Chapter 2

Computing Diverse Density

In this chapter we will explore different ways of computing Diverse Density. There will be two types of variations in how we compute it. One variation will involve different ways of combining evidence from instances in a bag. The other variation derives from our use of different concept classes. A concept class defines the set of possible concepts that can be learned by the algorithm. Diverse Density is a measure, of a member of the concept class, of the amount of intersection between the positive bags at that concept. It also measures the amount of negative evidence (instances from negatively labeled bags) at that concept.

2.1 Notation

The training data is presented as positive and negative bags. They are written as

$$B_1^+, \cdots, B_n^+$$

for positive bags, and as

$$B_1^-, \cdots, B_m^-$$

for negative bags. When the label on the bag does not matter, it will simply be referred to as B_i. A bag may contain any number of instances. For example, if the

the i^{th} positive bag contains p instances then they are written as

$$B_{i1}^+, \cdots, B_{ip}^+$$

and likewise for a negative bag. Each instance represents a point in a k-dimensional feature space. The individual feature values of the j^{th} instance of the i^{th} positive bag are

$$B_{ij1}^+, \cdots, B_{ijk}^+.$$

2.2 Diverse Density as probability

Given some concept class C, made of concepts $\{c_t\}$, we define a random variable T, whose domain is the set of concepts. $\Pr(T = t)$ is the probability that the t^{th} concept is correct. We will use $\Pr(t)$ as shorthand for that probability. We would like to compute the probability that a concept, c_t, is the target concept given the data. We call this probability the *Diverse Density* of the t^{th} concept:

$$DD(t) = \Pr(t \mid B_1^+, \cdots, B_n^+, B_1^-, \cdots, B_m^-) \tag{2.1}$$

We would like to maximize this probability with respect to t, or find the target concept c_t that is most likely to agree with the data. In Chapter 3, we will examine methods for maximizing Diverse Density. In this chapter, we will examine how to compute Diverse Density for a particular concept c_t.

Using Bayes' rule, we turn Formula 2.1 into

$$DD(t) = \frac{\Pr(B_1^+, \cdots, B_n^+, B_1^-, \cdots, B_m^- \mid t)\Pr(t)}{\Pr(B_1^+, \cdots, B_n^+, B_1^-, \cdots, B_m^-)}. \tag{2.2}$$

The quantity $\Pr(t)$ represents the prior knowledge of what concept is preferred. In this thesis, we will not use any prior knowledge, and therefore $\Pr(t)$ is constant. The quantity $\Pr(B_1^+, \cdots, B_n^+, B_1^-, \cdots, B_m^-)$ is also constant with respect to t. Since we will be maximizing with respect to t, this quantity can be thought of as a normalizing term and need not be calculated explicitly. We are therefore left to compute the likelihood:

$$\Pr(B_1^+, \cdots, B_n^+, B_1^-, \cdots, B_m^- \mid t) \tag{2.3}$$

We can further simplify Formula 2.3 by making the key assumption that all bags are conditionally independent given the true target concept. The validity of this assumption is discussed in Section 2.7. This allows us to decompose Formula 2.3 into

$$\prod_{1 \leq i \leq n} \Pr(B_i^+ \mid t) \prod_{1 \leq i \leq m} \Pr(B_i^- \mid t) \tag{2.4}$$

2.2.1 Exact generative models

If we had a generative model, describing how each bag was generated from the target concept, then we could plug it into Formula 2.4 and calculate Diverse Density exactly. We will now examine two such generative models, one for a discrete domain and another for a continuous domain. We do not have such a generative model for the applications described in Chapters 4, 5 and 6. However, the models examined here will hint at possible general approximations discussed in Section 2.3.

A discrete generative model example

Let us assume that all instances are taken from the discrete domain $\{1, \ldots, D\}$. The set of possible concepts is identical, where the meaning of concept c_2 is that any instance with value 2 should be labeled positive, and instances with values other than 2 should be labeled negative.

We will describe one possible generative model in this situation. The concept is picked uniformly at random from the D possible concepts. Each bag contains n instances. Positive bags have one instance equal to the concept and the rest are independently picked uniformly at random from D possible values. Negative bags have all instances independently chosen from the non-concept domain values. We can now calculate the following quantities:

$$
\begin{aligned}
\Pr(t) &= 1/D \\
\Pr(B_i^+ \mid t) &= \sum_{1\le j\le n} \frac{1}{n}\delta[B_{ij}^+ = t]\left(\frac{1}{D}\right)^{n-1} \\
\Pr(B_i^+) &= \left(\frac{1}{D}\right)^{n} \\
\Pr(t \mid B_i^+) &= \frac{1}{n}\sum_{1\le j\le n} \delta[B_{ij}^+ = t]
\end{aligned}
$$

If this was indeed the way in which bags were generated, then we can plug the above equations into Formula 2.4, and that in turn into Formula 2.2 for a complete definition of Diverse Density. Note that we can also compute the value of $\Pr(t \mid B_i^+)$, which is non-zero only when t is equal to at least one of the instances in the bag.

A continuous generative model example

Let us assume that each instance is described using a single real number. Once again, there are as many concepts as elements in the domain. We select the true concept to be c_t with probability $G_L(t)$, where $G_L(\cdot)$ is a zero-mean, L-variance Gaussian, and $L \gg 1$ is a large number (in other words, a roughly uniform distribution).

