Crowdsourced Nonparametric Density Estimation Using Relative Distances

Antti Ukkonen
Finnish Inst. of Occupational Health
Helsinki, Finland
antti.ukkonen@ttl.fi

Behrouz Derakhshan
Rovio Entertainment
Espoo, Finland
behrouz.derakhshan@gmail.com

Hannes Heikinheimo
Reaktor
Helsinki, Finland
hannes.heikinheimo@reaktor.com

Abstract

In this paper we address the following density estimation problem: given a number of relative similarity judgements over a set of items $\mathcal{D}$, assign a density value $p(x)$ to each item $x \in \mathcal{D}$. Our work is motivated by human computing applications where density can be interpreted e.g. as a measure of the rarity of an item. While humans are excellent at solving a range of different visual tasks, assessing absolute similarity (or distance) of two items (e.g. photographs) is difficult. Relative judgements of similarity, such as *A is more similar to B than to C*, on the other hand, are substantially easier to elicit from people. We provide two novel methods for density estimation that only use relative expressions of similarity. We give both theoretical justifications, as well as empirical evidence that the proposed methods produce good estimates.

Introduction

A common application of crowdsourcing is to collect training labels for machine learning algorithms. However, human computation can also be employed to solve computational problems directly (Amsterdamer et al. 2013; Chilton et al. 2013; Trushkowsky et al. 2013; Parameswaran et al. 2011; Bragg, Mausam, and Weld 2013). Some of this work is concerned with solving machine learning problems with the crowd (Gomes et al. 2011; Tamuz et al. 2011; van der Maaten and Weinberger 2012; Heikinheimo and Ukkonen 2013).

In this paper we focus on the problem of *nonparametric density estimation* given a finite sample from an underlying distribution. This is a fundamental problem in statistics and machine learning that has many applications, such as classification, regression, clustering, and outlier detection. Density can be understood simply as "the number of data points that are close to a given data point". Any item in a high density region should thus be very similar to a fairly large number of other items. We argue that in the context of crowdsourcing, density can be viewed for instance as a measure of "commonality" of the items being studied. That is, all items in a high density region can be thought of as having a large number of common aspects or features. Likewise, items from *low density* regions are outliers or in some sense unusual.

To give an example, consider a collection of photographs of galaxies (see e.g. (Lintott et al. 2008)). It is reasonable to assume that some types of galaxies are fairly common, while others are relatively rare. Suppose our task is to find the rare galaxies. However, in the absence of prior knowledge this can be tricky. One approach is to let workers label each galaxy as either common or rare according to the workers' expertise. We argue that this has two drawbacks. First, evaluating commonality in absolute terms in a consistent manner may be difficult. Second, the background knowledge of the workers may be inconsistent with the data distribution. Perhaps a galaxy that would be considered as rare under general circumstances is extremely common in the given data. We argue that in such circumstances density estimation may result in a more reliable method for identifying the rare galaxies. A galaxy should be considered as rare if there are very few (or none) other galaxies that are similar to it in the studied data.

The textbook approach for nonparametric density estimation are kernel density estimators (Hastie, Tibshirani, and Friedman 2009, p. 208ff). These methods usually consider the absolute distances between data points. That is, the distance between items A and B is given on some (possibly arbitrary) scale. Absolute distances between data points are used also in other elementary machine learning techniques, such as hierarchical clustering or dimensionality reduction. However, in the context of human computation, it is considerably easier to obtain information about *relative distances*. For example, statements of the form *"the distance between items A and B is shorter than the distance between items C and D"* are substantially easier to elicit from people than absolute distances. Moreover, such statements can be collected in an efficient manner via crowdsourcing using appropriately formulated HITs (Wilber, Kwak, and Belongie 2014). It is thus interesting to study *what is the expressive power of relative distances*, and are absolute distances even needed to solve some problems?

A common application of relative distances are algorithms for computing low-dimensional embeddings (representations of the data in $\mathbb{R}^m$) either directly (van der Maaten and Weinberger 2012) or via semi-supervised

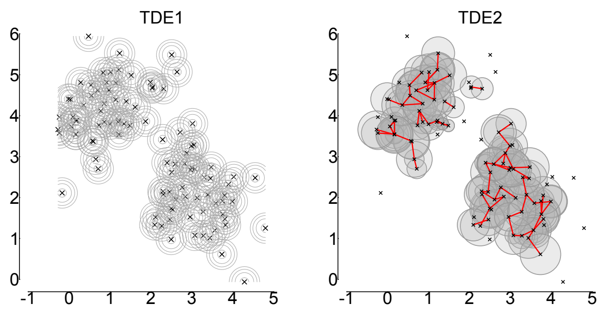

Figure 1: **Left:** Example of the TDE1 estimator we propose. **Right:** Example of the TDE2 estimator that we propose.

metric-learning (Schultz and Joachims 2003; Davis et al. 2007; Liu et al. 2012) if features are available. While embeddings can be used for various purposes, including also density estimation, we are more interested in understanding *what can be done using relative distances directly*, without first computing a representation of the items in some (arbitrary) feature space. This is because despite embeddings being powerful, they can be tricky to use in practical situations, e.g. if the true dimensionality of the data is unknown. Moreover, an embedding may have to be re-computed when new data is added to the model. We hope to avoid this by operating directly with relative distance judgements.

Our contributions

We consider *two approaches for density estimation using relative distances*. In particular, we provide the following contributions:

1. We describe how a standard kernel density estimator can be expressed using relative distance judgements, and describe how embeddings to $\mathbb{R}^m$ can be used to construct this estimator.
2. We propose a second estimator that is explicitly designed for human computation. The main upside of this estimator is that it is easier to learn and use. While the estimator is biased, the amount of the bias can be reduced at the cost of using more work to build the estimator.
3. We conduct experiments on simulated data to compare the two methods, and show that the second estimator is in general easier to learn than the first one, and that it produces better estimates.

High-level overview of the proposed methods

We propose two methods for density estimation using relative distances. The first is a simple variant of a standard kernel density estimator, while the other is a novel method that has some practical advantages over the first one. We call our estimators "triplet density estimators", because the relative distance judgements they use are defined on sets of *three* items.

Our input is a dataset $\mathcal{D}$, assumed to have been sampled iid. from some unknown density p, and we want to estimate the density $p(x)$ at an arbitrary point x. Both of the proposed methods are based on the observation that the density at a given point x is proportional to the number of close neighbours of x that belong to $\mathcal{D}$. If there are many other points in $\mathcal{D}$ that are at a close proximity to x, we say that x is from a high-density region. The question is thus how to identify and count the points that are close to x without measuring absolute distances.

The basic idea of the first method, called TDE1, is illustrated in the left panel of Figure 1. Suppose we have placed a circle with a short radius at every observed data point $u_i \in \mathcal{D}$. Given x, we can count how many of these circles cover x, i.e., the number of u_i that are close to x. Observe that to do this we only need to know whether the distance between x and some u_i is shorter or longer than the radius of the circle around u_i. This is a problem of comparing two distances, and does thus not require to know the absolute distances.

We can thus pair every $u_i \in \mathcal{D}$ with a reference point $u'_i \in \mathcal{D}$ so that the distance between u_i and u'_i equals the radius of the circle. However, as the circles at every u_i must have at least roughly the same radius, finding an appropriate reference point for every u_i can be difficult. We also need a means to find the reference points without access to absolute distances. This problem can be solved to some degree using a low-dimensional a embedding, but it adds to the complexity of the method.

