

DATA PROCESSING METHOD AND APPARATUS

[0001] This application claims priority to Chinese Patent Application No. 202011391497.6, filed with the China National Intellectual Property Administration on December 2, 2020 and entitled "DATA PROCESSING METHOD AND APPARATUS", which is incorporated herein by
5 reference in its entirety.

TECHNICAL FIELD

[0002] This application relates to the field of artificial intelligence, and in particular, to a data processing method and apparatus.

BACKGROUND

[0003] Artificial intelligence (Artificial Intelligence, AI) is a theory, a method, a technology, or an application system that simulates, extends, and expands human intelligence by using a digital computer or a machine controlled by a digital computer, to perceive an environment, obtain knowledge, and achieve an optimal result based on the knowledge. In other words, artificial intelligence is a branch of computer science and attempts to understand essence of intelligence and
10 produce a new intelligent machine that can react in a similar manner to human intelligence. Artificial intelligence is to research design principles and implementation methods of various intelligent machines, so that the machines have perception, inference, and decision-making functions. Research in the field of artificial intelligence includes robotics, natural language processing, computer vision, decision-making and inference, human-machine interaction,
15 recommendation and search, AI basic theories, and the like.

[0004] Currently, various AI functions, such as natural language processing, image processing, and speech processing, are usually implemented by using a neural network. With development of AI technologies, AI functions are gradually diversified. However, these functions are implemented based on input data and labels. Parameters of a machine learning model are trained by using
20 optimization methods such as a gradient descent method. After the parameters of the model

converge, the model can be used to predict unknown data. Therefore, processing of the input data and labels can be considered as a basis of AI. Based on data type differences, data to be input can be classified into a continuous feature and a discrete feature. Currently, for the discrete feature, one-hot encoding (one-hot encoding) is usually used. For the continuous feature, three common feature processing methods: a categorization method, a normalization method, and a discretization method are used. The discretization method is widely used in the industry, and includes an equal-frequency discretization method, an isometric discretization method, a logarithm method, a tree-based method, and the like.

[0005] An existing continuous feature processing method is to: discretize continuous feature values into different ranges (such as buckets, buckets) according to a discretization policy (such as a heuristic rule or model), replace original feature values with numbers of the ranges, and then obtain a vectorized representation in a manner same as that of the discrete feature. However, in this method, a large amount of manpower and time are usually required to try and optimize the discretization policy, so as to obtain an optimal discretization rule or model, and further obtain a final embedding vector representation. In addition, all discretization policies have the following disadvantages: a two-phase problem (two-phase problem, TPP), a similar value but different dissimilar embeddings (similar value but dissimilar embedding, SBD), and dissimilar values but a same embedding (dissimilar value but same embedding, DBS). For example, age features are divided into several groups including a group of ages of 18 to 40 and another group of ages of 40 to 60. A same embedding is used for ages of 18 and 40 with a large age difference, and cannot reflect the difference between the two ages. However, ages of 40 and 41 that are close to each other are divided into two groups, and embeddings may be significantly different. Therefore, vector representation values of the continuous feature in the existing solution are insufficient.

SUMMARY

[0006] Embodiments of this application provide a data processing method and apparatus to better learn a vector representation value of each feature value in a continuous feature, so that the vector representation value has a better representation capability.

[0007] According to a first aspect, an embodiment of this application provides a data processing method. The method specifically includes: A data processing apparatus obtains a

continuous feature from sample data, and then performs discretization processing on the continuous feature by using a discretization model, to obtain N discretization probabilities corresponding to the continuous feature. The N discretization probabilities correspond to N preset meta-embeddings, and N is an integer greater than 1. Finally, the data processing apparatus
5 determines a vector representation value of the continuous feature based on the N discretization probabilities and the N meta-embeddings.

[0008] In this embodiment, the continuous feature is feature data having a continuous statistical characteristic value in the sample data. In an example solution, the continuous feature includes, but is not limited to an age feature, a click count feature, and a score feature in a
10 recommendation model. A value of the click count feature may be a value range of an entire positive integer, and the value is a continuous feature, for example, the value is 1, 2, 3, 4, A value of the age feature may be 0 to an existing recorded maximum age. Discrete features in the sample data may be usually listed one by one only in a specific sequence. For example, a value of a gender feature is only male or female.

[0009] Vector representation means that a specific feature is represented by a vector. In this embodiment, the meta-embedding may be a preset initialized vector representation value, or may be an optimized vector representation value. The vector representation value of the continuous feature is a vector value that is determined based on the meta-embedding and that is used to
15 represent a specific feature of the continuous feature. It may be understood that a dimension of the vector representation value of the continuous feature is the same as that of the meta-embedding. In an example solution, the meta-embedding may be a five-dimensional vector value, for example, (01010). The vector representation value corresponding to the continuous feature is also a five-dimensional vector value, for example, (11010).
20

[0010] In this embodiment, the data processing apparatus calculates a discretization probability, that has more than one dimension, for a feature value of each continuous feature by
25 using the discretization model, presets a meta-embedding, that has more than one dimension, for each continuous feature field in the continuous feature, and determines, for a feature value, a vector representation value from the meta-embedding by using an aggregate function and the discretization probability. In this way, compared with the conventional technology, in this
30 embodiment, the vector representation value obtained through learning has a better representation capability, thereby helping improve accuracy of a prediction result.

[0011] Optionally, a specific manner in which the data processing apparatus performs discretization processing on the continuous feature by using a discretization model, to obtain N discretization probabilities corresponding to the continuous feature may be as follows: The data processing apparatus presets an initial variable in the discretization model, then determines, based on the initial variable, N mapping values corresponding to the continuous feature, and finally calculates the N discretization probabilities of the continuous feature based on the mapping value. In this embodiment, the initial variable may be an initial mapping field. For the continuous feature, the N mapping values corresponding to the continuous feature may be calculated based on the initial variable. Then, one corresponding probability is calculated based on each of the N mapping values, to obtain N probabilities. In this case, the N probabilities are used as the N discretization probabilities corresponding to the continuous feature.

[0012] Optionally, the discretization model may be a multiclass neural network, an attention network, or linear mapping and softmax. In this embodiment, the discretization model only needs to implement that the feature value corresponds to a plurality of discretization probabilities. The discretization model is not specifically limited herein. In addition, the discretization model may also be correspondingly selected based on different application scenarios. For example, in a system in which classification processing can be performed on continuous features, the discretization model may be the discretization model provided above.

[0013] Optionally, based on the foregoing manner, when the discretization model is linear mapping and softmax, the obtaining N discretization probabilities of the continuous feature by using a discretization model may be specifically:

presetting an initialized linear mapping variable to $W_{logit} = R^{1 \times h}$, where the initialized linear mapping variable is the initial variable; determining, according to a linear mapping formula, the N mapping values corresponding to the continuous feature, where

the linear mapping formula is $cont_{logit} = cont \cdot W_{logit}$; and calculating, according to a discretization formula, a probability corresponding to each of the N mapping values to obtain N

probabilities, where the discretization formula is $cont_p_k = e^{\frac{1}{\tau} cont_{logit}^k} / \sum_{i=1}^h e^{\frac{1}{\tau} cont_{logit}^i}$, and the N probabilities are used as the N discretization probabilities.

[0014] W_{logit} indicates a linear mapping variable, R indicates a real number field, h

indicates a quantity of buckets into which the continuous feature is discretized, h is equal to N , $cont_{logit}$ indicates a representation obtained after linear mapping of the continuous feature, $cont_p_k$ indicates a probability that the continuous feature is discretized to a k^{th} bucket, $cont_{logit}^k$ indicates a k^{th} neuron output after linear mapping of the continuous feature, τ indicates a temperature control coefficient of softmax, and $cont_{logit}^i$ indicates an i^{th} neuron output after linear mapping of the continuous feature.