One possible generative model is as follows. Each bag contains n instances. A positive bag contains one instance that was generated from a Gaussian distribution with mean equal to the concept and a variance of 1. The rest of the instances were generated independently from zero-mean, L-variance Gaussian. We can now calculate the following quantities:

$$
\begin{aligned}
\Pr(t) &= G_L(t) \\
\Pr(B_i^+ \mid t) &= \sum_{1\le j\le n} \left(\frac{1}{n} G_1(B_{ij}^+ - t) \prod_{\substack{1\le l\le n \\ l\ne j}} G_L(B_{il}^+) \right)
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{\prod_{1\le l\le n} G_L(B_{il}^+)}{\prod_{1\le l\le n} G_L(B_{il}^+)} \sum_{1\le j\le n} \left(\frac{1}{n} G_1(B_{ij}^+ - t) \prod_{\substack{1\le l\le n \\ l\ne j}} G_L(B_{il}^+) \right) \\
&= \prod_{1\le j\le n} G_L(B_{ij}^+) \sum_{1\le j\le n} \left(\frac{1}{n} G_1(B_{ij}^+ - t) \prod_{\substack{1\le l\le n \\ l\ne j}} G_L(B_{il}^+) \frac{1}{\prod_{1\le l\le n} G_L(B_{il}^+)} \right) \\
&= \frac{\prod_{1\le j\le n} G_L(B_{ij}^+)}{nL\sqrt{2\pi}} \sum_{1\le j\le n} \left(G_1(B_{ij}^+ - t) \exp\left(\frac{(B_{ij}^+)^2}{2L^2} \right) \right) \\
\Pr(B_i^+) &= \prod_{1\le j\le n} G_L(B_{ij}^+) \\
\Pr(t \mid B_i^+) &= \frac{G_L(t)}{nL\sqrt{2\pi}} \sum_{1\le j\le n} \left(G_1(B_{ij}^+ - t) \exp\left(\frac{(B_{ij}^+)^2}{2L^2} \right) \right)
\end{aligned}
$$

Once again, we can plug the generative model's calculation of $\Pr(B_i^+ \mid t)$ to get a complete definition of Diverse Density. As seen, we can also calculate $\Pr(t \mid B_i^+)$, which turns out to be proportional to a mixture of Gaussians. The Gaussians are centered at the instances and the mixture coefficients are $\exp\left(\frac{(B_{ij}^+)^2}{2L^2}\right)$. Note that when L is large, these coefficients are all approximately one.

2.2.2 Defining Diverse Density in terms of $\Pr(t \mid B_i)$

Because we will not have a generative model of $\Pr(B_i \mid t)$, we will instead attempt to use a general estimator for $\Pr(t \mid B_i)$. We describe several such estimators in Section 2.3. For now, we derive Diverse Density in terms of $\Pr(t \mid B_i)$. Using Bayes' rule again on each of the terms in Formula 2.4, we get

$$
\prod_{1\le i\le n} \frac{\Pr(t \mid B_i^+)\Pr(B_i^+)}{\Pr(t)} \prod_{1\le i\le m} \frac{\Pr(t \mid B_i^-)\Pr(B_i^-)}{\Pr(t)}
$$

Inserting this back into Formula 2.2, we get the following definition for Diverse Density:

$$
\begin{aligned}
DD(t) &= \left[\prod_{1\le i\le n} \Pr(t \mid B_i^+) \prod_{1\le i\le m} \Pr(t \mid B_i^-)\right] \times \\
&\left[\frac{\prod_{1\le i\le n} \Pr(B_i^+) \prod_{1\le i\le m} \Pr(B_i^-)}{\Pr(B_1^+,\cdots,B_n^+,B_1^-,\cdots,B_m^-)}\right] \times \left[\frac{1}{Pr(t)^{n+m-1}}\right] \qquad (2.5)
\end{aligned}
$$

Once again, the last term in Formula 2.5 is a constant if we assume a uniform prior. The middle term is also constant with respect to t, and disappears if we assume that the bags are generated independently. Neither the middle nor last terms in Formula 2.5 need to be computed if the operation we want to perform is a comparison between different hypotheses. This is the only operation (also known as Maximum Likelihood) needed when we optimize Diverse Density in Chapter 3. The last term in Formula 2.5 might seem odd because it indicates that concepts with a low prior become more probable with additional examples. However, there is a canceling prior term in each of the $\Pr(t \mid B_i)$ terms, as can be explicitly seen in the continuous generative model example.

The first line of Formula 2.5 agrees with the intuition established in Chapter 1 because the multiplication of the terms represents the notion that we want to find a concept that intersects the first positive bag *and* the second positive bag, etc., *and* is not in the first negative bag, *and* is not in the second negative bag, etc. This is a general definition of Diverse Density, but we have yet to specify how to compute $\Pr(t \mid B_i)$. In the following sections, we give several ways of computing this term, each of which gives a different instantiation of Diverse Density. The intuitive idea behind $\Pr(t \mid B_i^+)$ is that we would like it to be high if the i^{th} positive bag is close to c_t, and low otherwise. Similarly, we would like $\Pr(t \mid B_i^-)$ to be low if c_t is "far" from the i^{th} negative bag, and high otherwise. The two exact generative models described earlier indicate that $\Pr(t \mid B_i^+)$ should be estimated by a combination of the distances from each instance to the potential concept c_t.

2.3 Various ways of estimating $\Pr(t \mid B_i)$

2.3.1 Using noisy-or to estimate a density

One way of estimating the likelihood of a hypothesized concept given a single bag is to use *noisy-or* [Pearl, 1988]. Noisy-or is an idea from Bayesian Networks, where it is necessary to calculate the probability of an event with multiple causations. In the noisy-or model, it is assumed that the event can only happen if at least one of the causations occurred. It is also assumed that the probability of any cause failing to trigger the event is independent of any other cause. Therefore, if we denote the event as E and the causes as $C_1, \cdots, C_p$, then

$$\Pr(E \mid C_1, \cdots, C_p) = 1 - \prod_{1 \le j \le p} (1 - \Pr(E \mid C_j)),$$

where $\Pr(E \mid C_j)$ is the causal probability of the j^{th} cause triggering event E.

