

# HAC-models: a Novel Approach to Continuous Speech Recognition

*Hugo Van hamme*

Katholieke Universiteit Leuven, dept. ESAT, Belgium

Hugo.vanhamme@esat.kuleuven.be

## Abstract

In this paper, a bottom-up, activation-based paradigm for continuous speech recognition is described. Speech is described by co-occurrence statistics of acoustic events over an analysis window of variable length, leading to a vectorial representation of high but fixed dimension called “Histogram of Acoustic Co-occurrence” (HAC). During training, recurring acoustic patterns are discovered and associated to words through non-negative matrix factorisation. During testing, word activations are computed from the HAC-representation and their time of occurrence is estimated. Hence, words in a continuous utterance can be detected, ordered and located.

**Index Terms:** speech recognition, information discovery, non-negative matrix factorisation, co-occurrence statistics

## 1. Introduction

Hidden Markov Models (HMMs) have dominated automatic speech recognition (ASR) research for many decades. They have shown to be fairly adequate models for speech, but recent research has also uncovered some of their weaknesses such as poor robustness to pronunciation variation, co-articulation, assimilation and noise. Psycholinguistic studies on human speech recognition (HSR) [1] have also revealed differences with the way current ASR systems work. To deal with the intrinsic variation that is found in speech, ASR systems are based on statistical models that are learned from examples. However, the model structure is still hardwired and engineered: sentences are built of words, which are built of phonemes, which map to different allophones, which have statistical models. One of the goals of the ACORNS project [2] is to discover the structure in speech from data, much like a baby does not need linguistic theories to understand language. Although still outperformed by HMMs, this contribution presents encouraging results along this line.

The HAC-model (histogram of acoustic co-occurrence) with its associated learning algorithm based on non-negative matrix factorisation (NMF) [3] is able to discover recurring acoustic patterns in speech without supervision [4] or with weak supervision [5]. In the former case, the algorithm will identify which acoustic patterns reappear and therewith find the elements or latent structures that speech is composed of, and this without any guidance. With weak supervision, the utterances are accompanied by unordered information that relates to the spoken words, much like a baby receives e.g. visual and tactile information that relates to the audio. This is a weaker form of supervision than is used in the training of HMM-based ASR systems, where utterances are described hierarchically in terms of word sequences down to HMM state sequences.

Once the HAC-models are trained, they can be used to decompose utterances in terms of the discovered latent structures, which will be words in this paper, but could also be phone-like units. This is a bottom-up process: acoustic

inputs activate words which compete, much like in the Shortlist model [6] of HSR. (There are differences with this model such as the absence of a pre-lexical level and “possible word constraints”). In this paper, HAC-models are extended to not only detect *which* words an utterance is composed of, but also in *which order* the words occur.

This paper is organized as follows: section 2 recaptures the ideas behind HAC-models, showing how words can be detected. In section 3, it is explained how the detected words can be ordered in time. Section 4 expands on the idea to build an activation-driven decoder. In section 5, the HAC-model is discussed and related to human speech recognition.

## 2. HAC-models

### 2.1. Histograms of acoustic co-occurrence

In the present bottom-up approach, recognition is driven by the co-occurrence of acoustic events. In general, these events are the occurrence of specific patterns in the time-frequency plane, ranging from patterns with a local time and frequency extent, such as the *patches* described in [7], to fullband patterns that span several 100ms like phones or even syllable-sized units. Specifically, in previous work [5], phones as well as vector-quantised (VQ) fullband spectra were used. The acoustic events are represented by discrete symbols from a set  $\Sigma$ , such as phone identities or a VQ labels, their time of occurrence and an estimate of their posterior probability. In [4], acoustic events are represented as the edges in a directed acyclic graph, because it is then easy to define the distance or *lag*  $\tau$  between edges  $\alpha$  and  $\beta$  as the minimal number of vertices that need to be visited to travel from  $\alpha$  to  $\beta$ , as well as the joint weight  $p_{\alpha\beta}$  of the edge pair  $(\alpha,\beta)$  as the sum of the posterior probabilities of partial paths starting with edge  $\alpha$  and ending with edge  $\beta$ . An alternative *lag* measure could simply be the difference in time of occurrence of the acoustic events, which would eliminate the need to represent them in a lattice. However, for notational convenience, the lattice will be maintained below.

