

# Maximum Entropy Model

In subject area: [Computer Science](#)

The Maximum Entropy Model is defined as a model that maximizes entropy while satisfying statistical constraints by assuming the network state probability distribution is an exponential function of the network energy.

AI generated definition based on: [Neural Networks, 2018](#)

## On this page

### Chapters and Articles

You might find these chapters and articles relevant to this topic.

Review article

#### Connectivity inference from neural recording data: Challenges, mathematical bases and research directions

[2018, Neural Networks](#)

Ildefons Magrans de Abril, ... Kenji Doya

#### 5.1.8 Maximum entropy model

The maximum-entropy model (Roudi et al., 2015; Yeh et al., 2010) assumes that the network state [probability](#) distribution is given by an exponential function of the network energy

$E[S_i] = E[(x_1, \dots, x_n)]$  such that entropy is maximized while satisfying any statistical constraints. When first and second-order [statistics](#) are given, the state probability distribution is given by

(26)

$$P(x_1, \dots, x_n) = \frac{1}{Z} \exp(-E[(x_1, \dots, x_n)]) = \frac{1}{Z} \exp\left(\sum_i$$



temporal correlations.

The main limitation of the [maximum entropy](#) model is its [computational complexity](#). Recent studies demonstrated its

[View article](#)

[Read full article](#)

URL: <https://www.sciencedirect.com/science/article/pii/S0893608018300704>

Chapter

## Handbook of Statistics

2013, [Handbook of Statistics](#)

N. Mohanty, ... T.M. Rath

### 3.4 Maximum entropy

The [maximum entropy](#) (maxent) approach is rooted in [information theory](#) and has been successfully applied to many fields including [physics](#) and [natural language processing](#). It creates a model that best accounts for the available data but with a constraint that without any additional information the model should maximize entropy. In other words, the model prefers a uniform distribution by maximizing the [conditional entropy](#). The maximum entropy model was originally developed by Berger et al. (1996) for natural language applications such as information retrieval and speech recognition. Jeon and Manmatha (2004) adapted the model for images. We follow the derivation of Jeon and Manmatha (2004) applying maxent to the problem of shape classification.

Given pairs of [training data](#),  $(\vec{s}_1, y_1), (\vec{s}_1, y_1), \dots, (\vec{s}_n, y_n)$  where  $\vec{s}_i$  represents the image and  $y_i$  is the class label we need to create feature functions or predicates as follows. In order to apply the discrete maximum entropy model to classify the images we discretized the feature vector  $\vec{s}$  to get  $K$  discrete tokens,  $s_k$ . Maxent requires the creation of predicates,  $f(s_k, y_i)$ , where the value of the predicate is 1 if the discretized tokens for the image contain  $s_k$  and the image has the class label  $y_i$  and 0 otherwise. The expected value of the predicate with respect to the available

Show more

[View chapter](#)

[Explore book](#)



Chapter

# Large-Scale Machine Learning

2011, GPU Computing Gems Emerald Edition

Jerod J. Weinman, ... Shitanshu Aggarwal

## 19.2.1 Background: Maximum Entropy Theory

As its name suggests, the [maximum entropy](#) model is intimately related to [probability](#) theory. We now give details on the model in order to understand and implement the objective function being optimized. Let  $\mathbf{X}$  be the  $D \times N$  matrix of [training data](#), as before, so that  $\mathbf{x}_j$  is a column vector formed by the  $j$ th column of  $\mathbf{X}$ . Let  $\mathbf{W}$  be the  $L \times D$  [weight matrix](#), with  $\mathbf{w}_i$  the row vector formed by the  $i$ th row of  $\mathbf{W}$ . The probability of category label  $i$  given the data  $\mathbf{x}_j$  is

$$p_{i,j} = \frac{1}{Z_j} \exp w_i x_j. \quad (19.1)$$

The constant  $Z_j$  is the sum of the exponentiated inner products for all the category labels,

$$Z_j = \sum_{i=1}^L \exp(w_i x_j), \quad (19.2)$$

so that  $\mathbf{p}_j$  represents a properly normalized probability distribution over labels with  $\sum_{i=1}^L p_{i,j} = 1$ .