The second method we propose, called TDE2, is simpler in this regard. Rather than having to find a reference point for every $u_i \in \mathcal{D}$, it only uses a random set of *pairs of points* from $\mathcal{D}$. The only constraint is that the pairs should be "short". We say a pair is short if it is short in comparison to all possible pairs of items from $\mathcal{D}$. The intuition is that short pairs occur more frequently in high density regions of the density function p. The second method thus counts the number of short pairs that are close to x.

Here the notion of "closeness" also depends on the length of the pair. This is illustrated in the right panel of Figure 1. The pairs are shown in red, with gray circles around their endpoints. The radius of the circles is equal to the length of the respective pair. The pair is considered to be close to x if x is covered by at least one of the circles that are adjacent to the pair. Very short pairs must be very close to x in order to cover it, while longer pairs also cover points that are further away.

It is easy to see that TDE1 is essentially a standard kernel density estimator that uses the uniform kernel. The TDE2 estimator is designed for relative distances from the ground up and has thus some advantages over TDE1. In particular, TDE2 is simpler to construct.

TDE1: A simple triplet density estimator

Our first method can be viewed as a standard kernel density estimator that does not use absolute distances. First, we recall the basic definition of a kernel density estimator as originally proposed (Rosenblatt 1956; Parzen 1962), and as usually given in textbooks, e.g. (Hastie, Tibshirani, and Friedman 2009, p. 208ff). Then, we describe how a simple instance of this can be implemented only using relative distances.

Basics of kernel density estimation

Let $\mathcal{D} = \{u_1, \ldots, u_n\}$ denote a data set of size n drawn from a distribution having the density function p. For the moment, assume that $\mathcal{D}$ consists of points in $\mathbb{R}^m$. Given $\mathcal{D}$, the kernel density estimate (KDE) of p at any point x is given by

$$\mathsf{KDE}_n(x) = \frac{1}{nh^m} \sum_{i=1}^{n} K(\frac{1}{h}(x - u_i)), \tag{1}$$

where K is the *kernel function* and h is the *bandwidth* parameter. Fairly simple conditions on the kernel function K and the bandwidth h are sufficient to guarantee that the kernel density estimate is *an unbiased and consistent* estimate of p. That is, as the size of $\mathcal{D}$ increases, we have

$$\lim_{n\to\infty} \mathrm{E}_{\mathcal{D}\sim p}[\mathsf{KDE}_n(x)] = p(x), \tag{2}$$

where the expectation is taken over random samples from p of size n. Most importantly, the kernel function K must satisfy

$$K(y) \geq 0 \;\forall y \quad \text{and} \quad \int_{\mathbb{R}^m} K(y)dy = 1. \tag{3}$$

In this paper we consider *radial-symmetric* kernels. In addition to Equation 3 these satisfy $K(y) = K(y') \Leftrightarrow \|y\| = \|y'\|$. That is, the kernel is a function of only the length of its input vector. Also, we want $K(y)$ to be *non-increasing* in $\|y\|$. Now, assume that there exists a distance function d between the items in $\mathcal{D}$. For the moment, let $d(x, y) = \|x-y\|$. We can rewrite Equation 1 as follows:

$$\mathsf{KDE}_n(x) = \frac{1}{nh^m} \sum_{i=1}^{n} K(\frac{d(x, u_i)}{h}). \tag{4}$$

The bandwidth h is a parameter that must be specified by the user. There are a number of methods for choosing h in an appropriate manner. In practice the bandwidth h should decrease as n increases. In fact, we must have $\lim_{n\to\infty} h(n) = 0$ for Eq. 2 to hold. In the experiments we use cross validation to select an appropriate value of h when needed.

Normalised densities and relative distances

Before continuing, we point out a property of density functions and relative distances. Recall that the integral of a density function over its entire domain must by definition be equal to 1. This means that the value $p(x)$ is *not scale invariant*: the precise value of $p(x)$ depends on the "unit" we use to measure distance. If we measure distance in different units[1], the absolute value of $p(x)$ must change accordingly as well. This is reflected in Equation 1 through the bandwidth parameter h that must have the same unit as the distance $d(u, v)$.

However, when dealing with relative distances there is no distance unit at all. We only know that some distance is shorter (or longer) than some other distance, but the unit in which the distances are measured is in fact irrelevant. An important consequence of this is that *it is not possible to devise a normalised density estimator based on relative distances*. In other words, Equation 2 can not hold with equality. The best we can hope for is to find an estimator that is proportional to $p(x)$. Later we show how to derive a normalised estimator for one-dimensional data, but this makes use of absolute distances in the normalisation constant, much in the same way that h appears in the normalisation constant of Equation 1. In practice we must thus consider non-normalised estimates of $p(x)$ when using relative distances.

[1]Concretely, suppose we use feet instead of meters.

Uniform kernel with relative distances

The simplest kernel function K that satisfies the requirements given above is the *uniform kernel*. It takes some constant positive value if x is at most at distance h from the data point $u_i \in \mathcal{D}$ at which the kernel is located, and is otherwise equal to zero. Formally the m-dimensional uniform kernel is defined as

$$K^{\mathrm{unif}}(y) = \begin{cases} V(m)^{-1} & : y \leq 1, \\ 0 & : y > 1, \end{cases} \tag{5}$$

where $V(m)$ is the volume of the m-dimensional unit-sphere. This is defined as $V(m) = \frac{\pi^{\frac{m}{2}}}{\Gamma(\frac{m}{2}+1)}$, where Γ is the Gamma function.

In this case the density estimate at x is simply proportional to the number data points that are at most at a distance of h from x. We say the point u *covers* x *with bandwidth* h, whenever $K^{\mathrm{unif}}(\frac{d(x,u)}{h}) \leq 1$, This can be seen as a straightforward generalisation of a histogram that has unspecified bin boundaries. In Figure 1 (left) we can say that the density at x is proportional to the number of "discs" that cover x. It is easy to see that as n increases and h goes to zero (i.e., there are more points covered by smaller and smaller discs), this quantity will indeed approach the underlying density p.

We continue by designing a triplet density estimator that uses the uniform kernel K^{unif}. To compute K^{unif}, we must only determine if a data point u_i near enough to x. However, we cannot evaluate $d(u, v)$ explicitly for any $u, v \in \mathcal{D}$. Instead, we only have access to an "oracle" that can give statements about the relative distance between items. That is, we can issue tasks such as

"Is the distance $d(u, x)$ shorter than the distance $d(u, v)$?"

Or, in more natural terms:

"Of the items x and v, which is more similar to u?"

We can use this task to compute the uniform kernel as follows. Suppose that for every $u_i \in \mathcal{D}$, we have found some other $u'_i \in \mathcal{D}$, so that $d(u_i, u'_i)$ is equal to h. Denote this pairing by $\mathcal{P}(h)$. Note that several $u \in \mathcal{D}$ can be paired with the same u', $\mathcal{P}(h)$ is not required to be a proper *matching*. Given $\mathcal{P}(h)$, by the definition of the uniform kernel given in Equation 5, we have

$$K^{\mathrm{unif}}\left(\frac{d(x, u_i)}{h}\right) = V(m)^{-1}\, \mathrm{I}\{d(u_i, x) \leq d(u_i, u'_i)\}. \tag{6}$$

This, together with Equation 4 gives the triplet density estimator

$$\mathsf{TDE1}_n(x) = \frac{1}{V(m)nh^m} \sum_{(u,u') \in \mathcal{P}(h)} \mathrm{I}\{d(u,x) < d(u,u')\}. \tag{7}$$

This simply counts the number of $u \in \mathcal{D}$ that cover x with the specified bandwidth h. As we only need an oracle that can provide relative distance judgements to evaluate the indicator function above, Equation 7 provides a kernel density estimator that is suitable for human computation. Moreover, all known results that apply to Equation 1 (e.g. the one of Equation 2) directly carry over to Equation 7.