In our case, the event E is "c_t is the underlying concept". This event can happen if at least one of the instances in the bag are in c_t, so the causes are represented by instances in the bag. Specifically,

$$\begin{aligned} \Pr(t \mid B_i^+) &= \Pr(t \mid B_{i1}^+, \cdots, B_{ip}^+) \\ &= \frac{1 - \prod_{1 \le j \le p} (1 - \Pr(B_{ij}^+ \in c_t))}{z} \end{aligned} \tag{2.6}$$

Where z is a normalizing constant that depends on the concept class, and can be computed as $\int_t \Pr(t \mid B_i)$. If even one of the instances in the bag is likely to be in the concept c_t, then $\Pr(t \mid B_i^+)$ will be high. If none of the instances are likely to be in c_t, then $\Pr_c(t \mid B_i^+)$ will be close to zero. Computing $\Pr(B_{ij}^+ \in c_t)$ depends on the concept class, and will be discussed in Section 2.4. Negative bags are handled in a similar manner, with Equation 2.6 being replaced with

$$\Pr(t \mid B_i^-) = \frac{\prod_{1 \le j \le p} (1 - \Pr(B_{ij}^- \in c_t))}{z}.$$

2.3.2 Other density estimators

All-or-nothing density estimator

If we assume that there is no noise involved in any of the instances, then one interpretation of $\Pr(t \mid B_i)$ is

$$\Pr(t \mid B_i^+) = \begin{cases} 1/z & \text{if } \exists j \text{ such that } B_{ij}^+ \in c_t \\ 0 & \text{otherwise} \end{cases}$$
$$\Pr(t \mid B_i^-) = \begin{cases} 0 & \text{if } \exists j \text{ such that } B_{ij}^- \in c_t \\ 1/z & \text{otherwise} \end{cases} \tag{2.7}$$

This "all-or-nothing" approach means that the distribution of $\Pr(t \mid B_i^+)$ is zero everywhere except for spikes where c_t overlaps positive instances. Likewise, $\Pr(t \mid B_i^-)$ is zero only where c_t overlaps the negative instances. When this approach is used with Formula 2.5, Diverse Density is computing a strict intersection between the positive bags minus a union of the negative bags. If the data is not noisy at all and precisely agrees with our assumptions, then this strict approach should work. However, in most real world problems there are noisy measurements of the instances, irrelevant attributes, and perhaps mislabeling of the bags. In that case, strict intersection would return the empty set, but a softer approach such as described in the next few sections would be able to handle some amount of noise.

Most-likely-cause estimator

An approach similar to noisy-or, but one which requires fewer independence assumptions, is the most-likely-cause method. $\Pr(t \mid B_i)$ is estimated by only looking at the

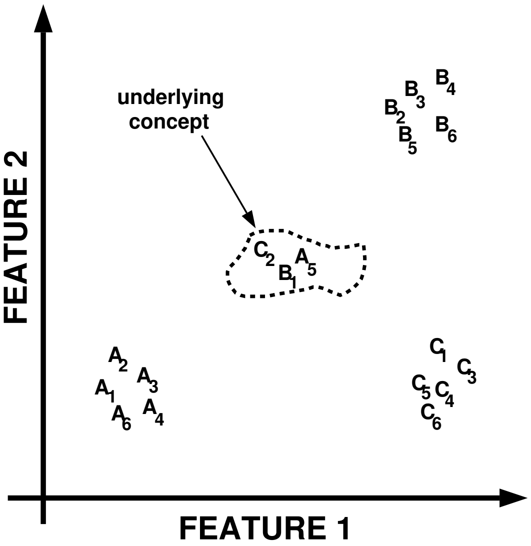


Figure 2-1: There are three bags (A, B, C), each with six instances $(A_1, \cdots, A_6, B_1, \cdots, B_6, C_1, \cdots, C_6)$. Every bag is positive because it has at least one instance within the true concept. A strict intersection of the bags would be empty. An intersection of the estimated densities of the bag would work only if A_5, B_1, and C_2 are not ignored as outliers.

instance in the bag which is most likely to be in the concept c_t. Specifically,

$$
\begin{aligned}
\Pr(t \mid B_i^+) &= \max_j \{\Pr(B_{ij}^+ \in c_t)\}/z \\
\Pr(t \mid B_i^-) &= (1 - \max_j \{\Pr(B_{ij}^- \in c_t)\})/z
\end{aligned}
\tag{2.8}
$$

Using domain knowledge in the estimator

So far, we have tried to make minimal assumptions about how the bag was generated. If there is any domain-specific knowledge though, other density estimators may be better and more computationally efficient than the ones described here. However, without that knowledge, some density estimators are likely to be very bad choices for computing Diverse Density. For example, let us suppose that we tried to estimate the density of each of the bags in Figure 2-1 using a single Gaussian for each bag. For bag A, the Gaussian would be centered in the cluster of A's instances, and instance A_5 — the very instance that makes the bag positive — would be ignored as an outlier. The same process would recur for the other bags, and Diverse Density would fail as an indicator of intersection. The lesson to be learned from this example is that, without problem-specific knowledge, one cannot afford to treat *any* instance as an outlier.

One example of using domain knowledge to create a better evidence combination function occurs in [Keeler *et al.*, 1991a] and [Dayan and Zemel, 1995]. Unlike the noisy-or model, which assumes that there is at least one cause, they assume that there is exactly one cause. They use a sum of likelihood-ratios estimate of of event E given a set of causes $C_1, \cdots, C_p$:

$$\Pr(E \mid C_1, \cdots, C_p) = \sum_{1 \le j \le p} \frac{\Pr(E \mid C_j)}{\Pr(\overline{E} \mid C_j)}$$

In Multiple-Instance learning, this translates to the assumption that there is exactly one instance which is a "true positive" in every positive bag. Section 2.5 compares some of the density estimators discussed here.

In the next section, we describe the final piece needed to compute Diverse Density: the probability that an instance B_{ij} is in the concept c_t. This probability will depend on the concept class from which the desired concept is to be learned.

2.4 Computing $\Pr(B_{ij} \in c_t)$ for various concept classes

Previous algorithms for Multiple-Instance learning, [Dietterich *et al.*, 1997, Auer, 1997, Long and Tan, 1996], tried to learn from a hypothesis class of axis-parallel hyper-rectangles. This led to concepts that were clearly defined: either the instance was in the rectangle or it was not. Here we will attempt to learn concepts that lead to a probabilistic measure of whether an instance is in the concept or not. We will assume that instances are corrupted by some noise. However, the farther the instance is from the concept, the less likely it is to be a corrupted part of the concept.

We will describe three concept classes of increasing complexity. All three were used in the applications of Chapters 4, 5 and 6. All three are also relatively simple concepts, and some of the difficulties of more complicated concepts are discussed in Chapter 8. Finally, it is straightforward to measure *distance* between an instance and the concept in all three cases. The probability of an instance being in the concept will depend on that distance.

