Learning Classifiers From Positive and Unlabeled Data

Charles Elkan and Keith Noto
University of California–San Diego

UCSD

The "Positive-only" Issue

Traditionally, classifiers are learned from **positive and negative** examples.

However, often positive examples correspond to observed relationships or phenomena, *i.e.* example X belongs to class Y, but the absence of such an observation does not imply that the relationship does not hold.

Therefore, we have **positive** labeled examples, and no reliably labeled **negative** examples. We do, however, have **unlabeled** examples.

- Online commerce: Would person X buy book/CD/DVD Y? We know all previous purchases.
- Social networks: Does relationship type T exist between persons A and B? We are only given 'yes' examples.
- Ecology: Can organism X live in location/environment Y? We observe only 'yes' examples.
- Cyber security: Is URL X malicious? We record only 'yes' examples.

Suppose the labeled positive examples are a random subset of all positive examples, *i.e.* there is no bias over *which* positive examples have known labels.

Then, we can train a standard classifier to distinguish between positive and **unlabeled** examples and:

(i) The classifier's prediction scores will be off only by a constant factor, and
(ii) The ranked order of future examples will be correct

Lemma

Let x represent the features of an example.

Let y represent its class, *i.e.* positive examples have $y = 1$, negative examples have $y = 0$.

Let s represent whether or not an example is **labeled positive**, *i.e.* some examples have $s = 1$ and $y = 1$, some have $s = 0$ and y unknown.

Lemma 1:

Suppose the "selected at random" assumption holds, *i.e.* $p(s = 1|y = 1, x) = p(s = 1|y = 1)$.

Then $p(y = 1|x) = p(s = 1|x)/c$ where $c = p(s = 1|y = 1)$.

Proof:

$$\begin{aligned} p(s = 1|x) &= p(y = 1 \wedge s = 1|x) && \textit{(only positive examples are labeled)} \\ &= p(y = 1|x)p(s = 1|y = 1, x) && \textit{(chain rule)} \\ &= p(y = 1|x)p(s = 1|y = 1) && \textit{(apply our assumption).} \end{aligned}$$

The result follows by dividing each side by $p(s = 1|y = 1)$. ■

Lemma 1 implies that a classifier trained on labeled positive vs. unlabeled examples **rank** future examples as if it were trained on positive vs. negative examples, since $p(y = 1|x)$ and $p(s = 1|x)$ only differ by a constant factor.

Using Lemma 1

Estimating $p(s = 1|y = 1)$:

- To use Lemma 1 in a classification task, we need a classifier that returns a probability, *i.e.* $p(y = 1|x)$.
- Since we train on **labeled** examples, our classifier actually learns $g(x) = p(s = 1|x)$, *i.e.* the probability that an example is *labeled*, as well as positive.
- We estimate $e \sim p(s = 1|y = 1)$ in three different ways (where V is a validation set):

$$e_1 = \frac{1}{n}\sum_{x \in P} g(x)$$
$$e_2 = \sum_{x \in P} g(x) / \sum_{x \in V} g(x)$$
$$e_3 = \max_{x \in V} g(x)$$

- We then predict $p(y = 1|x)$ as $g(x)/e$
- e_1 is preferable in practice, since it is based on n examples and has the lowest variance. e_3 is based on a single example and has high variance, and e_2 is exposed to additional variance via its denominator.

Double Weighting of Unlabeled Examples:

A second way to use Lemma 1 is to consider each unlabeled example as two weighted labeled examples:

- First, train a classifier on labeled and unlabeled examples and use it to estimate $p(s = 1|y = 1)$, as above.
- Second, label each **unlabeled** example as positive with a weight proportional to $p(y = 1|x, s = 0)$, and as negative with a weight proportional to $1 - p(y = 1|x, s = 0)$.
- Finally, retrain the classifier on the labeled data, and use it as usual.

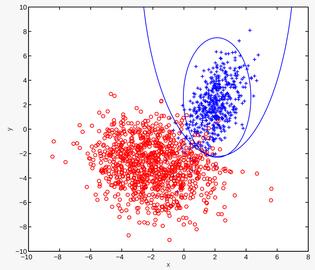
An Illustration

We generate 500 positive examples (blue pluses) and 1,000 negative examples (red circles), each from a two-dimensional Gaussian.

The **large ellipse** is the result of logistic regression trained on **all data, with labels**. It shows the set of points x for which $p(y|x) = 0.5$ is estimated.

The **small ellipse** is the result of logistic regression trained on **20% of the positive data versus all other data**, then transformed following Lemma 1 (using the e_1 estimate).

The two ellipses represent similar classifiers in practice since they agree closely everywhere that data points of both classes have high density.



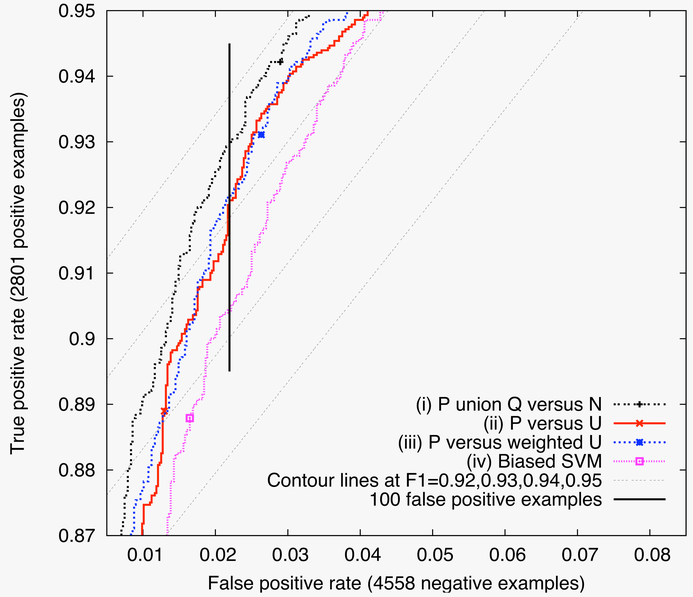
Experiments

Take three sets of Swiss-Prot database records.
Task is to distinguish proteins related to transmembrane and intracellular transport.

- P: Positive-labeled (2,453 examples)
- Q: Positive, but **unlabeled** (348 examples)
- N: Negative, but unlabeled (4,558)

Four methods:

(i) Train $P \cup Q$ vs. N (*i.e.* as if you know true labels); threshold on $p(y = 1|x) = 0.5$.
(ii) Train P vs. $Q \cup N$; threshold on $p(s = 1|x)/e_1 = 0.5$, based on e_1 estimate of $p(s = 1|y = 1)$.
(iii) Train P vs. $Q \cup N$; reweight examples based on estimated $p(s = 1|y = 1)$.
(iv) Train using biased-SVM, current state-of-the-art; Penalties C_U and C_P chosen using validation sets.



- Part of ROC space shown above. Reweighing examples based on estimated $p(s = 1|y = 1)$ or simply adjusting the threshold is more accurate than current state-of-the-art.
- e_1 estimator of $p(s = 1|y = 1)$ within 0.15% (!) of $|P|/|P \cup Q|$ on all cross-validation folds.
- After observing 100 false positives, we miss 9.6% of positive records using the biased SVM, but only 7.6% using the reweighing method–a 21% reduction of error rate, and 55 additional relevant articles in this case.

Computer Science & Engineering - University of California, San Diego - La Jolla, CA, 92093 United States of America
Mail: {elkan,knoto}@cs.ucsd.edu