There are some issues with this result, however. As discussed above, in practice we cannot evaluate the normalisation constant: both m and the exact value of h are unknown if our input only consists of relative distance comparisons.

More importantly, it is unlikely that for a given $\mathcal{D}$ we would find a pairing $\mathcal{P}(h)$ where the distances between all pairs are exactly equal to h. It is more realistic to assume that we can find a pairing $\tilde{\mathcal{P}}(h)$ where this holds approximately. That is, for (almost) every $(u,u') \in \tilde{\mathcal{P}}(h)$, we have $d(u,u') \approx h$. This implies that results about unbiasedness and convergence that hold for Equation 1 are no longer obviously true. While *variable bandwidth* density estimators have been proposed in literature, our problem differs from those in the sense that we do not try to select an optimal local bandwidth around each $u \in \mathcal{D}$, but the bandwidth at u randomly varies around h. In general it is nontrivial to even quantify this variance, or show how it affects the resulting estimator.

This raises one important question, however. If we are not able to evaluate the absolute value of $d(u,v)$, how can we find pairs that are (at least approximately) of length h?

Using an embedding to find $\tilde{\mathcal{P}}(h)$

As mentioned in the Introduction, by embedding the items to a low-dimensional Euclidean space, we can trivially estimate density using existing techniques. Such embeddings can be computed from relative distances e.g. using the algorithm described in (van der Maaten and Weinberger 2012). This has the downside that to estimate density at a new point x, we in general must recompute the embedding to see where x is located. Using the technique described above, however, it is possible to estimate density directly, provided we have the set of pairs $\mathcal{P}(h)$.

Therefore, we use the embedding only for constructing the set $\tilde{\mathcal{P}}(h)$ as follows:

1. Embed all items in $\mathcal{D}$ to $\mathbb{R}^m$ using some suitable algorithm. We use the method proposed in (van der Maaten and Weinberger 2012). This results in a set of points X. Note that the dimensionality m of the embedding is a parameter that must be specified by the user.
2. Use any existing technique for bandwidth selection to find a good h given the points $X \in \mathbb{R}^m$. We use cross validation in the experiments later.
3. For every item $u \in X$, find the item $v \in X$ so that the distance between u and v is as close to h as possible. This yields the set $\tilde{\mathcal{P}}(h)$.

The pairs (u,v) in $\tilde{\mathcal{P}}(h)$ have all approximately length h in the embedding. Of course this length is not the same as the unknown distance $d(u,v)$, but we can assume that $d(u,v)$ is approximately the same for every $(u,v) \in \tilde{\mathcal{P}}(h)$.

$\mathsf{TDE2}$: A second triplet density estimator

The density estimator proposed above (Equation 7) must construct a low dimensional embedding of the items to find the pairs $\tilde{\mathcal{P}}(h)$. While there are a number of techniques for finding such embeddings, they are not necessarily optimal for density estimation. Their objective may be to create visually pleasing scatterplots, which requires to introduce additional constraints that will favor embeddings with certain properties. For instance, the methods may aim to focus on short distances so that the local neighbourhood of each data point is preserved better. This emphasises clusters the data may have. As a consequence the resulting distances may become disproportionally skewed towards short distances. However, for the bandwidth selection process to produce good results, we would prefer the resulting distances to be as truthful as possible, possibly at the cost of the embedding resulting in a less clear visualisation. Moreover, the embedding algorithms tend to be sensitive to initialisation.

Motivated by this, we propose a second triplet density estimator that does not need an embedding. In particular, we show that it is possible to produce good density estimates, provided we can find a sufficient number of *short pairs*. Unlike in the method proposed above, they do not all have to be approximately of the same length h. Instead, it is sufficient to find pairs that are all at most of a certain length that we will denote by $\delta_{\max}$. We first describe the estimator and prove some of its properties, and continue by discussing the problem of finding short pairs without an embedding.

Basic idea

This variant of the density estimator uses a slightly different human intelligence task:

"Which of the three items x, u, and v is an outlier?"

The worker must identify one of the three items as the one being the most dissimilar to the other two. In (Heikinheimo and Ukkonen 2013) this task was used to compute the centroid of a set of items.

We model this HIT by a function Ω that takes as input the item x and the pair (u,v), and returns 1 if the distance between x and u or v is shorter than the distance between u and v, and 0 otherwise. That is, we have $\Omega(x,(u,v)) = 0$ if x is chosen as the outlier and $\Omega(x,(u,v)) = 1$ if either u or v is chosen as the outlier. More formally, we have

$$\Omega(x,(u,v)) = \mathrm{I}\{\min(d(x,u), d(x,v)) \le d(u,v)\},$$

and we say that the pair (u,v) *covers* the item x whenever $\Omega(x,(u,v)) = 1$.

Our method is based on the following observation. Let $\mathcal{P} = \{(u_i,v_i)\}_{i=1}^{l}$ denote a set of pairs, where every u_i and v_i is drawn from $\mathcal{D}$ uniformly at random. Moreover, let $\mathcal{P}^{\delta_{\max}} \subset \mathcal{P}$ denote the set of "short pairs", meaning those for which we know the distance $d(u, v)$ to be below some threshold $\delta_{\max}$. Now, fix some $x \in \mathcal{D}$, and consider the number of short pairs $(u, v) \in \mathcal{P}^{\delta_{\max}}$ that cover x. If x resides in a high density region of p, we would expect there to be many short pairs that cover x, while there are less of such pairs if x is from a low density region. This is illustrated in the right panel of Fig. 1.

The estimator that we propose, called TDE2, is defined as described above. We first draw an iid. sample of pairs $\mathcal{P}$ from p, and then remove all such pairs from $\mathcal{P}$ that are longer than $\delta_{\max}$ to obtain $\mathcal{P}^{\delta_{\max}}$. As we discuss in the remainder of this section, we can estimate density $p(x)$ at any given x by computing the number of pairs in $\mathcal{P}^{\delta_{\max}}$ that cover x and taking a simple transformation of the resulting number.

Preliminary definitions and some simple results

As discussed above, the estimate that TDE2 gives for $p(x)$ is based on the number of short pairs that cover x when the pairs are drawn independently from p. Suppose x is fixed, and we obtain the pair (u, v) by first drawing u according to p, and then draw v independently of u also from p. Let $\delta_{\max}$ denote some distance. Consider the events

$$\begin{aligned} A &= \text{“\textit{the item } } x \text{ \textit{is covered by the pair} } (u,v)\text{“, and} \\ B &= \text{“\textit{the distance} } d(u,v) \text{ \textit{is at most} } \delta_{\max}\text{“.} \end{aligned}$$

Let $q(x, \delta_{\max})$ denote the probability $\Pr[A \cap B]$. The TDE2 estimator actually estimates $q(x, \delta_{\max})$. Observe that we have

$$q(x, \delta_{\max}) = \Pr[A \cap B] = \Pr[A \mid B] \Pr[B]. \tag{8}$$

Now, let $\mathcal{P}$ be a set of pairs sampled iid. from p, and denote by $\mathcal{P}^{\delta_{\max}}$ the subset of $\mathcal{P}$ that only contains pairs up to length $\delta_{\max}$, i.e., $\mathcal{P}^{\delta_{\max}} = \{(u, v) \in \mathcal{P} \mid d(u, v) \le \delta_{\max}\}$. Observe that $\frac{|\mathcal{P}^{\delta_{\max}}|}{|\mathcal{P}|}$ is an unbiased and consistent estimate of $\Pr[B]$, i.e., we have

$$\lim_{|\mathcal{P}| \to \infty} \frac{|\mathcal{P}^{\delta_{\max}}|}{|\mathcal{P}|} = \Pr[B]. \tag{9}$$