2.4.1 Single point concept class

The simplest concept class is one where every concept corresponds to a a single point in feature space. This class assumes that given an underlying true concept P, every positive bag has at least one instance that is equal to P corrupted by some Gaussian noise. Every negative has no instances that are equal to P corrupted by some Gaussian noise.

The concept c_t is therefore described as a k-dimensional vector $c_{t_1}, \cdots, c_{t_k}$, where k is the number of dimensions in feature space. We calculate the probability that an instance B_{ij} is in the concept c_t with a Gaussian-like distribution:

$$\Pr(B_{ij} \in c_t) = \exp(-\sum_{1 \leq l \leq k} (B_{ijl} - c_{t_l})^2) \tag{2.9}$$

The closer instance B_{ij} is to the t^{th} concept (which in this case is location t in feature space), the more likely it is to be in that concept.

2.4.2 Single point-and-scaling concept class

Whenever a distance metric is imposed over a feature space (as in Equation 2.9), the question arises whether the scaling of the dimensions is correct. In most real world learning problems, the data has many irrelevant or redundant attributes, and even among the relevant attributes some are more important than others. We would like to learn the best weighting of the attributes so that irrelevant ones are ignored (their weight goes to zero) and important ones are made to stand out (their weight goes up).

We expand our concept class to include feature scaling in the following manner. We denote each member of the new concept class as c_{ts}. Each concept is described as two k-dimensional vectors, where k is the dimensionality of feature space. The first vector, denoted $\{c_{t_1}, \cdots, c_{t_k}\}$, is a point in feature space. The second vector, denoted $\{c_{s_1}, \cdots, c_{s_k}\}$, is a weighting of each feature. This concept class assumes that given an underlying true concept P, every positive bag has at least one instance that is equal to P corrupted by some Gaussian noise with P's distance metric.

We calculate the probability that an instance B_{ij} is in the concept c_{ts} as follows:

$$\Pr(B_{ij} \in c_{ts}) = \exp(-\sum_{1 \leq l \leq k} (c_{s_l}(B_{ijl} - c_{t_l}))^2) \tag{2.10}$$

The closer (using distance metric c_s) instance B_{ij} is to the location c_t in feature space, the more likely it is to be in that concept. We can think of the two concept classes described so far as being parameterized. The single point concept class was parameterized by a position vector, and the single point-and-scaling concept class is parameterized by both position and scaling vectors. In Chapter 3 we will show that we find the best parameters (i.e. the best concept) in the same way for both concept classes — by maximizing Diverse Density.

This definition of the point-and-scaling concept class only allows independent scaling of the features. In other words, it only allows stretching of the distance metric along the feature axis. It is possible to scale with a full covariance matrix, thereby allowing a stretch along arbitrary axis. However, the number of parameters needed to specify c_s becomes k^2, which results in a large computational expense when searching for the best concept.

2.4.3 Disjunctive point-and-scaling concept class

So far we have assumed that the target concept is a single area in feature space. One way of allowing more complicated concept classes is to allow a disjunction of d single-point concepts. In other words, a bag is positive if at least one of its instances is in concept c_{ts}^1 or in concept c_{ts}^2 or ... or in concept c_{ts}^d. A bag is negative if none of its instances are in any of the d concepts. Figure 2-2 shows an example of some positive bags that agree with a 2-disjunct concept.

It is still fairly straightforward to compute the probability that an instance B_{ij} is in the concept $c_{ts}^1 \vee c_{ts}^2 \vee \cdots \vee c_{ts}^d$. It can be measured as the maximum probability over being in any one of the disjuncts. Namely,

$$\Pr(B_{ij} \in c_{ts}^1 \vee c_{ts}^2 \vee \cdots \vee c_{ts}^d) = \max_{1 \leq a \leq d} \{\Pr(B_{ij} \in c_{ts}^a)\} \tag{2.11}$$

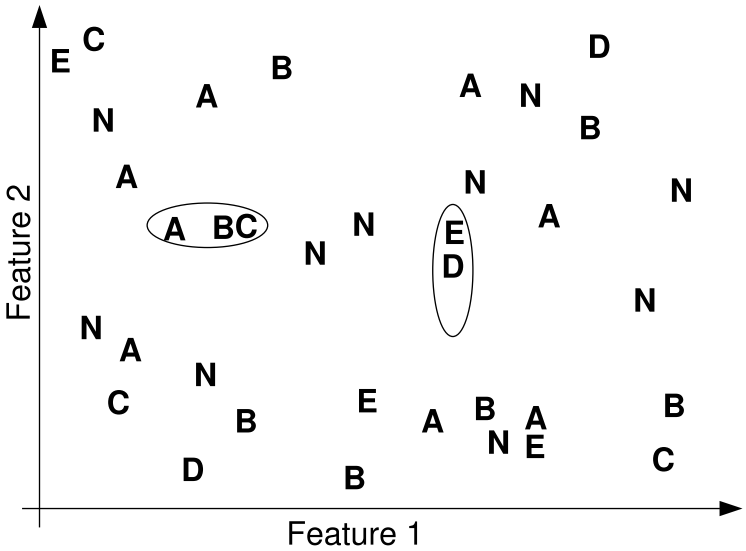


Figure 2-2: Example of a disjunctive concept. There are 5 positive bags, called A,B,C,D,E. An instance is labeled by its bags' name. Negative instances are marked as N. The 2-disjunct concept shown has different feature weightings in each disjunct. A new bag will be labeled positive if one of its instances falls near either disjunct.

Where $\Pr(B_{ij} \in c^a_{st})$ is calculated as in Formula 2.10. The disjunctive concept class is parameterized by $2d$ k-dimensional vectors. As we will see in Chapter 3, this increase in complexity leads to a more computationally expensive algorithm, but it can also lead to better concepts.