[0015] Optionally, in this embodiment, the data processing apparatus may determine, by using an aggregate function and based on the N discretization probabilities and the N meta-embeddings, the vector representation value corresponding to the continuous feature. The aggregate function is Max-Pooling, Top-K-Sum, or Weighted-Average.

[0016] It may be understood that the aggregate function is used to aggregate N meta-embeddings corresponding to the N discretization probabilities into one embedding corresponding to the continuous feature. In an example solution, the discretization probabilities are (a1, a2, a3, a4), and the meta-embeddings are (b1, b2, b3, b4). In this case, a1 corresponds to b1, a2 corresponds to b2, a3 corresponds to b3, and a4 corresponds to b4. The aggregate function is used to aggregate (b1, b2, b3, b4) into one embedding, and the continuous feature is represented by the embedding. For example, (b1, b2, b3, b4) may be aggregated to obtain b3 as the vector representation value corresponding to the continuous feature.

[0017] Max-Pooling is calculated in a manner: obtaining, from the meta-embeddings based on an index corresponding to a largest value in the discretization probabilities, the corresponding embedding as the vector representation value corresponding to the continuous feature. A calculation formula of the vector representation value is $vx_cont = E^k$, where $k = \arg \max_h \{cont_p_h\}$. For example, it is assumed that the discretization probabilities are (a1, a2, a3, a4), and the meta-embeddings are (b1, b2, b3, b4). In this case, a1 corresponds to b1, a2 corresponds to b2, a3 corresponds to b3, and a4 corresponds to b4. If a value of a3 is largest, b3 is used as the vector representation value of the continuous feature.

[0018] Top-K-Sum is calculated in a manner: obtaining indexes corresponding to k largest values in the discretization probabilities, then obtaining corresponding embeddings from the meta-embeddings, and summing up the indexes as the vector representation value corresponding to the

continuous feature. A calculation formula of the vector representation value is $vx_cont = \sum_{k=1}^K E^k$, where $k = \arg \text{topk}_h \{cont - p_h\}$. For example, it is assumed that the discretization probabilities are (a1, a2, a3, a4), and the meta-embeddings are (b1, b2, b3, b4). In this case, a1 corresponds to b1, a2 corresponds to b2, a3 corresponds to b3, and a4 corresponds to b4. If values of a2 and a3 are largest, a sum of b2 and b3 is used as the vector representation value of the continuous feature.

[0019] Weighted-Average is calculated in a manner: performing weighted summation on the N probabilities and the meta-embeddings, and using a weighted sum of the N probabilities and the meta-embeddings as the vector representation value corresponding to the continuous feature. A

calculation formula of the vector representation value is $vx_cont = \sum_{k=1}^h cont - p_k \times E^k$. It may be understood that h in the calculation formula is equal to N. For example, it is assumed that the discretization probabilities are (a1, a2, a3, a4), and the meta-embeddings are (b1, b2, b3, b4). In this case, a1 corresponds to b1, a2 corresponds to b2, a3 corresponds to b3, and a4 corresponds to b4. The vector representation value of the continuous feature is equal to (a1×b1+a2×b2+a3×b3+a4×b4).

[0020] Optionally, the method further includes: inputting a user feature and an object feature into a recommendation model or a search model to obtain a prediction result. The user feature or the object feature includes the vector representation value. The user feature or the to-be-recommended object feature represents the continuous feature by using the vector representation value. To be specific, the data processing method provided in this embodiment of this application may be further applied to a specific application scenario. When the data processing method is applied to the recommendation model or the search model, the user feature and the object feature need to be input into the recommendation model or the search model. If the user feature or the object feature includes the continuous feature, the data processing model may output the vector representation value of the continuous feature by using the foregoing method, the vector representation value is included in the user feature or the object feature and input to the recommendation model or the search model.

[0021] Optionally, the vector representation value may be directly included in the user feature or the object feature, may be spliced with another continuous feature to be used as an entire input feature representation, or may be spliced with another continuous feature and another discrete

feature to be used as an entire input feature representation. As long as the vector representation value can be used as an input of model application or model training, this is not specifically limited herein. In this way, because the continuous feature has a better vector representation value, the input feature representation also has a better representation capability. Therefore, in a model training and model application process, a function of the model can be implemented more accurately.

[0022] Optionally, in an application or training process of the foregoing model, the data processing apparatus may further obtain an actual result in the application process, and then adjust a weight parameter in the discretization model based on the prediction result and the actual result by using a loss function in the model training process. The weight parameter may be understood as another parameter that does not include N, for example, the meta-embedding. In this way, the discretization model and a machine learning model that uses the discretization model affect each other, so that the weight parameter in the discretization model is adjusted based on real-time data distribution, to optimize the discretization model.

[0023] Optionally, N is greater than or equal to 20 and less than or equal to 100. Within this range, the discretization model may have good application effect.

[0024] Optionally, the continuous feature may be a normalized continuous feature. In this way, discretization of the continuous feature can be implemented more quickly.

[0025] According to a second aspect, this application provides a data processing apparatus. The apparatus has a function of implementing behavior of the data processing apparatus in the first aspect. The function may be implemented by hardware, or may be implemented by hardware executing corresponding software. The hardware or the software includes one or more modules corresponding to the foregoing function.

[0026] In a possible implementation, the apparatus includes units or modules configured to perform the steps in the first aspect. For example, the apparatus includes: an obtaining module, configured to obtain a continuous feature; and

a processing module, configured to: perform discretization processing on the continuous feature by using a discretization model, to obtain N discretization probabilities corresponding to the continuous feature, where N is an integer greater than 1, and the N discretization probabilities correspond to N preset meta-embeddings; and determine a vector representation value of the continuous feature based on the N discretization probabilities and the

N meta-embeddings.

[0027] Optionally, the data processing apparatus further includes a storage module, configured to store program instructions and data that are necessary for the data processing apparatus.

[0028] In a possible implementation, the apparatus includes a processor and a transceiver. The processor is configured to support the data processing apparatus in performing a corresponding function in the method provided in the first aspect. The transceiver is configured to indicate communication between the data processing apparatus and a sample data storage apparatus, for example, obtain the continuous feature from the sample data storage apparatus. Optionally, the apparatus further includes a memory. The memory is configured to be coupled to the processor, and stores program instruction and data that are necessary for the data processing apparatus.

[0029] In a possible implementation, when the apparatus is a chip in the data processing apparatus, the chip includes a processing module and a transceiver module. The transceiver module may be, for example, an input/output interface, a pin, or a circuit on the chip, and transmits the continuous feature to another chip or module coupled to the chip. The processing module may be, for example, a processor. The processor is configured to: perform discretization processing on the continuous feature by using a discretization model, to obtain N discretization probabilities corresponding to the continuous feature, where N is an integer greater than 1, and the N discretization probabilities correspond to N preset meta-embeddings; and determine a vector representation value of the continuous feature based on the N discretization probabilities and the N meta-embeddings. The processing module may execute computer-executable instructions stored in a storage unit, to support the data processing apparatus in performing the method provided in the first aspect. Optionally, the storage unit may be a storage unit in the chip, for example, a register or a cache. Alternatively, the storage unit may be a storage unit outside the chip, for example, a read-only memory (read-only memory, ROM), another type of static storage device capable of storing static information and instructions, or a random access memory (random access memory, RAM).