To estimate $\Pr[A \mid B]$, we define the *triplet score* of the item x given set of pairs $\mathcal{P}$ as

$$S(x, \mathcal{P}^{\delta_{\max}}) = \sum_{(u,v) \in \mathcal{P}^{\delta_{\max}}} \Omega(x, (u, v)), \tag{10}$$

and find that $\frac{S(x,\mathcal{P}^{\delta_{\max}})}{|\mathcal{P}^{\delta_{\max}}|}$ is an estimate of the probability $\Pr[A \mid B]$, i.e.,

$$\lim_{|\mathcal{P}| \to \infty} \frac{S(x, \mathcal{P}^{\delta_{\max}})}{|\mathcal{P}^{\delta_{\max}}|} = \Pr[A \mid B]. \tag{11}$$

By combining equations 8, 9 and 11, we obtain the following:

Proposition 1. *Given the set $\mathcal{P}$ of pairs sampled independently from p, we have*

$$\lim_{|\mathcal{P}| \to \infty} \frac{S(x, \mathcal{P}^{\delta_{\max}})}{|\mathcal{P}|} = q(x, \delta_{\max}).$$

This shows that TDE2 is an unbiased and consistent estimate of $q(x, \delta_{\max})$.

Results for 1-dimensional data

Next, we discuss the connection between $q(x, \delta_{\max})$ and $p(x)$. Our objective is to show that by taking a suitable transformation of our estimate for $q(x, \delta_{\max})$ we can obtain a reasonable estimate for $p(x)$.

For one-dimensional data it is fairly easy to see when events A and B happen simultaneously and derive an expression for $q(x, \delta_{\max})$. Consider all pairs of length δ starting from $\delta = 0$ up to $\delta = \delta_{\max}$. For every such pair (u, v) of length δ, item x is covered by (u, v) exactly when u is contained in the interval $[x - 2\delta, x + \delta]$, and v is at $u + \delta$. (Thus v must be in the interval $[x - \delta, x + 2\delta]$.) If this happens we have $\min\{d(x, u), d(x, v)\} \le d(u, v)$ and $\Omega(x, (u, v)) = 1$. Next, suppose the *midpoint* of the pair (u, v) is at j. Clearly we must have $u = j - \delta/2$ and $v = j + \delta/2$. Using these we can write $q(x, \delta_{\max})$ as

$$q(x, \delta_{\max}) = 2 \int_{\delta=0}^{\delta_{\max}} \int_{j=x-\frac{3}{2}\delta}^{x+\frac{3}{2}\delta} p(j - \frac{\delta}{2}) p(j + \frac{\delta}{2})\, dj\, di. \tag{12}$$

The constant 2 is needed because we can obtain (u, v) by first drawing u and then v, or vice versa. The following proposition shows how $q(x, \delta_{\max})$ is connected to $p(x)$.

Proposition 2. *Let the function $q(x, \delta_{\max})$ be defined as in Equation 12. We have*

$$\lim_{\delta_{\max} \to 0} \sqrt{\frac{q(x, \delta_{\max})}{3{\delta_{\max}}^2}} = p(x). \tag{13}$$

Proof. See Appendix. □

This shows that by choosing a sufficiently small $\delta_{\max}$, we can obtain approximate estimates for $p(x)$ by taking a simple transformation of the estimate for $q(x, \delta_{\max})$. We conclude this section by combining this observation with propositions 1 and 2 to the following result.

Corollary 1. *Let p be a univariate density function, and let $\mathcal{P}$ be a set of pairs sampled independently from p. Let $\mathcal{P}^{\delta_{\max}} \subseteq \mathcal{P}$ contain those pairs in $\mathcal{P}$ having length at most $\delta_{\max}$. For a sufficiently small $\delta_{\max}$ we have*

$$\hat{p}(x, \delta_{\max}) = \sqrt{\frac{S(x, \mathcal{P}^{\delta_{\max}})}{3{\delta_{\max}}^2 |\mathcal{P}|}} \approx p(x).$$

Observe that $\hat{p}(x, \delta_{\max})$ as defined above yields estimates that are appropriately normalised. The estimates are guaranteed to converge to $q(x, \delta_{\max})$, and have a bias wrt. $p(x)$ that depends on $\delta_{\max}$. By “sufficiently small” we mean that $\delta_{\max}$ should be as small as possible without making the set $\mathcal{P}^{\delta_{\max}}$ too small. We continue with a simple study of the effect of $\delta_{\max}$ on the bias and variance of $\hat{p}(x, \delta_{\max})$.

Bias and variance of the 1-dimensional estimator, an empirical study

The estimator $\hat{p}(x, \delta_{\max})$ (Corollary 1) is biased, with the bias increasing as $\delta_{\max}$ increases. We studied the effect of the bias empirically. The top panel in Fig. 2 shows the shape of $\sqrt{\frac{q(x,\delta_{\max})}{3{\delta_{\max}}^2}}$ for different values of $\delta_{\max}$ when p is the

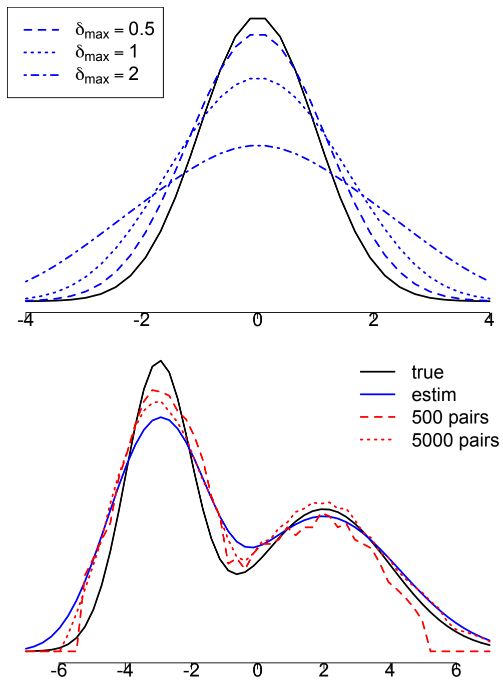

Figure 2: **Top:** Examples of the estimator $\hat{p}(x, \delta_{\max})$ for different values of $\delta_{\max}$ when the underlying density is a standard normal distribution. **Bottom:** Examples of estimating a bimodal distribution from a sample of 500 data points with 500 and 5000 pairs (dashed lines) when $\delta_{\max} = 1.0$.

standard normal distribution. These are what $\hat{p}(x, \delta_{\max})$ will converge to as the size of $\mathcal{P}$ increases. On the other hand, for "small" $\delta_{\max}$ the estimate for $q(x, \delta_{\max})$ will have a high variance as there are only few pairs of length at most $\delta_{\max}$. The top panel of Fig. 3 shows how the errors behave due to bias and variance as a function of $\delta_{\max}$. The estimate is computed from 200 points sampled from a standard normal distribution. For small $\delta_{\max}$ the total error consists mainly of variance due to the small size of $\mathcal{P}^{\delta_{\max}}$. As $\delta_{\max}$ increases the variance component diminishes, and most of the remaining error is due to bias.

