1 Probability Assessment

Each of N photographs is assessed by an expert (this includes assessments made by students which are subsequently reviewed by another person knowledgeable in the subject). During these assessments, experts will assign some number, K_i, of labels to photograph number i. The implication of this action is that any photograph showing similar earthquake damage should receive the same set of labels.

The purpose of this note is to define a way to:

a. assign a metric to each photograph that represents (in some fashion) the probability that a set of labels is correctly assigned, and

b. to incorporate into the metric some notion of the quality of the expert's assessment.

1.1 Basic Assignment. Rather than having the metric produce a single probability to each assessment, a probability density function, $f(x)$, is assigned to each (image,label)= (Φ, L) pair at hand. That is, a random variable, V is defined by setting

$$V = V_{(\Phi,L)} = P[\text{photo } \Phi \text{ possesses label } L].$$

The distribution associated with $V_{(\Phi,L)}$ will have a density function, $f_{(\Phi,L)}(x)$ defined in such a way as to provide a reasonable solution to the question *how likely is it that* V (the likelihood of a (image,label) pairing) *lies between two given values:* v_1 *and* v_2. This probability may also be expressed by writing $P[v_1 \leq V \leq v_2]$ and the function $f_{(\Phi,L)}(x)$ supplies this value of this probability through the computation:

$$P[v_1 \leq V \leq v_2] = \int_{v_1}^{v_2} f_{(\Phi,L)}(x)\, dx.$$

1.2 Density Function Shape. Everything defined in the previous section clearly depends upon the expert who is making the evaluation. All of the probabilities mentioned are therefore conditional probabilities, conditioned on the expert at hand. However, whenever there is only a single expert involved, the notation used to denote a conditional probability or expectation will be dropped for clarity.

The density functions, $f_{(\Phi,L)}(x) = f_{(\Phi,L)|E}(x|E))$, employed here are going to be triangular functions (as shown below in Figure 1). The width and position of the triangle's base expresses the range of certainty of the (Φ, L) pairing. Each (image,label) pair will be evaluated by one or more experts. The experts are required, when making the (Φ, L) association, to assign a subjective estimate to the random variable V defined above. Let $V^*_{(\Phi,L)}$ (or just V^* when the (image,label) pair involved is clear) denote the expert's subjective estimate of V.

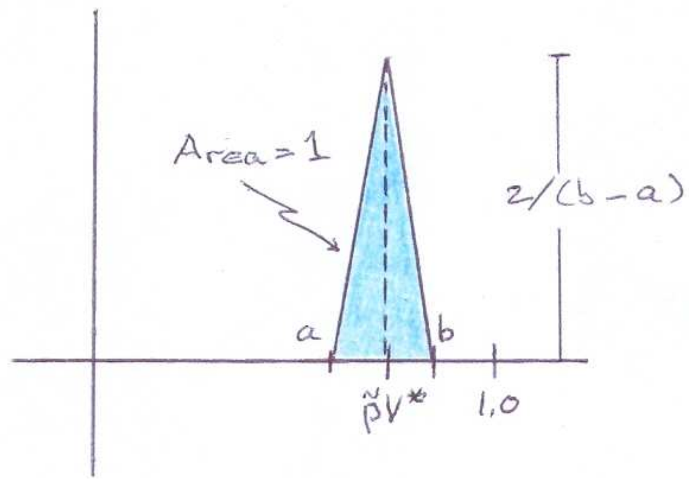


Figure 1

As stated above, the density function is given a triangular shape with the triangle's peak centered on the weighted value $\widetilde{p}V^*$, where V^* is the expert's self-assessment and $\widetilde{p}$ is a measure of the expert's quality (explained below):

$$f(x) = \begin{cases} \text{triangle over interval } [a,b] & \text{with } 0 < a < \widetilde{p}\,V^* < b < 1 \\ 0 & \text{otherwise.} \end{cases}$$

The width of the base of the triangle should be a measure of the uncertainty in the expert's judgment. Call this product $(\widetilde{p}\,V^*)$ the weighted self-assessment of the (Φ, L) pair.

1.3 Weighted Self-Assessment: V^* Component .

As part of the assessment process, each reviewer is required to supply a self-assessment of their certitude. According to [1], a self-assessment using qualifiers such as

```
Definitely        Almost          Probably      Maybe
                 Certainly

                          Unknown

Maybe not       Probably not       Almost       Definitely not
                                Certainly not
```

provides a familiar mechanism for self-evaluation. In general, the behind-the-scenes algorithms will assign numerical values to each qualitative judgment. In a *non-probabilistic* schema, values might be assigned such as:

$$\begin{gathered} \texttt{D:}0.9\pm \quad \texttt{AC:}0.7\pm \quad \texttt{P:}0.5\pm \quad \texttt{M:}0.3\pm \\ \texttt{U:}0 \pm 0.2 \\ \neg\texttt{M:}-0.3\pm \quad \neg\texttt{P:}-0.5\pm \quad \neg\texttt{AC:}-0.7\pm \quad \neg\texttt{D:}-0.9\pm \end{gathered}$$

where $x\pm$ is a shorthand for the interval $[x-0.1, x+0.1]$ with 0 ± 0.2 denoting the interval $[-0.2, 0.2]$. There are two things to note in the earthquake assessments at hand:

a) It is desired to make the schema probabilistic. A better choice, therefore, is to use values in the range $[0, 1]$ to represent the values assigned to the qualitative descriptors.

b) Experts are not evaluating each photograph against the list of all labels but only the labels they feel are relevant. Because of this, all of the negative qualifiers may be eliminated along with the qualifier `Unknown`.

With these two caveats a possible choice for the descriptor values might be

$$\texttt{D}:0.95\pm \quad \texttt{AC}:0.725\pm \quad \texttt{P}:0.275\pm \quad \texttt{M}:0.05\pm$$

In this case the $x\pm$ indicates the interval of $x \pm 0.1125$ intersected with the interval $[0, 1]$. The value 0.1125 creates a uniform spacing between the principle values $0.95, 0.725, 0.275, 0.05$.

If finer gradations are deemed necessary, the expert might be asked to express their certitude on a scale of 1 to 10 (for example). Behind the scenes these designations might represent the intervals of certitude such as[1] : $(0, 0.1), (0.1, 0.2), \ldots, (0.9, 1.0)$.

For purposes here, the only value of interest are going to be the center points, C_i, of the intervals: in the first case (four equally spaced gradations of certainty) the values are $C_1 = 0.95, C_2 = 0.725, C_3 = 0.275, C_4 = 0.05$; in the second case (10 equally spaced gradations) the values could be $C_1 = 0.5, C_2 = 0.15, C_3 = 0.25, \ldots, C_{10} = 0.95$. When the expert makes their selection of certainty for a given (Φ, L) pair, assign $V^* = C_s$ where C_s is the center point of the interval the expert selected.

1.4 Weighted Self Assessment: $\widetilde{p}$ Component .

The quality of an expert may be measured as follows. The expert, E, is presumed to generate a large number (more than 30 if possible) of (image,label) pairs. On one extreme, the expert may evaluate only one photo but associate many labels with it. On the other extreme, the expert may evaluate many photos put associate only a single label with each photo. Most likely, the expert will evaluate several photos and associate several labels with each. All of these possibilities are permitted.

In conjunction with the expert's photographic evaluations described above, there are going to be field reports compiled from first-hand inspections of the damage site(s). From these reports, a second set of (image,label) associations could be assembled for each of the photographs in the collection. Call this collection the truth standard.

For each photograph Φ evaluated by expert E, there will be a set of (image,label) pairs, (Φ, L), produced by E. Call this standard $\mathcal{E}(\Phi)$. The truth collection should also contain (image,label) pairs for Φ. Call this collection $\mathcal{T}(\Phi)$. Three numbers are produced by comparing the collections $\mathcal{E}(\Phi)$ and $\mathcal{T}(\Phi)$:

$$\begin{aligned} e_\Phi &= \text{the total size of } \mathcal{E}(\Phi) \\ t_\Phi &= \text{the total size of } \mathcal{T}(\Phi) \\ c_\Phi &= \text{the total size of } \mathcal{E}(\Phi) \cap \mathcal{T}(\Phi) \end{aligned}$$

In words, e_Φ is the total number of labels assigned to photograph Φ by expert E; t_Φ is the total number of labels assigned to photograph Φ by the field inspection reports; and c_Φ is the total number

[1] Note this difficulty: with 10 gradations, can an individual evaluator distinguish between a 7 and an 8?

of common (Φ, L) pairs shared between the expert's assessment of Φ and the field reports. Here is an example: if the expert determines photo Φ_1 should be associated with labels A, B, and C and the field reports show that photo Φ_1 contains damage captured by labels A, B, D and G, then $e_{\Phi_1} = 3$, $t_{\Phi_1} = 4$, and $c_{\Phi_1} = 2$.

To create a success value for expert E for photo Φ compute the ratio of successful identifications (c_Φ) to total identifications. Note that the total identifications must include three components: successes, omissions, and false inclusions. For photo Φ this ratio is:

$$\frac{c_\Phi}{e_\Phi + t_\Phi - c_\Phi}.$$

Note that in the denominator, e_Φ counts the common pairs and the false inclusions, t_Φ counts the common pairs and the omissions, and the subtraction of c_Φ eliminates the duplication in the count of the common pairs.

For the complete collection of photos evaluated by expert E, the ratio is

$$\widetilde{p} = \frac{\sum_\Phi c_\Phi}{\sum_\Phi (e_\Phi + t_\Phi - c_\Phi)},$$

where the sums are taken over all the photos evaluated by the expert. This ratio, denoted $\widetilde{p}$, measures the fraction of correct evaluations done by expert E.

Remark 1. It isn't necessary to actually create the truth standard since all that is needed are the values of c_Φ, e_Φ, and t_Φ for each photograph evaluated by E. This could be a lot of work for someone knowledgeable in reading field reports unless the field teams are required to create (Φ, L) associations as part of their reports.

Remark 2. For each photograph, a trial (in the sense of a Bernoulli trial) is seen to be either a successful pairing of photograph with a label or a mistaken pairing (either by inclusion or omission). In the example above, the number of trials associated with that photograph is $3 + 4 - 2 = 5$. For the entire set of photographs evaluated by E, the value $\widetilde{p}$ is used to rate the quality of the expert.

Notation. In what follows, the letters S and M will be used to denote the numerator and denominator of the quantity $\widetilde{p}$:

$$\begin{aligned} S &= \sum_\Phi c_\Phi \\ M &= \sum_\Phi (e_\Phi + t_\Phi - c_\Phi), \end{aligned}$$

where $\displaystyle\sum_\Phi$ denotes the sum over all photographs evaluated by expert E.

1.5 Triangle Base Width. The proportion of successes, $\widetilde{p} = S/M$, will be used as an estimate of the probability, p, that the expert E assigns the correct label(s) to an image. The items counted by M represent a sequence of Bernoulli trials, M in number, possessing a single trial probability of success given by p. In general, for any sequence of Bernoulli trials, the ratio of the number of successes (#`Succ`) to the total number of trials (#`Trls`) is known to be a good estimator for the parameter p: namely $\widehat{p} = \#\texttt{Succ}/\#\texttt{Trls}$. The estimator $\widehat{p}$ is a random variable used to produce a numerical estimate, $\widetilde{p}$, for p once a set of observations have been made. The estimator $\widehat{p}$ is a good estimator for p because if sufficient data is collected, the ensuing estimate $\widetilde{p}$ will be numerically close to the actual parameter p.

To determine how close $\widetilde{p}$ is to p a confidence interval, $[L, R]$, will be constructed based on the estimator $\widehat{p}$. The confidence interval to be used is that for a proportion: well-known and given by the interval

$$\begin{aligned}
[L, R] &= [\ \widetilde{p} - z_{\alpha/2}\sqrt{\widetilde{p}(1-\widetilde{p})/M}\ ,\ \widetilde{p} + z_{\alpha/2}\sqrt{\widetilde{p}(1-\widetilde{p})/M}\ ] \\
&= [\ \widetilde{p}\big(1 - z_{\alpha/2}\sqrt{\frac{1-\widetilde{p}}{\widetilde{p}M}}\big)\ ,\ \widetilde{p}\big(1 + z_{\alpha/2}\sqrt{\frac{1-\widetilde{p}}{\widetilde{p}M}}\big)\ ]
\end{aligned}$$

where $\widetilde{p}$ and M are as stated above (M is assumed to be large, i.e. $M \geq 30$) and $(1-\alpha)$ is the desired confidence. This result is particularly meaningful when the interval $[L, R]$ is small. Typical choices for α are 0.1, 0.05, or 0.01 yielding 90%, 95% and 99% confidence intervals respectively. The expression $z_{\alpha/2}$ is easily determined from the standard normal distribution function — in the three cases cited here $z_{\alpha/2}$ is $1.644853627, 1.959963985, 2.575829304$ for $\alpha = 0.1$, 0.05, and 0.01 respectively.

Once the interval $[L, R]$ has been determined, (shown in figure 1) the endpoints of the base of the triangle are set to $a = V^*L$ and $b = V^*R$ provided that $0 < V^*L < V^*R < 1$. If $V^*L < 0$ (resp. $1 < V^*R$) then set $a = 0$ (resp $b = 1$). In summary:

$$\begin{aligned}
a &= \max(\ 0, V^*\widetilde{p}\big(1 - z_{\alpha/2}\sqrt{\frac{1-\widetilde{p}}{\widetilde{p}M}}\big)\ ) \\
b &= \min(\ 1, V^*\widetilde{p}(1 + z_{\alpha/2}\sqrt{\frac{1-\widetilde{p}}{\widetilde{p}M}})\ ).
\end{aligned}$$

The max and min functions guarantee that the base of the triangle will fall within the interval $[0, 1]$. Also note that with respect to the weighted self-assessment the interval appears to have the shape $[V^*\widetilde{p}(1-\epsilon)\ ,\ V^*\widetilde{p}(1+\epsilon)]$, where the quantity $\epsilon > 0$ and is

(1) small when $\widetilde{p}$ is near 1 – the expert is good, and

(2) large when $\widetilde{p}$ is near 0 – the expert is not good.

1.6 Triangle Height : Uncertainty. To make this triangular function into a density function, the area under this triangle must equal one. Therefore, the height of the triangle must be $2/(b-a)$. Outside of this triangle the density function is zero. Most of the time, this triangle will be isosceles unless $a = 0$ or $b = 1$ (truncation occurs as indicated above by the use of the $\max(\,,)$ and $\min(\,,)$ functions). Since the area under the triangle has been forced equal to 1, a probability density function results. Note that tacit in this analysis is the assumption that the random variables V and p are continuous variables (that is, they produce values over a continuum of numbers).

1.7 Triangle Height : Certainty . This is the case where knowledge of an implication or value is certain. The corresponding certainty functions are degenerate triangles (that is, triangles with a base width of zero). In this case the density function will have this shape

$$f(x) = \delta(x - \widetilde{p}V^*),$$

where $\delta()$ is the Dirac delta (an impulse). This case is treated in more detail below.

2 Belief Network Approach

2.1 Rationale. Three expert system types were considered for use in this analysis of methods to be used in a system of automated earthquake damage assessments.

Certainty Factors. The approach used by the FIESTA satellite troubleshooting expert system (built by Stanford Telecommunications, Inc. in the 1980s) is the starting point for this analysis. This expert system used the MYCIN model with certainty factors. David Hackerman [2] points out the difficulties of attaching certainty factors to evaluations made by expert systems. He suggests the use of belief networks. These permit the computation of a probability density function associated with a conclusion made by an expert system.

According to Hackerman, certainty factors may become artificially large or small depending on the number and type of inferences in the total expert system. Too many statements such as $e_i \Rightarrow h$ with differing evidence e_i but the same concluding hypothesis h can skew the associated certainty factor low while other constructs may cause the certainty factor of the hypothesis to skew high.

Fuzzy Sets/Fuzzy Logic. The use of fuzzy sets and fuzzy logic requires that something like a probability density function be associated with observations (developed by Lotfi A. Zadeh [4]). Fuzzy sets provide ways for developing logical relationships among inexact propositions (”this person is tall” for example) and give algorithms for dealing with the propositions associated with such a system. For example, if $A =$ fuzzy set of tall people, $B =$ fuzzy set of thin people, the rules of fuzzy logic gives mechanisms for evaluating the results of logical statements such as: $A \vee B, A \wedge B, A \Rightarrow B$ and so forth.

A fuzzy set A will have an inclusion function, $i_A(x)$, associated with it that assigns to each x in some universe a real value that describes the likelihood that $x \in A$. The function $i_A(x)$ is not necessarily a probability density function. In fact in some systems, $i_A(x)$ may produce values between -1 and 1 where values near -1 indicate that x is not likely in A and values near 1 indicates that x is likely in A.

If the fuzzy set B possesses a similar inclusion function $i_B(x)$ then the rules of fuzzy logic describe how to compute an inclusion function, $i_C(x)$, for the fuzzy set C where C is some logical combination of A and B (such as $C = (A \vee B)$ or $C = (A \Rightarrow B)$).

Belief Networks. A belief network (described by J. Pearl [3]) gets around a lot of the difficulties inherent with certainty factors and fuzzy sets. In the envisioned earthquake analysis system, this seems like a best attack on the problem. This approach is similar to the fuzzy sets/fuzzy logic method but uses probability density functions to value objects, sets, relationships, implications, and so forth. One big difficulty with this method is that it relies on knowledge of a global joint probability density function of the objects involved. This is often nearly impossible to compute. However, for specific networks of relationships, many work arounds exist that free the system from having to have more than general (abstract) knowledge of the complete joint distribution.

The framework described so far (i.e. the definition of the (image,label) density function) may also serve as a basis for either an analysis using belief networks or an analysis via the theory fuzzy Sets.

2.2 Basics. A belief network may be envisioned as a directed graph with the nodes representing various objects, events, states, etc. and the directed edges corresponding to implications or inferences of various sorts.

For this problem the following network will serve as the starting point.