[0030] In a possible implementation, the apparatus includes a communication interface and a logic circuit. The communication interface is configured to obtain a continuous feature. The logic circuit is configured to: perform discretization processing on the continuous feature by using a discretization model, to obtain N discretization probabilities corresponding to the continuous feature, where N is an integer greater than 1, and the N discretization probabilities correspond to

N preset meta-embeddings; and determine a vector representation value of the continuous feature based on the N discretization probabilities and the N meta-embeddings.

[0031] The processor mentioned anywhere above may be a general-purpose central processing unit (Central Processing Unit, CPU), a microprocessor, an application-specific integrated circuit (application-specific integrated circuit, ASIC), or one or more integrated circuits configured to control program execution of the data transmission method in the foregoing aspects.

[0032] According to a third aspect, an embodiment of this application provides a computer-readable storage medium. The computer storage medium stores computer instructions, and the computer instructions are used to perform the method according to any one of the possible implementations of the foregoing aspects.

[0033] According to a fourth aspect, an embodiment of this application provides a computer program product including instructions. When the computer program product runs on a computer, the computer is enabled to perform the method in any one of the foregoing aspects.

[0034] According to a fifth aspect, this application provides a chip system. The chip system includes a processor, configured to support a data processing apparatus in implementing functions described in the foregoing aspects, for example, generating or processing data and/or information in the foregoing aspects. In a possible design, the chip system further includes a memory. The memory is configured to store program instructions and data that are necessary for the data processing apparatus, to implement functions in any one of the foregoing aspects. The chip system may include a chip, or may include a chip and another discrete component.

BRIEF DESCRIPTION OF DRAWINGS

[0035] FIG. 1 is a schematic diagram of an artificial intelligence main framework;

[0036] FIG. 2 is a schematic diagram of a processing procedure of a recommendation system;

[0037] FIG. 3 is a schematic diagram of a structure of a recommendation system;

[0038] FIG. 4 is a diagram of an example model architecture for structures of a discretization model and a deep learning model according to an embodiment of this application;

[0039] FIG. 5 is a schematic diagram of an embodiment of a data processing apparatus according to an embodiment of this application;

[0040] FIG. 6 is a schematic diagram of another embodiment of a data processing apparatus

according to an embodiment of this application;

[0041] FIG. 7 is a schematic diagram of an embodiment of a data processing method according to an embodiment of this application;

[0042] FIG. 8 is a schematic diagram of an application scenario of application recommendation display according to an embodiment of this application;

[0043] FIG. 9 is a schematic diagram of another embodiment of a data processing method according to an embodiment of this application; and

[0044] FIG. 10 is a schematic diagram of another embodiment of a data processing method according to an embodiment of this application.

DESCRIPTION OF EMBODIMENTS

[0045] To make objectives, technical solutions, and advantages of this application clearer, the following describes embodiments of this application with reference to accompanying drawings. It is clear that the described embodiments are merely some rather than all of the embodiments of this application. A person of ordinary skill in the art may learn that, as a new application scenario emerges, the technical solutions provided in embodiments of this application are also applicable to a similar technical problem.

[0046] In this specification, claims, and accompanying drawings of this application, the terms "first", "second", and the like are intended to distinguish between similar objects but do not necessarily indicate a specific order or sequence. It should be understood that the data termed in such a way is interchangeable in proper circumstances so that embodiments described herein can be implemented in orders other than the order illustrated or described herein. Moreover, the terms "include", "contain" and any other variants mean to cover the non-exclusive inclusion, for example, a process, method, system, product, or device that includes a list of steps or modules is not necessarily limited to those steps or modules, but may include other steps or modules not expressly listed or inherent to such a process, method, system, product, or device. Names or numbers of steps in this application do not mean that the steps in the method procedure need to be performed in a time/logical sequence indicated by the names or numbers. An execution sequence of the steps in the procedure that have been named or numbered can be changed based on a technical objective to be achieved, provided that same or similar technical effect can be achieved. Division into units

in this application is logical division and may be other division in an actual application. For example, a plurality of units may be combined or integrated into another system, or some features may be ignored or not performed. In addition, the displayed or discussed mutual couplings or direct couplings or communication connections may be implemented through some interfaces. The indirect couplings or communication connections between the units may be implemented in electronic or other similar forms. This is not limited in this application. In addition, the units or subunits described as separate parts may or may not be physically separate, may or may not be physical units, or may be distributed on a plurality of circuit units. Some or all of the units may be selected based on an actual requirement to implement the objectives of the solutions of this application.

[0047] To better understand embodiments of the present invention, concepts that may be used in the following embodiments are first explained herein:

[0048] A machine learning system trains parameters of a machine learning model based on input data and labels by using optimization methods such as a gradient descent method, and finally predicts unknown data by using a model obtained through training.

[0049] A personalized recommendation system is a system that performs analysis and modeling based on historical data of a user and according to a machine learning algorithm, and predicts a new request by using a model obtained through modeling, to provide a personalized recommendation result.

[0050] Continuous feature: Features can be classified into continuous features and discrete features based on whether feature values are continuous. A feature that can have any value within a specific range is referred to as a continuous feature. Values of the continuous feature are continuous, and two adjacent values may be infinitely divided, that is, an infinite quantity of values may be obtained.

[0051] A discrete feature is a feature whose feature values can be listed one by one in a specific order, and the values are usually integers, such as, user gender, nationality, and object type. For some scenarios that are continuous features in nature, values are also integers, that is, these features can be considered as discrete features.

[0052] Feature field and feature value: The feature field is a set of feature values. For example, gender is a feature field. The feature value is a value in a feature field. For example, in a gender feature field, both male and female are corresponding feature values.

[0053] Continuous feature discretization: Discretization processing is a common data preprocessing method and is used to convert a continuous value attribute into a discrete value attribute.

[0054] Vector representation means that a specific feature is represented by a vector. In this embodiment, the meta-embedding may be a preset initialized vector representation value, or may be an optimized vector representation value. The vector representation value of the continuous feature is a vector value that is determined based on the meta-embedding and that is used to represent a specific feature of the continuous feature. It may be understood that a dimension of the vector representation value of the continuous feature is the same as that of the meta-embedding. In an example solution, the meta-embedding may be a five-dimensional vector value, for example, (01010). The vector representation value corresponding to the continuous feature is also a five-dimensional vector value, for example, (11010).

[0055] FIG. 1 is a schematic diagram of an artificial intelligence main framework. The main framework describes an overall working procedure of an artificial intelligence system, and is applicable to a requirement of a general artificial intelligence field.

[0056] The following describes the artificial intelligence main framework from two dimensions: an "intelligent information chain" (a horizontal axis) and an "IT value chain" (a vertical axis).

[0057] The "intelligent information chain" reflects a series of processes from obtaining data to processing the data. For example, it may be a general process of intelligent information perception, intelligent information representation and formation, intelligent inference, intelligent decision-making, and intelligent execution and output. In this process, the data undergoes a refinement process of "data-information-knowledge-intelligence".

[0058] The "IT value chain" reflects a value brought by artificial intelligence to the information technology industry from an underlying infrastructure and information (providing and processing technology implementation) of human intelligence to an industrial ecological process of a system.

[0059] (1) Infrastructure

[0060] The infrastructure provides computing capability support for an artificial intelligence system, implements communication with the external world, and implements support by using a basic platform. The infrastructure communicates with the outside by using a sensor. A computing

capability is provided by a smart chip (a hardware acceleration chip such as a CPU, an NPU, a GPU, an ASIC, or an FPGA). The basic platform includes related platforms, for example, a distributed computing framework and a network, for assurance and support, including cloud storage and computing, an interconnection network, and the like. For example, the sensor communicates with the outside to obtain data, and the data is provided to a smart chip in a distributed computing system provided by the basic platform for computing.