In practice it is possible to set $\delta_{\max}$ so that the estimator will produce useful results. The bottom panel of Fig. 2 shows an example of a mixture of two normal distributions (black curve), the limit to which $\hat{p}(x, 1.0)$ would converge (blue curve), as well as two estimates computed with 500 and 5000 pairs in $\mathcal{P}$, respectively (the red dashed curves). We can observe that even 500 pairs are enough to obtain a reasonable estimate. Finally, the bottom panel of Fig. 3 shows how the estimation accuracy (correlation between the estimate and the true density) behaves as the number of pairs in $\mathcal{P}$ is increased from 250 to 2500.

Generalisation to higher dimensions

Above we showed that for one-dimensional data the square root of $q(x, \delta_{\max})$ is proportional to $p(x)$ as $\delta_{\max}$ approaches zero. We can show a similar result also for arbitrary

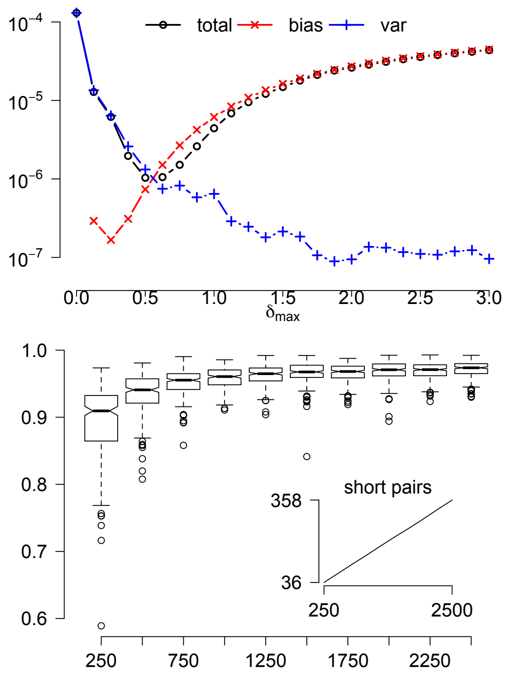

Figure 3: **Top:** Estimation error (MSE, wrt. the true density) as a function of $\delta_{\max}$ when the estimate is computed using 1000 pairs drawn from a data of 200 data points sampled from a standard normal distribution. **Bottom:** Correlation of the estimate ($\delta_{\max} = 0.5$) with true density (the bimodal distribution shown on the left) as a function of the number of pairs in $\mathcal{P}$. The inset shows the number of short pairs that remain in $\mathcal{P}^{\delta_{\max}}$ after removing those longer than $\delta_{\max}$.

dimensional inputs.

Proposition 3. *Let $q(x, \delta_{\max})$ be defined as in Equation 8, and let $C(\delta_{\max})$ be some function of $\delta_{\max}$. We have*

$$\lim_{\delta_{\max} \to 0} \sqrt{\frac{q(x, \delta_{\max})}{C(\delta_{\max})}} = p(x). \quad (14)$$

Proof. See Appendix. □

This result is important because in general we cannot know the true, underlying dimensionality of the data. Hence, our estimator should be independent of the dimensionality, and Proposition 3 shows this to be the case. Finally, since $q(x, \delta_{\max})$ is in practice proportional to the triplet score $S(x, \mathcal{P}^{\delta_{\max}})$ (Eq. 10), our estimator can be defined simply as $\sqrt{S(x, \mathcal{P}^{\delta_{\max}})}$, because both $C(\delta_{\max})$ as well as $|\mathcal{P}|$ are constants that do not depend on x.

Finding short pairs without embeddings

Above we showed how the length of the pairs that are used to compute $\hat{p}(x, \delta_{\max})$ affects the quality of the estimate. In particular, we should only use pairs that are "short" in comparison to the average length of the pairs. In practice we do *not* have the absolute lengths of the pairs, however.

We can only obtain relative distance information, and cannot thus simply filter the pairs based on their absolute length. Instead, we must find short pairs by some other means. Next we discuss a simple method for doing this.

Let $\mathcal{P}^{\delta_{\max}}$ denote the set of pairs that are at most of length $\delta_{\max}$. Since all pairs in $\mathcal{P}$ are drawn uniformly at random from p, we know that $\frac{|\mathcal{P}^{\delta_{\max}}|}{|\mathcal{P}|}$ is an unbiased estimate of the probability $\Pr[d(u,v) \leq \delta_{\max}]$ as was discussed above. Inversely, by fixing some value $\alpha \in [0, 100]$, we can consider all pairs in $\mathcal{P}$ for which the length falls below the α:th percentile of the length distribution of all pairs in $\mathcal{P}$. By setting α to some small value, e.g. $\alpha = 10$, we obtain pairs that correspond to some unknown "small" value of $\delta_{\max}$, so that $\frac{\alpha}{100} = \Pr[d(u,v) \leq \delta_{\max}]$. Let $\mathcal{P}^{\alpha}$ denote the set of pairs selected for a given α. This is simply an alternative formulation for "short" pairs without considering absolute distances.

Finding the α:th percentile of a set of values can be done efficiently using a *selection algorithm*. In general, these find the i:th largest value in a given list of values without sorting the values. They commonly run in linear time in the size of their input, which is $|\mathcal{P}|$ in this case. Note that e.g. the Quickselect algorithm (Hoare 1961) (and its variants, e.g. the Floyd-Rivest algorithm (Floyd and Rivest 1975)) also find all pairs that are below the percentile as a simple side-effect. No separate filtering step is thus needed.

Most selection algorithms are based on pairwise comparisons, and can thus be implemented without explicitly evaluating $d(u,v)$ for any $(u,v) \in \mathcal{P}$. We only need an oracle that can evaluate the indicator function $\mathrm{I}\{d(u,v) < d(u',v')\}$. This is easily expressed as the following human intelligence task:

"Which of the two pairs, (u,v) or (u',v'), are more similar?"

Moreover, by definition of the percentile, we will select $\frac{\alpha}{100}|\mathcal{P}|$ pairs. This can be used to control the size of the resulting estimator (the set $\mathcal{P}^{\alpha}$). The TDE1 estimator proposed above uses by definition $n = |\mathcal{D}|$ pairs. To make the TDE2 estimator of comparable in complexity, we should use $\frac{100n}{\alpha}$ pairs in $\mathcal{P}$ to have $\mathcal{P}^{\alpha}$ of the desired size. Note that both $\mathcal{P}$ and $\mathcal{P}^{\alpha}$ are in this case *linear* in n. This is a desirable property as the number of pairs will directly translate to the number of HITs needed to construct as well as to evaluate the estimator. For this it is crucial that we do not need *all* pairs that are shorter than $\delta_{\max}$, but a uniform sample of these is sufficient.

Using a standard selection algorithm for finding $\mathcal{P}^{\alpha}$ has the downside that the necessary HITs cannot be solved independently. E.g. Quickselect operates in the same way as Quicksort by placing items to either side of a pivot item. This creates dependencies among the tasks, where one task must be solved before others can even be submitted to the workers. Developing a selection algorithm without such dependencies is an interesting open question.

Experiments

In this Section we compare the proposed density estimators, TDE1 and TDE2, using simulations.