The training examples must have category labels assigned to the data. Let  $\mathbf{y}$  be a  $N \times 1$  column vector with entries  $y_j \in \{1, \dots, L\}$ .

With all of this, the objective function is the conditional log-likelihood of the labels given the [observed data](#),

$$\begin{aligned} \mathcal{L}(\mathbf{W}; \mathbf{y}, \mathbf{X}) &\equiv \sum_{j=1}^N \log p_{y_j, j} \\ &= \sum_{j=1}^N (w_{y_j} x_j - \log Z_j), \end{aligned} \quad (19.3)$$

where  $p_{y_j, j}$  indicates the probability of the correct label  $y_j$  for the  $j$ th instance and  $\mathbf{w}_{y_j}$  is the vector of weights for the correct label of the instance. Typically, a [regularization](#) term is added to the function in order to prevent overfitting [4, 7]. We omit the details of this practice and point out where in the parallel implementation they are handled. For a particular label  $i$  and feature  $k$ , the gradient of the objective function, needed by the optimization routine, is given by

[View chapter](#)[Explore book](#) >[Read full chapter](#)URL: <https://www.sciencedirect.com/science/article/pii/B978012384988500019X>

Chapter

## Big Data Analytics

2015, *Handbook of Statistics*

Venkat N. Gudivada, ... Vijay V. Raghavan

### 6.2 Better Models with More Data

[Maximum Entropy](#) (MaxEnt) [language models](#) are linear models which are typically regularized using the L1 or L2 terms in the likelihood objective. This obviates the need for smoothed  $n$ -gram language models. In Biadisy et al. (2014), the effect of adding backoff features and its variants to MaxEnt models is investigated. This approach results in better language models with lower perplexity even with [training data](#) in the order of hundreds of billions of words and hundreds of millions of features.

[View chapter](#)[Explore book](#) >[Read full chapter](#)URL: <https://www.sciencedirect.com/science/article/pii/B9780444634924000095>

Chapter

## Large-Scale Machine Learning

2011, *GPU Computing Gems Emerald Edition*

Jerod J. Weinman, ... Shitanshu Aggarwal


### 19.2 Core Technology

A [maximum entropy](#) model consists of an  $L \times D$  matrix  $\mathbf{W}$  that determines how much weight to assign each of  $D$  features in an input vector for each of  $L$  potential category labels. A  $D \times 1$  column vector of [input features](#)  $\mathbf{x}$  to be classified must simply be multiplied by the [weight matrix](#). The label with the largest value in the product  $\mathbf{W}\mathbf{x}$  can be taken as the assigned or [predicted value](#). Thus, classification is a relatively efficient task. The goal of the learning algorithm is to find a weight matrix  $\mathbf{W}$  that gives high values to correct labels and relatively low values to the rest for all input vectors.



requires an optimization over many parameters. Though the [optimization problem](#) is convex, representing the second derivative of the objective function with the [Hessian matrix](#) is space prohibitive. Therefore, limited-memory quasi-Newton [optimization algorithms](#) (e.g., L-BFGS [3]) must be used to find the optimal weights. The bottleneck in the process is simply evaluating the objective function and its gradient when the number of training instances  $N$  is extremely large.

Fortunately, both the objective function and the gradient (detailed later in this chapter) are sums over the  $N$  training instances

Show more 

View chapter

Explore book 

[Read full chapter](#)

URL: <https://www.sciencedirect.com/science/article/pii/B978012384988500019X>

Chapter

## Big Data Analytics for Social Media

2016, Big Data

S. Kannan, ... A. Kejariwal

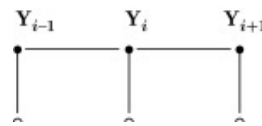
### Statistical NLP methods

[NER](#) can be treated as a standard classification problem, for example, identifying if a token is an entity or not. Statistical [machine learning algorithms](#) can be applied to solve this problem. Supervised classifiers yielded better NER systems compared to rule-based ones, as they are easier to maintain and adapt to different domains. Hidden Markov models (HMMs) [22] and [support vector machine](#) (SVM)-based models [23] were used in NER tasks with varying degree of success.