[0061] (2) Data

[0062] The data at an upper layer of an infrastructure indicates a data source in the field of artificial intelligence. The data relates to a graph, an image, speech, and text, further relates to internet of things data of a conventional device, and includes service data of an existing system and perception data such as force, displacement, a liquid level, a temperature, and humidity.

[0063] (3) Data processing

[0064] Data processing usually includes a manner such as data training, machine learning, deep learning, searching, inference, or decision-making.

[0065] Machine learning and deep learning may mean performing symbolic and formalized intelligent information modeling, extraction, preprocessing, training, and the like on data.

[0066] Inference is a process in which a human intelligent inference manner is simulated in a computer or an intelligent system, and machine thinking and problem resolving are performed based on formal information according to an inference control policy. A typical function is searching and matching.

[0067] Decision-making is a process in which a decision is made after intelligent information is inferred, and usually provides functions such as classification, ranking, and prediction.

[0068] (4) General capability

[0069] After data processing mentioned above is performed on data, some general capabilities may further be formed based on a data processing result, for example, an algorithm or a general system, such as translation, text analysis, computer vision processing, speech recognition, and image recognition.

[0070] (5) Smart product and industry application

[0071] The smart product and the industry application are a product and an application of an artificial intelligence system in various fields, and are package of an overall solution of artificial intelligence, so that decision-making for intelligent information is productized and an application

is implemented. Application fields mainly include smart manufacturing, smart transportation, smart home, smart health care, smart security protection, autonomous driving, a safe city, a smart terminal, and the like.

[0072] A continuous feature processing method provided in embodiments of this application is applied to application scenarios of various machine learning systems. In an example solution, the machine learning system may be a personalized recommendation system. In the personalized recommendation system, a specific processing procedure of the personalized recommendation system may be shown in FIG. 2. Raw data is first collected, and then feature processing is performed to obtain feature data that can be input to a model for training. The feature data is input to an initialized model, and a final recommendation model is obtained through model training. Finally, an online service module of the recommendation model generates a recommendation list for a user. A main structure of the recommendation system may be shown in FIG. 3. The structure includes user data collection, a log storage module, an offline training module, a prediction model, and an online prediction module. Basic running logic of the recommendation system as follows: The user performs a series of behaviors such as browsing, clicking, commenting, and downloading in a front-end display list to generate behavior data (that is, a front end collects user data), and then the behavior data is stored in the log storage module. The personalized recommendation system performs offline model training by using data including a user behavior log, generates the prediction model after training convergence, deploys the model in an online service environment, and provides a recommendation result based on an access requested by the user, an object feature, and context information. Then the user generates feedback to the recommendation result to form a new round of user data. In the flowchart shown in FIG. 2 or the diagram of the structure shown in FIG. 3, feature processing of the raw data or user data (that is, log data) is a basis of model training and an online service, and is important in the machine learning system.

[0073] The following describes in detail a function and a relationship between feature processing and a machine learning model or a deep learning model with reference to a diagram of a model architecture. An example solution is shown in FIG. 4. Deep learning is used as an example of the diagram of the model architecture. The diagram of the model architecture includes an input layer, a vectorized representation layer (Embedding), a multi-layer neural network (Multi-layer Perceptron, MLP) and feature interaction layer, and an output layer. The input layer inputs a data feature, including a continuous feature and a discrete feature. The data feature is processed at the

vectorized representation layer. Details are as follows: For the discrete feature, sparse coding is performed by using one-hot, then a corresponding embedding vector representation is extracted from an embedding table based on an ID of sparse coding, and finally embedding vector representations of all discrete features are sequentially spliced. For the continuous feature, vectorized representation is performed after discretization. After the vectorized representation of each feature is obtained, embeddings of all the continuous features and discrete features are spliced as input data of the neural network. The input data is processed by using the MLP and feature interaction layer and the output layer to obtain a predicted value. It can be learned that discrete processing of the continuous feature is not only used for representation modeling of the continuous feature, but also directly affects parameter learning of an upper-layer MLP and feature interaction layer. In addition, gradient backpropagation indirectly affects learning of parameters related to the discrete feature. This plays an important role in final model prediction effect.

[0074] As shown in FIG. 5, an embodiment of this application provides a data processing apparatus 500, including a log data storage module 501, a continuous feature discretization module 502, a vector representation aggregation module 503, a vector representation splicing module 504, and an output module 505. The log data storage module 501 is configured to collect and store behavior data fed back by a front-end user. The continuous feature discretization module 502 outputs N discretization probabilities for a continuous feature by using linear transformation and softmax, a multiclass neural network, or an attention network. The vector representation aggregation module 503 is configured to determine, based on the N discretization probabilities and N preset meta-embeddings, a vector representation value corresponding to the continuous feature. Finally, the vector representation splicing module 504 sequentially splices vector representation values of all features as input data of the machine learning or deep learning model. Finally, the output module 505 outputs the input data to the machine learning model or the deep learning model.

[0075] In a feature processing process, the log data storage module 501 further includes the discrete feature. Therefore, the data processing apparatus 500 further needs to process the discrete feature. In addition, the continuous feature in log data may also be first normalized. Therefore, in an example solution, a data processing apparatus 600 combined with machine learning or deep learning may be shown in FIG. 6, and includes: a log data storage module 601, a normalization module 602, a continuous feature discretization module 603, a vector representation aggregation module 604, a discrete feature processing module 605, a vector representation splicing module

606, and an output module 607. The log data storage module 601 is configured to collect and store behavior data fed back by a front-end user. The normalization module 602 is configured to normalize a continuous feature in the behavior data, and then input a normalized continuous feature to the continuous feature discretization module 603. The continuous feature discretization module 603 outputs N discretization probabilities for the continuous feature by using linear mapping and softmax, a multiclass neural network, or an attention network. The vector representation aggregation module 604 is configured to determine, based on the N discretization probabilities and N preset meta-embeddings, a vector representation value corresponding to the continuous feature. The discrete feature processing module 605 is configured to perform sparse coding on a discrete feature in the behavior data by using one-hot, and then extract a corresponding embedding vector representation value from an embedding table based on an ID of sparse coding. Finally, the vector representation splicing module 606 sequentially splices the vector representation value of the discrete feature and the vector representation value of the continuous feature as input data of the machine learning model or the deep learning model. Finally, the output module 607 outputs the input data to the machine learning model or the deep learning model.

[0076] An embodiment of this application provides a continuous feature processing method. For details, refer to FIG. 7. Specific steps of the continuous feature processing method are as follows.

[0077] 701: Obtain a continuous feature.

[0078] In a model training process or a model application process, behavior data of a front-end user is stored as sample data in the model training process or the model application process. The sample data includes a continuous feature and/or a discrete feature. This embodiment mainly describes the continuous feature processing method. The data processing apparatus obtains the continuous feature from the sample data.

[0079] Optionally, the continuous feature may be a continuous feature normalized by the data processing apparatus. In an example solution, the continuous feature can be normalized according to a formula $\bar{X} = (x - \min) / (\max - \min)$ $\bar{x} = (x - Min) / (Max - Min)$.

[0080] 702: Perform discretization processing on the continuous feature by using a discretization model, to obtain N corresponding discretization probabilities in the continuous feature, where N is an integer greater than 1, and the N discretization probabilities correspond to N preset meta-embeddings.

[0081] After obtaining the continuous feature, the data processing apparatus inputs the continuous feature into the discretization model, and performs discretization processing on the continuous feature to obtain the N corresponding discretization probabilities in the continuous feature.