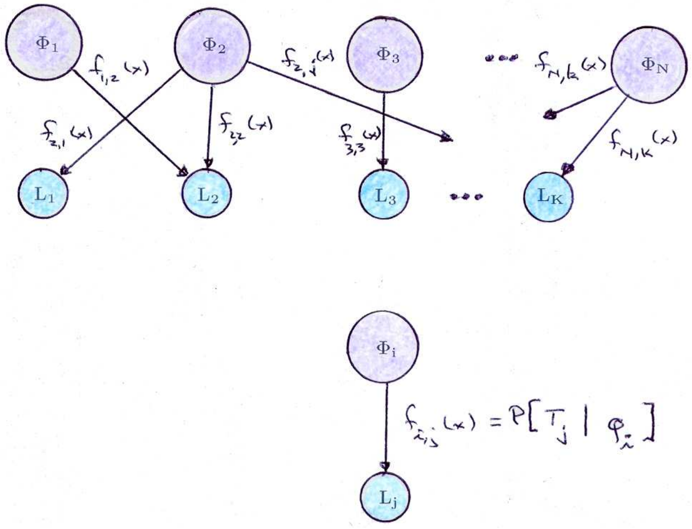


Figure 2

Call this network, the photo-base. Each of the circles in the top row contains a designation, Φ_i, representing the $i-$th photograph. The circles in the second row represent the labels, L_j, that are available for assignment to each photograph by an expert. The connecting, directed line segment (also called an edge) represents the probabilistic assessment made by an expert that the label belongs to the photograph. The lower portion of Figure 1 illustrates a typical pairing of a photograph with a label, showing the associated inference

$$f_{i,j}(x) = P[T_j \,|\, \Phi_i, r; x],$$

where T_j is the event *label j is assigned*, Φ_i is a given photograph, r represents any other (possibly empty) relevant information about the assignment[2], and x is the certainty inherent in the expert's evaluations ($0 \leq x \leq 1$). The expression $f_{i,j}(x)$ is actually a probability density function dependent on the certainty, x, of the expert evaluations and is identical to the description given in section 1.2. Note also that the absence of a directed edge between a photograph and a label is equivalent to the assertion $f_{i,j}(x) = 0$ for all x.

The directed edges between nodes (photographs or labels) also indicate dependencies found in the situation. Lack of a directed edge indicates the existence of a conditional independence between the entities represented by the nodes of the graph. The task at hand will be to assign or compute probabilities (sometimes a single value, sometimes a probability density function) for each of the

[2] It is recognized, from this point forward, that all of this analysis might depend on some unspecified external information, r. The inclusion of r as part of any conditioning events will be dropped in the sequel unless specifically needed

vertices in the diagram. The set of initial probabilities assigned to the photographs in the photo-base will be denoted by ϕ_i for $i = 1, 2, \ldots, N$ and represent any a priori assignment of probabilities to the photographs. The only restriction on these probabilities is $\sum_i \phi_i \leq 1$. The quantity $1 - \sum_i \phi_i$ represents the probability that the set of initial photographs is incomplete (that is, doesn't capture the complete story of earthquake damage).

If it is believed that all of the photographs in the photo-base are of equal weight relative to some earthquake, setting $\phi_i = 1/N$ for all i and reading off the probabilities associated with the labels (the individual damage) may give a sense of the relative importance of each label (as represented by the photo-base).

If, on the other hand, it is desired to know the damage represented by a specific photograph in the photo-base (say, photo Φ_i), assigning weights $\phi_j = 0$ when $j \neq i$ and $\phi_j = 1$ when $j = i$ will reveal this.

2.3 Single Photograph.

To observe the contribution of a single photograph to the probability attached to a label, the joint density function of the two events *i-th photograph selected* (Φ_i) and *label j is assigned* (T_j) needs to be determined. It is known that for two events A and B that $P[A, B] = P[A \,|\, B]P[B]$ where the symbol A, B represents the conjunction of the two events (the event representing the occurrence of both events A and B). Therefore

$$P[T_j, \Phi_i; x] = P[T_j \,|\, \Phi_i; x]P[\Phi_i; x].$$

This probability, $P[T_j \,|\, \Phi_i; x]$, represents the contribution of photograph Φ_i (and possibly other information r) to the probability that the event T_j occurs (as assessed by an expert). But notice that this is the same as stating

$$P[T_j, \Phi_i; x] = f_{i,j}(x)P[\Phi_i; x] = f_{i,j}(x)\phi_i(x).$$

The probability $\phi_i(x)$ represents the relative importance of photograph Φ_i in whatever analysis is being conducted.

2.4 Label Probabilities.

From the basic relationship just expressed the label probabilities may be expressed by summing the joint density function $P[T_j, \Phi_i; x]$ over the index i (the complete set of photographs) yielding

$$P[T_j; x] = \sum_i P[T_j, \Phi_i; x] = \sum_i f_{i,j}(x)\phi_i(x) \qquad (*).$$

Note that in this case, the result $\ell_j(x) = P[T_j; x]$ is the probability density function associated with the label L_j expressed in terms of the certainty value of x. Also note that many of the summands in equation $(*)$ are zero since $f_{i,j}(x) = 0$ when no directed edge is shown.

2.5 An Example. Consider the following inference network.

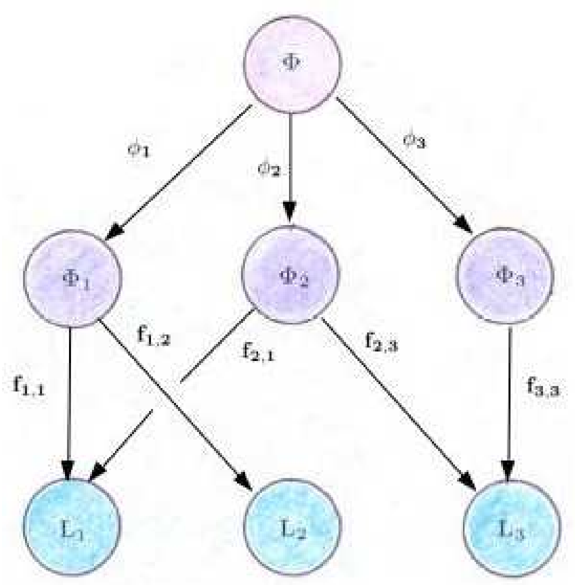


Figure 3

There are five inference probabilities in this network that link the photographs to the labels are denoted by $f_{1,1}, f_{1,2}, f_{2,1}, f_{2,3}, f_{3,3}$. The dependency on a variable x, the certainty of each implication, has been dropped for clarity but each is dependent on a range of probabilities (certainties) represented by the variable $x, 0 \leq x \leq 1$.

The context for this example is as follows: a new photograph, Ψ, is introduced. Comparisons between Ψ and the three photographs, Φ_i, in the photo-base are made. The probability densities $\phi_i(x)$ will express the relative certainty that Ψ is similar to Φ_i — the overriding assumption being that similar images reflect similar degrees of damage. That is

$$\widehat{\phi}_i(x) = P\big[\substack{\text{aspects of } \Psi \\ \text{are exhiited by } \Phi_i} \,\big|\, \Psi\big].$$

The relative certainty for Φ_i is given by

$$\phi_i(x) = \frac{\widehat{\phi}_i(x)}{\kappa(x)},$$

where $\kappa(x) = \sum_i \widehat{\phi}_i(x)$. For example, if similarity values assigned to the images Φ_1, Φ_2, Φ_3 are 70%, 20%, and 1% respectively (with a certainty of 1) then the relative similarities assigned to each image are

$$\phi_1 = \frac{70}{91} \qquad \phi_2 = \frac{20}{91} \qquad \phi_3 = \frac{1}{91}.$$

(with a certainty of 1). More likely, the ϕ_i will be certitude functions, $\phi(x)$, that measure a level of belief that the values 70%, 20%, and 1% represent some aspect of similarity between Ψ and Φ_i. In the

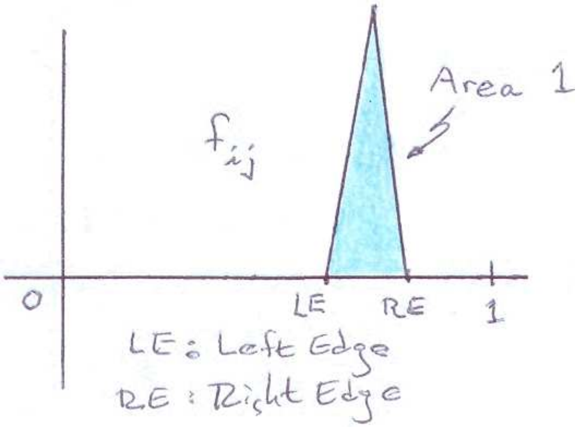


Figure 4

case above, $\phi_i(x)$ may be expressed using a Dirac delta: $\phi_1(x) = \delta(x - 70/91)$, $\phi_2(x) = \delta(x - 20/91)$, $\phi_3(x) = \delta(x - 1/91)$.

Continuing with this example, if it is believed that there is a 70% mean commonality ($\mu_1 = 0.70$) between Ψ and Φ_1, a 20% mean commonality ($\mu_2 = 0.20$) between Ψ and Φ_2, and a 1% mean commonality ($\mu_3 = 0.01$) between Ψ and Φ_3, the functions $\phi_i(x)$ will be represented as a triangular distribution (see Figure 1) with the triangle centered over its respective mean belief value μ_1, μ_2, μ_3 and base values given by $0 < a_i = \mu_i - h$ and $1 > b_i = \mu_i + h$ for some suitably small choice of $h > 0$. The assumption here is that there is some external mechanism for assessing the values of μ_i and h.

The density functions for the middle row of circles are therefore $\phi_i(x) = P[\Phi_i \,|\, \Psi; x]P[\Psi; x]$. In the application, the uncertainty lies in the expert's evaluations and not in the images Φ_i, Ψ, or the link between these two entities. Moreover, it is assumed that the photo Ψ is given so its probability of occurrence may be taken as 1. if no uncertainty is attached to the link from Ψ to Φ_i, the conditional probability of Φ_i given Ψ (call it π_i) is assumed to be a constant with respect to x: $\phi_i(x) = \delta(x - \pi_i)$.

The designations in the bottom row of circles in the figure represent three labels (call them L_1, L_2, L_3) and each of these possesses a probability density function (the certainty) represented by $\ell_j(x)$. In this example, assume that the certainty functions, $f_{i,j}$, are defined based on the general shape shown in Figure 4

with certitude functions defined as follows

Photo	**Label**	**Inference**	**Left End**	**Center**	**Right End**
Φ_1	L_1	$f_{1,1}$	0.5	0.75	1.0
	L_2	$f_{1,2}$	0.8	0.90	1.0
Φ_2	L_1	$f_{2,1}$	0.7	0.80	0.9
	L_2	$f_{2,3}$	0.3	0.40	0.5
Φ_3	L_3	$f_{3,3}$	0.9	0.95	1.0

Table 1: Certitude Functions

A similar table (shown below) defines the $\phi_i(x)$ functions. Here the left and right endpoints of the intervals ($a_i = \mu_i - h$ and $b_i = \mu_2 + h$) are given by **Center point** ± **Half-width** (see Figure 5). The table presents an illustration of the case that there is an 70% chance that the new photo illustrates the same damage as database photo 1, a 20% chance that the new photo shows damage similar to photo 2, and a 1% chance that it matches photo 3 (for technical reasons the certitude functions, $\phi_i(x)$, of the images are represented by triangular functions with a very small base (±0.01) rather than Dirac densities).

Photo	**Center Point**	**Half-Width**
Φ_1	0.70	0.01
Φ_2	0.20	0.01
Φ_3	0.01	0.01

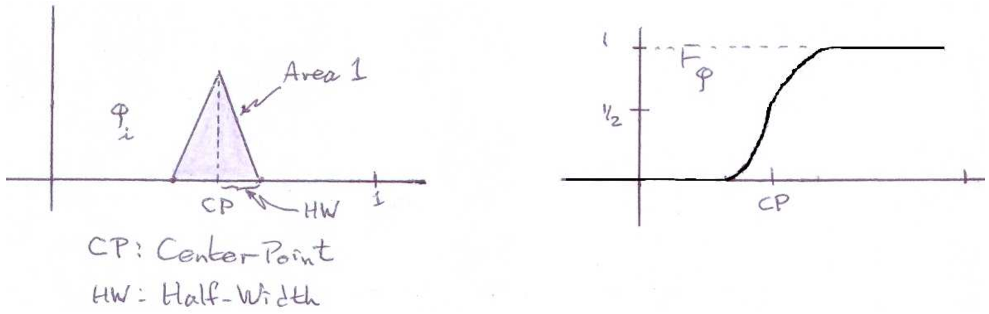


Figure 5

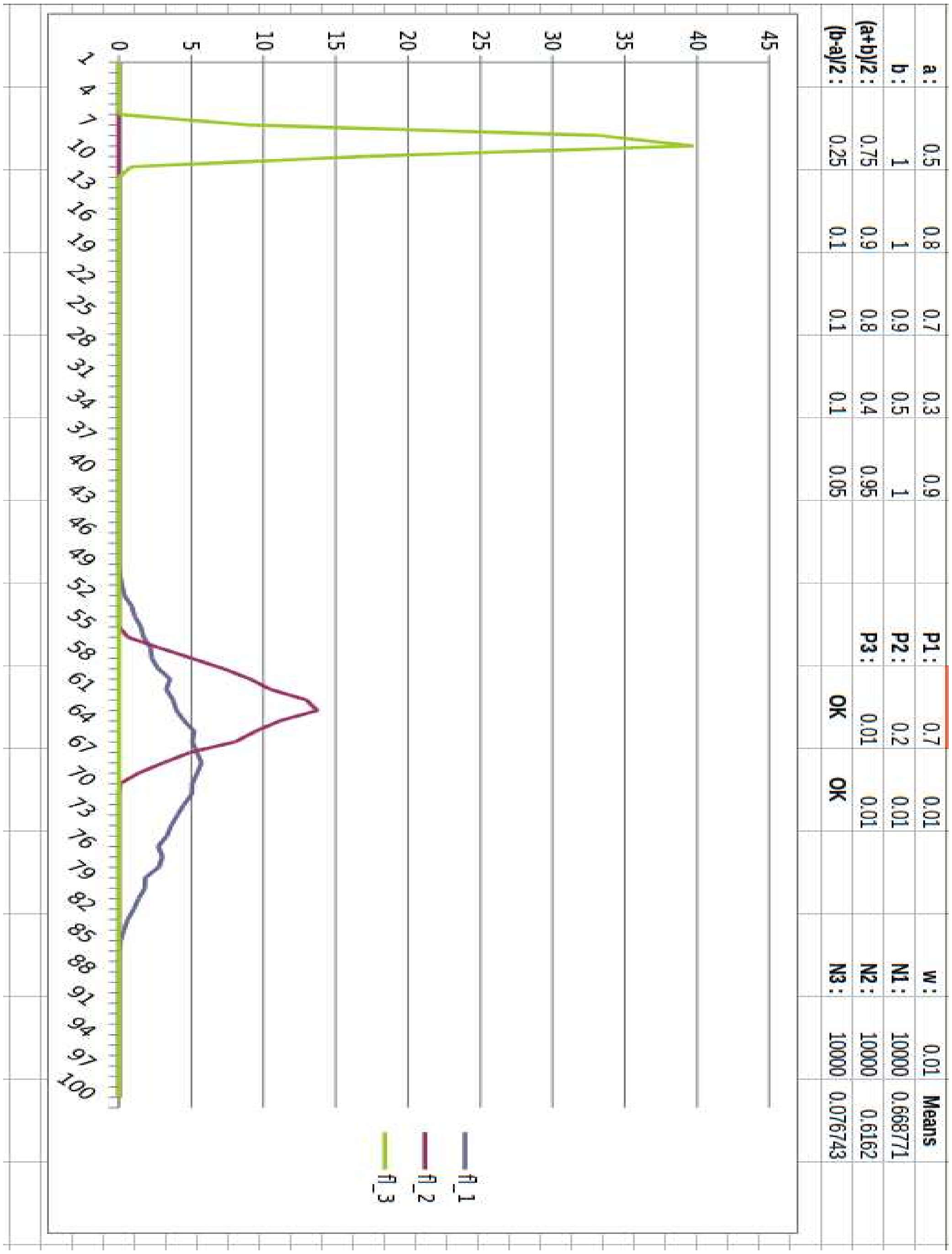
The sole requirements here are

1. the left endpoints of the functions $\phi_i(x)$ (here: 0.79, 0.09, 0.0) are greater than or equal to zero, and
2. the right endpoints (here: 0.81, 0.11, 0.02) are less than or equal to 1.

The basic relationships between all of the certainty functions is expressed by

$$\begin{aligned}\ell_1 &= \phi_1 f_{1,1} + \phi_2 f_{2,1}\\ \ell_2 &= \phi_1 f_{1,2}\\ \ell_3 &= \phi_2 f_{2,3} + \phi_3 f_{3,3}.\end{aligned}$$

In each of these equations, ℓ_j should really be written as $\ell_j(x)$ as should ϕ_i and $f_{i,j}$. The dependency on x is there but not shown for the sake of clarity. The relationship among ℓ_j, $f_{i,j}$, and ϕ_j is complex enough to make analytic computation of the certainty functions (aka probability density functions) for ℓ_j difficult without asserting additional independence assumptions. However, and estimation of ℓ_j may be computed through simulation. Figure 6 shows the simulation results for this set of certainty functions.



a :	0.5	0.8	0.7	0.3	0.9		P1 :	0.7	0.01		w :	0.01	Means
b :	1	1	0.9	0.5	1		P2 :	0.2	0.01		N1 :	10000	0.668771
(a+b)/2 :	0.75	0.9	0.8	0.4	0.95		P3 :	0.01	0.01		N2 :	10000	0.6162
(b-a)/2 :	0.25	0.1	0.1	0.1	0.05			OK	OK		N3 :	10000	0.076743

Figure 6
Sample Label Certitude Functions
Link certitudes: sideways rows 1–6 Image certitudes: sideways rows 8–10

Remark 1. The graphs presented in this chart represent simulated values of the functions ℓ_1, ℓ_2, and ℓ_3. These graphs appear to be triangular. This may be an indication that independence assumptions are warranted and that assumption can lead to a possible simplification in the simulation. Under independence assumptions, the functions ℓ_1, ℓ_2, and ℓ_3 may be worked out analytically.