[Maximum entropy](#) (ME) models were highly successful [24]. These models use [posterior probability](#) of a tag given a word within a limited window  $w_{n-i}^n = w_{n-i} \dots w_{n+i}$  around  $n$ th word  $w_n$  with a window size of  $2i+1$  [8]. The models comprised of set of [binary features](#) such as lexical, word based, transition, prior, compound, and dictionary features. Chieu and Ng added other features to the system such as useful word suffixes, class suffixes, and initial [capital letters](#) [25].



of-speech (POS) tagging and NER. A typical CRF structure is illustrated in Fig. 6. CRF provides many advantages over ME or HMMs.



[View chapter](#)

[Explore book](#) >

[Read full chapter](#)

URL: <https://www.sciencedirect.com/science/article/pii/B9780128053942000039>

Review article

## Mining smartphone data for app usage prediction and recommendations: A survey

2017, *Pervasive and Mobile Computing*

Hong Cao, Miao Lin

### 5 Other studies

In addition to the main research streams on app usage prediction and app recommendations, we also briefly touch on several following related topics, namely classifying the apps [67] and retrieving similar users in terms of usage patterns [113–115], etc. These works also mixed past and present discovery of the underlying app usage patterns from raw smartphone records, and the techniques proposed could benefit app usage prediction and recommendations.

Zhu et al. [67] applied a [Maximum Entropy](#) model (MaxEnt) to classify apps into different categories by combining context information from web and individuals' app usage information. The context information included both explicit and implicit feedback from web. The explicit feedback from each app was the top-searched results from a search engine. Based on the search outcomes, two measures, namely general label confidence score and general label entropy, were proposed to evaluate the likelihood of the app being classified into a given category. The implicit feedback of the apps was the latent topics, and these topics were learned from an [LDA](#) model [79] by considering similar meanings of words. Also, two pieces of contextual information, namely pseudo feedback from context vectors and frequency patterns, were extracted from individuals' app usage

[Show more](#) [View article](#)[Read full article](#)URL: <https://www.sciencedirect.com/science/article/pii/S1574119217300421>[Review article](#)

## Abstractive summarization: An overview of the state of the art

[2019, Expert Systems with Applications](#)

Som Gupta, S. K Gupta

### 2.3.2 Recent works on abstractive summarization using Deep Learning Models

Baral *et al.* (2013) used [neural networks](#) for [parsing](#) the semantic graphs for generating the abstractive summaries by deep [linguistic analysis](#). They used Minimal Recursion Semantics (MRS) for [semantic representation](#) of grammars. They performed disambiguation by using [maximum entropy](#) model. They developed this model to solve the alignment problem of AMR Graphs. MRS can be used both for parsing and the text generation. Niu *et al.* (2017) proposed a multi-document [abstractive summarization](#) approach by using chunk graphs and neural networks. They used a Recurrent Neural Network [Language Model](#) which helped evaluate the linguistic quality of sentence, which further helped create good readable abstractive summaries.

Simple sequence-to-sequence model map the input sequence to the [output sequence](#). Jobson and Gutierrez (2018) used encoder-decoder RNN along with [LSTM](#) to create the summaries. They used the word-embedding for the training purpose and attention function for creating the context vector at each time step. Nallapati, Zhou, dos Santos, Gulehre, and Lapata (2016) used the attention model along with the RNN to handle the issues of modeling keywords and capturing of the hierarchical structure between the sentence and word. They used the bidirectional

[Show more](#) [View article](#)[Read full article](#)URL: <https://www.sciencedirect.com/topics/computer-science/maximum-entropy-model>



# COINCIDENCE RESOLUTION: A REVIEW OF general methodologies and applications in the clinical domain

2011, *Journal of Biomedical Informatics*

Jiaping Zheng, ... Guergana K. Savova

## 4.1.3 Partitioning

The binary [classification results](#) from the mention-pair model or the entity-mention model are only the first step in resolving coreference. The markables need to be clustered into chains based on the predictions from the classifier.