5 **[0082]** In this embodiment, a specific manner in which the data processing apparatus performs discretization processing on the continuous feature by using the discretization model, to obtain the N discretization probabilities corresponding to the continuous feature may be as follows: The data processing apparatus presets an initial variable in the discretization model, determines, based on the initial variable, N mapping values corresponding to the continuous feature, and finally
10 calculates the N discretization probabilities of the continuous feature based on the mapping value. In this embodiment, the initial variable may be an initial mapping field. For the continuous feature, the N mapping values corresponding to the continuous feature may be calculated based on the initial variable. Then, one corresponding probability is calculated based on each of the N mapping values, to obtain N probabilities. In this case, the N probabilities are used as the N discretization
15 probabilities corresponding to the continuous feature.

[0083] Optionally, N is greater than or equal to 20 and less than or equal to 100. Within this range, the discretization model may have good application effect.

[0084] In the recommendation system provided in this embodiment of this application, the discretization model may be a multiclass neural network, an attention network, or linear mapping
20 and softmax.

[0085] In an example solution, linear mapping and softmax are used as an example for description.

[0086] A quantity of buckets into which the continuous feature is discretized is preset to h (in other words, it is equivalent to that a quantity of distribution areas into which the continuous
25 feature is discretized is preset to h, and it may be understood that h is equal to N), where temperature is τ , and an initialized linear mapping variable is $W_{logit} = R^{1 \times h}$.

[0087] Then, in a first step, linear mapping is performed on the continuous feature according to a linear mapping formula to obtain the N mapping values, where the linear mapping formula is
$$cont_{logit} = cont \cdot W_{logit}.$$

30 **[0088]** In a second step, the N discretization probabilities corresponding to the continuous

feature are calculated according to a discretization formula and based on the N mapping values,

where the discretization formula is $cont_p_k = e^{\frac{1}{\tau} cont_{logit}^k} / \sum_{i=1}^h e^{\frac{1}{\tau} cont_{logit}^i} \cdot W_{logit}$ indicates a

linear mapping variable, R indicates a real number field, h indicates the quantity of buckets into which the continuous feature is discretized, $cont_{logit}$ indicates a representation obtained

5 after linear mapping of the continuous feature, $cont_p_k$ indicates a probability that the continuous feature is discretized to a k^{th} bucket, $cont_{logit}^k$ indicates a k^{th} neuron output after linear mapping of the continuous feature, τ indicates a temperature control coefficient of softmax, and $cont_{logit}^i$ indicates an i^{th} neuron output after linear mapping of the continuous feature.

[0089] After the first step and the second step, the data processing apparatus obtains
10 probability distribution $cont_p$ with a size of $1 \times h$, which indicates a probability that the continuous feature is discretized to different buckets. In an example solution, age is used as an example. It is assumed that an age value is 20, and four buckets h1, h2, h3, and h4 are allocated to this age field. The foregoing steps are performed to obtain 1×4 probability distribution: 0.1, 0.15, 0.7, and 0.05 1×4 的概率分布, 就是分别0.1, 0.15, 0.7, 0.05, 此时4个值. That is, the
15 probability distribution of the age value 20 in the four buckets is (0.1, 0.15, 0.7, and 0.05). In addition, it can be learned from the foregoing result that a probability that the age value 20 is distributed in a third bucket is highest.

[0090] In another example solution, the multiclass neural network is used as an example for description.

20 [0091] A quantity of buckets into which the continuous feature is discretized is preset to h (in other words, it is equivalent to that a quantity of distribution areas into which the continuous feature is discretized is preset to h). Then, in a first step, all feature values $cont$ in the continuous feature are input into a multilayer perceptron MLP. A formula of an l^{th} layer of the MLP is $O^l = \sigma(O^{l-1} \cdot W^l + b^l)$ $O^l = \sigma(O^{l-1} \cdot W^l + b^l)$, where σ is an activation function, and may
25 be a sigmoid function or a tanh function. W^l is a weight parameter of the l^{th} layer, b^l is a deviation parameter of the l^{th} layer, O^{l-1} is an output of a previous layer, and $O^0 = cont$ $O^0 = cont$, to be specific, an original feature value is an input of a first layer.

[0092] In a second step, an activation function at a last layer of the MLP is set to softmax, a formula is

$$\text{cont}_p = O^L = \text{softmax}(O^{L-1} \cdot W^L + b^L)$$

$\text{cont}_p = O^L = \text{softmax}(O^{L-1} \cdot W^L + b^L)$, and a formula of softmax is $S_k = e^k / \sum_{i=1}^h e^i$

$S_k = e^k / \sum_{i=1}^h e^i$, where cont_p indicates a probability that the continuous feature is discretized to

5 each bucket. $W_{\text{logit}} = R^{1 \times h}$

[0093] 703: Determine a vector representation value of the continuous feature based on the N discretization probabilities and the N meta-embeddings.

[0094] The data processing apparatus initializes N vector representation values (also referred to as Meta Embedding) for each continuous feature field in the continuous feature, that is,

10 $V = R^{h \times e}$, where e represents a description dimension (also referred to as Embedding size) of

the vector representation value. The vector representation value represents all feature values of one continuous feature field. Then, the data processing apparatus aggregates, based on the discretization probability obtained in the step 702, the meta-embeddings by using an aggregate function, to obtain the corresponding vector representation value (also referred to as an embedding

15 vx_cont vx_cont) in the continuous feature. The aggregate function is used to aggregate the N corresponding meta-embeddings in the continuous feature field into one embedding corresponding to the feature value.

[0095] In an example solution, a vector representation in the meta-embedding corresponding to the continuous feature field one-to-one corresponds to a discretization probability of the feature

20 value. It is assumed that the discretization probabilities are (a1, a2, a3, a4), and the vector representation values are (b1, b2, b3, b4). In this case, a1 corresponds to b1, a2 corresponds to b2,

a3 corresponds to b3, and a4 corresponds to b4. The aggregate function is used to aggregate the meta-embeddings (b1, b2, b3, b4) corresponding to the feature value into one embedding. In an example solution, (b1, b2, b3, b4) may be aggregated to obtain b3 as the vector representation

25 value corresponding to the feature value.

[0096] Optionally, there may be a plurality of aggregate functions. Details may be as follows:

[0097] In a possible implementation, the aggregate function is Max-Pooling. A largest value is determined from the discretization probabilities obtained in the step 702, and then an embedding

corresponding to the largest value is obtained from the meta-embeddings as the vector representation value corresponding to the feature value. A calculation formula of the vector representation value is $vx_cont = E^k$, where $k = \arg \max_h \{cont_p_h\}$. For example, it is assumed that the discretization probabilities are (a1, a2, a3, a4), and the meta-embeddings are (b1, b2, b3, b4). In this case, a1 corresponds to b1, a2 corresponds to b2, a3 corresponds to b3, and a4 corresponds to b4. If a value of a3 is largest, b3 is used as the vector representation value of the feature value.

[0098] In another possible implementation, the aggregate function is Top-K-Sum. k largest probabilities are selected from the discretization probabilities obtained in the step 702, then embeddings corresponding to the probabilities are obtained from the meta-embeddings, and the embeddings are summed up to be used as the vector representation value corresponding to the feature value. A calculation formula of the vector representation value is $vx_cont = \sum_{k=1}^K E^k$, where $k = \arg \text{top}k_h \{cont_p_h\}$. For example, it is assumed that the discretization probabilities are (a1, a2, a3, a4), and the meta-embeddings are (b1, b2, b3, b4). In this case, a1 corresponds to b1, a2 corresponds to b2, a3 corresponds to b3, and a4 corresponds to b4. If a value of k is 2, and values of a2 and a3 are respectively first two digits, the vector representation value of the feature value is b2+b3.