2.5 Comparing density estimators

Figures 2-3 (a), (b) and (c) show a comparison between the all-or-nothing, noisy-or, and most-likely-cause approaches to estimating $\Pr(t \mid B^+_i)$. The positive bag contains five instances, each one described by a single continuous feature. They are marked as points along the x-axis. We use the concept class consisting of all locations in feature space (i.e. the single point concept class). The graph shows, for every concept c_t ($t \in [0, 10]$), how likely that concept is.

The all-or-nothing estimator only allows concepts that are equal to one of the instances in the bag. Both the noisy-or and the most-likely-cause estimators behave qualitatively like a kernel density estimator [Duda and Hart, 1973]. Figure 2-4 ex-

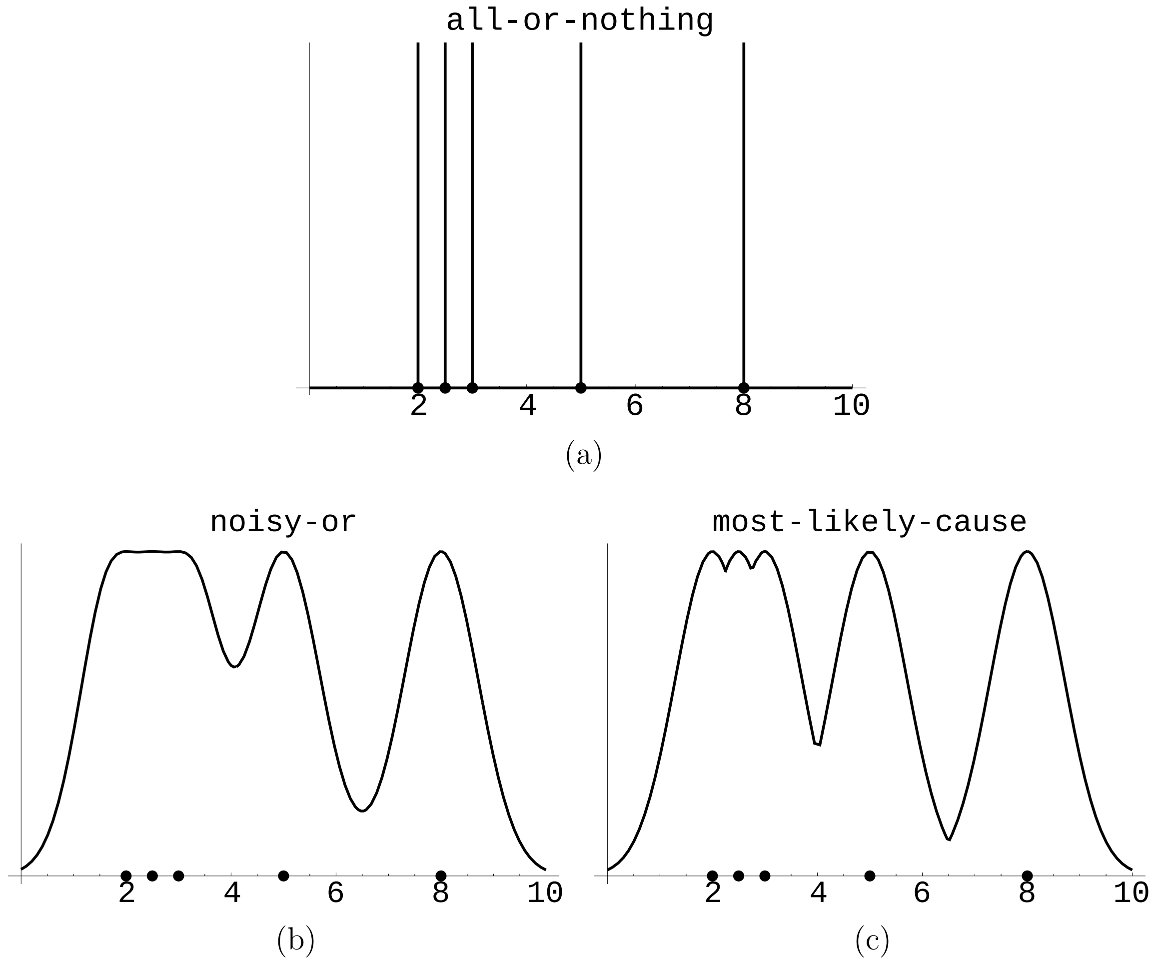


Figure 2-3: Comparison between (a) all-or-nothing, (b) noisy-or, and (c) most-likely-cause approaches to approximating $\Pr(t \mid B_i^+)$. In our example, the positive bag has instances at $2.0, 2.5, 3.0, 5.0$, and 8.0 (in a one-dimensional feature space). They are marked as points along the x-axis. We use a single point concept class, so each graph shows the value of $\Pr(t \mid B_i^+)$ for every concept c_t, $t \in [0, 10]$.