Remark 2. The column labeled **Means** shows the mean certainty assigned to each L_j as implied by the certainty function $\ell_j(x)$. That is, the average certainty assigned to the statement *label L_j is attached to photograph* Φ is 0.668 ($i = 1$), 0.616 ($i = 2$), and 0.076 ($i = 3$).

However, this is not the only metric that could be used to assess the certainty of the association of L_j with Φ. Another candidate metric might be the value of x where $\ell_j(x)$ is maximum. Also, the variance (or standard deviation) of the certainty function $\ell_j(x)$ might also be reported.

2.6 Simulation. The basic relationships above may be used to generate the density functions associated with ℓ_j. Notice that values expressed by ℓ_j are dependent on eight random quantities: the five $f_{i,j}$ and the three ϕ_i. Samples of $f_{i,j}$ and ϕ_j are generated in accordance with their respective certainty distributions.

By way example, if $g(x)$ is one of the certainty densities, begin by forming the distribution function, $G(x)$, of $g(x)$ as follows:

$$G(x) = \int_{-\infty}^{x} g(t)\,dt = \begin{cases} 0 & \text{if } x < 0 \\ \displaystyle\int_{0}^{x} g(t)\,dt & \text{if } 0 \leq x \leq 1 \\ 1 & \text{if } 1 < x \end{cases} \cdot$$

It is well-known that a random number, R, extracted from a probability distribution with density function $g(x)$ and distribution function $G(x)$ can be produced by starting with a value, U, taken from a uniform distribution over $[0, 1)$ and then computing the value $R = G^{-1}(U)$. Note that the distribution function for R is

$$F_R(r) = P[R \leq r] = P[G^{-1}(U) \leq r] = P[U \leq G(r)] = G(r),$$

because for the random variable U (uniform over $[0, 1)$), has a distribution function given by $P[U \leq u] = u$. Therefore, the distribution function for the variable R must be $G(r)$.

Remark 1. The interval $[0, 1)$ is used because most uniform random number generators found in software libraries produce random numbers in this range.

Remark 2. In the rare case where $U = 0$, the inverse function $G^{-1}(U)$ is not well-defined. That is, there are infinitely many pre-images of the value 0. The value $R = 0$ is used in this case.

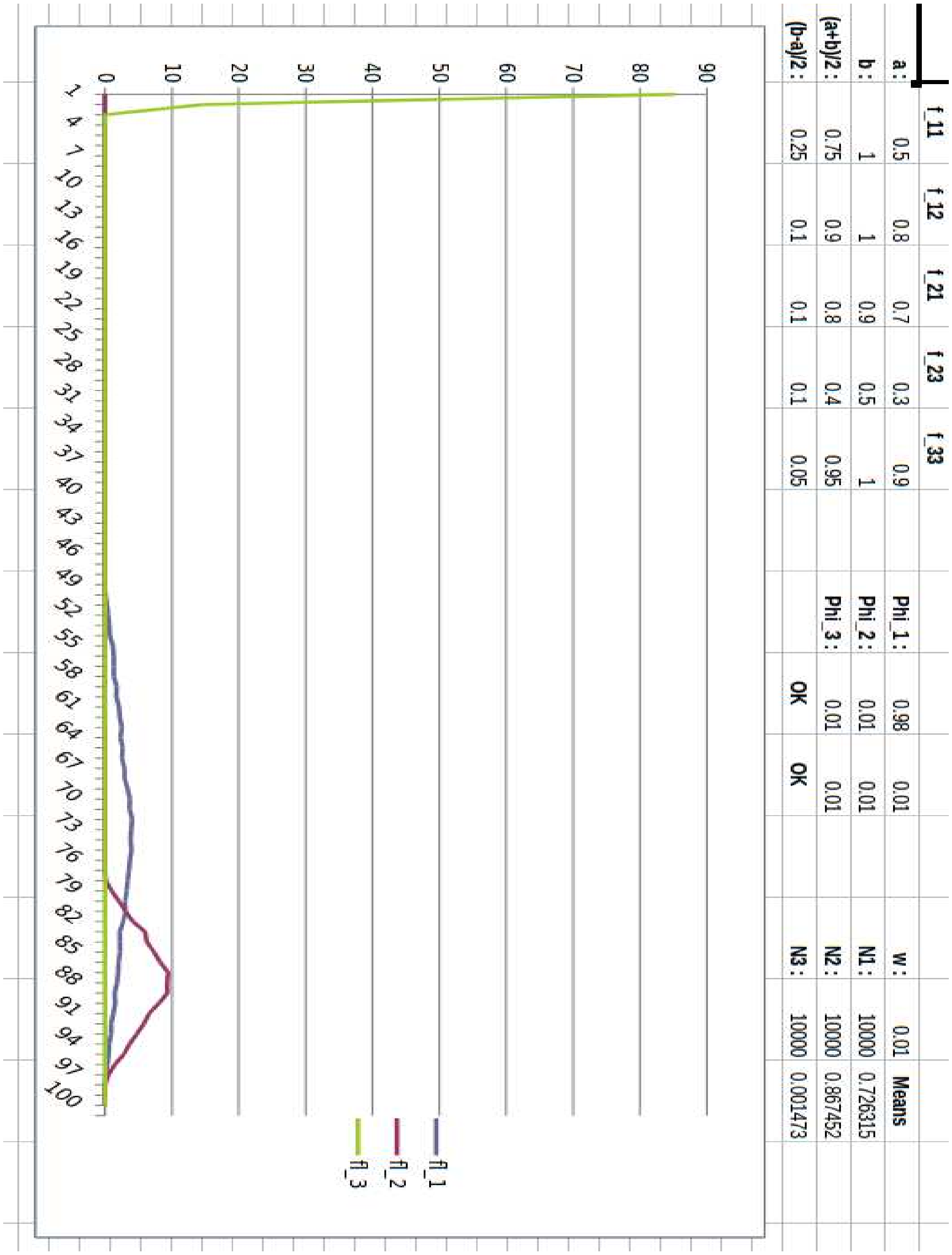
Remark 3. This mechanism works equally well with impulsive density functions described in section **1.7**. In this case $g(x) = \delta(x - a)$ where a is a constant and

$$G(x) = \int_{-\infty}^{x} g(t)\,dt = \begin{cases} 0 & \text{if } x < a \\ 1 & \text{if } a \leq x. \end{cases}$$

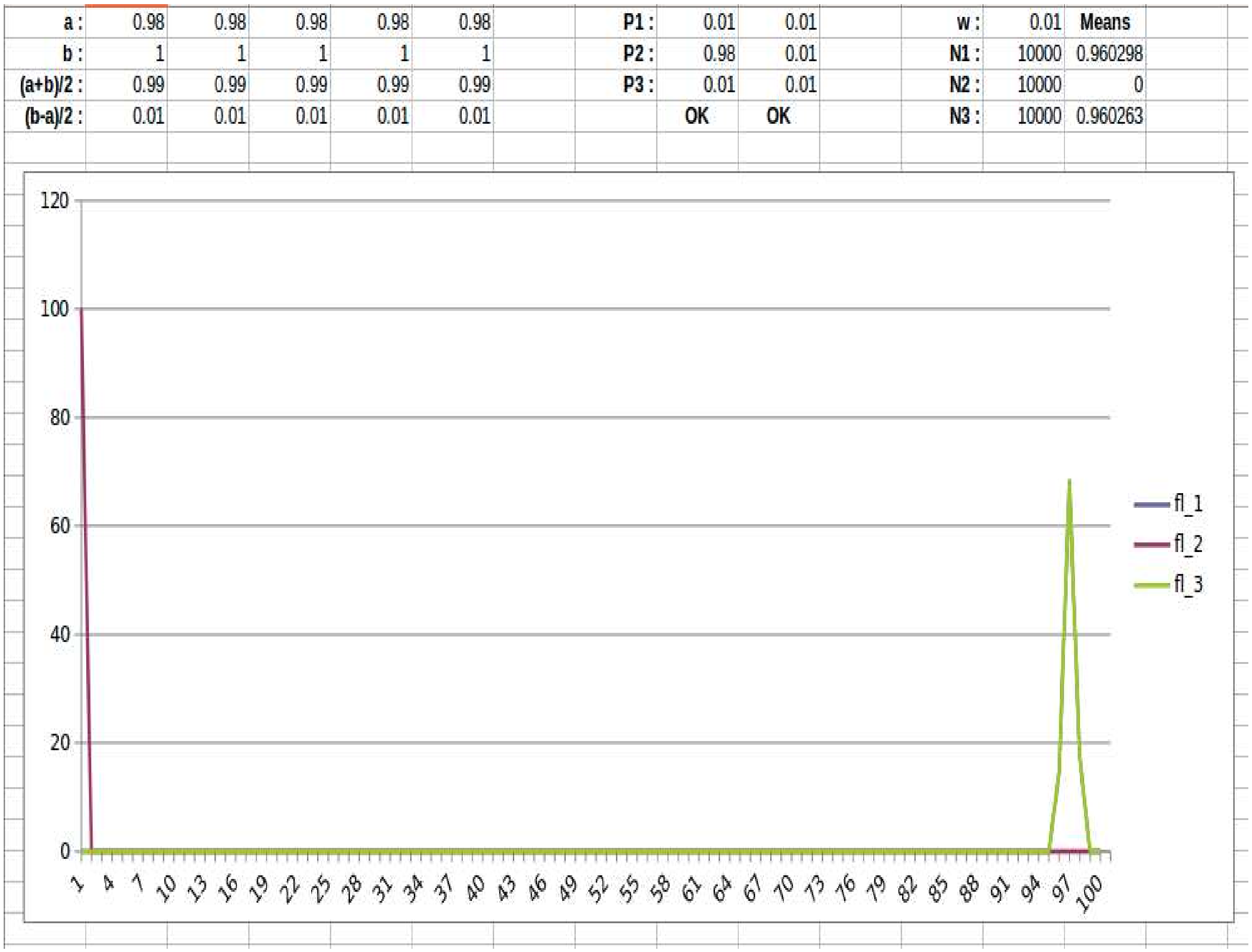
That is, $R = 1$ if and only if $U \geq a$.

Begin the simulation by producing certainty samples for each of the random variables Φ_j and $f_{i,j}$ according to their respective distribution functions (following convention, call these $\tilde{f}_{i,j}$ and $\tilde{\phi}_j$. Combine these eight values together to form $\tilde{\ell}_1$, $\tilde{\ell}_2$, and $\tilde{\ell}_3$. Begin three lists of values, one holding $\tilde{\ell}_1$, one holding $\tilde{\ell}_2$, and one holding $\tilde{\ell}_3$. Repeat this generation scheme N times adding each new value $\tilde{\ell}_j$ to the corresponding list. Each list may contain many duplicated values. From these lists generate a normalized histogram (that is, a histogram normalized to unit area). This type of histogram represents an approximation to the density of the expression ℓ_j.

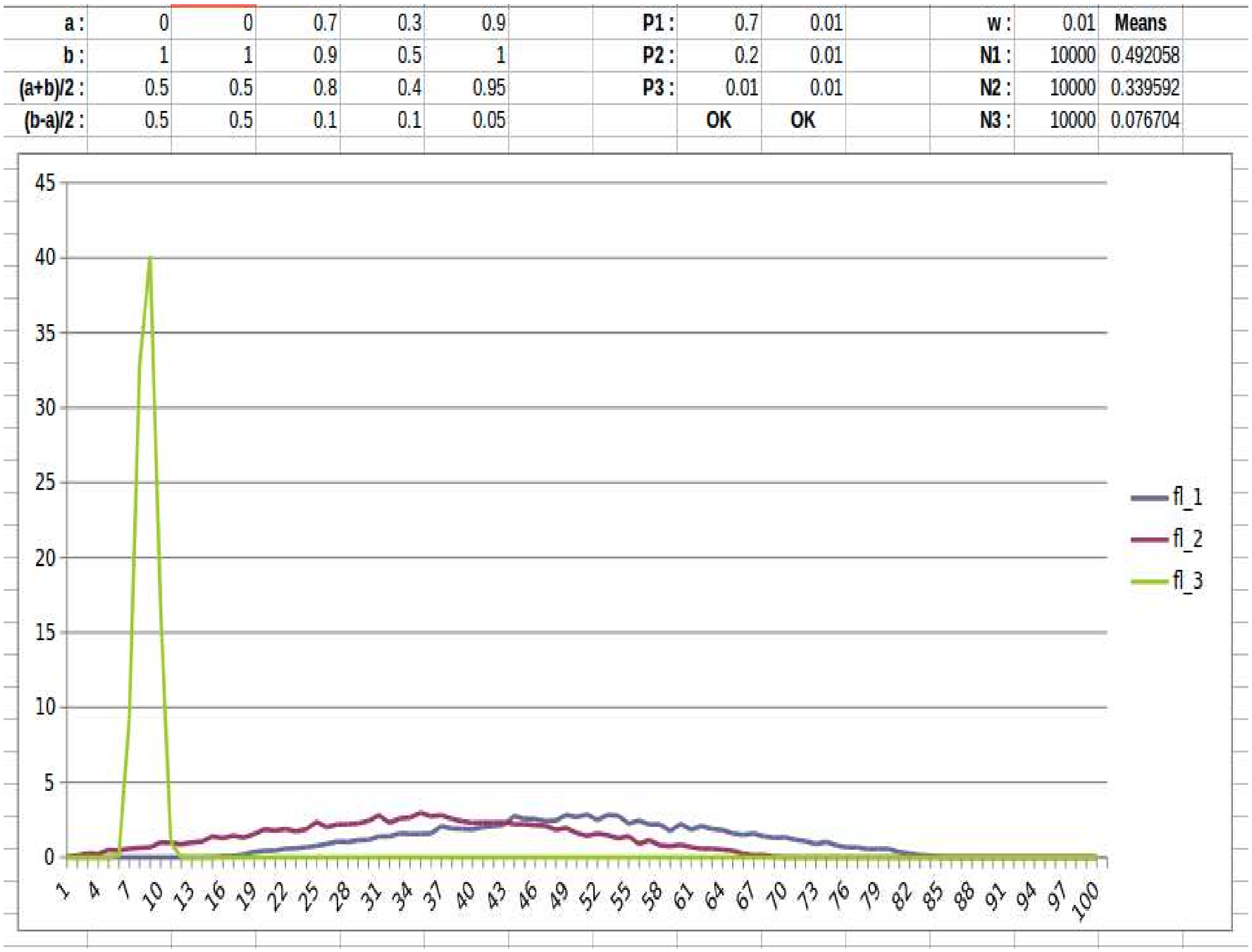
2.7 More Results. Here are more runs of the simulation with changes made to the inference functions ($(f_{ij}(x))$) as well as the functions $\phi_i(x)$.



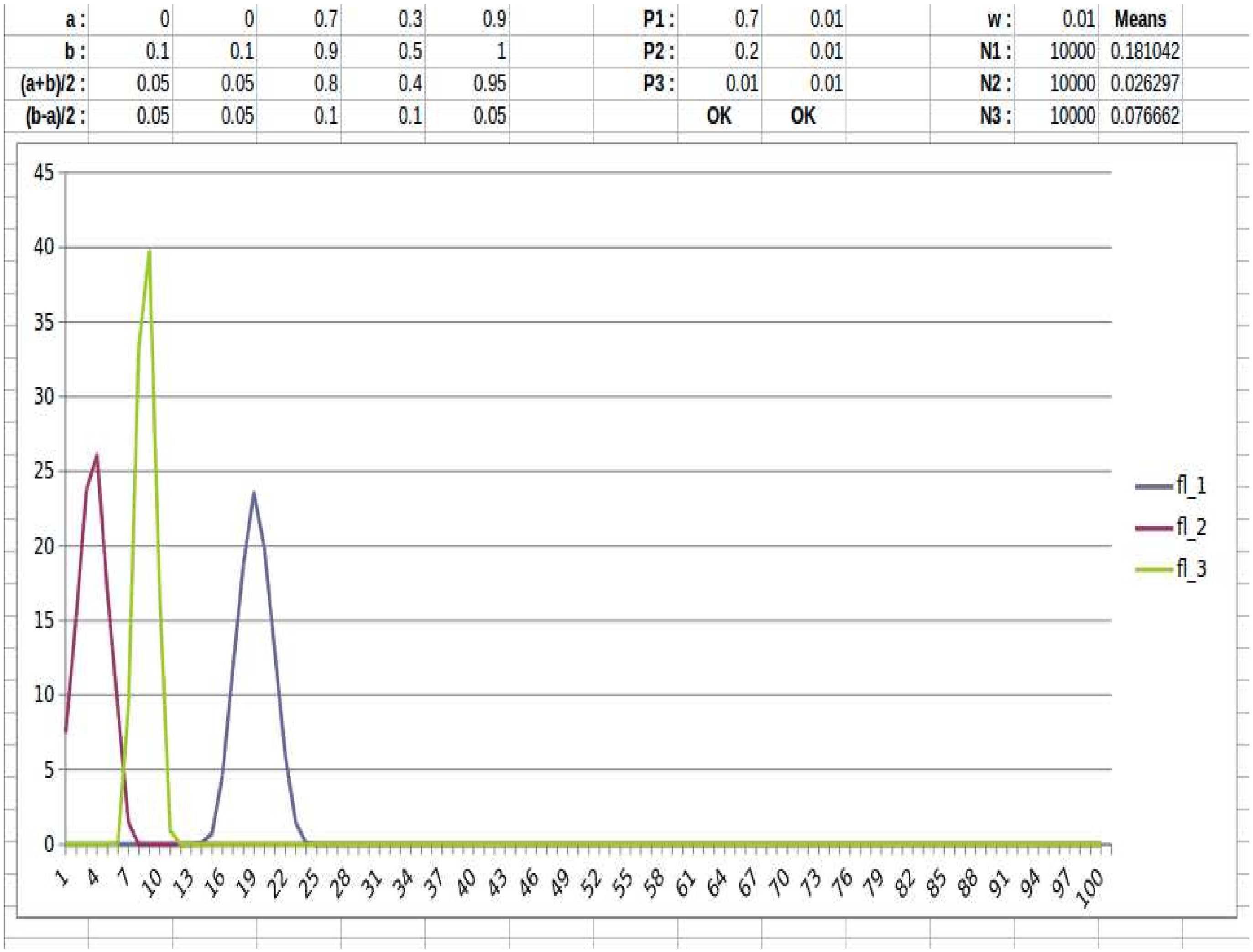
	f_11	f_12	f_21	f_23	f_33							
a :	0.5	0.8	0.7	0.3	0.9	Phi_1 :	0.98	0.01		w :	0.01	Means
b :	1	1	0.9	0.5	1	Phi_2 :	0.01	0.01		N1 :	10000	0.726315
(a+b)/2 :	0.75	0.9	0.8	0.4	0.95	Phi_3 :	0.01	0.01		N2 :	10000	0.867452
(b-a)/2 :	0.25	0.1	0.1	0.1	0.05		OK	OK		N3 :	10000	0.001473