[0099] In another possible implementation, the aggregate function is Weighted-Average. The discretization probabilities are obtained in the step 702, and then, weighted summation is performed on the N probabilities and the meta-embeddings to obtain a weighted sum as the vector representation value corresponding to the feature value. A calculation formula of the vector representation value is $vx_cont = \sum_{k=1}^h cont_p_k \times E^k$. For example, it is assumed that the discretization probabilities are (a1, a2, a3, a4), and the meta-embeddings are (b1, b2, b3, b4). In this case, a1 corresponds to b1, a2 corresponds to b2, a3 corresponds to b3, and a4 corresponds to b4. The vector representation value of the feature value is equal to (a1×b1+a2×b2+a3×b3+a4×b4).

[00100] In an example solution, age is used as an example. It is assumed that an age value is 20, and four buckets h1, h2, h3, and h4 are allocated to this age field. The foregoing steps are performed to obtain 1×4 probability distribution: 0.1, 0.15, 0.7, and 0.05. 1×4的概率分布，就是分别0.1， 0.15， 0.7， 0.05，此时4个值. That is, it can be learned from the probability distribution of the age value 20 in the four buckets that a probability that the

age value 20 is distributed in a third bucket is highest. If the data processing apparatus selects the aggregate function Max-Pooling for calculation, the data processing apparatus selects a bucket whose probability is 0.7, and uses an embedding corresponding to the bucket as the vector representation value of the feature value.

5 **[00101]** In this embodiment, the data processing apparatus calculates a discretization probability, that has more than one dimension, for a feature value of each continuous feature by using the discretization model, presets a meta-embedding meta-embedding, that has more than one dimension, for each continuous feature field in the continuous feature, determines, for a feature value, a vector representation value from the meta-embedding by using an aggregate function and
10 the discretization probability. In this way, for the continuous feature, compared with the conventional technology, in this embodiment, the vector representation value obtained through learning has a better representation capability, thereby helping improve accuracy of a prediction result.

[00102] In this embodiment, the data processing method shown in FIG. 7 may be applied to a
15 plurality of application scenarios, for example, a recommendation model or a search model. The following describes an application scenario of the data processing method provided in this embodiment of this application by using a click-through rate prediction scenario in a mobile phone application market recommendation system shown in FIG. 8 as an example. In the application scenario shown in FIG. 8, a specific data model of the application scenario is a click-through rate
20 prediction model (or a recommendation model), and the click-through rate prediction model is mainly used in a "Top apps" column shown in FIG. 8 to recommend, based on a user feature (for example, a user age or a user gender) and an object feature (an application), corresponding applications (as shown in FIG. 8, applications such as "App 1" and "App 2" displayed in top applications) to the user. A specific processing procedure of the data model may be as follows:
25 obtaining the user feature and the object feature, and then processing a discrete feature in the user feature and the object feature by using conventional processing. That is, one-hot encoding is first performed, and then an embedding representation is obtained through an embedding lookup operation. A continuous feature in the user feature and the object feature is processed by using the method shown in FIG. 7 to obtain a corresponding vector representation value, and then a vector
30 representation value of the discrete feature and the vector representation value of the continuous feature in the user feature and the object feature are input to the recommendation model

corresponding to the application scenario shown in FIG. 8 as input feature representations of the model, to obtain a recommendation result.

[00103] In this embodiment, the recommendation model may further calculate, based on a prediction result and an actual result, a loss value (loss) by using a loss function, and complete parameter update of the recommendation model and the discretization model based on the loss. During an online service, the data processing apparatus may be used as a part of the recommendation model, to complete discretization of the continuous feature online and learn an embedding of each continuous feature. Compared with technologies, such as an artificial feature engineering technology and a bucket discretization preprocessing technology, in this embodiment, processing time can be saved. When incremental training is used, a weight parameter of the discretization model may be adjusted with latest data distribution, so that data utilization efficiency is higher.

[00104] The continuous feature processing method provided in this application may be described below by using specific experimental data. This embodiment provides three datasets: a Criteo dataset, an AutoML dataset, and a Huawei industrial dataset. In an example solution, statistical information of each dataset is shown in Table 1.

Table 1

Dataset name	Dataset size	Quantity of all features	Quantity of discrete features	Quantity of continuous features
Criteo	45.8 M	39	26	13
AutoML	4.69 M	74	51	23
Huawei industry	8.75 M	85	44	41

[00105] M is equal to 10 raised to the power of 6.

[00106] In this embodiment, an experiment evaluation indication is an AUC (that is, area under curve), and continuous feature processing technologies are a normalization method, an isometric discretization method, a logarithm method, DeepGBM, and the continuous feature processing technology provided in this embodiment of this application. Experiments are performed on the foregoing three datasets. For example, DeepFM is used as a top-level depth model. Experimental results are shown in Table 2.

Table 2

	Criteo	AutoML	Industrial
DeepFM-Norm	0.8107	0.7523	0.7248
DeepFM-EDD	0.8125	0.7545	0.7251
DeepFM-LD	0.8138	0.7527	0.7265
DeepFM-TD	0.8130	0.7531	0.7262
DeepFM-AutoDis	0.8149	0.7556	0.7277
% Impr.	0.14%	0.15%	0.17%

[00107] AutoDis indicates a framework or an apparatus for performing the data processing method in embodiments of this application. It can be learned from the foregoing results that the technical solution provided in this embodiment can achieve a better result.

[00108] In addition, the technical solution provided in this embodiment may be applied to different models, and also has improvement effect. In this embodiment, several common depth models in the industry are selected for click-through rate (click-through-rate, CTR) prediction, including a feedforward neural network (factorisation-machine supported neural networks, FNN), Wide&Deep (that is, a joint training model of a wide model with logistic regression having a sparse feature and transformation and a deep model of a feedforward neural network having an embedding layer and a plurality of hidden layers), DeepFM, a DCN, an IPNN, and the like. Experimental results are shown in Table 3.

Table 3

	Criteo		AutoML		Industrial	
	Basic model	+AutoDis	Basic model	+AutoDis	Basic model	+AutoDis
FNN	0.8059	0.8091	0.7383	0.7448	0.7271	0.7286
Wide&Deep	0.8097	0.8121	0.7407	0.7442	0.7275	0.7287
DeepFM	0.8108	0.8149	0.7525	0.7556	0.7262	0.7277
DCN	0.8091	0.8128	0.7489	0.7508	0.7262	0.7281
IPNN	0.8101	0.8135	0.7519	0.7541	0.7269	0.7283

[00109] It can be learned from the foregoing results shown in Table 3 that the continuous feature processing method provided in this embodiment is added to these common depth models. This can significantly improve model performance, and show that the continuous feature processing method has good compatibility.

5 **[00110]** FIG. 9 is a possible schematic diagram of a structure of a data processing apparatus 900 in the foregoing embodiment. The data processing apparatus 900 may be configured as the foregoing data processing apparatus. The data processing apparatus 900 may include a processor 902, a computer-readable storage medium/memory 903, a transceiver 904, an input device 905, an output device 906, and a bus 901. The processor, the transceiver, the computer-readable storage medium, and the like are connected by using the bus. A specific connection medium between the
10 foregoing components is not limited in this embodiment of this application.

[00111] In an example, the transceiver 904 obtains a continuous feature.

[00112] The processor 902 performs discretization processing on the continuous feature by using a discretization model, to obtain N discretization probabilities corresponding to the
15 continuous feature, where N is an integer greater than 1, and the N discretization probabilities correspond to N preset meta-embeddings; and determines a vector representation value of the continuous feature based on the N discretization probabilities and the N meta-embeddings.

[00113] In still another example, the processor 902 may run an operating system to control functions between devices and components. The transceiver 904 may include a baseband circuit
20 and a radio frequency circuit. For example, the vector representation value may be processed by using the baseband circuit and the radio frequency circuit, and then sent to a recommendation system or a search system.