For example, for the markables in Example 8, a system may generate the following results for the pairs:  $\langle m_1, m_2 \rangle$  non-coreferential,  $\langle m_1, m_3 \rangle$  non-coreferential,  $\langle m_2, m_3 \rangle$  non-coreferential,  $\langle m_2, m_4 \rangle$  non-coreferential,  $\langle m_2, m_5 \rangle$  coreferential, etc. The partitioning algorithm is responsible to cluster the five markables into three sets:  $\{m_1\}$ ,  $\{m_2, m_4\}$ , and  $\{m_3, m_5\}$ , from the imperfect classification results.

Let  $\langle m_i, m_j \rangle^+$  denote that the classification output for markables  $m_i$  and  $m_j$  is coreferential, and  $\langle m_i, m_j \rangle^-$  otherwise. Suppose the results for four markables  $m_1 \dots m_4$  are  $\langle m_1, m_2 \rangle^+$ ,  $\langle m_1, m_3 \rangle^-$ ,  $\langle m_1, m_4 \rangle^+$ ,  $\langle m_2, m_3 \rangle^+$ ,  $\langle m_2, m_4 \rangle^-$ , and  $\langle m_3, m_4 \rangle^+$ . There is not a natural grouping of the four markables that is consistent with the pair-wise result.

Similarly in the entity-mention case, let  $\langle \{m_i \dots m_j\}, m_k \rangle^\pm$  denote the classification result for a partial cluster  $\{m_i \dots m_j\}$  and a markable  $m_k$ . Given results of  $\langle \{m_1\}, m_2 \rangle^-$ ,  $\langle \{m_2\}, m_3 \rangle^+$ , and  $\langle \{m_1, m_3\}, m_4 \rangle^+$ , there is not a partition of the four markables that satisfies the three individual results.

Show more

View article

[Read full article](#)

URL: <https://www.sciencedirect.com/science/article/pii/S153204641100133X>

Review article





# general methodologies and applications in the clinical domain

2011, *Journal of Biomedical Informatics*

Jiaping Zheng, ... Guergana K. Savova

## 4.4 Specialized models

It is worth noting that many coreference resolution applications focus on a single type of NP, or handle different types of NPs separately, based on the observation that different types of NPs exhibit different patterns in terms of coreference participation [70,71]. Strube et al. [72] provided empirical evidence by examining the performance of the same set of features on different types of NPs, and obtained disparate results on pronouns, proper names, and definite NPs (in the order from highest to lowest). Ge et al. [54], and Yang et al. [46] built systems that only resolve pronouns. Morton [51] trained a [maximum entropy](#) model to resolve pronouns, and applied simple string matching to resolve proper nouns. Denis and Baldridge [43] learned separate models for third person pronouns, speech pronouns (first and second person), proper names, definite NPs, and other anaphoras that do not fall into one of the previous categories. Zelenko et al. [73] also trained five classifiers to handle names, nominal NPs, first person pronouns, second person pronouns, "it", singular third person pronouns, and plural third

[View article](#)

[Read full article](#)

URL: <https://www.sciencedirect.com/science/article/pii/S153204641100133X>

## Related terms:

[Support Vector Machine](#), [Machine Learning](#),

[Maximum Entropy](#),

[Generative Adversarial Networks](#),

[Conditional Random Field](#), [Training Data](#),

[Network Traffic](#), [Deep Transfer Learning](#),

[Language Modeling](#), [crame rao low bound](#).

**Expert Systems with Applications**

Journal

**Pattern Recognition Letters**

Journal

**Neurocomputing**

Journal

**Journal of Biomedical Informatics**

Journal

All content on this site: Copyright © 2025 or its licensors and contributors. All rights are reserved, including those for text and data mining, AI training, and similar technologies. For all open access content, the relevant licensing terms apply.