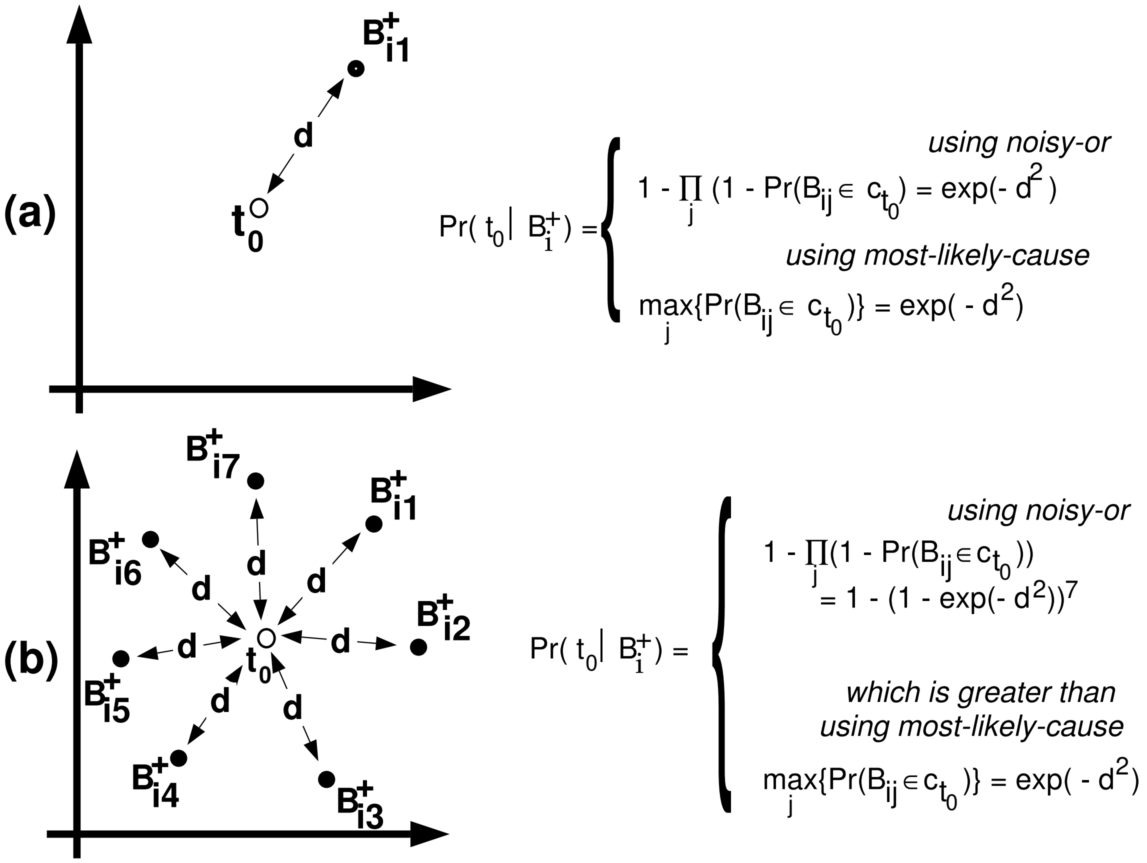


Figure 2-4: Two different situations are examined: (a) in which a positive bag contains a single instance and we try to estimate $\Pr(t \mid B_i^+)$ at the concept t_0 which is distance d away from B_i^+'s single instance. (b) in which a positive bag contains many instances, all of which fall along a circle with radius d and center t_0. We once again try to estimate $\Pr(t \mid B_i^+)$ at t_0. Noisy-or and most-likely-cause are equivalent when there is a single minimally close instance. However, the noisy-or estimate increases as minimally equidistant instances are added, whereas the most-likely-cause estimate remains constant.

amines more carefully the difference between the noisy-or and the most-likely-cause approaches. In Figure 2-4(a), a positive bag B_i^+ contains only one instance. If we use the single point concept class, both noisy-or and most-likely-cause compute the same value for $\Pr(t_0 \mid B_i^+)$, the likelihood of concept c_{t_0}. However, if that bag contains 7 instances which are all equidistant from t_0, then the likelihood of concept c_{t_0} increases with the noisy-or estimator. It remains constant with the most-likely-cause estimator.

2.6 Examples

In this section we present two example data sets. Both are learned using the single point concept class with a noisy-or estimator, and both have a one or two dimensional feature space. This allows us to plot the Diverse Density surface. This plot shows, for every concept c_t, the value of $\prod_{1\leq i\leq n} \Pr(t \mid B_i^+) \prod_{1\leq i\leq m} \Pr(t \mid B_i^-)$, which is proportional to $DD(t)$. The peak of this plot corresponds to the most likely concept. We can also plot each term in the above product. In the applications of Chapters 4, 5 and 6 the complexity of the concept class and the feature space will prohibit such a plot.

2.6.1 Example: A simple data set

We constructed three positive bags (B_1^+, B_2^+, B_3^+) and one negative bag (B_1^-). The feature space is made of one dimension. The instances in each bag are shown as dots along the x-axis in Figure 2-5. The concept class is single point, which means that the concept could be at any one feature location. We use noisy-or to compute $\Pr(t \mid B_i)$. Each plot in Figure 2-5 shows the probability of a concept being at location t (where t ranges from 0 to 10), given some evidence.

We can see how the Diverse Density computation progresses as more evidence is combined. The top line in Figure 2-5 shows the Diverse Density surface after combining evidence from the first two positive bags. Good candidate location for the most likely concept appear to be at $t = 3$ and $t = 8$. The second line shows the

