Non-Parametric Semi-Supervised Learning by Bayesian Label Distribution Propagation

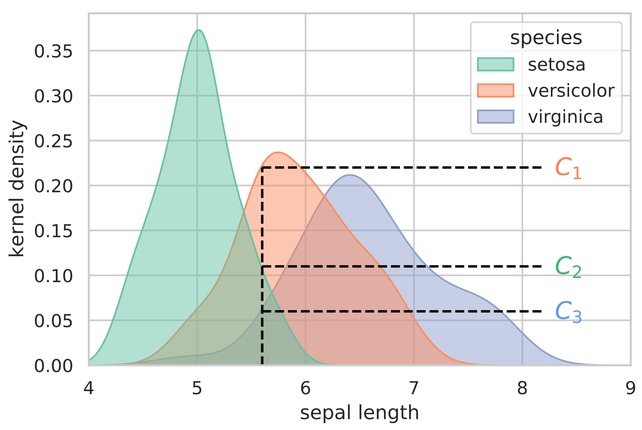
Jonatan M. N. Gøttcke[1], Arthur Zimek [1], Ricardo J. G. B. Campello

[1]University of Southern Denmark, Denmark.
[2]University of Newcastle, Callaghan, Australia.

Abstract

Semi-supervised classification methods are specialized to use a very limited amount of labelled data for training and ultimately for assigning labels to the vast majority of unlabelled data. Label propagation is such a technique that assigns labels to those parts of unlabelled data that are in some sense close to labelled examples and then uses these predicted labels in turn to predict labels of more remote data. Here we propose to not propagate an immediate label decision to neighbors but to propagate the label probability distribution. This way we keep more information and take into account the remaining uncertainty of the classifier. We employ a Bayesian schema that is simpler and more straightforward than existing methods. As a consequence we avoid to propagate errors by decisions taken too early. A crisp decision can be derived from the propagated label distributions at will. We implement and test this strategy with a probabilistic k-nearest neighbor classifier, proving competitive with several state-of-the-art competitors in quality and more efficient in terms of computational resources.

General Schema



Here we see three probability density functions for the three classes over the sepal length attribute in the Iris dataset. We propose to propagate label probability distributions to unlabelled instances in a semi-supervised manner using the (estimated) probability density for each class starting from a straightforward Bayesian schema:

With the prior class probability $\Pr(c)$ for each class $c \in C$ and Bayes theorem, the probability for instance x to belong to a class c can be estimated by:

$$\Pr(c|x) \propto \frac{\hat{f}(x|c)\Pr(c)}{\sum_{c_i \in C} \hat{f}(x|c_i)\Pr(c_i)} \quad (1)$$

where $\hat{f}$ is some estimate of the probability density (which could be a direct probability estimate, if some classifier delivers that).

Such estimated label probability distributions are assigned to all instances $x \in U$, such that the label y_i of instance x_i is in fact a label distribution:

$$y_i = (\Pr(c_1|x_i), \Pr(c_2|x_i), \cdots, \Pr(c_n|x_i))^\top \quad (2)$$

Example model using kNN

To test this concept in semi-supervised classification we employ a probabilistic *k*NN classifier to estimate label probability distributions in a non-parametric way and describe an algorithm to propagate label probability density distributions using *k*NN, resulting in an algorithm *k*NN *Label Distribution Propagation* (*k*NN-LDP). While we keep all information of the label probability distributions as long as possible, we can at any point derive a crisp labelling of a query object if needed, taking the maximum of the assigned class probabilities. In the supervised scenario of using the *k*NN classifier, the label probability distribution for some instance x is given by the class-conditional density estimates based on the k nearest neighbors of x taken over the labeled training data L [1, 2]:

$$\hat{f}(x|c_j) = \frac{|\{x_\ell \in \textit{k}\text{NN}(x) \cap c_j\}|}{|\{x_\ell \in L \cap c_j\}| \cdot \textit{Vol}_{\textit{k}\text{NN}(x)}} \quad (3)$$

where $|\cdot|$ denotes the cardinality of a set and $\textit{Vol}_{\textit{k}\text{NN}(x)}$ denotes the volume needed to cover k nearest neighbors of x, centered at x. The shape of this volume will depend on the employed distance function. Note, however, that the volume cancels nicely out when putting this into Eq. (1).

In the semi-supervised scenario tackled here, an instance among the nearest neighbors might not have a crisp label but a label probability distribution itself, or no label for instances $\in U$. For getting a well-defined probability distribution we can treat the "unknown" case as a special class. Accounting for partial labels in Eq. 3 thus yields

$$\hat{f}(x|c_j) = \frac{\sum_{x_\ell \in \textit{k}\text{NN}(x)} \Pr(c_j|x_\ell)}{\sum_{x_\ell \in L} \Pr(c_j|x_\ell) \cdot \textit{Vol}_{\textit{k}\text{NN}(x)}} \quad (4)$$

Using this in Eq. (1), the probability for each class c in the label distribution, depending on the label probability distributions of the k nearest neighbors, is therefore given by

$$\Pr(c|x) = \frac{\sum_{x_\ell \in \textit{k}\text{NN}(x)} \Pr(c|x_\ell)}{|\,\textit{k}\text{NN}(x)|} \quad (5)$$