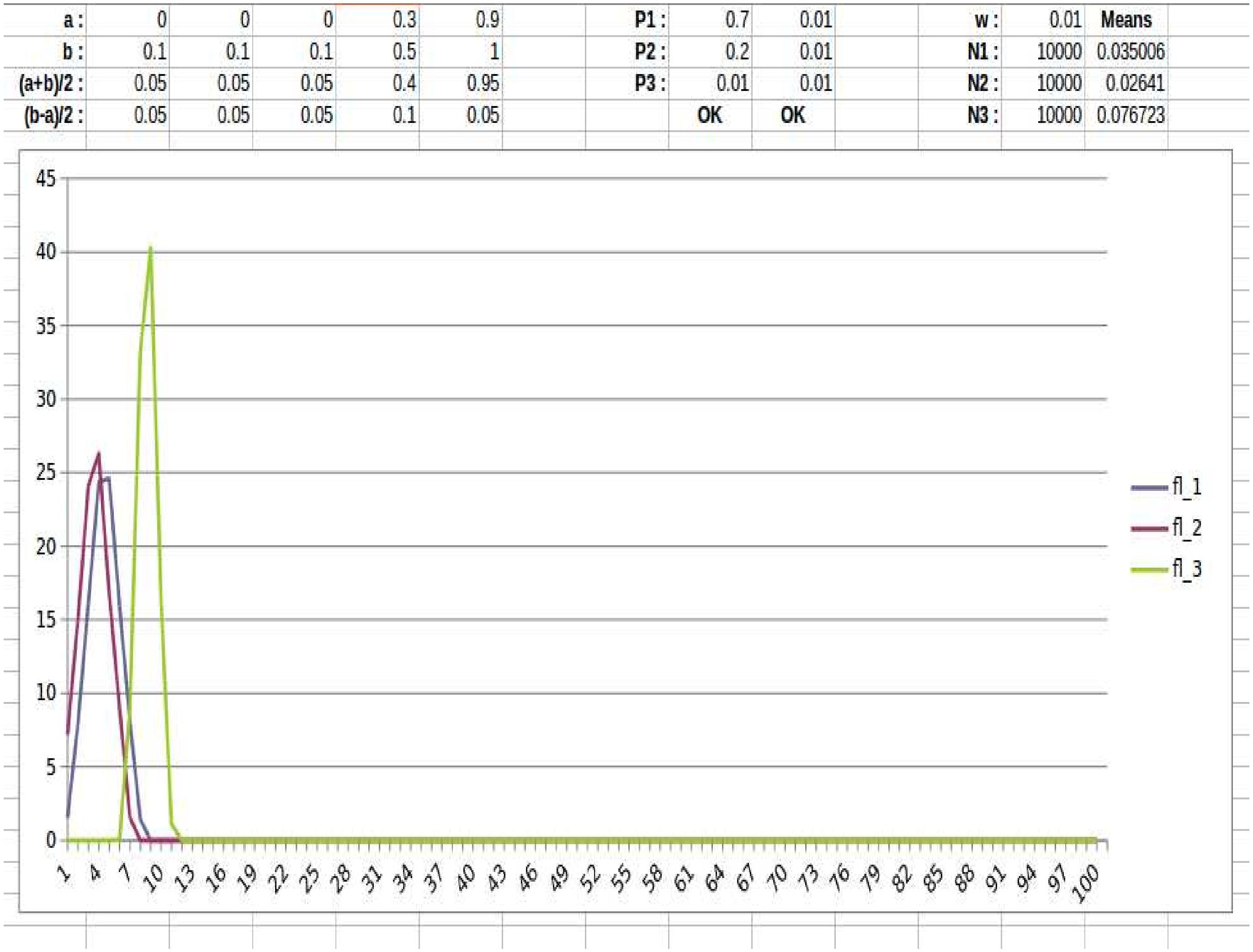
a :	0.98	0.98	0.98	0.98	0.98		P1 :	0.01	0.01		w :	0.01	Means
b :	1	1	1	1	1		P2 :	0.98	0.01		N1 :	10000	0.960298
(a+b)/2 :	0.99	0.99	0.99	0.99	0.99		P3 :	0.01	0.01		N2 :	10000	0
(b-a)/2 :	0.01	0.01	0.01	0.01	0.01			OK	OK		N3 :	10000	0.960263



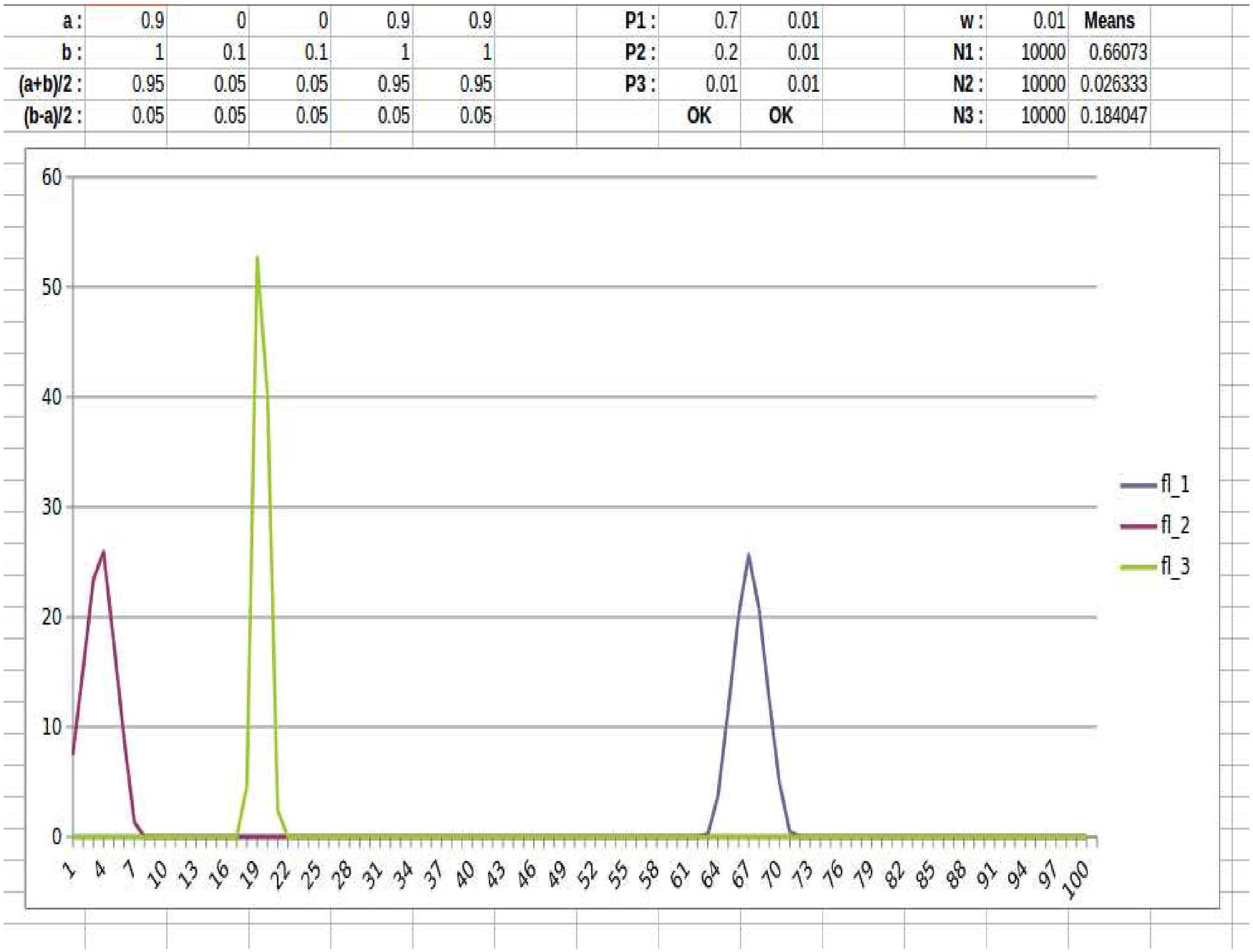
a :	0	0	0.7	0.3	0.9		**P1 :**	0.7	0.01		**w :**	0.01	**Means**
b :	1	1	0.9	0.5	1		**P2 :**	0.2	0.01		**N1 :**	10000	0.492058
(a+b)/2 :	0.5	0.5	0.8	0.4	0.95		**P3 :**	0.01	0.01		**N2 :**	10000	0.339592
(b-a)/2 :	0.5	0.5	0.1	0.1	0.05			**OK**	**OK**		**N3 :**	10000	0.076704



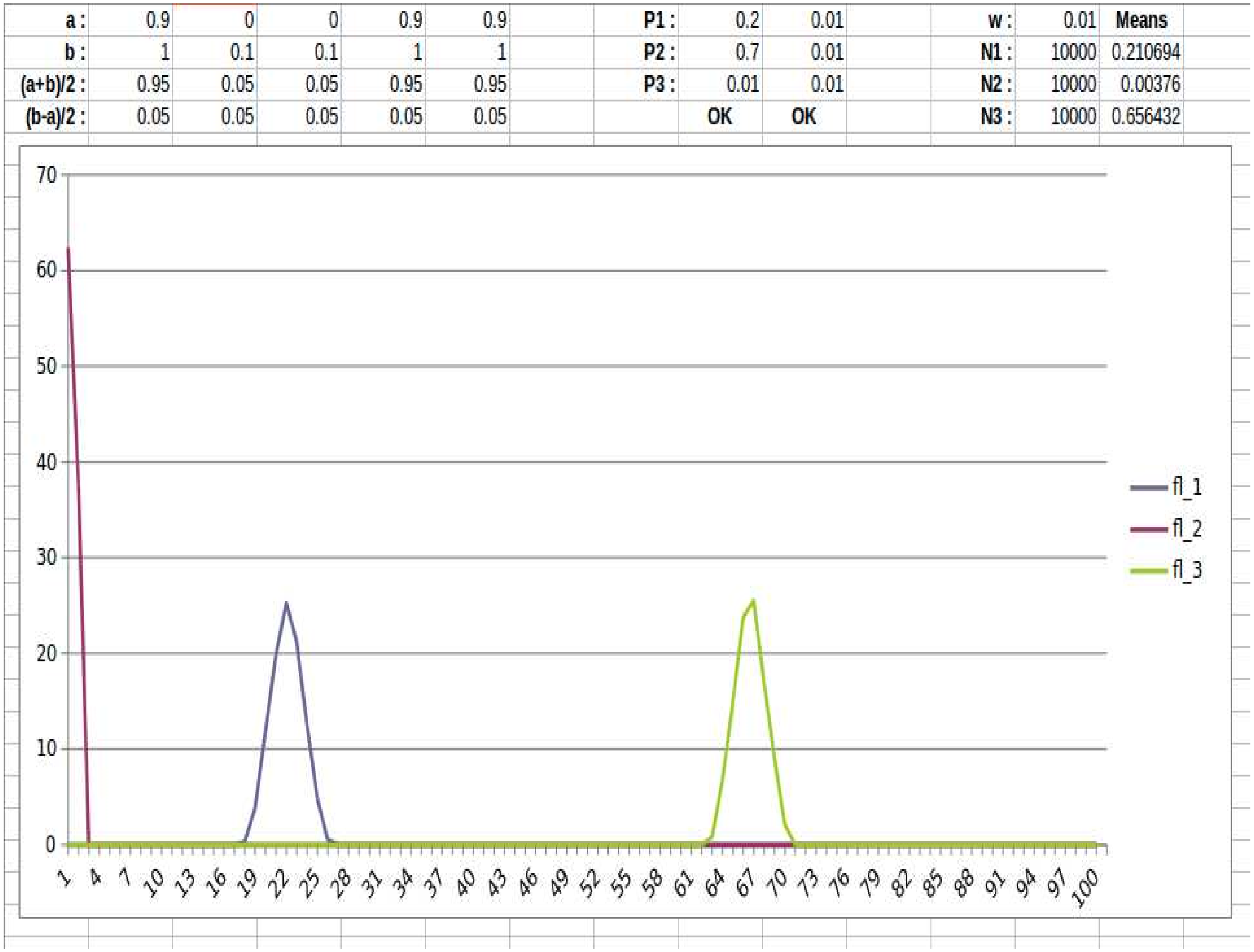
a :	0	0	0.7	0.3	0.9		P1 :	0.7	0.01		w :	0.01	Means
b :	0.1	0.1	0.9	0.5	1		P2 :	0.2	0.01		N1 :	10000	0.181042
(a+b)/2 :	0.05	0.05	0.8	0.4	0.95		P3 :	0.01	0.01		N2 :	10000	0.026297
(b-a)/2 :	0.05	0.05	0.1	0.1	0.05			OK	OK		N3 :	10000	0.076662



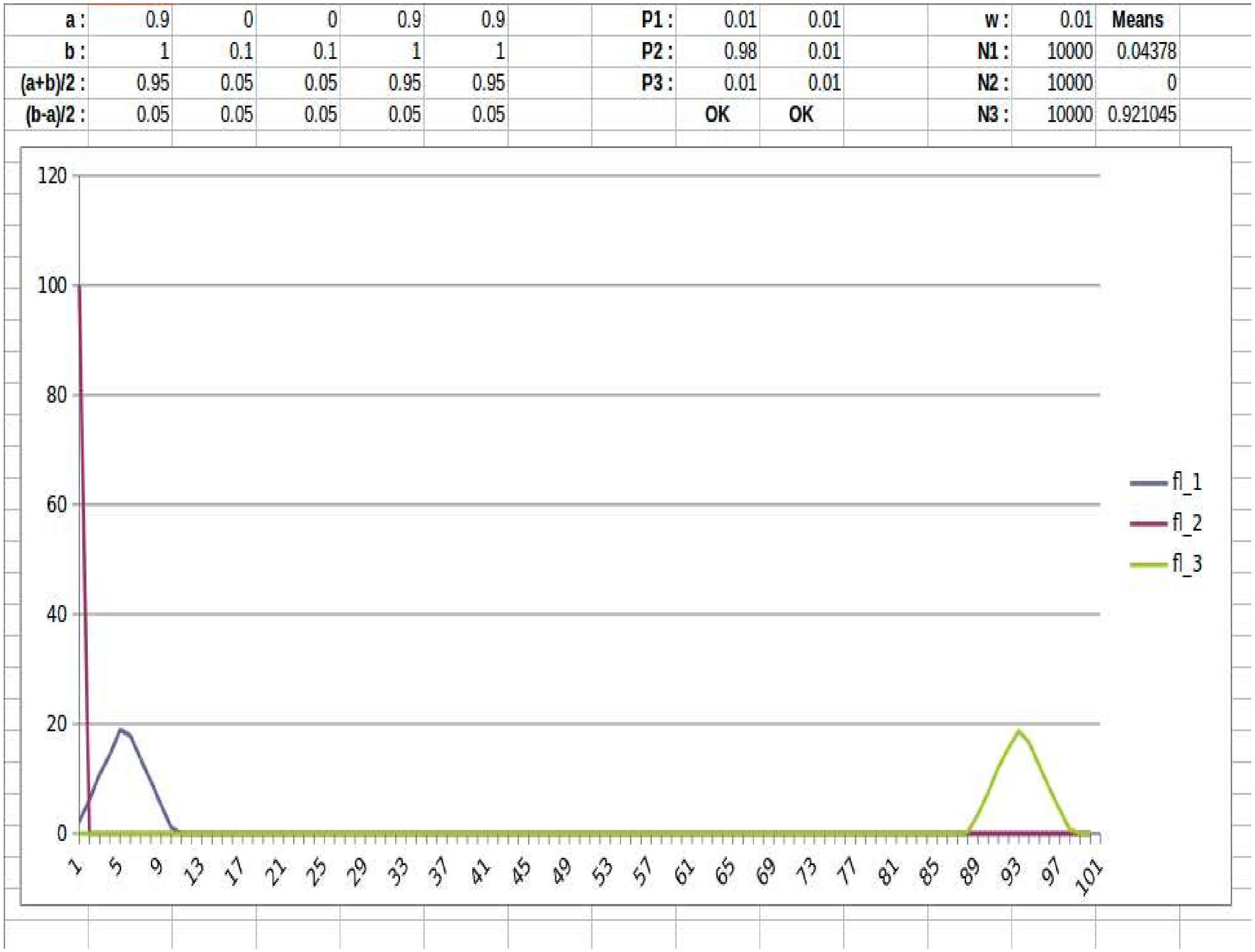
a :	0	0	0	0.3	0.9		**P1 :**	0.7	0.01		**w :**	0.01	**Means**
b :	0.1	0.1	0.1	0.5	1		**P2 :**	0.2	0.01		**N1 :**	10000	0.035006
(a+b)/2 :	0.05	0.05	0.05	0.4	0.95		**P3 :**	0.01	0.01		**N2 :**	10000	0.02641
(b-a)/2 :	0.05	0.05	0.05	0.1	0.05			**OK**	**OK**		**N3 :**	10000	0.076723