Next, all pairs of acoustic events  $(A,B) \in \Sigma \times \Sigma$  are considered and the weight of all edge pairs with lag  $\tau$  and carrying symbols A and B respectively is accumulated over the graph. Mathematically, the  $n$ -th utterance is characterized by

$$\mathbf{V}_{in} = \sum_{(\alpha,\beta) \in \Theta_i} p_{\alpha\beta} \quad (1)$$

where  $\Theta_i$  is the set of edge pairs with lag  $\tau$  and labels A and B respectively. The index  $i$  is a one-to-one mapping of all combinations  $(A,B)$  to the integers  $1 \dots |\Sigma|^2$ . In other words, a co-occurrence histogram of all possible acoustic event pairs is constructed. This results in a shift-invariant representation of fixed dimension (the square of the number of different symbols), independent of the length of the analysis window,

which will be referred to as *HAC* (histogram of acoustic co-occurrence) or simply *histogram*.

In this paper, the acoustic events will be quantised full-band spectra computed at regular time instants or frames. The lattice of acoustic events then degenerates to a chain where each arc carries a VQ label with unity posterior probability. The HAC-representation is then the number of times all VQ symbol label pairs are observed  $\tau$  frames apart.

For a given segment of speech, a unique high-dimensional HAC representation can be computed. Both the actual acoustic events occurring in the utterance as well as their order affect the HAC-representation. Conversely, a HAC representation does however *not* map to a unique symbol sequence or symbol lattice. For example, the histogram of the symbol sequence ABCDA and ABCDABCD differ only by a scaling factor. The histograms of ABCDA and its (almost) cyclic permutation BCDAB are identical. In general, order is weakly represented in the HAC-model, much like a bigram language model only weakly represents grammar. Taking this analogy further, histograms over more than two symbols could alleviate this weakness at the cost of complexity. It will however be shown below that even with histograms of co-occurrence pairs, encouraging speech recognition results can be obtained. HAC-models are different from *convolutional NMF* [8], which describes traces in the acoustic space by an impulse response, a very rigid model for word-sized units.

## 2.2. Matrix factorisation for pattern discovery and recognition

If utterances are composed of one or more out of  $R$  recurring acoustic patterns such as words, the histograms have a linearity property that is essential in the proposed method: each such acoustic pattern is characterized by a HAC and hence the HAC of each utterance will be a (integer) linear combination of histograms. Define  $\Theta_r$  as the subset of  $\Theta_i$  that originates from the  $r$ -th acoustic pattern. Then:

$$\mathbf{V}_{in} \approx \sum_{r=1}^R \sum_{(\alpha,\beta) \in \Theta_r} P_{\alpha\beta} = \sum_{r=1}^R \mathbf{W}_{ir} \mathbf{H}_{rn}$$

where  $\mathbf{H}_{rn}$  is the number of times pattern  $r$  occurs in utterance  $n$  and column  $\mathbf{W}_{ir}$  is the histogram of acoustic co-occurrences for the  $r$ -th pattern. In matrix form:

$$\mathbf{V} \approx \mathbf{W} \mathbf{H} \quad (2)$$

Given their interpretation, all entries of  $\mathbf{W}$  and  $\mathbf{H}$  are constrained to be positive or zero. Because of these constraints, equation (2) is known as *Non-negative Matrix Factorisation* (NMF) [3]. Since the observed symbols are subject to variability and uncertainty, equation (2) is only approximate and  $\mathbf{W}$  and  $\mathbf{H}$  are estimated as the positive matrices that minimize the divergence metric

$$D(\mathbf{V} \parallel \mathbf{W} \mathbf{H}) = \sum_{i,j} \left( \mathbf{V}_{ij} \log \frac{\mathbf{V}_{ij}}{[\mathbf{W} \mathbf{H}]_{ij}} - \mathbf{V}_{ij} + [\mathbf{W} \mathbf{H}]_{ij} \right) \quad (3)$$

An algorithm for finding  $\mathbf{W}$  and  $\mathbf{H}$  given  $\mathbf{V}$  based on multiplicative updates is given in [3]. This way, a small vocabulary can be discovered without supervision in a collection of utterances of continuously spoken words [4].

