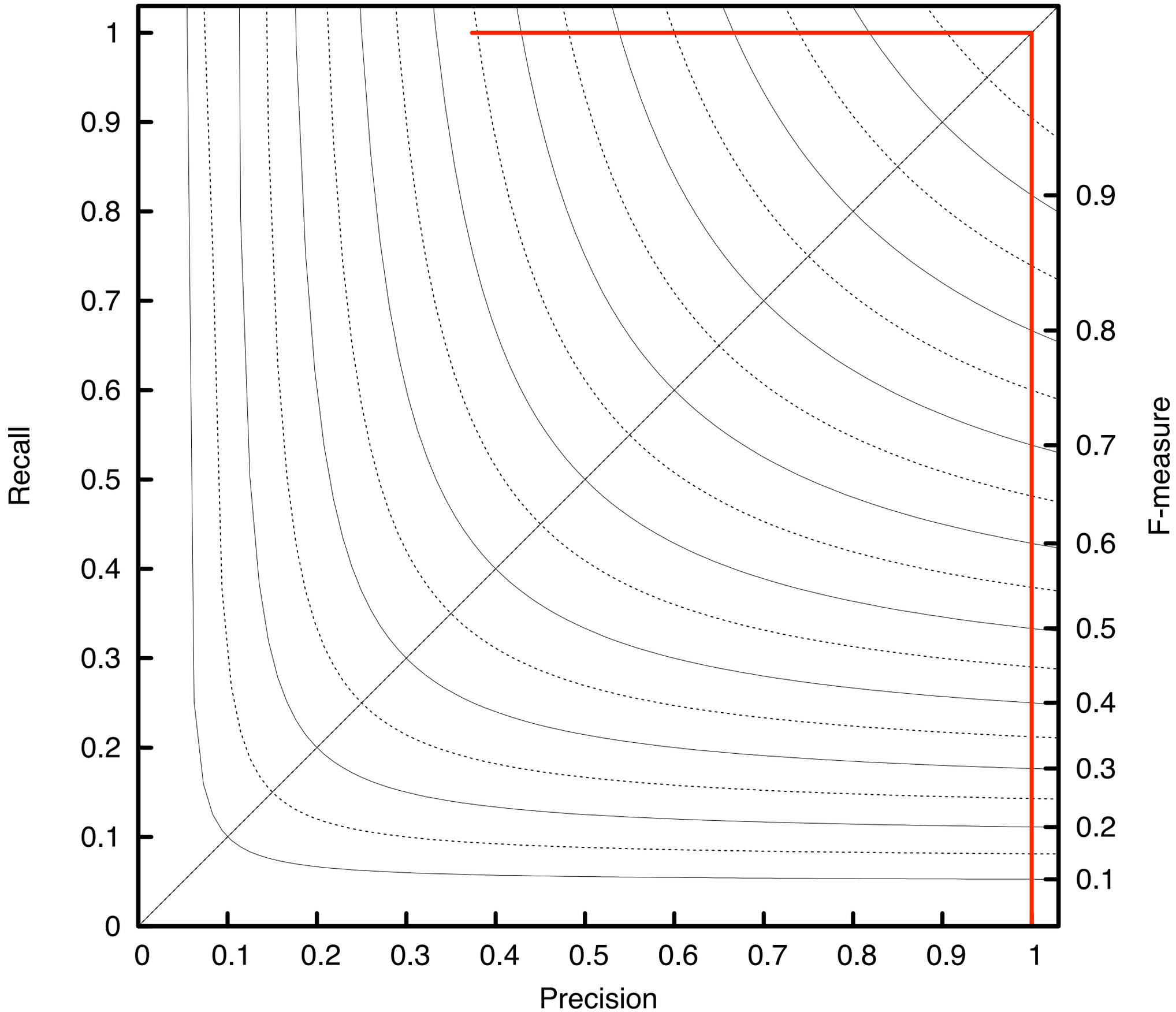
- 150 instances, 3 class labels
- 4 numeric features (plus intercept)
- Binarized: one-against-rest