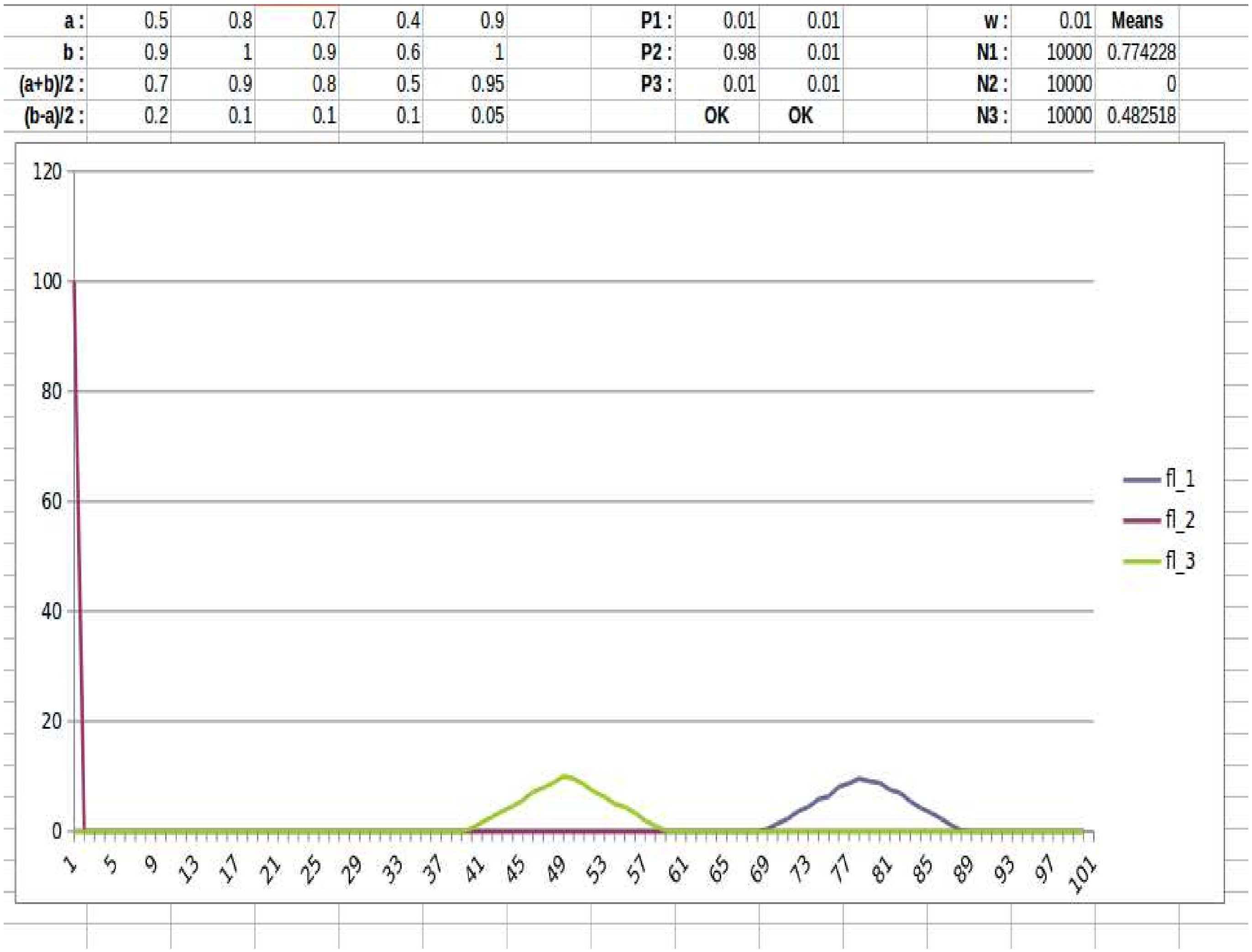
a :	0.9	0	0	0.9	0.9		P1 :	0.7	0.01		w :	0.01	Means
b :	1	0.1	0.1	1	1		P2 :	0.2	0.01		N1 :	10000	0.66073
(a+b)/2 :	0.95	0.05	0.05	0.95	0.95		P3 :	0.01	0.01		N2 :	10000	0.026333
(b-a)/2 :	0.05	0.05	0.05	0.05	0.05			OK	OK		N3 :	10000	0.184047



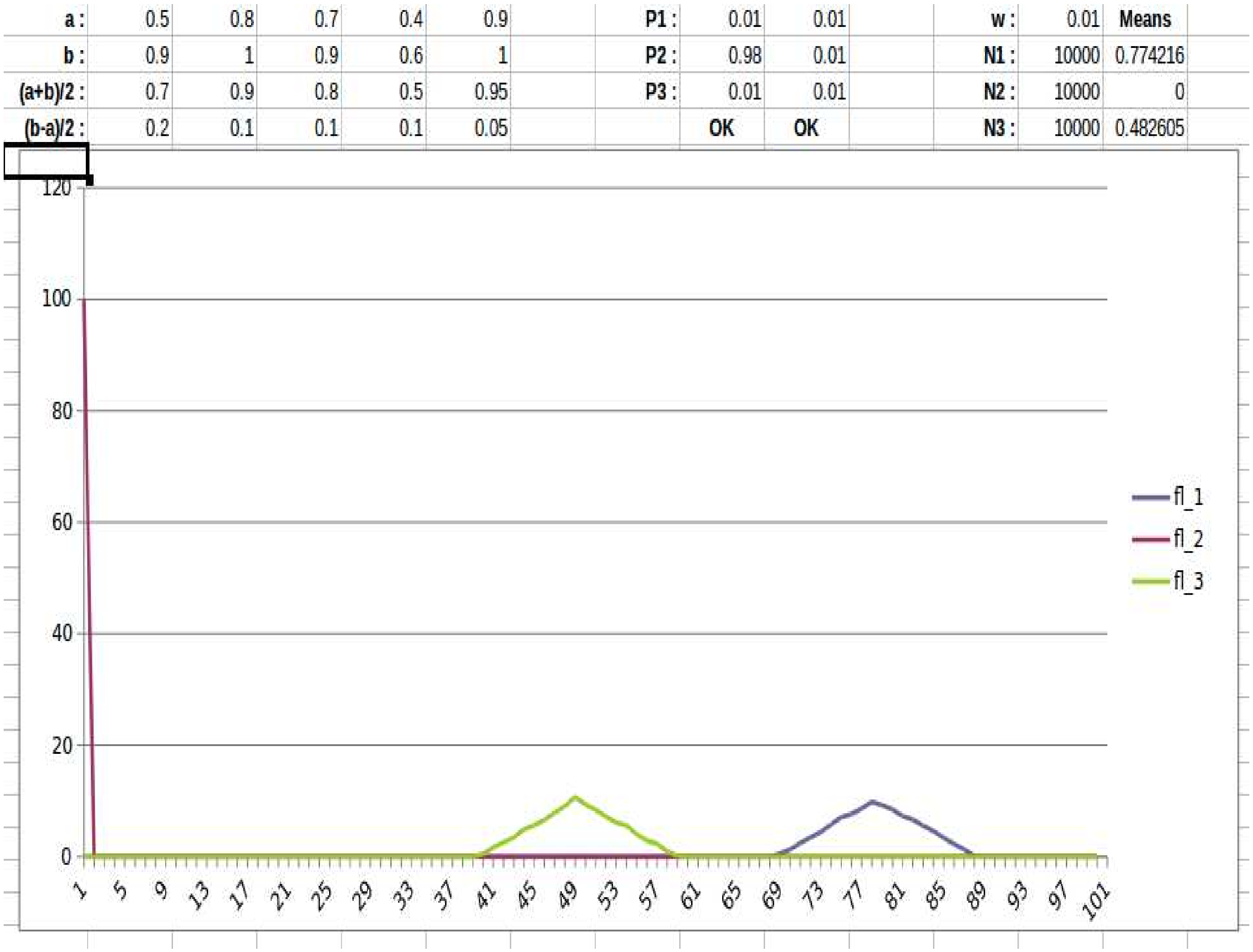
a :	0.9	0	0	0.9	0.9		**P1 :**	0.2	0.01		**w :**	0.01 **Means**
b :	1	0.1	0.1	1	1		**P2 :**	0.7	0.01		**N1 :**	10000 0.210694
(a+b)/2 :	0.95	0.05	0.05	0.95	0.95		**P3 :**	0.01	0.01		**N2 :**	10000 0.00376
(b-a)/2 :	0.05	0.05	0.05	0.05	0.05			**OK**	**OK**		**N3 :**	10000 0.656432



a :	0.9	0	0	0.9	0.9		P1 :	0.01	0.01		w :	0.01	Means
b :	1	0.1	0.1	1	1		P2 :	0.98	0.01		N1 :	10000	0.04378
(a+b)/2 :	0.95	0.05	0.05	0.95	0.95		P3 :	0.01	0.01		N2 :	10000	0
(b-a)/2 :	0.05	0.05	0.05	0.05	0.05			OK	OK		N3 :	10000	0.921045



a :	0.5	0.8	0.7	0.4	0.9		P1 :	0.01	0.01		w :	0.01	Means
b :	0.9	1	0.9	0.6	1		P2 :	0.98	0.01		N1 :	10000	0.774228
(a+b)/2 :	0.7	0.9	0.8	0.5	0.95		P3 :	0.01	0.01		N2 :	10000	0
(b-a)/2 :	0.2	0.1	0.1	0.1	0.05			OK	OK		N3 :	10000	0.482518



a :	0.5	0.8	0.7	0.4	0.9		**P1 :**	0.01	0.01		**w :**	0.01	**Means**
b :	0.9	1	0.9	0.6	1		**P2 :**	0.98	0.01		**N1 :**	10000	0.774216
(a+b)/2 :	0.7	0.9	0.8	0.5	0.95		**P3 :**	0.01	0.01		**N2 :**	10000	0
(b-a)/2 :	0.2	0.1	0.1	0.1	0.05			**OK**	**OK**		**N3 :**	10000	0.482605

2.8 Multiple Experts at Work. Suppose that a single photograph, Φ_s, in the photo-base is evaluated by K different experts. This section describes how to incorporate all opinions concerning a single (image,label) pair (Φ_s, L_j) into a single inference function $f_{sj}(x)$.

Let E_i be the event *expert i makes an assessment*. Then

$$\begin{aligned} P[\Phi_s, L_j] &= P[\big(\Phi_s \cap \bigcup_i E_i\big), L_j] \\ &= \sum_i P[\Phi_s, E_i, L_j] \\ &= \sum_i P[L_j \,|\, \Phi_s, E_i] P[\Phi_s, E_i] \\ &= \sum_i P[L_j \,|\, \Phi_s, E_i] P[E_i \,|\, \Phi_s] P[\Phi_s]. \end{aligned}$$

The left-hand side of this last equation may be written as $P[\Phi_s, L_j] = P[L_j \,|\, \Phi_s] P[\Phi_s]$. Removing the common factor[3] $P[\Phi_s] > 0$ from both the left- and right-hand sides leaves

$$P[L_j \,|\, \Phi_s] = \sum_i P[L_j \,|\, \Phi_s, E_i] P[E_i \,|\, \Phi_s] \qquad (*).$$

The term $P[L_j \,|\, \Phi_s, E_i]$ is the probability that label L_j is assigned given that photograph Φ_s is under consideration. The term $P[L_j \,|\, \Phi_s, E_i]$ is the probability that label L_j is assigned given that photograph Φ_i is under consideration and that expert E_j is performing the evaluation. And the term $P[E_i \,|\, \Phi_s]$ is the probability that expert E_i performs the evaluation given the photograph Φ_s. Assuming for the moment that expert E_i is just as likely to be chosen for this task as any other of the K experts who evaluate this photograph, set $P[E_i \,|\, \Phi_s] = 1/K$.

Note that the factor of $1/K$ may be considered equivalent to the statement that no expert is favored over any other expert in performing the review. As a quality profile develops for the complete set of experts it may be better to make the choice factor $P[E_i \,|\, \Phi_s]$ for each expert more reflective of the expert's quality (say, perhaps, proportional to the expert's quality factor).

Relationship $(*)$ then becomes

$$P[L_j \,|\, \Phi_s] = \frac{1}{K} \sum_i P[L_j \,|\, \Phi_s, E_i]$$

or

$$KX = \sum_i X_i,$$

where $X_i = P[L_j \,|\, \Phi_s, E_i]$ and $X = P[L_j \,|\, \Phi_s]$. So far, the two quantities of interest, X and X_i, have been treated as single, random variables. From this last equation, a certainty (probability density) function may be computed by assuming that each expert's assessment is independent of all the other expert's assessments. That is, the random variables X_i are conditionally independent with respect to each expert.

[3] Taken to be positive.

The density function of the sum of two independent random variables may be computed by finding the convolution product of the two density functions:

$$f_{(X+Y)}(x) = \big(f_X \star f_Y\big)(x) = \int_{-\infty}^{\infty} f(t)g(x-t)\,dt.$$

Therefore the certainty function associated with $KX = KP[L_j \mid \Phi_s]$ (call it $f_{KX}(x)$) will be the K-fold convolution of the certainty functions associated with the variables $X_i = P[L_j \mid \Phi_s, E_i]$ (call them $f_{X_i}(x)$):

$$f_{KX}(x) = \big(f_{X_1} \star f_{X_2} \star \ldots \star f_{X_K}\big)(x).$$

Finally, the relationship between $f_{KX}(x)$ and $f_X(x)$ is given, in general, by

$$f_X(x) = K f_{KX}(Kx) \qquad 0 \le x \le 1/K.$$

Note that $X_i = P[L_j \mid \Phi_s, E_i]$ has a certainty (probability density) function given by the expression $f_{X_i}(x) = f_{s,j,i}(x)$ for the i-th expert where $f_{s,j,i}(x)$ is the triangular density defined in section 1.2 for the i-th expert.

The convolution product of two densities may be computed by a direct calculation of the convolution integral. For a K-fold product this integral becomes

$$\begin{aligned}&\big(f_{X_1} \star f_{X_2} \star \ldots \star f_{X_K}\big)(x)\\ &= \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_{X_1}(x_1) f_{X_2}(x_2 - x_1) f_{X_3}(x_3 - x_2) \ldots f_{X_K}(x - x_{K-1})\,dx_{K-1} \ldots dx_2 dx_1.\end{aligned}$$

The convolution integral may also be computed using a Fourier transform. Define the Fourier transform of the function $f(t)$ by

$$\widehat{f}(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t}\,dt.$$

The inverse Fourier transform is given by

$$\widetilde{f}(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \widehat{f}(\omega) e^{i\omega t}\,d\omega.$$

If conditions are right for the function $f(t)$ (as they are when $f(t)$ is a probability density function) then $\widetilde{f}(t) = f(t)$ almost everywhere. This ensures that integrals involving the function $f(t)$ are the same as integrals involving $\widetilde{f}(t)$. The convolution theorem states that the Fourier transform of $\big(f \star g\big)(t)$ is the same as the product of the Fourier transform of $f(t)$ with the Fourier transform of $g(t)$. That is, $\big(f \star g\big)(t)$ equals the inverse transform of the product $\widehat{f}(\omega)\widehat{g}(\omega)$. This transform-multiply-inverse transform operation is usually quicker that computing the two-fold convolution integral. A K-fold convolution product will require the Fourier transforms of K functions, a K-fold product, and then a single inverse transform.

If each of the functions, $f(x)$, to be convolved is represented discretely by a collection of $n = 2^m$ points, a Discrete Fast Fourier Transform may be used to perform the computations (See [5]). Each of the initial K Fourier transforms requires $O(\frac{1}{2}n\log n)$ multiply/add operations, followed by $O(Kn)$ multiplies to implement the convolution theorem operations, and $O(\frac{1}{2}n\log n)$ operations to perform the inverse transform. The total complexity of the algorithm is then on the order of

$$\frac{1}{2}(K+1)n\log n + Kn.$$

2.9 Certainty. If it is believed that an expert's rating $X_i = P[L_j \,|\, \Phi_s, E_i]$ of a photograph is certain (that is, is to be taken as truth) the certainty distribution function of X_i may be taken as

$$F_X(x) = F_{s,j,i}(x) = \begin{cases} 1 & \text{if } x \geq x_i \\ 0 & \text{Otherwise,} \end{cases}$$

where x_i is the actual likelihood assigned to the (image,label) link given the opinion expressed by the expert. For example, the expert E_i could assert that $1 = P[L_j \,|\, \Phi_s, E_i]$ ("image Φ_s absolutely shows damage identified by label L_j"). Since no one can be absolutely certain, the value assigned to x_i may be exactly 1 but is more likely a value close to 1 (e.g. 0.97) – the later choice may ease computational problems by moving slightly away from the extreme value of 1.