Without additional constraints,  $\mathbf{W}$  (and therefore  $\mathbf{H}$ ) is determined within a positive  $R$ -by- $R$  scaling matrix  $\mathbf{S}$  with a positive inverse:  $\mathbf{W} \mathbf{H} = \mathbf{W} \mathbf{S} \mathbf{S}^{-1} \mathbf{H}$ . Examples of such scaling matrices are diagonal matrices with a strictly positive diagonal

(scaling of the columns of  $\mathbf{W}$ ) and permutation matrices. To address scaling, the constraint that each column of  $\mathbf{W}$  sums to unity is imposed, while permutation will not affect the results.

Once  $\mathbf{W}$  is estimated on a training set, new utterances can be analysed with factorisation (2) by estimating  $\mathbf{H}$ , whose columns reveal the degree to which each discovered acoustic pattern is present in each new utterance.

## 2.3. Supervised learning

If it is known which words occur in each utterance, this information can be exploited to associate each column of  $\mathbf{W}$  to a word identity. Therefore, the  $M \times N$  *grounding* matrix  $\mathbf{G}$  is formed, which holds in its  $m$ -th row and  $n$ -th column the number of times the  $m$ -th word occurs in the  $n$ -th utterance. Here,  $M$  is the number of word identities and  $N$  is the number of utterances available for supervised training. Subsequently, compute:

$$\begin{bmatrix} \mathbf{G} \\ \mathbf{V} \end{bmatrix} \approx \begin{bmatrix} \mathbf{W}_g \\ \mathbf{W}_v \end{bmatrix} \mathbf{H} \quad (4)$$

which expresses that word identity needs to be explained jointly with the acoustic data by common model activations  $\mathbf{H}$ . The common dimension  $R$  is chosen  $R \geq M$ . Experiments, not reported in this paper, have shown that overestimation of  $R$  is an acceptable strategy. The resulting columns of  $\mathbf{W}_g$  will be zero, i.e. be associated with acoustic events that have no relevance to grounding, e.g. to a model for silence or filler words.

After supervised training, i.e. computing factorisation (4), recognition on unseen data is achieved by first computing  $\hat{\mathbf{H}}$  in  $\mathbf{V} \approx \mathbf{W}_v \hat{\mathbf{H}}$  using only the acoustic co-occurrence data and with fixed  $\mathbf{W}_v$ . The presence of words or their *activation* (i.e. an estimate of the grounding information: typically 1 for a word that is present and 0 if it is absent) in the test utterances is subsequently estimated as:

$$\hat{\mathbf{G}} = \mathbf{W}_g \hat{\mathbf{H}} \quad (5)$$

Notice an important difference with an HMM-based speech recogniser: each column of the matrix  $\hat{\mathbf{G}}$  will reveal to which extent each trained word is present in the corresponding test utterance. However, it will say nothing about the *order* in which the words occur in the utterance, a problem that is addressed in the next section and forms the first novelty of this paper.

## 3. Word order and timing

The method of section 2 uses a “bag of words” approach, in that words can be discovered at training time, linked to word identities and subsequently can be spotted in given test utterance. However, word order, which is essential in language, is not modelled. A mechanism to find out in which order words occur in the test utterance is required.

### 3.1. Time-scaled histograms

Next to the histogram of acoustic co-occurrences, the edge pair weights are also multiplied with the time of occurrence of the first arc  $t_\alpha$  (e.g. its ending time) and are accumulated over the lattice. Hence, with the notations of equation (1):

$$\mathbf{T}_{in} = \sum_{(\alpha,\beta) \in \Theta_i} t_\alpha P_{\alpha\beta} \quad (6)$$

With a similar reasoning as the one of section 2.2 for histograms, these *time-scaled* histograms (6) of an utterance are shown to be composed of the time-scaled histograms of acoustic patterns, but the weighting time is offset by the starting time of the pattern. Then:

$$\mathbf{T}_{in} = \sum_{j=1}^R \sum_{(\alpha,\beta) \in \Theta_j} t_{\alpha} p_{\alpha\beta} = \sum_{r=1}^R \mathbf{X}_{ir} \mathbf{H}_{rn} + \sum_{r=1}^R \mathbf{W}_{v,ir} \mathbf{U}_{rn} \quad (7)$$

where  $\mathbf{X}_{ir}$  is the time-scaled histogram of the  $r$ -th pattern when aligned to some reference time and  $\mathbf{U}_{rn}$  is the time offset of the  $r$ -th pattern in the  $n$ -th utterance. Hence, for supervised learning mode:

$$\begin{bmatrix} \mathbf{G} \\ \mathbf{V} \\ \mathbf{T} \end{bmatrix} \approx \begin{bmatrix} \mathbf{W}_g & \mathbf{0} \\ \mathbf{W}_v & \mathbf{0} \\ \mathbf{X} & \mathbf{W}_v \end{bmatrix} \begin{bmatrix} \mathbf{H} \\ \mathbf{U} \end{bmatrix} \quad (8)$$