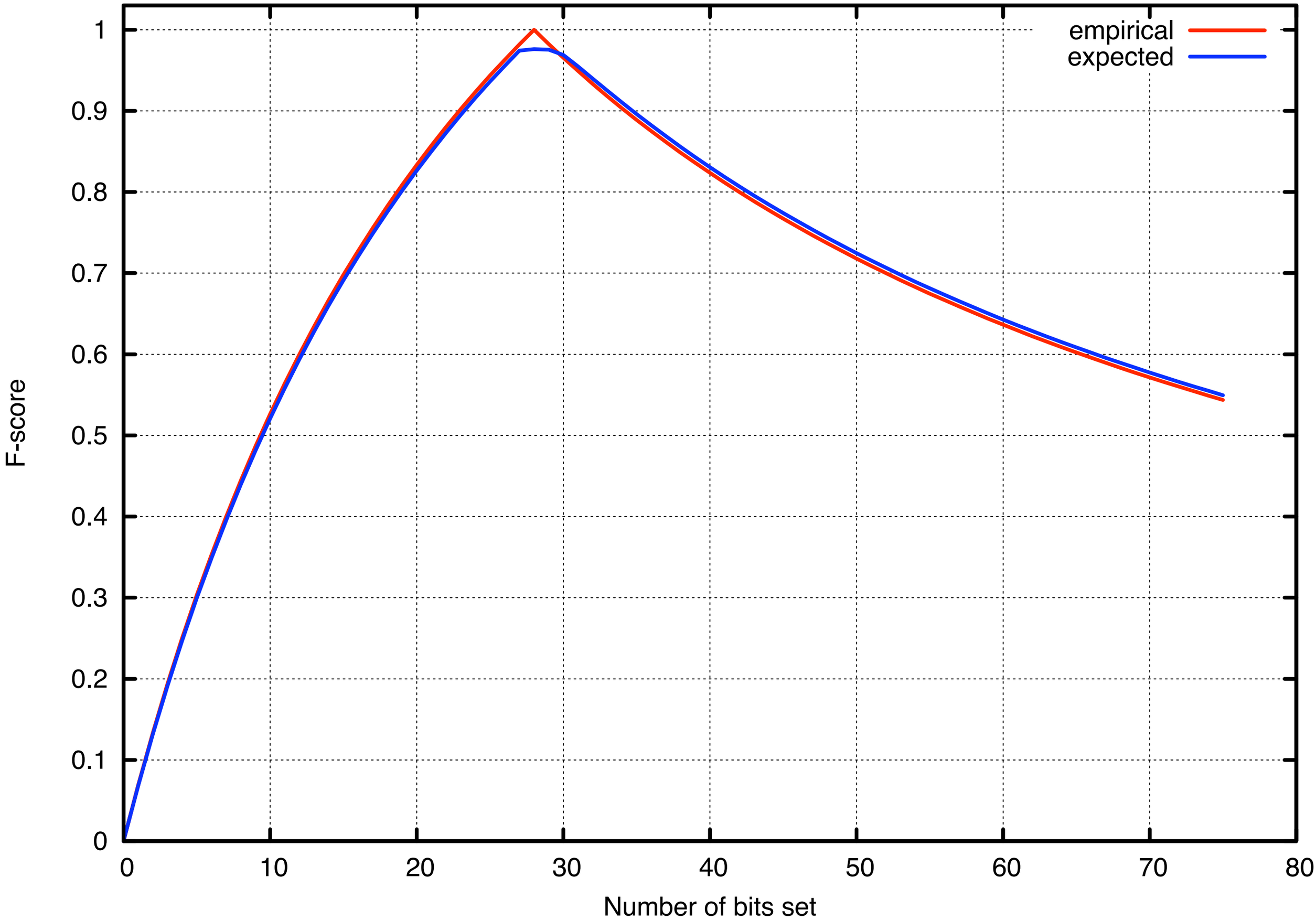

0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
True positive rate
False positive rate
I. versicolor


Recall
Precision
F-measure
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


empirical
expected
F-score
Number of bits set
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
10
20
30
40
50
60
70
80