Experimental setup

We measure cost as the number of HITs that are needed to *construct* the estimator. The methods are implemented as follows:

- TDE1: The estimator is constructed by first embedding a set of items into $\mathbb{R}^2$ with the method of (van der Maaten and Weinberger 2012). (In practice we have to make some guess about the input data dimensionality when using this method. In these experiments we assume a fixed dimensionality of 2 independently of the true data dimensionality.) Given the embedding, we find an optimal bandwidth h using cross validation, and select the set $\mathcal{P}(h)$ by pairing every $u \in \mathcal{D}$ with the $v \in \mathcal{D}$ so that the distance between u and v in the embedding is as close to h as possible (see Section). The estimates are computed with Equation 7 (by omitting the normalisation).
- TDE2: We use the selection algorithm based method of Section to compute the set $\mathcal{P}^{\alpha}$. To make the cost of computing an estimate for a single x equal to the cost of TDE1, we let $|\mathcal{P}| = \frac{100}{\alpha}|\mathcal{D}|$. This will result in $\mathcal{P}^{\alpha} = |\mathcal{D}|$. We used $\alpha \in \{0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35\}$. The estimates are computed as $\sqrt{S(x,\alpha)}$ where $S(x,\alpha)$ is defined analogously to $S(x, \delta_{\max})$.

We use a separate training data $\mathcal{D}^{\mathrm{train}}$ to learn the estimator, and evaluate it on the test data $\mathcal{D}^{\mathrm{test}}$ that is drawn from the same distribution as $\mathcal{D}^{\mathrm{train}}$. In all cases we let $|\mathcal{D}^{\mathrm{train}}| = |\mathcal{D}^{\mathrm{test}}| = 200$ items sampled iid. from the density in question. We use the following synthetic datasets that are mixtures of multivariate normal distributions:

Gaussian-5D : A single standard multivariate normal distribution in a 1-dimensional space.

Gaussian-5D : A single standard multivariate normal distribution in a 5-dimensional space.

2Gaussians-3D : Two standard multivariate normal distributions in a 3-dimensional space, located at $(0,0,0)$ and $(3,3,3)$.

2Gaussians-5D : As 2Gaussians-3D but in a 5-dimensional space.

4Gaussians-2D : Four standard multivariate normal distributions in a 2-dimensional space, located at $(0,0)$, $(0,5)$, $(5,0)$ and $(5,5)$.

4Gaussians-3D : Four standard multivariate normal distributions in a 3-dimensional space, located at $(0,0,0)$, $(5,0,0)$, $(0,5,0)$ and $(0,0,5)$.

We only consider fairly low-dimensional data with $k \leq 5$. This is because density estimation becomes less meaningful for high-dimensional data due to the curse of dimensionality. We think this is a sufficient study of the problem also from a practical point of view. In crowdsourcing applications we do not expect the workers to use more than a few dimensions when giving similarity judgements.

Also, we assume that computation is noise-free. That is, we obtain consistent and correct answers to each HIT. In practice some quality control mechanism such as (Dawid and Skene 1979; Raykar and Yu 2012) can be used to reduce the effect of erroneous solutions to tasks. We assume

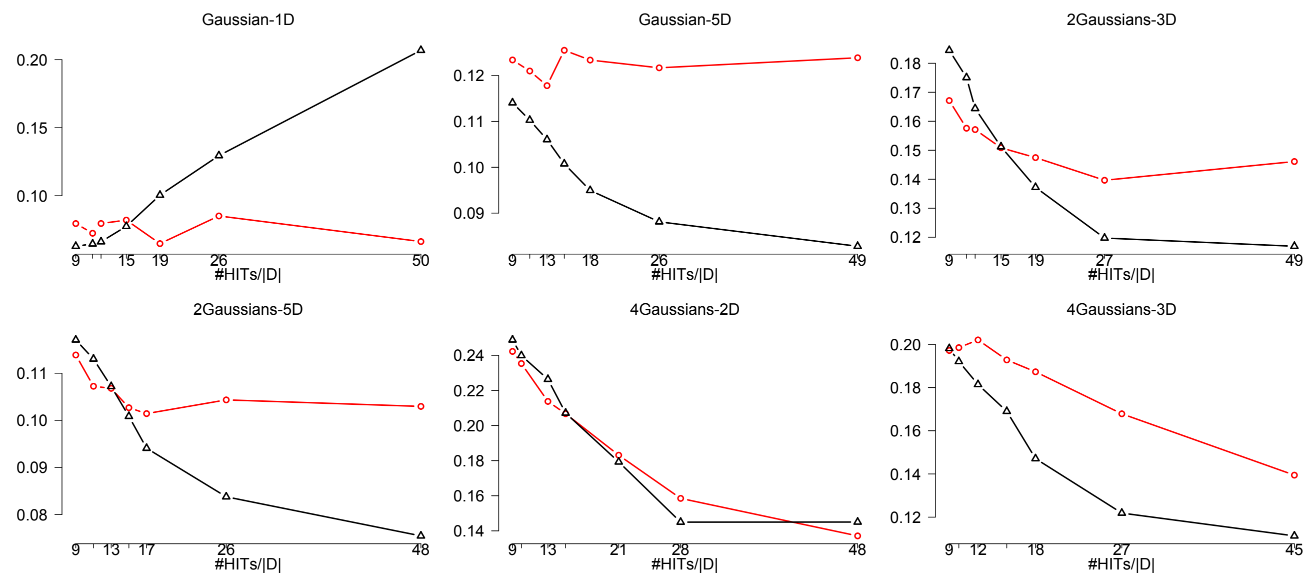

Figure 4: Experimental results. Estimation error (MAE) as a function of the number of HITs used to construct the estimator. TDE1 is shown in **red**, TDE2 is shown in **black**.

this can be taken care of by the crowdsourcing platform, and may increase the costs by a small constant factor.

Both methods use $|\mathcal{D}|$ HITs to compute a single estimate. We first run TDE2 100 times for every value of α. Then, we compute the average number of HITs (pairwise comparisons of two item pairs) that were needed to find $\mathcal{P}^{\alpha}$. Then, we allow TDE1 to the same number of HITs to compute the embedding. This way we can compare the methods given the same amount of work. We report the amount of work as the number of HITs per data point in $\mathcal{D}^{\text{train}}$.

We measure the quality of the resulting estimates using mean absolute error, defined as

$$\text{MAE} = \frac{\sum_{x \in \mathcal{D}^{\text{test}}} |\operatorname{estimate}(x) - p(x)|}{|\mathcal{D}^{\text{test}}| \max_x \{p(x)\}},$$

where $p(x)$ is the true density at x. The estimate and p are both normalised so that $\sum_{x \in \mathcal{D}^{\text{test}}} \operatorname{estimate}(x) = \sum_{x \in \mathcal{D}^{\text{test}}} p(x) = 1$. We also normalise the error with $\max_x \{p(x)\}$ so that it can be understood a fraction of the largest density value in $\mathcal{D}^{\text{test}}$.

Results

Results are shown in Figure 4. We show how the error behaves as a function of the number of HITs required per data item to build the estimator. (Here the largest amount of work corresponds to using $\alpha = 0.05$ with TDE2.) In this plot we show TDE1 in red, and TDE2 in black. We observe that both methods perform better when they are allowed to use more work. Increasing the amount of work starts to have diminishing effects after some point. TDE1 performs slightly better than TDE2 when only a very small amount of work is used. However, as the number of HITs is increased, TDE2 tends to perform better than TDE1. This is mainly because TDE2 makes no assumptions about the dimensionality of the input unlike TDE1. Notice that with 4Gaussians-2D the methods perform almost equally well. Of course in practice the dimensionality of the input data is unknown, and making an educated guess about this can be difficult.

Conclusions

In this work we presented methods, TDE1 and TDE2, for density estimation using judgements of relative similarity. TDE1 is an implementation of a standard kernel density estimator with relative distances. TDE2 is a novel method that has some practical advantages over TDE1. In particular, our experiments suggest that TDE2 is more robust when the true dimensionality of the input is unknown.