The estimation of  $\mathbf{X}$  in (8) appears to be a non-trivial problem. Only with knowledge of the time of occurrence of the words within the training utterances, an accurate estimate of  $\mathbf{X}$  was obtained. In order to avoid to have to resort to exploiting additional (timing) information, the following approximation was considered: if the time-scaled histogram of the utterance is modelled as the sum of the time-scaled histograms of the acoustic patterns, but where event occurrence time  $t_{\alpha}$  is approximated by the word occurrence time,  $\mathbf{X}$  will equal  $\mathbf{W}_v$  after normalisation. Accepting an offset in  $\mathbf{U}$ , (8) can be replaced by (supervised mode)

$$\begin{bmatrix} \mathbf{G} \\ \mathbf{V} \\ \mathbf{T} \end{bmatrix} \approx \begin{bmatrix} \mathbf{W}_g \mathbf{H} \\ \mathbf{W}_v \mathbf{H} \\ \mathbf{W}_v \mathbf{U} \end{bmatrix} \quad (9)$$

Rather than estimating  $\mathbf{W}_g$ ,  $\mathbf{W}_v$ ,  $\mathbf{H}$  and  $\mathbf{U}$  jointly on the training data using (9), slightly better results were obtained when training the  $\mathbf{W}$ -matrices from (4). This is not surprising since the approximation in the lower partition ( $\mathbf{T}$ ) negatively influences the factorization of  $\mathbf{V}$ . During testing,  $\mathbf{H}$  and  $\mathbf{U}$  are estimated separately with fixed  $\mathbf{W}_v$  and the time of occurrence of a pattern is found as  $\mathbf{U}/\mathbf{H}$  (element-wise division).

### 3.2. Multiplicity

If a word occurs twice or more within an utterance, its number of occurrences as well as its times of occurrence will be added in one value for  $\mathbf{H}$  and  $\mathbf{U}$  and hence, the estimated time of occurrence for the pattern will turn out to be an estimate of the “average” occurrence time of the pattern, which makes ordering the words impossible. Although the multiplicity of a word can in principle be determined by inspecting  $\mathbf{H}$  (or in case of supervised learning  $\mathbf{G}$ ), the patterns cannot be separated. This is intrinsically related to the property that HAC-models do not make an explicit segmentation of the data. The problem will be partially solved in section 4, where a method for handling disjoint repetitions of a word will be described.

### 3.3. Experiments

The training speech data are taken from the TI-DIGITS corpus which contains recordings of 55 male and 57 female US-American adults, downsampled to 16 kHz. Since including examples of the words in isolation would simplify the problem of discovery of acoustic patterns, the isolated digit

strings were removed from training and test, totaling 6159 connected digit sequences of length 2 through 7 for training.

For the acoustic information, 12 MFCC’s plus log-energy are computed at a 100 Hz frame rate. A codebook of respectively 150, 150 and 100 for static, velocity and acceleration parameters is trained on the training set using the K-means algorithm. All training utterances are then processed resulting in a VQ-label for static, velocity and acceleration features per 10 ms analysis frame. Per utterance, the label co-occurrence histograms for the three streams are computed with a lag value  $\tau = 5$ , resulting in a 22500-dimensional vector for the static and the velocity stream and 10000-dimensional vector for the acceleration stream. To improve discrimination between speech and non-speech, leading and trailing silence were stripped off each training utterance with an energy-based voice activation detector and considered as additional training utterances. Hence, a (very sparse)  $55000 \times 12318$  data matrix  $\mathbf{V}$  is obtained.  $\mathbf{T}$  is obtained similarly. The VQ histogram counts are divided by a fixed constant (100) such that the acoustic and grounding information have roughly the same weight in the cost function in supervised learning. Experiments have shown that the value of this constant is not critical: it can be changed over several orders of magnitude without significant impact. Subsequently, factorisation (4) is computed for  $R = 12$  and  $\mathbf{W}_v$  and  $\mathbf{W}_g$  are stored for recognition.