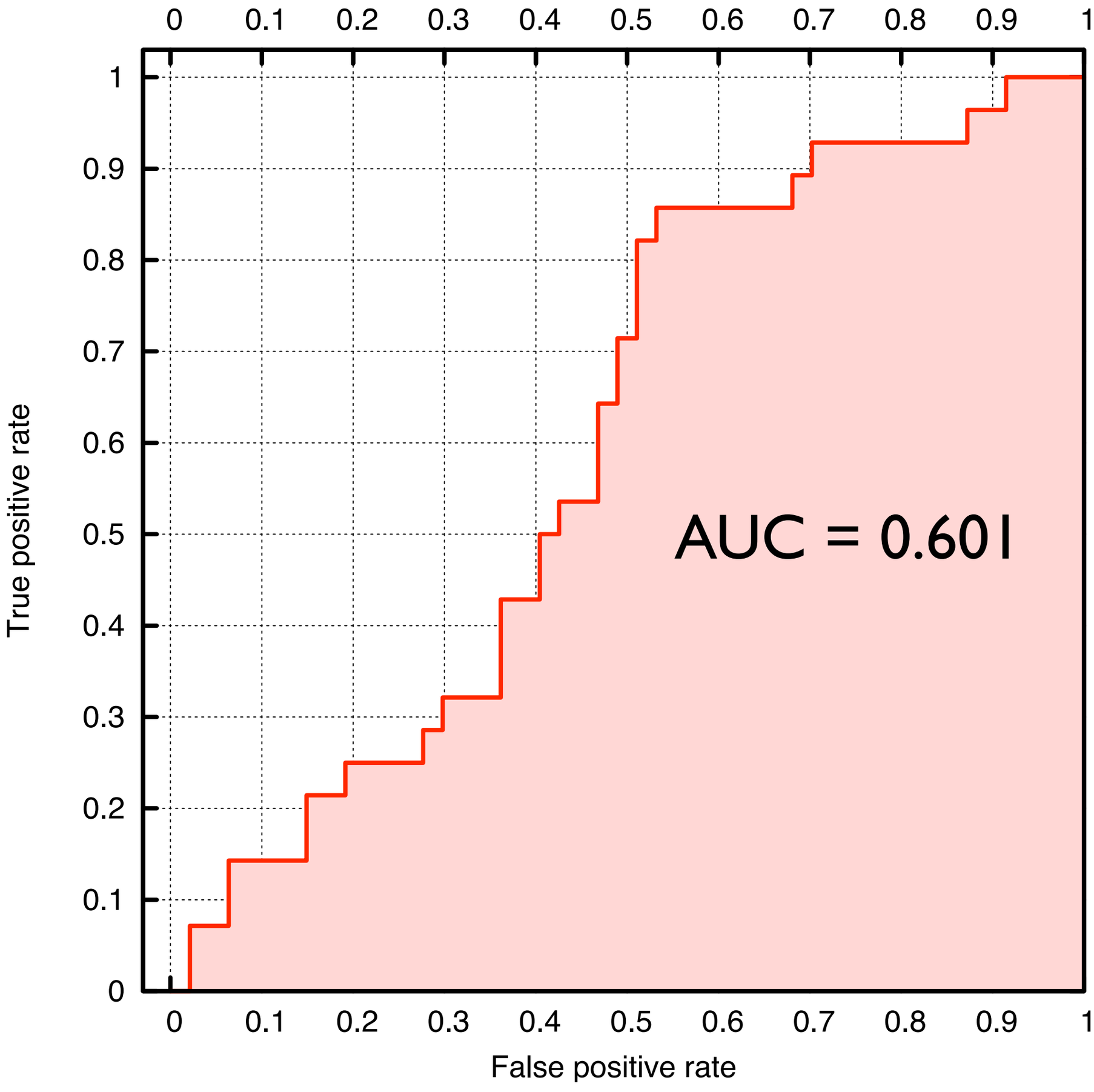

0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
True positive rate
False positive rate
I. setosa