There are multiple avenues for further research. Both TDE1 and TDE2 currently use $|\mathcal{D}|$ HITs to estimate density at a single point x. It seems likely that this could be improved by optimising the pairs against which x is compared. Another interesting question is whether we could learn estimators like TDE1 without embeddings? That is, can we find pairs of approximately the same length directly with e.g. the kind of pairwise comparisons as used for TDE2. One possible solution to this are partial orders with ties (Ukkonen et al. 2009). Also the theoretical results presented in this paper could be improved. In particular, proving bounds on the estimation error as a function of α would be interesting.

References

Amsterdamer, Y.; Grossman, Y.; Milo, T.; and Senellart, P. 2013. Crowd mining. In *SIGMOD Conference*, 241–252.

Bragg, J.; Mausam; and Weld, D. S. 2013. Crowdsourcing multi-label classification for taxonomy creation. In *HCOMP 2013*.

Chilton, L. B.; Little, G.; Edge, D.; Weld, D. S.; and Landay,

J. A. 2013. Cascade: crowdsourcing taxonomy creation. In *CHI 2013*, 1999–2008.

Davis, J. V.; Kulis, B.; Jain, P.; Sra, S.; and Dhillon, I. S. 2007. Information-theoretic metric learning. In *ICML 2007*.

Dawid, A. P., and Skene, A. M. 1979. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics* 20–28.

Floyd, R. W., and Rivest, R. L. 1975. Algorithm 489: the algorithm selectfor finding the i:th smallest of n elements. *Communications of the ACM* 18(3):173.

Gomes, R. G.; Welinder, P.; Krause, A.; and Perona, P. 2011. Crowdclustering. In *Advances in Neural Information Processing Systems*, 558–566.

Hastie, T.; Tibshirani, R.; and Friedman, J. 2009. *The elements of statistical learning*, volume 2. Springer.

Heikinheimo, H., and Ukkonen, A. 2013. The crowd-median algorithm. In *First AAAI Conference on Human Computation and Crowdsourcing*.

Hoare, C. A. R. 1961. Algorithm 65: Find. *Communications of the ACM* 4(7):321–322.

Lintott, C. J.; Schawinski, K.; Slosar, A.; Land, K.; Bamford, S.; Thomas, D.; Raddick, M. J.; Nichol, R. C.; Szalay, A.; Andreescu, D.; et al. 2008. Galaxy zoo: morphologies derived from visual inspection of galaxies from the sloan digital sky survey. *Monthly Notices of the Royal Astronomical Society* 389(3):1179–1189.

Liu, E. Y.; Guo, Z.; Zhang, X.; Jojic, V.; and Wang, W. 2012. Metric learning from relative comparisons by minimizing squared residual. In *ICDM 2012*.

Parameswaran, A. G.; Sarma, A. D.; Garcia-Molina, H.; Polyzotis, N.; and Widom, J. 2011. Human-assisted graph search: it's okay to ask questions. *PVLDB* 4(5):267–278.

Parzen, E. 1962. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics* 33(3):1065–1076.

Raykar, V. C., and Yu, S. 2012. Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *Journal of Machine Learning Research* 13:491–518.

Rosenblatt, M. 1956. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics* 27(3):832–837.

Schultz, M., and Joachims, T. 2003. Learning a distance metric from relative comparisons. In *NIPS 2003*.

Tamuz, O.; Liu, C.; Belongie, S.; Shamir, O.; and Kalai, A. 2011. Adaptively learning the crowd kernel. In *ICML 2011*, 673–680.

Trushkowsky, B.; Kraska, T.; Franklin, M. J.; and Sarkar, P. 2013. Crowdsourced enumeration queries. In *ICDE 2013*, 673–684.

Ukkonen, A.; Puolamäki, K.; Gionis, A.; and Mannila, H. 2009. A randomized approximation algorithm for computing bucket orders. *Inf. Process. Lett.* 109(7):356–359.

van der Maaten, L., and Weinberger, K. 2012. Stochastic triplet embedding. In *Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on*, 1–6. IEEE.

Wilber, M. J.; Kwak, I. S.; and Belongie, S. J. 2014. Cost-effective hits for relative similarity comparisons. In *Second AAAI Conference on Human Computation and Crowdsourcing*.

Appendix

Proof of Proposition 2

Proof. We first upper bound $q(x, \delta_{\max})$ by letting

$$\begin{aligned} q(x,\delta_{\max}) &\leq q'(x,\delta_{\max}) \\ &= 2\int_{\delta=0}^{\delta_{\max}} \int_{j=x-\frac{3}{2}\delta}^{x+\frac{3}{2}\delta} p(u^*)p(u^*+\delta^*)\,dj\,di, \end{aligned}$$

where u^* and δ^* are the solution to $\mathrm{argmax}_{u,\delta}\, p(u)p(u+\delta)$ st. $u \in [x-2\delta_{\max}, x+\delta_{\max}]$ and $\delta \leq \delta_{\max}$. Since u^* and δ^* only depend on $\delta_{\max}$ and not on δ or j, $p(u^*)p(u^*+\delta^*)$ can be taken out from the integrals, and we have

$$q'(x,\delta_{\max}) = 2p(u^*)p(u^*+\delta^*) \int_{\delta=0}^{\delta_{\max}} \int_{j=x-\frac{3}{2}\delta}^{x+\frac{3}{2}\delta} 1\,dj\,di.$$

A straightforward calculation reveals that the integrals simplify to $\frac{3}{2}{\delta_{\max}}^2$, and we obtain

$$q'(x,\delta_{\max}) = 3{\delta_{\max}}^2 p(u^*)p(u^*+\delta^*).$$

Now, as $\delta_{\max} \to 0$, the $x \pm 2\delta_{\max}$ interval becomes tighter and tighter around x. It follows that $u^* \to x$ and $\delta^* \to 0$, and thus $p(u^*)p(u^*+\delta^*) \to p(x)^2$. By combining these observations we find that

$$\begin{aligned} \lim_{\delta_{\max}\to 0} \frac{q'(x,\delta_{\max})}{3{\delta_{\max}}^2} &= \lim_{\delta_{\max}\to 0} \frac{3{\delta_{\max}}^2 p(u^*)p(u^*+\delta^*)}{3{\delta_{\max}}^2} \\ &= \lim_{\delta_{\max}\to 0} p(u^*)p(u^*+\delta^*) = p(x)^2. \end{aligned}$$

An upper bound of $q(x, \delta_{\max})$ thus approaches $p(x)^2$ from above as $\delta_{\max} \to 0$. We can make a similar reasoning in terms of the pair $(u_*, u_* + \delta_*)$ that has the *lowest* probability in the interval $x \pm 2\delta_{\max}$. This will show that a lower bound to $q(x, \delta_{\max})$ approaches $p(x)^2$ from below. By combining these we obtain that $q(x, \delta_{\max})/3{\delta_{\max}}^2$ must approach $p(x)^2$ as $\delta_{\max} \to 0$. □

Proof of Proposition 3

Proof. We present the proof for 2-dimensional data, but the argument generalises to k-dimensions in a fairly straightforward manner. Fix some point x with density $p(x)$, and consider a random pair (u, v) drawn from p given that $d(u, v)$ is fixed to some $\delta \leq \delta_{\max}$. We sample the pair by first drawing u from p, and then draw v *given* u so that $d(u, v) = \delta$. The point x is covered by the pair (u, v) when one of the following events happens:

$$\begin{aligned} \mathcal{U}_\delta: &\quad d(x,u) \leq \delta \\ \mathcal{V}_\delta: &\quad \delta < d(x,u) < 2\delta \text{ and } d(x,v) \leq \delta. \end{aligned}$$

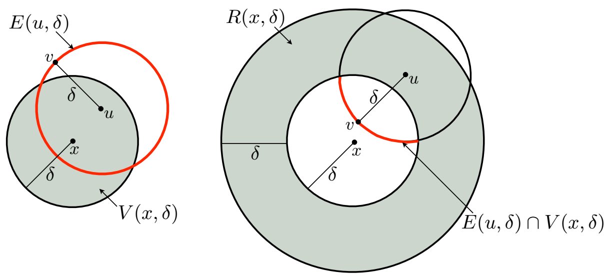

Figure 5: For the proof of Proposition 3. Left: Example of deriving $\Pr[\mathcal{U}]$. Right: Example of deriving $\Pr[\mathcal{V}]$.