In this first evaluation, recognition and subsequently ordering of digits is attempted on a per-utterance basis. However, as outlined in section 3.2, it is not straightforward to locate multiple occurrences of the same word within a sentence, since (roughly) their average time of occurrence would be estimated. For consecutive repetitions of a digit, this is not a problem for order estimation. Therefore, utterances with non-adjacent repetitions of any digit were removed for testing, resulting in 4163 utterances. For example, “998” is retained, but “989” is removed as a test utterance. Given the number  $K_n$  of different digits occurring in the  $n$ -th test utterance, the  $K_n$  candidates with highest activation according to equation (5) are selected, yielding a word error rate of 2.83% and an *unordered* string error rate of 8.62%. Notice that this recognition result is unordered, so word error rate is defined as the sum of the number of incorrect digits that end up in the top  $K_n$ , divided by the sum of  $K_n$  over the complete test set. A string is incorrect if it contains any incorrect digit.

The recognised digits of each utterance are subsequently ordered by their estimated time of occurrence, i.e.  $\mathbf{H}$  and  $\mathbf{U}$  are estimated based on equation (9) and the digit position  $\mathbf{U}/\mathbf{H}$  is formed. For the  $n$ -th utterance, the recognised top  $K_n$  candidates are ordered according to their estimated time of occurrence. This yields an *ordered* string error rate of 11.72%. Hence, only in 3.1% of the strings, the ordering process introduced additional errors.

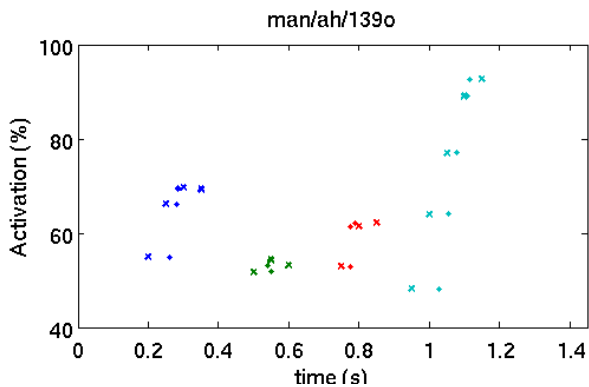
## 4. A sliding window decoder

The approach of section 3 takes a holistic approach to recognition in the sense that a complete utterance is analysed in terms of the components (words) it is made of and subsequently those components are located in time. At no point there is an attempt to break down the utterance in segments. For long utterances, one can expect that *unmixing* the components becomes ill-conditioned.

In this section, a more local implementation of the same idea is examined. A sliding window of 400ms is moved over the utterance in steps of 50ms, word activation is computed and location of the best candidate is performed. This single

best decoding strategy is viable only because no grammatical constraints need to be taken into account here. The candidate is accepted if its activation is higher than 0.25 and if its estimated location is within 40ms of the window center. Examples of the activation levels and estimated times of accepted candidates by this process are shown in *Figure 1*. If in subsequent 50ms frames, the same word is detected, it is considered as a continuation of the same word and not a repetition of it. However, when the same digit is repeated, this often – but not always – fails to produce an interruption in the activation of candidates, hence leading to deletion errors. Thus, the detection of adjacent repetitions of the same word is error-prone and not attempted in the sequel: adjacent repetitions of the same digit are mapped to a single occurrence in reference and in hypothesis during scoring.

This decoder is evaluated on a subset of the TI-DIGITS test set containing 6214 digit strings of length 2 through 7. As such, the algorithm leads to a word error rate of 7.40% (4.41% insertions, 2.19% deletions and 0.81% substitutions). It is observed that “oh” is often inserted after “zero”, which is not unexpected, since the former is the last phone of the latter and the decoder does not have a constraint to find complete words. Invariantly, the word “oh” receives a large activation when the sliding window reaches the end of “zero”. Therefore, all occurrences of “zero oh” were mapped to “zero”. Similarly, “six” is often inserted before “seven” and “zero” and was only accepted in this context if its activation exceeds 0.5. This leads to a word error rate of 5.57% (2.56% insertions, 2.23% deletions and 0.78% substitutions). For comparison, a discrete density HMM was trained and tested on the same material, using the same VQ data and 7 states per digit (3 for silence). This yields 3.75% word error rate (1.53% insertions, 1.19% deletions and 1.02% substitutions). The lower performance of the HAC-model is mainly caused by insertions and deletions, which is not surprising given its extremely simple decoding strategy based on word activation without dynamic programming.