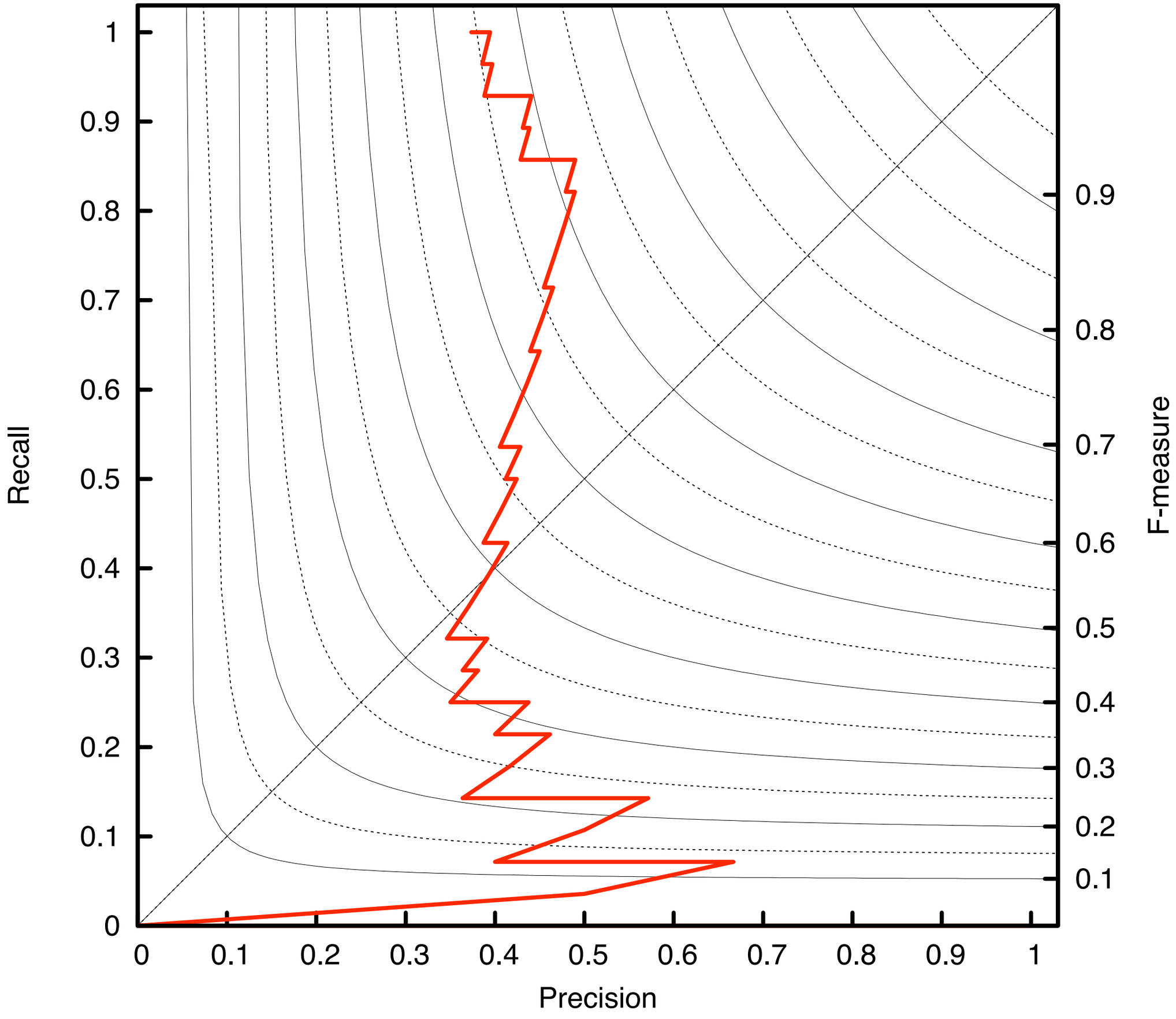

Recall
Precision
F-measure
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

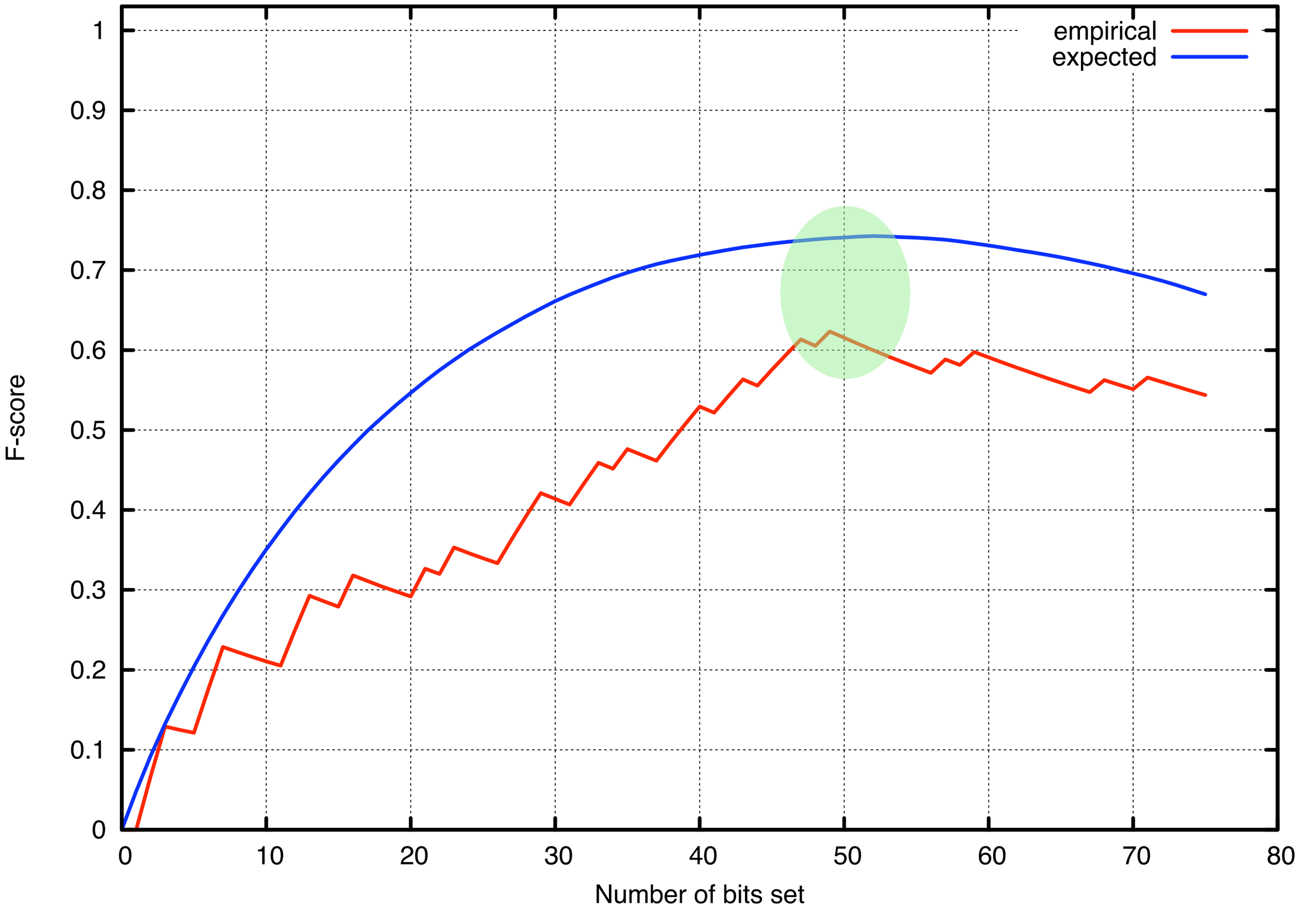

empirical
expected
F-score
Number of bits set
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
10
20
30
40
50
60
70
80

When Good Models Go Bad

- Take *Iris setosa* test set, artificially introduce noise in predictions
- For positive label, draw probability at random, centered at 0.55 with sd 0.4
- For negative label, draw probability at random, centered at 0.45 with sd 0.4


0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
True positive rate
False positive rate
AUC = 0.601


Recall
Precision
F-measure
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


empirical
expected
F-score
Number of bits set
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
10
20
30
40
50
60
70
80

Open Questions

- The Flip Lemma of the decoding algorithm makes a zero-order Markov assumption. Can this be relaxed?
- Many loss functions in NLP can be evaluated efficiently (e.g. *Pk* loss for segmentation tasks, *F* score for ternary sequence labeling), but the existence of efficient inference methods remains open.

- Efficient decoding of ternary IOB label strings is possible if we can solve the following problem:

  Given an acyclic NFA *A*, find an equivalent unambiguous NFA *B* which is at most polynomially larger than *A*.

Conclusions

- We have described a general framework for evaluating the expectations of certain simple loss functions. Our method relies on the fact that many loss functions are sparse, in the sense of having a range that is much smaller than their codomain.
- We have described an efficient method for optimizing the expected *F* score by simplifying the problem analytically.

- We have demonstrated empirically that MEU decoding is effective at predicting extraction quality at different operating points.

- Python source code can be downloaded from purl.org/net/jansche/meu_framework/