Now we can write $q(x, \delta_{\max})$ in terms of $\mathcal{U}_\delta$ and $\mathcal{V}_\delta$ by integrating over all values of δ up to $\delta_{\max}$:

$$q(x, \delta_{\max}) = \int_0^{\delta_{\max}} (\Pr[\mathcal{U}_\delta] + \Pr[\mathcal{V}_\delta])\, d\delta.$$

The proof relies on a similar argument as the proof of Proposition 2. We show that $q(x, \delta_{\max})$ can be both upper and lower bounded so that these upper and lower bounds both go to $C(\delta_{\max})p(x)^2$ as $\delta_{\max} \to 0$ for some $C(\delta_{\max})$.

We proceed to derive upper bounds for $\Pr[\mathcal{U}_\delta]$ and $\Pr[\mathcal{V}_\delta]$. Let $V(y, \delta)$ denote the area of a disc centered at point y with radius δ. Also, let $E(y, \delta)$ denote the 1-dimensional circular subspace centered at point y with radius δ. That is, the set $E(y, \delta)$ is the "edge" of the set $V(y, \delta)$. Next, let $p^{\delta_{\max}}(x)$ denote the maximum density inside the set $V(x, 2\delta_{\max})$, that is,

$$p^{\delta_{\max}}(x) = \max_{y \in V(x, 2\delta_{\max})} p(y). \qquad (15)$$

Observe that the probability of event $\mathcal{U}_\delta$ is the total density inside $V(x, \delta)$ (so that the point u is close enough to x) times the probability that v is drawn from the circle around u with radius δ, i.e., the set $E(u, \delta)$ (so that we have $d(u, v) = \delta$ as required). That is,

$$\Pr[\mathcal{U}_\delta] = \int_{V(x,\delta)} p(u) \left(\int_{E(u,\delta)} p(v)\, dv \right) du.$$

Also see left side of Figure 5. Since $\delta \le \delta_{\max}$ and $d(x, u) \le \delta$, we have $d(x, v) \le 2\delta$, and we can thus upper bound $\Pr[\mathcal{U}_\delta]$ by replacing both $p(u)$ and $p(v)$ with $p^{\delta_{\max}}(x)$. This yields

$$\Pr[\mathcal{U}_\delta] \le p^{\delta_{\max}}(x)^2 \int_{V(x,\delta)} \left(\int_{E(u,\delta)} 1\, dv \right) du.$$

The value of the integrals no longer depends on the density p, only on the length δ of the pair (u, v). We can thus replace those with a function of δ and obtain

$$\Pr[\mathcal{U}_\delta] \le p^{\delta_{\max}}(x)^2 C_1(\delta).$$

Next, consider the event $\mathcal{V}_\delta$. It happens when u falls in the "ring" around x, defined as the set $R(x, \delta) = V(x, 2\delta) \setminus V(x, \delta)$, and v falls on the segment of $E(u, \delta)$ that intersects with $V(x, \delta)$. We have thus

$$\Pr[\mathcal{V}_\delta] = \int_{R(x,\delta)} p(u) \left(\int_{E(u,\delta) \cap V(x,\delta)} p(v)\, dv \right) du.$$

Also see the right side of Figure 5. Like above with $\mathcal{U}_\delta$, we upper bound $\Pr[\mathcal{V}_\delta]$ by replacing $p(u)$ and $p(v)$ both with $p^{\delta_{\max}}(x)$. This is allowed as $d(x, u) \le 2\delta$ and $d(x, v) \le \delta$. We obtain

$$\Pr[\mathcal{V}_\delta] \le p^{\delta_{\max}}(x)^2 \int_{R(x,\delta)} \left(\int_{E(u,\delta) \cap V(x,\delta)} 1\, dv \right) du.$$

As above with $\mathcal{U}_\delta$, the value of the integrals only depends on δ and not p. We get the upper bound

$$\Pr[\mathcal{V}_\delta] \le p^{\delta_{\max}}(x)^2 C_2(\delta).$$

Next we combine these results to obtain a simple upper bound for $q(x, \delta_{\max})$:

$$\begin{aligned}
q(x, \delta_{\max}) &= \int_0^{\delta_{\max}} (\Pr[\mathcal{U}_\delta] + \Pr[\mathcal{V}_\delta])\, d\delta \\
&< \int_0^{\delta_{\max}} \left(p^{\delta_{\max}}(x)^2 C_1(\delta) + p^{\delta_{\max}}(x)^2 C_2(\delta) \right) d\delta \\
&= \int_0^{\delta_{\max}} p^{\delta_{\max}}(x)^2 \left(C_1(\delta) + C_2(\delta) \right) d\delta \\
&= p^{\delta_{\max}}(x)^2 \int_0^{\delta_{\max}} \left(C_1(\delta) + C_2(\delta) \right) d\delta \\
&= p^{\delta_{\max}}(x)^2 C(\delta_{\max}).
\end{aligned}$$

The $C(\delta_{\max})$ that appears above is the one we use in the claim (Eq. 14) of the proposition. Now, we can replace $q(x, \delta_{\max})$ with this upper bound in Equation 14, and observe that

$$\lim_{\delta_{\max} \to 0} \frac{p^{\delta_{\max}}(x)^2 C(\delta_{\max})}{C(\delta_{\max})} = \lim_{\delta_{\max} \to 0} p^{\delta_{\max}}(x)^2 = p(x)^2.$$

Here the second equality simply follows from the definition of $p^{\delta_{\max}}(x)$ as given in Equation 15 above. As $\delta_{\max}$ goes to zero, the area from which we take $p^{\delta_{\max}}(x)$ shrinks until only x is left. The upper bound of $q(x, \delta_{\max})/C(\delta_{\max})$ thus approaches $p(x)^2$ from above as $\delta_{\max} \to 0$. We can do the same reasoning in terms of a lower bound of $q(x, \delta_{\max})$, by replacing max with a min in Equation 15, i.e.,

$$p^{\delta_{\max}}(x) = \min_{y \in V(x, 2\delta_{\max})} p(y).$$

This lower bound will approach $p(x)^2$ from below as $\delta_{\max} \to 0$ in the same manner.

Finally, notice that the argument used in this proof, while described for $k = 2$, generalises to arbitrary k. We only redefine the set $V(y, \delta)$ as the volume of a k-dimensional sphere with radius δ, and $E(y, \delta)$ as the $k - 1$ dimensional surface of $V(y, \delta)$. It follows that we can always derive a $C(\delta_{\max})$ that is independent of $p(x)$ which is enough to complete the proof. $\square$