[00114] The transceiver 904 and the processor 902 may implement a corresponding step in any one of the embodiments in FIG. 7 to FIG. 8. Details are not described herein again.

25 **[00115]** It may be understood that FIG. 9 shows only a simplified design of the data processing apparatus. In an actual application, the data processing apparatus may include any quantity of transceivers, processors, memories, and the like, and all data processing apparatuses that can implement this application fall within the protection scope of this application.

[00116] The processor 902 in the foregoing apparatus 900 may be a general-purpose processor,
30 for example, a CPU, a network processor (network processor, NP), or a microprocessor, or may be an ASIC, or one or more integrated circuits configured to control program execution in the

solutions of this application. Alternatively, the processor 902 may be a digital signal processor (digital signal processor, DSP), a field-programmable gate array (field-programmable gate array, FPGA), or another programmable logic device, a discrete gate or a transistor logic device, or a discrete hardware component. Alternatively, a controller/processor may be a combination of processors implementing a computing function, for example, a combination of one or more microprocessors, or a combination of the DSP and the microprocessor. The processor usually performs logical and arithmetic operations based on program instructions stored in the memory.

[00117] The bus 901 may be a peripheral component interconnect (peripheral component interconnect, PCI for short) bus, an extended industry standard architecture (extended industry standard architecture, EISA for short) bus, or the like. The bus may be classified into an address bus, a data bus, a control bus, or the like. For ease of representation, only one thick line is used to represent the bus in FIG. 8, but this does not mean that there is only one bus or only one type of bus.

[00118] The computer-readable storage medium/memory 903 may further store an operating system and another application. Specifically, the program may include program code, and the program code includes computer operation instructions. More specifically, the memory may be a ROM, another type of static storage device that can store static information and instructions, a RAM, another type of dynamic storage device that can store information and instructions, a magnetic disk memory, or the like. The memory 903 may be a combination of the foregoing memories. In addition, the computer-readable storage medium/memory may be located in the processor, or may be located outside the processor, or distributed in a plurality of entities including a processor or a processing circuit. The computer-readable storage medium/memory may be specifically embodied in a computer program product. For example, the computer program product may include a computer-readable medium in a packaging material.

[00119] Alternatively, this embodiment of this application provides a universal processing system. For example, the universal processing system is usually referred to as a chip. The universal processing system includes one or more microprocessors that provide a processor function and an external memory that provides at least a part of a storage medium. All these components are connected to other supporting circuits by using an external bus architecture. When instructions stored in the memory are executed by the processor, the processor is enabled to perform some or all of the steps of the data retransmission method performed by a data processing apparatus in the

embodiments shown in FIG. 7 and FIG. 8, and/or another process of the technology described in this application.

[00120] Method or algorithm steps described in combination with the content disclosed in this application may be implemented by hardware, or may be implemented by a processor by executing software instructions. The software instructions may include a corresponding software module. The software module may be located in a RAM memory, a flash memory, a ROM memory, an EPROM memory, an EEPROM memory, a register, a hard disk, a removable magnetic disk, a CD-ROM, or a storage medium in any other form known in the art. For example, a storage medium is coupled to a processor, so that the processor can read information from the storage medium and write information into the storage medium. Certainly, the storage medium may alternatively be a component of the processor. The processor and the storage medium may be disposed in an ASIC. In addition, the ASIC may be disposed in a terminal. Certainly, the processor and the storage medium may alternatively exist in the data processing apparatus as discrete components.

[00121] For details, refer to FIG. 10. An embodiment of this application is a possible schematic diagram of a structure of a data processing apparatus 1000 in the foregoing embodiment. The data processing apparatus 1000 includes an obtaining module 1001 and a processing module 1002. The obtaining module 1001 is connected to the processing module 1002 by using a bus. The data processing apparatus 1000 may be the data processing apparatus in the foregoing method embodiment, or may be configured as one or more chips in the data processing apparatus. The data processing apparatus 1000 may be configured to perform some or all functions of the data processing apparatus in the foregoing method embodiment. In addition, FIG. 10 shows only some modules of the data processing apparatus in this embodiment of this application.

[00122] The obtaining module 1001 is configured to obtain a continuous feature.

[00123] The processing module 1002 is configured to: perform discretization processing on the continuous feature by using a discretization model, to obtain N discretization probabilities corresponding to the continuous feature, where N is an integer greater than 1, and the N discretization probabilities correspond to N preset meta-embeddings; and determine a vector representation value of the continuous feature based on the N discretization probabilities and the N meta-embeddings. The processing module 1002 may further perform the method performed by the continuous feature discretization module 502 and the vector representation aggregation module 503 in FIG. 5. Alternatively, the processing module 1002 may further perform the method

performed by the continuous feature discretization module 603 and the vector representation aggregation module 604 in FIG. 6.

[00124] Optionally, the data processing apparatus 1000 further includes a storage module. The storage module may store computer-executable instructions. In this case, the storage module is coupled to the processing module, so that the processing module can execute the computer-executable instructions stored in the storage module, to implement functions of the data processing apparatus in the foregoing method embodiment. In an example, the storage module optionally included in the data processing apparatus 1000 may be a storage unit in the chip, for example, a register or a cache. Alternatively, the storage unit may be a storage unit outside the chip, for example, a ROM, another type of static storage device that can store static information and instructions, or a RAM.

[00125] It should be understood that a procedure performed between the modules of the data processing apparatus in the embodiment corresponding to FIG. 10 is similar to a procedure performed by the data processing apparatus in the method embodiment corresponding to FIG. 7. Details are not described herein again.

[00126] It may be clearly understood by a person skilled in the art that, for ease and brevity of description, for a detailed working process of the foregoing system, apparatus, and unit, refer to a corresponding process in the foregoing method embodiment. Details are not described herein again.

[00127] In the several embodiments provided in this application, it should be understood that the disclosed system, apparatus, and method may be implemented in another manner. For example, the described apparatus embodiments are merely examples. For example, division into the units is merely logical function division and may be other division during actual implementation. For example, a plurality of units or components may be combined or integrated into another system, or some features may be ignored or not performed. In addition, the displayed or discussed mutual couplings or direct couplings or communication connections may be implemented through some interfaces. The indirect couplings or communication connections between the apparatuses or units may be implemented in an electrical form, a mechanical form, or another form.

[00128] The units described as separate parts may or may not be physically separate, and parts displayed as units may or may not be physical units, that is, may be located in one location, or may be distributed on a plurality of network units. Some or all of the units may be selected based on an actual requirement to achieve the objectives of the solutions of embodiments.

[00129] In addition, functional units in embodiments of this application may be integrated into one processing unit, or each of the units may exist alone physically, or two or more units are integrated into one unit. The integrated unit may be implemented in a form of hardware, or may be implemented in a form of a software functional unit.

5 **[00130]** When the integrated unit is implemented in the form of the software functional unit and sold or used as an independent product, the integrated unit may be stored in a computer-readable storage medium. Based on such an understanding, the technical solutions of this application essentially, or the part contributing to the conventional technology, or all or some of the technical solutions may be implemented in a form of a software product. The computer software product is
10 stored in a storage medium and includes several instructions for instructing a computer device (which may be a personal computer, a server, or a network device) to perform all or some of the steps of the method described in embodiments of this application. The foregoing storage medium includes any medium that can store program code, such as a USB flash drive, a removable hard disk, a read-only memory (ROM, Read-Only Memory), a random access memory (RAM, Random
15 Access Memory), a magnetic disk, or an optical disc.