Experimental Evaluation

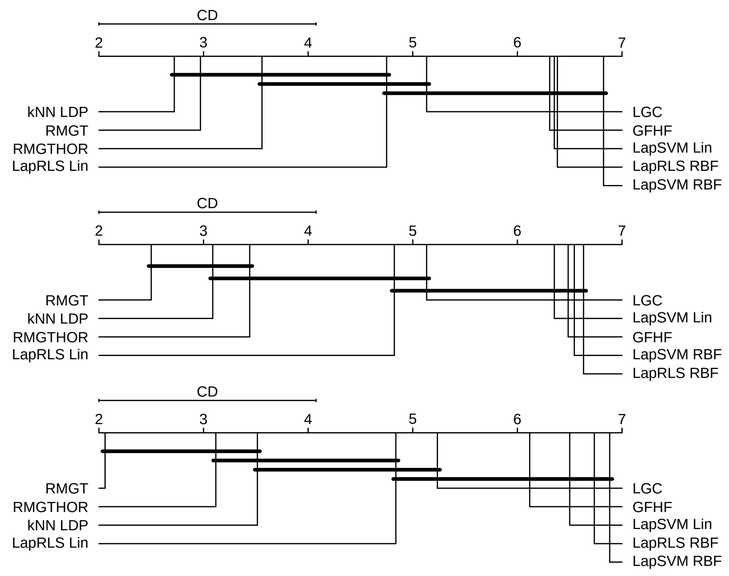


Figure 1: Critical difference plots over a 10-fold cross validation using 5%,10% and 20% labelled data.

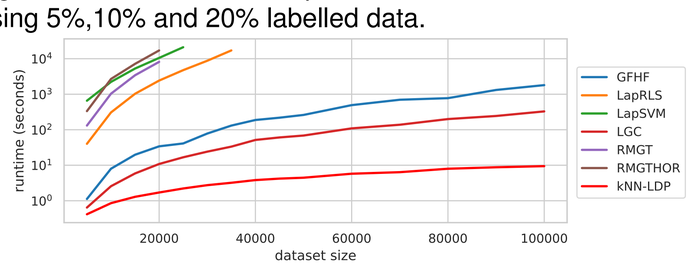


Figure 2: Runtime in seconds scaling with dataset size.

Propagation Algorithm

To use as much information as possible for the label assignment in just two passes over the data we start with the instances where most information is available, that is where the sum of the class probabilities over the neighbors (except for the class "unknown") is maximal, and continue to process instances with decreasing order w.r.t. this available information (which might change over time). This requires checking all neighborhoods in advance. For the sake of efficiency, the forward and reverse k nearest neighbors should be indexed in this first pass.

$$w(x) = \sum_{c \in C \setminus \{\text{"unknown"}\}} \Pr(c|x) \quad (6)$$

for all $x \in U$ **do**
  index forward and reverse k nearest neighbors (*k*NN, R*k*NN)
  $x.w \leftarrow \sum_{c \in C \setminus \{\text{unknown}\}} \Pr(c|x)$ {Eq. (6)}
end for
$PQ_U \leftarrow$ priority-queue(U) {decreasing order w.r.t. $x.w$}
while PQ_U.size > 0 **do**
  $x \leftarrow PQ_U$.getMax()
  if $x.w > 0$ **then**
    $x.y \leftarrow (\Pr(c|x))_{c \in C}$ {Eqs. (2) and (5)}
    for all $p \in$ R*k*NN(x) **do**
      $p.w \leftarrow \sum_{c \in C \setminus \{\text{unknown}\}} \Pr(c|p)$ {Eq. (6)}
      PQ_U.update(p)
    end for
    $L \leftarrow L \cup \{x\}$
  else
    $x.y \leftarrow$ unknown
    for all $p \in PQ_U$ **do**
      $p.y \leftarrow$ unknown
      $L \leftarrow L \cup \{p\}$
    end for
    $PQ_U \leftarrow \emptyset$
  end if
end while

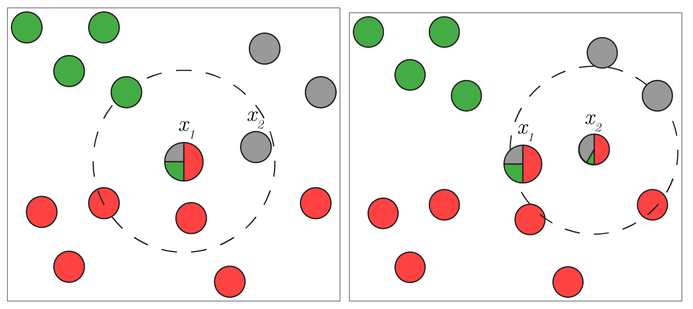


Figure 3: Figure showing the two first iterations of the label distribution propagation algorithm. Instances are being labelled in the order from most to least label information (Eq 6).

Conclusion

We studied an elegant non-parametric method with a clear interpretation in terms of density estimation and Bayesian reasoning here that performs as good as or better than state-of-the-art methods on a large collection of datasets even though it was put on a disadvantage compared to other methods in two aspects:

1. In cases where a vertex in the *k*NN graph belongs to a component with no label information our method simply abstains from making a decision which as such will always count as an error.
2. We performed grid search optimization of several parameters for the competing Laplacian methods and the methods using an RBF kernel. Without such parameter tuning, these methods would not be able to achieve reasonable performance. No such parameter optimization was done for the k-value for methods using nearest neighbor information.

In terms of efficiency and scalability, our method is clearly outperforming the competitors.

References

[1] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern classification*. Wiley, 2nd edition, 2001.

[2] Mohammed J. Zaki and Wagner Meira, Jr. *Data Mining and Analysis: Fundamental Concepts and Algorithms*. Cambridge University Press, 2014. ISBN 9780521766333.

github.com/goettcke/kNN_LDP