2.9.1 Convolution of Two Certainties. Suppose two independent certainties X_a and X_b are to be used to form a composite certainty $Y = \alpha X_a + \beta X_b$. Both certainty components have distribution function and density functions as follows:

$$\begin{aligned} F_{X_a}(x) &= \begin{cases} 1 & \text{if } x \geq a \\ 0 & \text{Otherwise,} \end{cases} \\ f_{X_a}(x) &= \delta(t-a) \end{aligned}$$

Computation of $F_Y(y)$ is now given by

$$\begin{aligned} F_Y(y) &= \int_{-\infty}^{\infty} \int_{-\infty}^{(y-\alpha s)/\beta} F_{X_a,X_b}(s,t)\, dtds \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{(y-\alpha s)/\beta} f_{X_a}(s) f_{X_b}(t)\, dtds \\ &= \int_{-\infty}^{\infty} f_{X_a}(s) \int_{-\infty}^{(y-\alpha s)/\beta} \delta(t-b)\, dtds \\ &= \int_{-\infty}^{\infty} \delta(s-a) I((y-\alpha s)/\beta \geq b))\, ds, \end{aligned}$$

where $I(z \geq b)$ is an indicator function that x is greater than or equal to b. Another way of expressing this is to write $I(z \geq b) = H(z-b)$ where $H(\cdot)$ is the Heaviside function. Then continuing

$$\begin{aligned} F_Y(y) &= \int_{-\infty}^{\infty} \delta(s-a) H((y-\alpha s)/\beta - b))\, ds \\ &= H((y-\alpha a)/\beta - b)) \\ &= H((y-\alpha a - \beta b)/\beta)) \\ &= H(y - \alpha a - \beta b) \end{aligned}$$

Consequently, $f_Y(y) = \delta(y - \alpha a - \beta b)$. Notice that the expression $f_Y(y) = \delta(y - c)$ is the same as asserting that Y is the constant c with probability 1 since for all sequences ϵ_n with $\epsilon_1 > \epsilon_2 > \ldots > 0$ and $\lim_n \epsilon_n = 0$:

$$\begin{aligned} P[Y = c] &= P[\bigcap_{n=1}^{\infty} \{c - \epsilon_n < Y \leq c + \epsilon_n\}] \\ &= \lim_{n\to\infty} P[c - \epsilon_n < Y \leq c + \epsilon_n] \\ &= \lim_{n\to\infty} \big(F(c + \epsilon_n) - F(c - \epsilon_n)\big) = 1 \end{aligned}$$

since $F_Y(y) = H(y - c)$. By repetition of this argument, it is clear that if $Y = \sum_i \alpha_i X_i$ where each of the $X_i = c_i$ are certainties, that $f_Y(y) = \sum_i \alpha_i c_i$ with probability 1.

2.9.2 Convolution of a Certainty with a non-Certainty.

A straightforward application of the convolution of a certainty function with a Dirac density. In the many of the computations that follow, certitude densities are represented by a discrete array of points, `Cert[i]`. The spacing, Δp, represented by the index `i`, now makes it difficult to distinguish between a triangular density centered at p_0 with a base width of $2p$ and the impulsive density $\delta(x - p_0)$. In the original analysis, triangular densities with a base width of $2p$ were used in place of Dirac impulses. This forced all non-Dirac densities to have a base width of $4p$ or more.

2.10 Recovery of $P[\Phi]$ from $P[L]$ and $P[L]$ from $P[\Phi]$.

If the values of $P[\Phi \,|\, L]$ were known, it would be possible to recover the probabilities $P[\Phi]$ from knowledge of the values of $P[L]$ from the relationship

$$P[\Phi] = \sum_L P[\Phi \,|\, L] P[L].$$

However, the values of $P[\Phi \,|\, L]$ in this application are not known. It must also be the case that the following *a priori* relationship exists

$$P[L] = \sum_\Phi P[L \,|\, \Phi] P[\Phi].$$

In this case. the values of $P[L \,|\, \Phi]$ are known through the assessments made by the evaluators but the values of Φ are not. Assuming that there are I photographs $(\Phi_i, i = 1, 2, \ldots, I)$ and J labels $(L_j, j = 1, 2, \ldots, J)$ with respective probabilities ϕ_i and ℓ_j yields the following matrix representations:

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_I \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1J} \\ g_{21} & g_{22} & \cdots & g_{2J} \\ \vdots & \vdots & \vdots & \vdots \\ g_{I1} & g_{I2} & \cdots & g_{IJ} \end{bmatrix} \begin{bmatrix} \ell_1 \\ \ell_2 \\ \vdots \\ \ell_J \end{bmatrix},$$

where $g_{ij} = P[\Phi_i \,|\, L_j]$ and

$$\mathbf{L} = \begin{bmatrix} \ell_1 \\ \ell_2 \\ \vdots \\ \ell_J \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1J} \\ f_{21} & f_{22} & \cdots & f_{2J} \\ \vdots & \vdots & \vdots & \vdots \\ f_{I1} & f_{I2} & \cdots & f_{IJ} \end{bmatrix}^T \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_I \end{bmatrix},$$

where $f_{ij} = P[L_j \,|\, \Phi_i]$. Put another way, $\mathbf{L} = \mathbf{F^T\Phi}$ and $\mathbf{\Phi} = \mathbf{GL}$ where $\mathbf{F} = [f_{ij}]_{I\times J}$ and $\mathbf{G} = [g_{ij}]_{I\times J}$. It may be assumed here that the number of photographs is much larger than the number of labels ($I >> J$). This is sufficient to guarantee that there are infinitely many solutions ϕ_i for the second matrix equation given values for ℓ_j. The task is to find at least one solution that also satisfies $1 = \sum_i \phi_i$ and $0 \le \phi_i \le 1$.

Such a problem as this may be solved using techniques from the theory of Linear Programming. The focus will be on the second matrix equation that expresses the vector $\mathbf{L} = [\ell_1, \ell_2, \ldots, \ell_J]^T$ as the product of a constant matrix, $\mathbf{F^T}$, with the vector $\mathbf{\Phi} = [\phi_1, \phi_2, \ldots, \phi_I]^T$. The problem at hand is to determine $\mathbf{\Phi}$ given $\mathbf{L}$ subject to the constraints defined by the matrix inequality $\mathbf{F^T\Phi} \le \mathbf{L}$ and the restrictions

$$\begin{aligned} 0 &\le \phi_1 \le 1 \\ 0 &\le \phi_2 \le 1 \\ &\vdots \\ 0 &\le \phi_I \le 1 \\ 1 &= \sum_{1\le i\le I} \phi_i \end{aligned}$$

2.11 Linear Programming. A general linear program is a system of m equations (and inequalities) in n variables defined as follows (all vectors are column vectors):

Given vectors $\mathbf{c}, \mathbf{b}$ and a matrix $\mathbf{A}$, find a vector $\mathbf{x}$ such that

$\mathbf{c^T x}$ (or length n) is maximized subject to

$\mathbf{Ax} \le \mathbf{b}$ ($\mathbf{A}$ is m rows by n columns)

$\mathbf{x} \ge \mathbf{0}$ (x has length n)

$\mathbf{b}$ is of length m.

The expression $z = \mathbf{c^T x}$ is called the objective function while $\mathbf{Ax} \le \mathbf{b}$ and $\mathbf{x} \ge \mathbf{0}$ are called the constraints. The set of all vectors $\mathbf{x}$ that satisfy the constraints are called feasible solutions and form a shape known as a convex polyhedron – a k dimensional solid formed by the intersection of some number of $(k-1)$ dimensional hyperplanes)[4].

Of specific interest is the case where the resulting polyhedron, $\mathcal{P}$, is bounded (that is, $\mathcal{P}$ can be placed inside of some k dimensional hypercube). In this case $\mathcal{P}$ is called a polytope. Major consequences in this special case include:

1. A convex polytope contains a finite number of vertices
2. Every feasible solution can be written as a weighted linear combination of the vertices (the weights must be non-negative and sum to 1). In general this representation is not unique. An exception to this rule occurs when the feasible solution is also a vertex.

[4] Linear programming began to take shape in 1939. During the course of its development the names of its basic components got exchanged, changed. reused. etc. This has led to great confusion when reading books and papers written on this subject.

3. All feasible solutions that maximize the objective function are found among the vertices.

In this situation, another name for a vertex is an extreme point: $\mathbf{x}$ is an extreme point of the convex set $\mathcal{C}$ provided that there is a hyperplane $\mathcal{H}$ with the property $\mathcal{H} \cap \mathcal{C} = \{\mathbf{x}\}$. Moreover, in the linear programming case, a vertex can also be defined as a point $\mathbf{v}$ that satisfies some linearly independent subcollection of n of the $(m+n)$ equations: $\mathbf{Av} = \mathbf{b}$ together with $\mathbf{b} = \mathbf{0}$.

In symbols, if $\mathbf{x}$ is a feasible solution and $\mathbf{v}_j$ are the set of vertices, then $\mathbf{x} = \sum_j \lambda_j \mathbf{v}_j$ where $\lambda_j \geq 0$ and $\sum_j \lambda_j = 1$. The converse is also true: every weighted sum $\sum_j \lambda_j \mathbf{v}_j$ is a feasible solution. Therefore, knowledge of the vertices of the feasible region suffices to determine the maximum of the objective function. Note also that the objective function may also be expresx sed as a minimization problem since minimizing $z = \mathbf{c^T x}$ is the same as maximizing $z = -\mathbf{c^T x}$.

2.11.1 The problem at hand.

Note that the condition $\phi_i \leq 1$ is captured by the last inequality: $\sum_{1 \leq i \leq I} \phi_i \leq 1$. Certainly, if there exists a solution to the set of linear equations

$$\begin{aligned} \mathbf{F^T \Phi} &= \mathbf{L} \\ \sum_{1 \leq i \leq I} \phi_i &\leq 1 \\ -\phi_j &\leq 0 \qquad j = 1, 2, \ldots, I \end{aligned}$$

it will be found in the solution space of the linear program defined by replacing $\mathbf{F^T \Phi} = \mathbf{L}$ with $\mathbf{F^T \Phi} \leq \mathbf{L}$. The positivity of the coefficients of the $\mathbf{F^T}$ matrix guarantees that the origin is *below* the half planes defined by each of the rows of $\mathbf{F^T}$. The intersection of these half-spaces (the solution space of the linear program) defines a t dimensional polytope where $t \leq \min(I, J)$. If in addition the rank of the matrix $\mathbf{F^T}$ is $\min(I, J)$, then the dimension of the solution polytope (call the polytope $\mathcal{P}$) will be exactly $t = \min(I, J)$ and will lie inside the finite hypercube $\{0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1, \ldots, 0 \leq x_t \leq 1\}$.

In the sequel, a generic element of $\mathcal{P}$ will be denoted by $\mathbf{\Phi}$ and a vertex of $\mathcal{P}$ will be denoted by $\mathbf{v}$. In this linear program there are $I + J + 1$ constraint equations, I non-basis variables, and $I + J + 1$ slack variables. In the notation of a linear program:

$$\begin{aligned} \text{Maximize } & \mathbf{c^T x} \\ & \mathbf{Ax} \leq \mathbf{b} \end{aligned}$$

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} \mathbf{I}_J & \mathbf{0}_J^T & \mathbf{F^T}_{J \times I} \\ \mathbf{u}_J & 1 & \mathbf{u}_I \\ \mathbf{0}_J & \mathbf{0}_1 & -\mathbf{I}_I \end{bmatrix} \\ \mathbf{b} &= (b_1, b_2, \ldots, b_{I+J+1})^T \\ \mathbf{c} &= (c_1, c_2 \ldots, c_{I+J+1}), \end{aligned}$$

where $\mathbf{A}$ is the matrix of coefficients, $\mathbf{b}$ is a vector of constraint values, $\mathbf{c}$ is the objective (cost) vector, $\mathbf{0}_x$ is a zero vector (resp. matrix) of length x (resp. dimension $x \times x$), $\mathbf{u}_y$ is a vector of 1s of length y, and $\mathbf{I}_x$ is an $x \times x$ identity matrix. In what follows, the constraints $\phi_i \geq 0$ will be ignored until needed.

2.11.2 Ideal Case. The linear program is now given an objective function to maximize as follows. Let the residual vector $\mathbf{r}$ be defined by

$$\mathbf{r} = \mathbf{L} - \mathbf{F^T}\boldsymbol{\Phi} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_J \end{bmatrix},$$

where $r_j = \ell_j - f_{1j}\phi_1 - f_{2j}\phi_2 - \ldots - f_{Ij}\phi_I$ with $1 \leq j \leq J$ and ϕ_i the i-th coordinate of some vector $\boldsymbol{\Phi}$. The notation $r_j = r(\boldsymbol{\Phi}, j)$ is used when it is necessary to emphasize that r_j depends on both j and $\boldsymbol{\Phi}$. Note that by construction $0 \leq r_j \leq \ell_j$ for all feasable $\boldsymbol{\Phi}$.

Two objective functions to be maximized will be considered:

$$\begin{array}{ll} (1) & z = -r_1 - r_2 - \ldots - r_J \\ (2) & z = -r_j \qquad \text{for some } j. \end{array}$$

Because $r_j \geq 0$ for every j, the maximum possible value of z in case (1) is 0 and occurs if and only if $r_j = 0$ for all j. However the maximum value of z may not be 0 indicating that there are no solutions to the system of linear equations. But if the magnitude of the maximum z is small, then all of the r_j will be smaller indicating that there are points in $\mathcal{P}$ that are nearly solutions to the system of linear equations. This case is the primary focus of this effort.

Case (2) is used to maximize each coordinate of a solution individually. This may be a quick fallback position if the maximum value of $|z|$ produced by (1) is unacceptably large.

Note that the vertices of the polytope solution region do not depend on the objective function z under consideration. In this initial research, the maximum z will be found by a traverse of the vertices, evaluating z at each vertex, and comparing each value of z to the largest z seen so far. While this is not the most efficient solution method it can be used to produce the maximum of the object function z. It is of particular interest to note that the maximum value of z may not be zero. In the sequel, two techniques are proposed to handle the case $z < 0$.

2.11.3 Non-ideal $z < 0$: Coordinate Minimization – A Detour. Note: This technique is documented as it will lead to a certitude function for each coordinate. However, it does not necessarily lead to a set of solution vectors $\boldsymbol{\Phi}$ that satisfy all contraints simultaneously.

The focus here were be on the j-th row of the constraint $\mathbf{L} = \mathbf{F^T}\boldsymbol{\Phi}$. This constraint corresponds to label j. By changing the objective function to $z = -r_j$, a vertex traverse is still a valid way to find the vertices that minimize r_j (note that the underlying vertices do not have to be recomputed if j is changed). Denote this minimum value by $\ell_j^\star$:

$$\ell_j^\star = \min_{\Phi}\{r(\Phi, j)\}.$$

By running the vertex traverse to completion all vertices that yield the minimum value $\ell_j^\star$ can be determined. Note that the vertices that determine $\ell_j^\star$ will likely change as j changes.

2.11.4 Non-ideal $z < 0$: Supremum Norm Minimization. Here the supremum norm is discussed for the case $z < 0$. In the last section the norm used was generated separately for each individual label. In this section the supremum norm is used as it is a global measurement of the size of the difference between the vector $\mathbf{L}$ and $\mathbf{F^T\Phi}$ for feasible $\mathbf{\Phi}$.

Define the supremum norm of a vertex by $M(\mathbf{\Phi}) = ||\mathbf{L} - \mathbf{F^T\Phi}||_\infty$. In general, the supremum norm of a vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is given by $||\mathbf{x}||_\infty = \max_i\{|\, x_i\,|\}$. There will be a collection of vectors $\Phi_1^\star, \Phi_2^\star, \ldots, \Phi_K^\star$ for which $M(\mathbf{\Phi})$ is the smallest (call this smallest value $M^\star \geq 0$). These vertices are the vertices whose supremum norm distance from $\mathbf{L}$ is the smallest. Ideally $M^\star = 0$ and the collection $\Phi_K^\star$ is a set of vertices that satisfies all of the constraints of the linear program with equality.

More typically, $M^\star > 0$. Let the associated label vector collection be $\mathbf{L}_i^\star = \mathbf{F}^T\Phi_i^\star$. By the construction, all of these vectors $\mathbf{L}^\star$ satisfy $M^\star = ||\mathbf{L} - \mathbf{L}^\star||_\infty$ and there is at least one such vector $\mathbf{L}^\star$ in this collection[5].

2.11.5 Max Norm vs. Coordinate Minimization. The following propositions show the relationship between the two previously described measures.

Proposition 1. $\ell_j^\star \leq M^\star$ for all j.

Proof: For each fixed j

$$\begin{aligned}\ell_j^\star &= \min_\Phi\{r(\mathbf{\Phi}, j\} \\ &\leq r(\mathbf{\Phi}, j) \qquad \text{(for all } \mathbf{\Phi}) \\ &\leq \max_j\{r(\mathbf{\Phi}, j)\} \qquad \text{(for all } \mathbf{\Phi}) \\ &= M(\mathbf{\Phi}) \qquad \text{(for all } \mathbf{\Phi}).\end{aligned}$$

Since $\ell_j^\star \leq M(\mathbf{\Phi})$ for all $\mathbf{\Phi}$ it must be the case that $\ell_j^\star \leq M^\star$ for all j. ⟦

The conclusion to be drawn from proposition 1 is that if there is a set of vertices of the $\mathbf{L}$ polytope that cause $M^\star = 0$ then these vertices also cause $\ell_j^\star = 0$ for every j.

[5] A non-subscripted $L^\star$ indicates a generic representative of the set $\{L_i^\star\}$

.........This is wrong .. ignore........

Proposition 2. If $\ell_j^\star = 0$ for all j, then $M^\star = 0$.

Proof:

$$\begin{aligned} M^\star &= \min_\Phi \{M(\mathbf{\Phi})\} \\ &= \min_\Phi \max_j \{r(\mathbf{\Phi}, j)\} \\ &\leq \max_j \{r(\mathbf{\Phi}, j)\} \quad \text{for all } \mathbf{\Phi} \\ &= r(\mathbf{\Phi}, j') \quad \text{for all } \mathbf{\Phi} \text{ and some } j' \\ \text{NO: } j' \text{depends on } \mathbf{\Phi} >> \quad &\leq \min_\Phi \{r(\mathbf{\Phi}, j')\} \quad \text{and some } j' \\ &= \ell_{j'} \end{aligned}$$

If $\ell_j = 0$ for all j then $0 \leq M^\star \leq \ell_{j'} = 0$. ▯

The example shown below in section 2.11.6 has every $\ell_j^\star = 0$ for some $\mathbf{\Phi}$ but $M^\star > 0$.