CLAIMS

What is claimed is:

1. A data processing method, comprising:

obtaining a continuous feature;

performing discretization processing on the continuous feature by using a discretization model, to obtain N discretization probabilities corresponding to the continuous feature, wherein N is an integer greater than 1, and

the N discretization probabilities correspond to N preset meta-embeddings; and

determining a vector representation value of the continuous feature based on the N discretization probabilities and the N meta-embeddings.

2. The method according to claim 1, wherein the performing discretization processing on the continuous feature by using a discretization model, to obtain N discretization probabilities corresponding to the continuous feature comprises:

presetting an initial variable in the discretization model;

determining, based on the initial variable, N mapping values corresponding to the continuous feature; and

calculating the N discretization probabilities of the continuous feature based on the mapping value.

3. The method according to claim 1 or 2, wherein the discretization model is a multiclass neural network, an attention network, or linear mapping and softmax.

4. The method according to claim 3, wherein when the discretization model is linear mapping and softmax, the presetting an initial variable in the discretization model comprises:

presetting an initialized linear mapping variable to $W_{logit} = R^{1 \times h}$, wherein the initialized linear mapping variable is the initial variable;

the determining, based on N, N mapping values corresponding to the continuous feature comprises:

determining, according to a linear mapping formula, the N mapping values corresponding to the continuous feature, wherein

the linear mapping formula is $cont_{logit} = cont \cdot W_{logit}$; and

the calculating the N discretization probabilities of the continuous feature based on the

mapping value comprises:

calculating, according to a discretization formula, a probability corresponding to each of the N mapping values to obtain N probabilities, wherein the discretization formula is

$$cont_p_k = e^{\frac{1}{\tau} cont_{logit}^k} / \sum_{i=1}^h e^{\frac{1}{\tau} cont_{logit}^i}$$

, and the N probabilities are used as the N discretization

5 probabilities; and

W_{logit} indicates a linear mapping variable, R indicates a real number field, h indicates a quantity of buckets into which the continuous feature is discretized, h is equal to N, $cont_{logit}$ indicates a representation obtained after linear mapping of the continuous feature, $cont_p_k$ indicates a probability that the continuous feature is discretized to a k^{th} bucket, $cont_{logit}^k$ indicates a k^{th} neuron output after linear mapping of the continuous feature, τ indicates a temperature control coefficient of softmax, and $cont_{logit}^i$ indicates an i^{th} neuron output after linear mapping of the continuous feature.

10

5. The method according to any one of claims 1 to 4, wherein the determining a vector representation value of the continuous feature based on the N discretization probabilities and the N meta-embeddings comprises:

15

determining the vector representation value of the continuous feature by using an aggregate function and based on the N discretization probabilities and the N meta-embeddings, wherein the aggregate function is Max-Pooling, Top-K-Sum, or Weighted-Average.

6. The method according to any one of claims 1 to 5, wherein the method further comprises: inputting a user feature and an object feature into a recommendation model or a search model to obtain a prediction result, wherein

20

the user feature or the object feature comprises the vector representation value; and the user feature or the to-be-recommended object feature represents the continuous feature by using the vector representation value.

25

7. The method according to claim 6, wherein the method further comprises: obtaining an actual result; and adjusting a weight parameter of the discretization model based on the prediction result and the actual result by using a loss function.

8. The method according to any one of claims 1 to 7, wherein N is greater than or equal to 20 and less than or equal to 100.

9. The method according to any one of claims 1 to 8, wherein the continuous feature is feature data having a continuous statistical feature value in sample data.

5 10. The method according to claim 9, wherein the continuous feature comprises, but is not limited to, an age feature, a click count feature, and a score feature in a recommendation system.

11. The method according to any one of claims 1 to 10, wherein the continuous feature is a normalized continuous feature.

12. A data processing apparatus, comprising:

10 an obtaining module, configured to obtain a continuous feature; and

a processing module, configured to: perform discretization processing on the continuous feature by using a discretization model, to obtain N discretization probabilities corresponding to the continuous feature, wherein N is an integer greater than 1, and the N discretization probabilities correspond to N preset meta-embeddings; and determine a vector representation value of the continuous feature based on the N discretization probabilities and the N meta-embeddings.

13. The apparatus according to claim 12, wherein the processing module is specifically configured to: preset an initial variable in the discretization model, determine, based on the initial variable, N mapping values corresponding to the continuous feature, and calculate the N discretization probabilities of the continuous feature based on the mapping value.

20 14. The apparatus according to claim 12 or 13, wherein the discretization model is a multiclass neural network, an attention network, or linear mapping and softmax.

15. The apparatus according to claim 14, wherein when the discretization model is linear mapping and softmax, the processing module is specifically configured to: preset an initialized linear mapping variable to $W_{logit} = R^{1 \times h}$, wherein the initialized linear mapping variable is the initial variable;

25 determine, according to a linear mapping formula, the N mapping values corresponding to the continuous feature, wherein

the linear mapping formula is $cont_{logit} = cont \cdot W_{logit}$; and

calculate, according to a discretization formula, a probability corresponding to each of the N mapping values to obtain N probabilities, wherein the discretization formula is

$cont_p_k = e^{\frac{1}{\tau} cont_{logit}^k} / \sum_{i=1}^h e^{\frac{1}{\tau} cont_{logit}^i}$, and the N probabilities are used as the N discretization probabilities; and

W_{logit} indicates a linear mapping variable, R indicates a real number field, h indicates a quantity of buckets into which the continuous feature is discretized, h is equal to N, $cont_{logit}$ indicates a representation obtained after linear mapping of the continuous feature, $cont_p_k$ indicates a probability that the continuous feature is discretized to a k^{th} bucket, $cont_{logit}^k$ indicates a k^{th} neuron output after linear mapping of the continuous feature, τ indicates a temperature control coefficient of softmax, and $cont_{logit}^i$ indicates an i^{th} neuron output after linear mapping of the continuous feature.

16. The apparatus according to any one of claims 12 to 15, wherein the processing module is specifically configured to: determine, by using an aggregate function and based on the N discretization probabilities and the N meta-embeddings, the vector representation value corresponding to the continuous feature, wherein the aggregate function is Max-Pooling, Top-K-Sum, or Weighted-Average.

17. The apparatus according to any one of claims 12 to 16, wherein the processing module is further configured to input a user feature and an object feature into a recommendation model or a search model to obtain a prediction result, wherein the user feature or the object feature comprises the vector representation value; and the user feature or the to-be-recommended object feature represents the continuous feature by using the vector representation value.

18. The apparatus according to claim 17, wherein the obtaining module is further configured to obtain an actual result; and

the processing module is further configured to adjust a weight parameter of the discretization model based on the prediction result and the actual result by using a loss function.

19. The apparatus according to any one of claims 12 to 18, wherein N is greater than or equal to 20 and less than or equal to 100.

20. The apparatus according to any one of claims 12 to 19, wherein the continuous feature is feature data having a continuous statistical feature value in sample data.

21. The apparatus according to claim 20, wherein the continuous feature comprises, but is not

limited to, an age feature, a click count feature, and a score feature in a recommendation system.

22. The apparatus according to any one of claims 12 to 20, wherein the continuous feature is a normalized continuous feature.

5 23. A data processing apparatus, comprising at least one processor and a memory, wherein the processor is configured to be coupled to the memory, and the processor invokes instructions stored in the memory to control the data processing apparatus to perform the method according to any one of claims 1 to 11.

10 24. A computer storage medium, wherein the computer-storage medium stores computer instructions, and the computer instructions are used to perform the method according to any one of claims 1 to 11.

25. A computer program product comprising instructions, wherein when the computer program product runs on a computer, the computer is enabled to perform the method according to any one of claims 1 to 11.