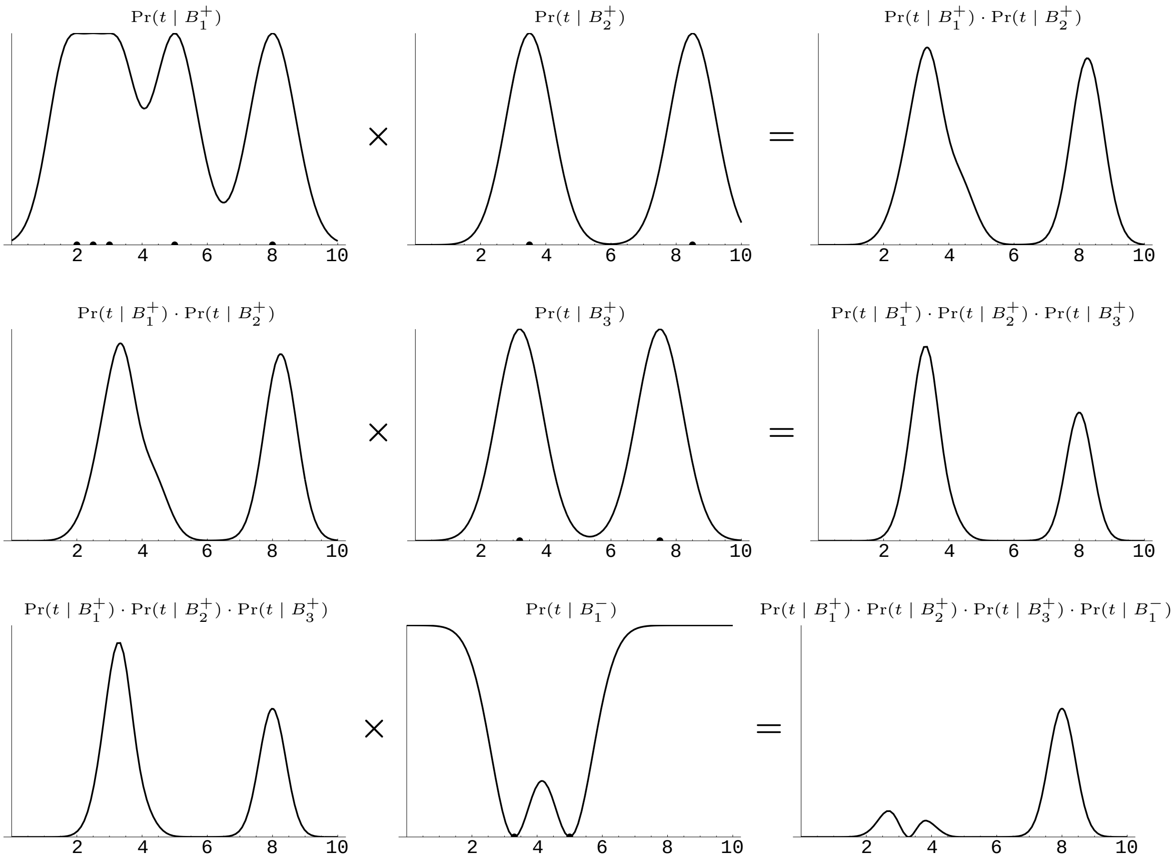


Figure 2-5: Computing $DD(t)$ for every concept c_t ($t \in [0, 10]$) on a simple data set.

surface after combining all three positive bags. The concept at $t = 3$ appears most probable. The third line combines that with evidence from the negative bag. Because there are negative instances close to $t = 3$, the $t = 8$ concept becomes the winner.

2.6.2 Example: A difficult artificial data set

To test more rigorously how Diverse Density works with a noisy-or estimator and the single point concept class, we generated the following difficult data set. 10 bags were generated, each with 50 instances. Every instance was generated uniformly at random from the space of $[0, 100] \times [0, 100]$. The underlying concept was a 5×5 square in the middle of the 100×100 feature space. A bag was labeled positive if at least one of its instances fell within the square, and negative otherwise. The set of bags is shown in Figure 2-6. The instances from the negative bags are all shown as dots (since it does not matter which instance came from which negative bag). Instances from the i^{th} positive bag are each labeled as i. This is a particularly difficult data set because every bag (both positive and negative) comes from the same underlying distribution.

Figure 2-7(b) shows the Diverse Density surface over the single point concept class $\{c_t : t \in [0, 100] \times [0, 100]\}$. Diverse Density at every point is calculated using the noisy-or density estimator. The peak at the middle of feature space falls in the true concept, so an algorithm that returns the concept with maximum Diverse Density as its learned hypothesis (as `maxDD` does in Chapter 3) would perform well on this example.

It is instructive to look at two approaches that would fail on this training set. If we had used an all-or-nothing estimator instead of noisy-or, Diverse Density would be zero everywhere. That is because the strict intersection of the positive bags is empty. If we had tried to solve this problem as if it were a traditional supervised learning problem, then every instance would receive the label of the bag to which it belongs. We can see the density surface that this would generate by *adding* the contributions of every bag (instead of multiplying them, as in Diverse Density), with negative bags contributing negative density. The Kernel Regression [Atkeson *et al.*, 1997]

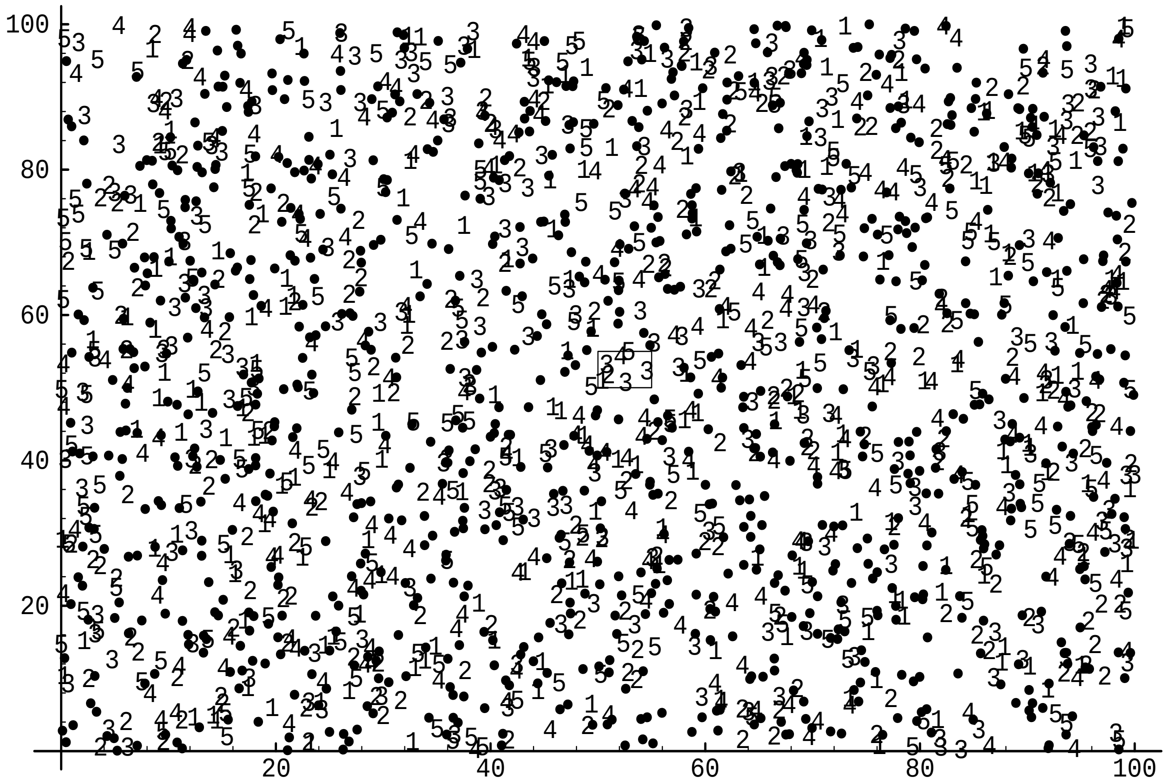
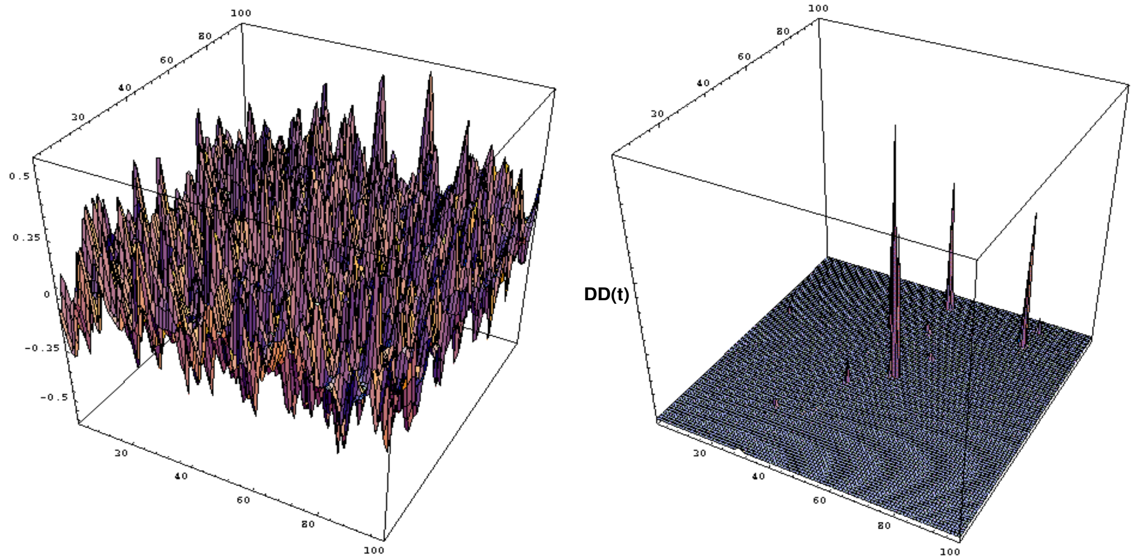