Proposition 3. Suppose that $f(\mathbf{x}) = \sum_{i=1}^{n} c_i x_i - k = 0$ is the equation of an affine hyperplane in n dimensional space. The minimum distance between this hyperplane and the point $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ is given by

$$\frac{f(\mathbf{a})}{||\mathbf{c}||} = \frac{\mathbf{a} \cdot \mathbf{c} - k}{||\mathbf{c}||},$$

where $\mathbf{c} = (c_1, c_2, \ldots, c_n)$ is a vector normal to the plane defined by $f(\mathbf{x})$.

Proof: Let $\mathbf{u} = \mathbf{c}/||\mathbf{c}||$ be a unit vector pointing in the direction of the vector $\mathbf{c}$ normal to the plane. Assume that the tip of $\mathbf{u}$ and $\mathbf{p}$ lie on the same side of $f(\mathbf{x}) = 0$ (replace $\mathbf{u}$ by $-\mathbf{u}$ otherwise). If λ is the magnitude of the minimum distance between the point $\mathbf{a}$ and the plane, then the point $\mathbf{p}$ in the plane closest to $\mathbf{a}$ satisfies $\mathbf{p} + \lambda \mathbf{u} = \mathbf{a}$. The coordinates of $\mathbf{p}$ are

$$p_i = a_i - \lambda u_i = a_i - \lambda \frac{c_i}{||\mathbf{c}||}$$

and since $\mathbf{p}$ lies in the plane normal to $\mathbf{c}$

$$\sum_i c_i \big(a_i - \lambda \frac{c_i}{||\mathbf{c}||}\big) = k.$$

This reduces to $\sum_i a_i c_i - \lambda ||\mathbf{c}|| = k$. Solving for λ yields the result[6]. ▯

In the case that $\mathbf{a} = (0, 0, \ldots, 0)$ the distance between the plane and the origin becomes $|\, k\,| / ||\mathbf{c}||$. The consequence of this is that changing the constant k by a factor of $d > 0$ moves the plane closer to $(d < 1)$ or farther away from $(d > 1)$ the origin through a distance proportional to d.

[6] Replacing $\mathbf{u}$ with $-\mathbf{u}$ yields the result with λ replaced by $-\lambda$

Proposition 4. Let $\mathcal{P}$ be the convex region defined by the linear program $\mathbf{F^T\Phi} \leq \mathbf{L}$ with vertices $\{\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_J}\}$. Let $M^\star = \min_{\mathbf{v}\in\mathcal{P}}\{||\mathbf{L} - \mathbf{F^T v}||_\infty\}$ and let $\mathbf{v}^\star$ be any vector satisfying $M^\star = ||\mathbf{L} - \mathbf{F^T v}^\star||_\infty$. Finally, let $\mathbf{L}^\star = \mathbf{F^T v}^\star$. Then, $\mathbf{v}^\star$ is a vertex vector of the convex space $\mathbf{F^T\Phi} \leq \mathbf{L}^\star$ and satisfies $\mathbf{F^T v}^\star = \mathbf{L}^\star$.

Proof: It is clear from the construction that $\mathbf{L}^\star \leq \mathbf{L}$. Let $\mathcal{P}^\star$ be the convex region defined by the linear program $\mathbf{F^T\Phi} \leq \mathbf{L}^\star$. Because $\mathbf{v}^\star \in \mathcal{P}$ maximizes the objective function $z = -\sum r^\star$ relative to $\mathcal{P}^\star$ (the vector $\mathbf{v}^\star$ is feasable and produces the value $z = 0$), the vector $\mathbf{v}^\star$ must be a vertex of $\mathcal{P}^\star$ [???] satisfying $\mathbf{F^T v}^\star = \mathbf{L}^\star$. ⟦

Resolution. Combining proposition 1, 3, and 4 together leads to the following disovery. If the original constraint equations $\mathbf{F^T\Phi} \leq \mathbf{L}$ are replaced by $\mathbf{F^T\Phi} \leq \mathbf{L}^\star$, where $\mathbf{L}^\star$ is a vector associated with the value $M^\star$, there will be at least one vertex of $\mathcal{P}^\star$ that forces $r_j^\star = 0$ for every j.

Note that replacing $\mathbf{L}$ with $\mathbf{L}^\star$ has moved the j-th hyperplane closer to the origin by a factor of $(1 - \ell_j^\star/\ell_j)$, a value between 0 and 1. The new constraint values $\ell_j^\star$ may be viewed as corrections to the NN supplied values ℓ_j. This results in certitude functions for each of the labels and a correction vector for the vector of values supplied by the neural network.

Question 1. Is $\mathbf{v}^\star$ unique?

Question 2. If $\mathbf{v}^\star$ is not unique, do all such candidates produce the same $\mathbf{L}^\star$?

Question 3. If $\mathbf{v}^\star$ is not unique, and not all candidates produce the same $\mathbf{L}^\star$, is there an optimal choice for $\mathbf{v}^\star$? Is there a way to use all $\mathbf{v}^\star$ in the certitude function construction?

Question 4. Making this adjustment to the NN supplied values breaks the constraint $\sum_j \ell_j = 1$. Could another label, called L_{other}, be invented to hold the residual probability $(1 - \sum_j \ell_j^\star)$?

2.11.6 $z < 0$ **: Objective Function Minimization.** To handle this case, it has been observed that the vector $\mathbf{L} - \mathbf{L}^\star$ typically has small coordinates because

$$||\mathbf{L} - \mathbf{L}^\star||_\infty = ||\mathbf{L} - \mathbf{F^T\Phi} + \mathbf{F^T\Phi} - \mathbf{L}^\star||_\infty \leq ||\mathbf{L} - \mathbf{F^T\Phi}||_\infty + ||\mathbf{F^T\Phi} - \mathbf{L}^\star||_\infty = M^\star + 0.$$

From Proposition 4, there are vectors $\mathbf{\Phi}$ that are feasible solutions to $\mathbf{F^T\Phi} \leq \mathbf{L}^\star$ that also satisfy $\mathbf{F^T\Phi} = \mathbf{L}^\star$. Therefore, there must also be vertices of the solution space polytope that also satisfy $\mathbf{F^T\Phi} = \mathbf{L}^\star$ and possess the same $M^\star$ value. Let $\mathbf{\Phi_1}, \mathbf{\Phi_2}, \ldots, \mathbf{\Phi_k}$ be these vertex solutions. That is, if $\mathbf{v_i}$ represents the complete set of vertices of the polytope solution space, then

$$M^\star = ||\mathbf{F^T\Phi_i} - \mathbf{L}^\star||_\infty = ||\mathbf{F^T\Phi_j} - \mathbf{L}^\star||_\infty \leq ||\mathbf{F^T v_k} - \mathbf{L}^\star||_\infty$$

for all i, j, k in their respective ranges.

Note that any convex combinations of the $\mathbf{\Phi_i}$ must also satisfy the equation $\mathbf{F^T\Phi} = \mathbf{L}^\star$. A lingering question if how to choose the *best* collection of these vertices.

2.12 LP Computations. In creating a computational algorithm for producing the certitude functions, three main techniques are used:

A. The Simplex Method

B. A vertex traversal algorithm (see [7])

C. Histogram construction.

2.12.1 The Simplex Method. The Avis algorithm ([7]) requires a data dictionary be established by following any of the various techniques of linear programming. The simplex method is used the create a data dictionary for the objective function $z = -\sum_j \phi_j$. Avis's algorithm also requires knowledge of at least one vertex of the polytope solution space. In this case, it is clear that the zero vector is a vertex.

2.12.2 Vertex Traversal. The data dictionary is now manipulated by Avis's algorithm to produce a non-repeating traversal of all of the vertices, $\mathbf{v}_j$, of the convex solution set $\mathcal{P}$. In practice, the transversal will have repeating vertices unless infinite precision arithmetic is used.

This is hard to do in a computer but can be approximated by starting the linear program using rational values. That is, the elements of $\mathbf{F^T}, \mathbf{L}, \mathbf{L}^\star$ and so on will be represented by rational numbers $\frac{a}{b}$ where a and b are integers. This will work until the approximation of a real number, r, by a rational number $\frac{a}{b}$ requires a value for b that exceeds the maximum integer of the computer used.

Note that routines will be needed to perform the fractional arithmetic with respect to addition, subtraction, multiplication, and division (quotients and remainders). A routine for finding the greatest common divisor of two integers will most likely also be needed.

Vertex traversal is a promising approach because of two key facts:

(1) every point of $\mathcal{P}$ is a finite, convex combination of the vertices of $\mathcal{P}$, and

(2) the vertices of $\mathcal{P}$ are not dependent on the objective function, only on the constraints.

Upon completion of the vertex traversal a subcollection of the vertices is identified that produces the value $M^\star$. If $M^\star = 0$ the traversal is completed and the histogram construction phase may begin. If $M^\star > 0$ the NN supplied values of $\mathbf{L}$ are adjusted to form $\mathbf{L}^\star$ and steps 1 and 2 are repeated. After this, the adjusted value of $M^\star = 0$ and histogram construction begins.

2.12.3 Histogram Construction. Suppose that a probabilistic value for each image has been determined. That is, $\phi_i = P[\text{Image } i \text{ exhibits some of the damage represented by the labels}]$ is known. It is therefore theoretically possible to know the probability distributions of each of the labels but this is a difficult problem complicated by the dependencies that exist between the elements of the $\mathbf{F}^T$ matrix – the same expert may have contributed opinions on several (image,label) pairs. To overcome this analytical encumbrance an approximation can be generated through the use of a histogram simulation.

Simulation method: *a Priori*. If $\mathbf{\Phi}$ is given, randomly select values, from their respective distribution, for the elements of $\mathbf{F^T}$ and $\mathbf{\Phi}$ and then generate label certitude vectors $\mathbf{L}$ via $\mathbf{L} = \mathbf{F^T}\mathbf{\Phi}$ (one

Ł vector for each simulation pass). Perform a large number of simulation passes. Finally, construct a histogram for each $\mathbf{L}$ vector coordinate.

Simulation method: ***a Posteriori*.** If fixed values for $\mathbf{L}$ are given replace $\mathbf{F^T}$ by its mean matrix $E[\mathbf{F^T}]$. Determine the vertices $\mathbf{v}_i$ of the solution polytope and form a feasible solution to the linear system $\mathbf{L} = \mathbf{F^T}\mathbf{\Phi}$ by randomly, uniformly selecting weights $\lambda_j > 0 \quad \sum_j \lambda_j = 1$ to build a candidate value for $\mathbf{\Phi} = \sum_j \lambda_j \mathbf{v_j}$ Now use this $\mathbf{\Phi}$ as described in the *a Priori* method to produce an observed $\mathbf{L}$: at each simulation pass, both the matrix $\mathbf{F^T}$ and the candidate $\mathbf{\Phi}$ are regenerated. Use these observations to build a vector histogram for $\mathbf{L}$.

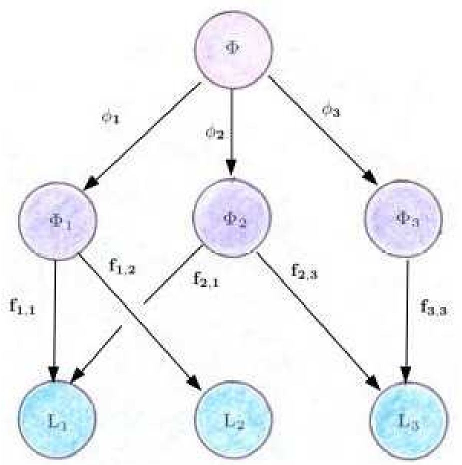
Ranking the Images. Here are some metrics with which to rank the contribution that each image makes in the construction of the label histograms.

- $\big(\max_{\mathbf{\Phi}}\{\phi_1\}, \max_{\mathbf{\Phi}}\{\phi_2\}, \ldots, \max_{\mathbf{\Phi}}\{\phi_I\}, \big)$
- Rank by the magnitudes of the centroid vector.
- Histogram the coordinates of the $\mathbf{\Phi}_i$. Rank my peak of density, mean of density, etc.

2.12.4 An Example. The following example will be used to illustrate the computations that are used in generating certitude functions. Consider this belief network topology (figure 7) coupled with the certitude functions developed from the evaluations of two experts (figure 8). The values assigned to each link were the ones presented at the CARLA conference:

Labels $\ell_1 = 0.70$ $\ell_2 = 0.26$ $\ell_3 = 0.04$

Expert	Image	Label	Left	Center	Right
1	1	1	0.026	0.06	0.094
1	1	2	0.208	0.48	0.752
1	2	3	0.026	0.06	0.094
1	3	3	0.208	0.48	0.752
2	1	1	0.9140625	0.921875	0.9296875
2	1	2	0.6328158	0.640625	0.6484375
2	2	1	0.9140625	0.921875	0.9296875

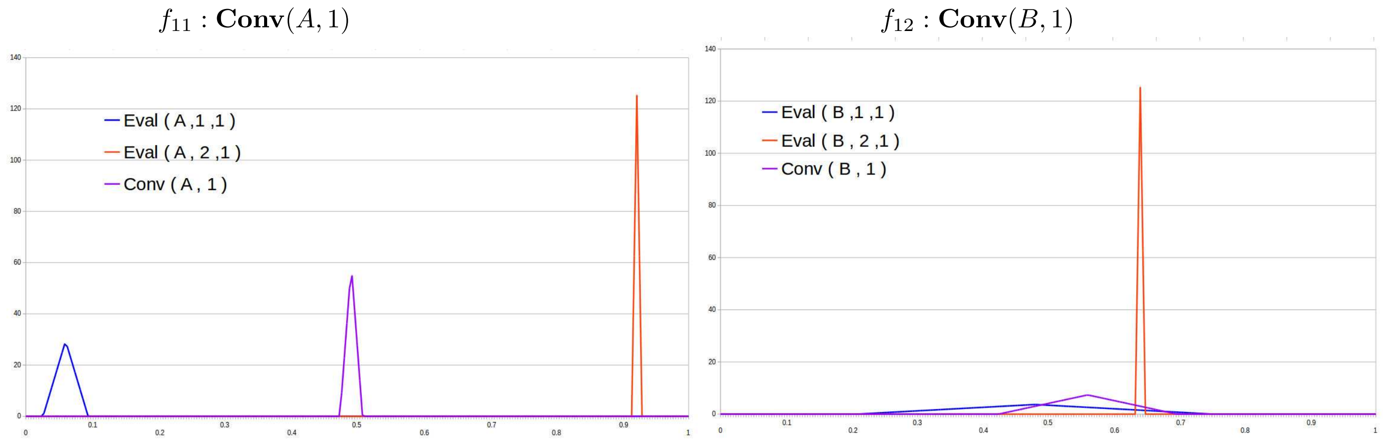


$$\begin{array}{llll} f_{11}: & \Phi_1 \to L_1 & \Leftrightarrow & \text{Conv}(A, 1) \\ f_{12}: & \Phi_1 \to L_2 & \Leftrightarrow & \text{Conv}(B, 1) \\ f_{21}: & \Phi_2 \to L_1 & \Leftrightarrow & \text{Eval}(D, 2, 2) \\ f_{23}: & \Phi_2 \to L_3 & \Leftrightarrow & \text{Eval}(C, 1, 1) \\ f_{33}: & \Phi_3 \to L_3 & \Leftrightarrow & \text{Eval}(D, 1, 2) \end{array}$$

$$\phi_1 = 6/15 \quad \phi_2 = 6/15 \quad \phi_3 = 3/15$$

Figure 7: Example Belief Network Topology

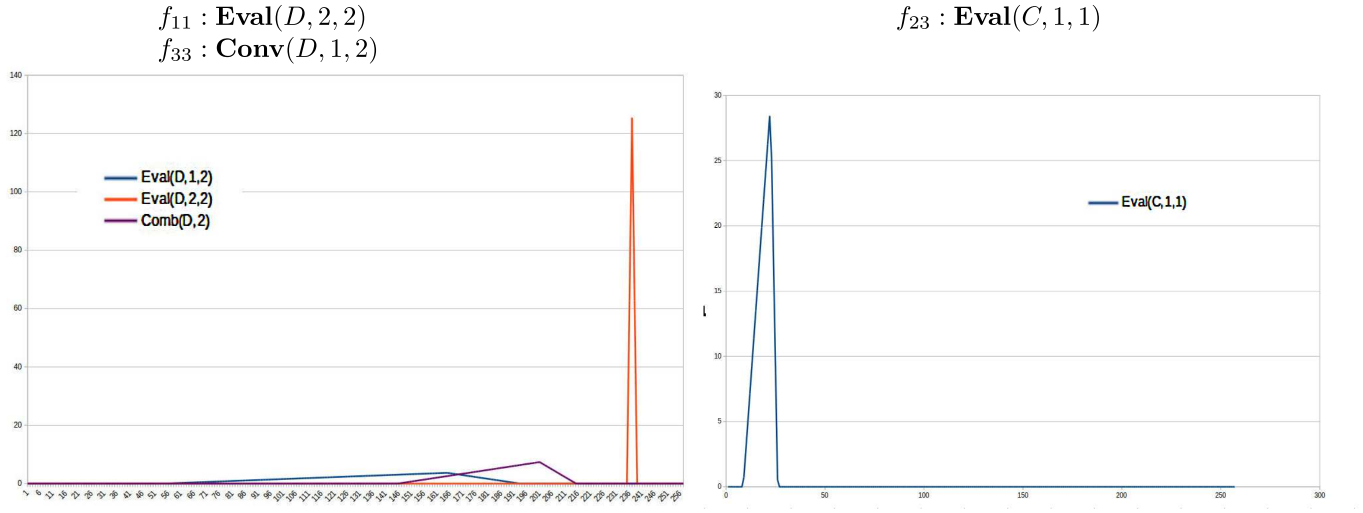
The notation `Eval(A , 2 , 1)` means that image A was evaluated by expert 2 with regards to label 1. The notation `Conv(A , 1)` represents the combining (through convolution) of the evaluation of image A with regard to label 1 by the two experts. The certitude functions for the five links shown in figure 7 are illustrated below.



f_{11} : $\mathbf{Conv}(A, 1)$

f_{12} : $\mathbf{Conv}(B, 1)$

Figure 8a: Links 11 and 12



f_{11} : $\mathbf{Eval}(D, 2, 2)$
f_{33} : $\mathbf{Conv}(D, 1, 2)$

f_{23} : $\mathbf{Eval}(C, 1, 1)$

Figure 8b: Links 21, 23, and 33

2.12.5 A Priori Evaluations. If the probabilities that measure the similarity between the target image and the truth set images, $\phi_i = P[\Phi \to \Phi_i]$, are known then a simulation experiment may be conducted to produce an estimate of the certitude functions of each label L_j. The photograph Φ is assumed to be a given upon which all results are conditioned. Consequently the image certitude functions (if any) and the label certitude functions all have the form $h(\cdot)P[\Phi]$ for some function $h(\cdot)$ specific to each image or label[7]. Therefore, it will be assumed that $P[\Phi] = 1$. When probabilistic information about Φ itself becomes available (that is when $P[\Phi]$ is known), all results given below may be adjusted simply by multiplying each result by $P[\Phi]$ to obtain unconditioned certitude functions.