*Figure 1: activation level versus center position of analysis window (x) and estimated location (.) for the utterance “139oh”. The activation of each digit is plotted in a different colour.*

## 5. Discussion and conclusions

The present approach shows some remarkable similarities with models of human speech recognition (HSR). Most remarkable is that in HAC-models the speech data are not segmented, but that a window of speech is considered. Words are activated and compete, much like is the case in the Shortlist model [6] of HSR. A holistic match of speech with high-dimensional models is made, which differs strongly from

the approach taken in HMMs. The implicit segmentations that are generated in HMMs lead to sharp boundaries between words, a concept that is not so clear in HSR and that also might explain our insensitivity to strong cross-word coarticulations. However, HAC’s failure to detect or even hypothesise word boundaries also leads to the problems with word multiplicity mentioned above. Progress on this front can probably be made if words are described as a sequence of subword units, which are then located and hence detect word beginnings and endings. This will be explored in further research.

Notice also that at no point any order information in the training data was exploited. At best, the presence/absence of words in the training data was used in supervised mode, though the NMF-based pattern discovery method can even work without supervision [4]. Eventually, it is capable of recognising and ordering the discovered acoustic patterns.

The current implementation contains only a single layer of representations, mapping the acoustic level directly to the lexical level. In ASR as well as in most models of HSR [6], [9], [10], [1], a pre-lexical (e.g. phonemic) level is assumed. This level is required for building larger vocabularies, since re-use of acoustic representations (phonemes) can then be achieved for learning parsimonious lexical representations that require less storage and which can be learned from a small number of examples. A pre-lexical level, however, could also help decreasing the ambiguity of the decoding process by imposing *possible word constraints* [6], [1].

## 6. Acknowledgements

This research was funded by the European Commission under contract FP6-034362 (ACORNS).

## 7. References

- [1] Scharenborg, O., Norris, D., ten Bosch, L., McQueen, J.M., "How should a speech recognizer work?", *Cognitive Science*, 29, 867-918, 2005.
- [2] Boves, L., ten Bosch, L., Moore, R.K., "ACORNS – Towards computational modeling of communication and recognition skills", *IEEE Int. Conf. on Cognitive Informatics*, 349-356, 6-8 Aug. 2007
- [3] Lee, D.D., Seung, H.S., "Learning the parts of objects by non-negative matrix factorization", *Nature*, Vol. 401, No. 6755, 788-791, 1999
- [4] Stouten, V., Demuynck, K., Van hamme, H., "Discovering Phone Patterns in Spoken Utterances by Non-negative Matrix Factorisation." *IEEE Signal Processing Letters*, Vol. 15, 131-134, 2008.
- [5] Van hamme, H, "Integration of Asynchronous Knowledge Sources in a Novel Speech Recognition Framework", *ISCA ITRW workshop on Speech Analysis and Processing for Knowledge Discovery*, Aalborg, Denmark, June 2008, accepted.
- [6] Norris, D. "Shortlist: A connectionist model of continuous speech recognition", *Cognition*, 52, 189-234, 1994.
- [7] Ezzat, T., Bouvrie, J., Poggio, T., "Spectro-Temporal Analysis of Speech Using 2-D Gabor Filters", In. *Proc. Interspeech*, 506-509, Antwerp, Belgium, August 2007
- [8] Smaragdīs, P., "Non-negative Matrix Factor Deconvolution; Extraction of Multiple Sound Sources from Monophonic Inputs", In C.G. Puntonet and A. Prieto (Eds.), *Independent Component Analysis and Blind Signal Separation*, 494-499, Springer Berlin / Heidelberg, 2004
- [9] McClelland, J.L., Elman, J.L., The TRACE model of speech perception. *Cognitive Psychology*, 18, 1-86, 1996
- [10] Norris, D., "The Bayesian Reader: Explaining word recognition as an optimal Bayesian decision process." *Psychological Review*, 113(2), 327-357, 2006