Figure 2-6: Negative and positive bags with all instances being drawn from the same distribution. The bags are labeled according to their intersection with the middle square. Negative instances are dots, positive instances are numbered according to which bag they are from. The square (underlying concept) contains at least one instance from every positive bag and no negatives.

algorithm[1] can be thought of as performing such a density estimation; it classifies a point in feature space as positive if the total density at that point is greater than zero. Figure 2-7(a) shows this density surface. Not only are there many areas that are mistakenly labeled positive (have density value greater than zero), but the true concept does not stand out. In fact, no supervised learning algorithm will perform well when roughly 49 out of every 50 positive examples are mislabeled.

The surface in Figure 2-7(b) is close to zero almost everywhere because of the exponential nature of Diverse Density. A point that is close to the intersection of n bags will have exponentially higher Diverse Density than a point that is close to

[1]Given a set of input-output training pairs $\{x_i, y_i\}$, kernel regression predicts the output at a new point, p, as a weighted average of all the outputs $\{y_i\}$, where the i^{th} weight is proportional to the zero-mean, σ^2-variance Gaussian $G(\parallel p - x_i \parallel)$. The kernel width, σ, indicates how much smoothing is performed. Radial kernels other than Gaussians can also be used.



(a) Surface using regular density (b) Surface using Diverse Density

Figure 2-7: Density surfaces over the feature space of the difficult artificial example.

the intersection of $n - q$ bags (exponential in q). The smaller peaks in Figure 2-7(b) are the result of chance concentrations of positive instances from different bags and chance sparsity of negative instances. It is possible that a different random collection of instances will have chance concentrations with even higher Diverse Density. However, as the number of training bags increases, the chance of having an accidental concentration with Diverse Density higher than at the true concept becomes lower and lower. Figure 2-8 demonstrates that as the number of positive or negative bags is increased, the chance that the maximum Diverse Density point is within the true concept converges to 1. For every $3 \leq n \leq 20$ positive bags and $0 \leq m \leq 20$ negative bags, 100 runs were produced. During each run, 50 instances were generated uniformly at random for each bag, and the run was called successful if the maximum Diverse Density point fell within the underlying concept. The small number of examples needed to learn consistently well on this task is compared against the MULTINST algorithm in Chapter 7.

2.7 Assumptions and their validity

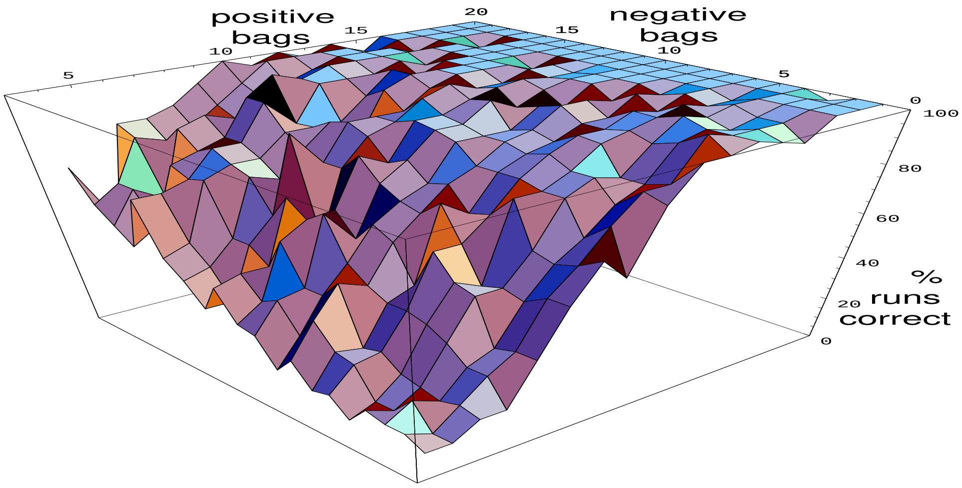


Figure 2-8: Success of Diverse Density vs. number of training bags. Percentage of success was computed from 100 runs, where the bags were generated randomly in each run.