[7] It is reasonable to assume that the images and labels of the truth set are independent of the target image Φ.

To produce an estimate of the certitude functions for the labels the following histogram construction algorithm is employed to construct a raw histogram. The notation ϕ_i is the certitude function of the image Φ_i, the notation $f_{i,j}$ is the certitude function of the link $\Phi_i \to L_j$, and the notation $\ell_j = \sum_i f_{i,j}\phi_i$ is the certitude function of the label L_j. In this example, it is assumed that the values ϕ_1, ϕ_2, and ϕ_3 are the constants $6/15 = 0.4,\ 6/15 = 0.4$, and $3/15 = 0.2$ respectively.

The goal of this algorithm is to create a vector of sample values $\underline{\ell} = (\ell_1, \ell_2, \ldots, \ell_J)$. To perform this task, a stand-in value is chosen for $f_{i,j}$ by choosing a random value, u, from a continuous uniform distribution over $[0, 1]$ and then computing $F_{i,j}^{-1}(u)$, where F is the distribution function of the variable $f_{i,j}$, see note[8]. A similar stand-in is created for the variable ϕ_i. These stand-ins are multiplied together and then summed over all photos, ϕ_i, to form a sample value for ℓ_j.

Algorithm 1

```
For each label $L_j$ repeat:
  Set $\ell_j \leftarrow 0$.
  For each $\mathbf{\Phi}_i$:
      Transform image certitude $\phi_i$ into a distribution function, $G_i$ for the node $\Phi_i$.
      Transform link certitude $f_{i,j}$ into a distribution function, $F_{i,j}$ for the link $\Phi_i \to L_j$.
      Choose $u$ and $v$ from CU$[0, 1]$
      $\ell_j \leftarrow \ell_j + F_{i,j}^{-1}(u)G_i^{-1}(v)$
  End loop $\mathbf{\Phi}_i$
End repeat $L_j$
Create $\underline{\ell} = (\ell_1, \ell_2, \ldots, \ell_J)$
```

Notes.

1. The expression CU$[0, 1]$ is the continuous uniform distribution over the interval $[0, 1]$.
2. If the certitude function ϕ is a certainty, use this constant value in place of $G_i^{-1}(v)$.
3. If the certitude function $f_{i,j}$ is a certainty, use this constant value in place of $F_{i,j}^{-1}(u)$.

The vector $\underline{\ell}$ determined by this algorithm forms a single group observation of the random processes defined for each label, L_j. Repeating this basic algorithm N times builds up a sample of N vector observations, $\underline{\ell_1}, \underline{\ell_2}, \ldots, \underline{\ell_N}$ from which raw histograms for each label $(L_1, L_2, \ldots, L_J)$, may be constructed. Scale each coordinate raw histogram so that the area under each histogram is approximately one. These normalized histograms are good approximations to label L_j's certitude function provided N is large.

For the example belief network discussed above, the certitude functions shown below (figure 9) were calculated following the *a Priori* procedure. The three triangularly shaped curves are estimates[9] for the certitude functions corresponding to the three labels given the constant image probabilities and the link certitudes. A check of the theoretical mean values yields $E[\ell_1] = 0.565$, $E[\ell_2] = 0.224$, $E[\ell_3] = 0.120$. Note that the chart shows the approximate locations of these mean values as 0.55, 0.23, and 0.13 respectively.

[8] Let $Y = F_X^{-1}(U)$. Then $F_Y(y) = P[Y \le y] = P[F_X^{-1}(U) \le y] = P[U \le F_X(y)] = F_X(y)$.

[9] Causality flows from the images to the labels.

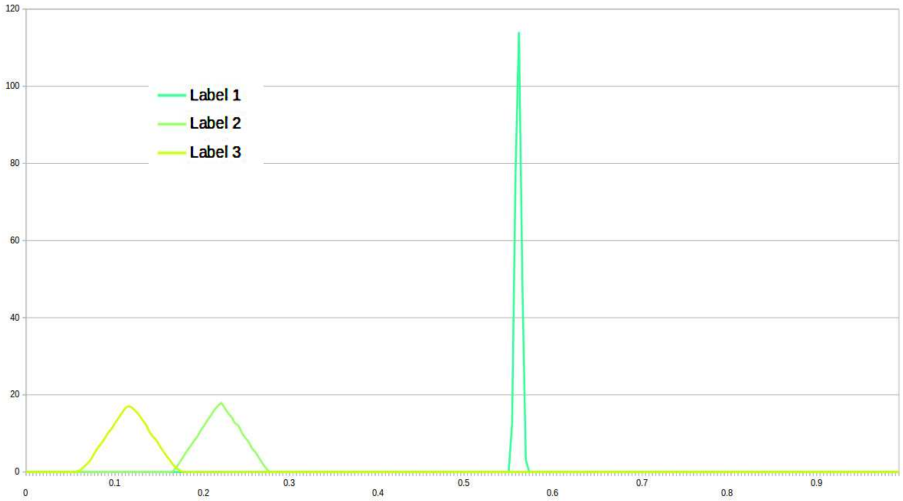


Figure 9: BN Generated a priori Certitude Functions

This analysis assumes that the NN provides probabilistic estimates for the similarity between the target photo and the images in the truth set. However, it may be (as in this application) that the starting point is the NN's estimation of the label probabilities ℓ_j and not the image probabilities ϕ_i.

2.12.6 Acceleration. When the size of $\mathbf{F^T}$ becomes large, the software may begin to slow down dramatically as the number of vertices discovered becomes larger and larger (the number of vertices exhibit exponential growth in I and J). However, a speed up may be achieved with the realization that only a small number of the labels will be assigned will be assigned non-zero label values by the neural network. In practice, all label probabilities assigned by the neural network that are below some small threshold should be set to zero as some neural networks will assigned positive values to all of the labels, making no reduction possible.

If label L_i receives a value of zero from the neural network ($\ell_i = P[L_i] = 0$), row i of $\mathbf{F^T}$ may be eliminated since $\ell_i = 0$ forces the values of ϕ_j that correspond to non-zero elements of the i row of $\mathbf{F^T}$ to be zero. This may cause inconsistencies in the solution to the linear program (for example, it is conceivable that if sufficient number of ℓ_i are zero that all values of ϕ_j may end up being zero). Both of these possibilities will be explored in the sequel:

1. elimination of rows of $\mathbf{F^T}$ corrresponding to $\ell_i = 0$, and
2. elimination of ϕ_j that do not participate heavily in *a priori/a posteriori* evaluations (that is, using only the dominant ϕs).

2.12.6.1 Acceleration Results – label elimination. Consider the linear program defined in section 2.11.1. If there is a $b_j = 0$,then it must be the case that $0 = b_j = \sum_i a_{ji} x_i$. Since all of the $a_{ji} \geq 0$ and all $x_i \geq 0$ it must be that all $x_j = 0$. That is equation j identifies the origin as a vertex. But so do all of the positivity constraints. Consequently, elimination of the $b_j = 0$ row in the matrix will have no effect on the determination of the vertices but will reduce the computational load present in the problem.

Here is a table that shows the effect of removing equations corresponding to $b_j = 0$:

Case	**Unfiltered**	**Filtered**
2F	9	1
3F	21	3
4F	170	1
4F-a	164	1
5F	5410	350
6F	???	299

??? indicates that the computer program did not finished after 2 hours of execution.

2.12.6.2 Acceleration Results – Photo elimination. Consider the linear program defined in section 2.11.1. If some column of $\mathbf{F}^T$ equals $(0, 0, 0, \ldots, 0)^T$ then perhaps elimination is possible ???

2.12.7 A Posteriori Evaluations. In this application the neural network processes a target photograph and produces a numerical estimate for the vector $\mathbf{L} = [\ell_1, \ell_2, \ldots, \ell_J]^T$, a constant vector. The task here is to work backwards and reconstruct a feasible vector $\mathbf{\Phi} = (\phi_1, \phi_2, \ldots, \phi_I)$ of image probabilities that then can be propagated forward through the belief network to produce a set of certainty functions for the vector $\mathbf{L}$ of labels. It will be assumed that the randomness to be attached to the vector $\mathbf{L}$ comes from the uncertainty in the expert's evaluations and not from any inherited randomness in the solution vector $\mathbf{\Phi}$.

Because the number of photographs is much larger than the number of labels ($I >> J$) it is guaranteed (barring inconsistencies in the equations) that there will be infinitely many vectors $\mathbf{\Phi}$ that satisfy the linear program for a given vector $\mathbf{L}$. The sum of the ϕ_i is permitted to be strictly smaller than one to allow for the fact that the image collection is likely incomplete. This condition is modeled by postulating the existence of an image Φ_{other} to hold the remaining probability. As there are no links connecting Φ_{other} to any labels, this image may be ignored in the computations. Moreover, only those labels identified by the NN as having a non-zero likelihood need be considered in this analysis.

<<**this paragraph subject to deletion**>> At this point, assume that the values supplied by the neural network, $\ell_j(\text{NN})$, represent the mean values of the label certitude functions: $\ell_j(\text{NN}) = E[L_j]$. The linear program can be simplified by applying the expectation operator to the equation $\mathbf{L} = \mathbf{F^T\Phi}$ to produce $E[\mathbf{L}] = E[\mathbf{F}^T]\mathbf{\Phi}$, where $E[\mathbf{F}^T] = \left[E[f_{j.i}]\right]_{J\times I}$. **This is what is being used in the current software . . . if $\mathbf{\Phi}$ is constant it is true but otherwise ???** This linear system will be used to construct a polytope solution space $\mathcal{P}$ relating the vectors $\mathbf{L} = [\ell_j(\text{NN})]^T$ to $\mathbf{\Phi}$. The certitude functions associated with the $\mathbf{\Phi}$ are assumed to be certainties (**constants . . . bad assumption? In what follows F is replaced by $E[\mathbf{F}]$ . . . this needs justification**).

A vertex traversal is performed next to locate the vertices $\mathbf{v}_k$ of the convex polytope solution space that satisfy the constraint equations exactly for label $\mathbf{L}_j$. As each vertex is examined, the minimum value of the vertex supremum norm is kept and the vertices $\mathbf{v}'_k$ that possess the minimum norm value are saved. Ideally, at the end of this pass through the vertex set, the minimum supremum norm is zero. If this happens, then any vector created by forming a convex combination of the $\mathbf{v}'_k$ will also exhibit a minimum supremum norm of zero.

The collection $\mathbf{v}'_k$ of vertices then are used as the basis for the histogram construction for all of the involved labels. Algorithm 2 describes the procedures for constructing this histogram.

Algorithm 2

```
$M^\star \leftarrow 2$;    $\mathbf{v}' \leftarrow 0$;    $\{\mathbf{v}'\}$ is empty;
Begin vertex walk
  Repeat
    Generate next vertex $\mathbf{v}'$
    Compute vector sample $\mathbf{L} \leftarrow \mathbf{F}^\mathbf{T}\mathbf{v}'$; find $M^\star$
    If $M^\star$ strictly decreased, empty $\{\mathbf{v}'\}$; remember new minimum $M^\star$ value, restart collection with $\mathbf{v}'$
    If $M^\star$ did not change, add $\mathbf{v}'$ to $\{\mathbf{v}'\}$ collection
    If $M^\star$ stricty increased, do nothing
    Vertex walk to next $\mathbf{v}'$ vertex
  Until $\mathbf{v}' = 0$
  Choose a large integer $N$.
  Transform link certitudes $f_{i,j}$ into a distribution functions, $F_{i,j}$ for each link $\Phi_i \rightarrow L_j$.
  Repeat $N$ number of times
    Select the values of the weights $\lambda_k$ uniformly, randomly: $\sum \lambda_k = 1, \lambda_k \geq 0$.
    Form a solution vector $\tilde{\mathbf{\Phi}} = \sum_{\{\mathbf{v}'\}} \lambda_k \mathbf{v}'$
    Make random selections for each $f_{i,j}$ based on certitude functions
    Compute observed $\tilde{\mathbf{L}} = \mathbf{F}^\mathbf{T}\tilde{\mathbf{\Phi}}$ and add to histogram
  End Repeat
Normalize the resulting raw histogram to unit area.
```

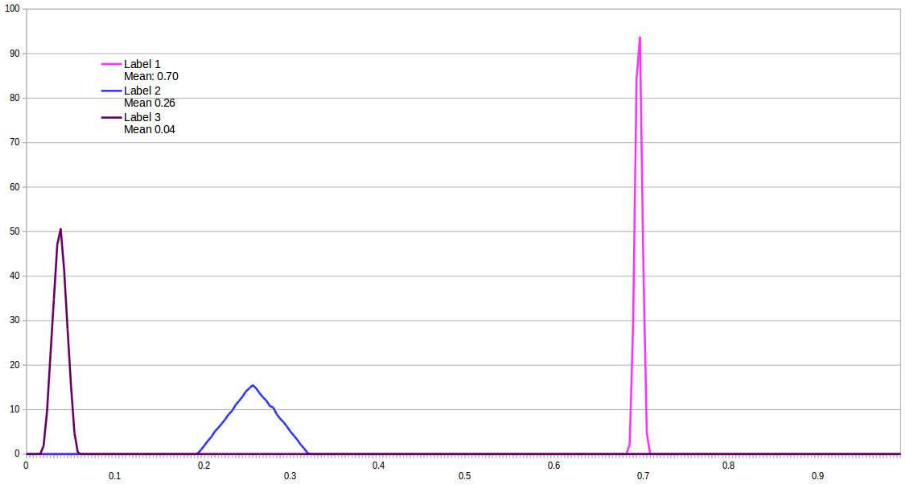


Figure 10: BN Generated a posteriori Certitude Functions

Here is the example run with a NN supplied label values, $\ell_j(\text{NN})$, of 0.75, 0.26, 0.04 for labels 1, 2, and 3 respectively. The simulation created 100,000 random solutions to the linear program all satisfying the constraint equations exactly $\mathbf{L} = \mathbf{F}^\mathbf{T}\mathbf{\Phi}$. It should be noted that with these choices for the NN output, the supremum norm was 0 as were all minimum coordinate norms, $\ell^\star$.

In the course of generating the previous certitude functions It was noted that when the NN supplied values were changed to 0.75, 0.25, 0.05 the supremum norm was not zero ($M^\star > 0$). The minimum supremum norm in this case is 0.003473 and was produced by a single vector $\mathbf{v}' = [0, 0, 0.003473]^T$. When the NN suplied probabilities are replaced with the values 0.75, 0.25, and 0.046526923 and the algorithms run again, the resulting supremum norm is 1.388×10^{-17}.

3 New Ideas to Explore – avoid reading

3.1 Multiple Experts Used to Rate a Single Expert – Under Development. Continuing with the measurement device analogy of the previous section, view the collection of assessments A_i as being a set of K independent and identically distributed (iid) random variables. Each assessment A_i consists of the selection of a valuation from a discrete list of values (certain, probably, etc.) according to some innate and unknown distribution function, F_A. If the number of measurements of the (Φ_s, L_j) pair is large, the central limit theorem may be invoked to assert that

$$\overline{A} = \frac{1}{K} \sum_{i=1}^{K} A_i$$

converges in distribution to a normal with a mean and variance equal to the mean and variance of F_A. It is known that then the value of $\overline{A}$ is a good estimate of μ_A and that S_A^2 is a good estimate of σ_A^2.

Some sort of distribution must be assigned to A. Suppose, arbitrarily and capriciously, the binomial distribution is elected to serve in this capacity. Then the binomial parameters n, p may be estimated from the equation

$$\begin{aligned} \mu_A &= np \\ \sigma_A^2 &= np(1-p), \end{aligned}$$

to produce values for the parameters n and p (rounding to the nearest integer will undoubtedly be required for the parameter n). If expert E_i has provided an evaluation, A_i, then the metric

$$\widetilde{p}_i = P[A = A_i]$$

might be a good substitute for the procedure described in section **1.4**.

4 References

[1] `www-compsci.swan.ac.uk\∼csphil\CS345\chapts5-9.pdf`

[2] Hackerman, David, *The Certainty-Factor Model*, Encyclopedia of Artificial Intelligence, Second Edition, Wiley, New York, pp. 131-138.

[3] Pearl, Judea, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Mateo, CA

[4] Klir, George J.(editor), Yuan, Bo(editor), *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems – Selected Papers by Lotfi A. Zadeh*, Advances in Fuzzy Systems – Applications and Theory, Vol 6, World Scientific.

[5] Knuth, D. E., The Art of Computer Programming, Vol 2, Section 4.3.3, pp 290-295.

[6] Press, W. H., et. al., Numerical Recipes in C, Section 8.10, pp 329-343.

[7] Avis, David and Fukuda, Komei, *A Pivoting Algorithm for Convex Hulls and Vertex Enumeration of Arrangements and Polyhedra*, Discrete & Computational Geometry, 1992, Sep 1, Vol. 8, Number 3, pp 295–313.