

DATA PRIVACY PROTECTION MULTI-PARTY JOINT TRAINING METHOD AND APPARATUS

TECHNICAL FIELD

[0001] One or more embodiments of this specification relate to the field of machine learning technologies, and in particular, to a data privacy protection multi-party joint training method and apparatus, a computer-readable storage medium, and a computing device.

BACKGROUND

[0002] With development of artificial intelligence (Artificial Intelligence, AI for short), a deep neural network (Deep Neural Network, DNN for short) is gradually applied to a plurality of fields such as risk assessment, speech recognition, face recognition, and natural language processing. Structures of DNN networks in different application scenarios are relatively fixed, and more training data is required to implement better model performance.

[0003] In many fields such as healthcare and finance, different enterprises or organizations have different data samples. Model accuracy is greatly improved if these data is jointly trained. This effectively improves quality of service and enhances user satisfaction. However, these raw training data contains a large amount of user privacy and trade secrets. Irreversible negative impacts are caused if the information is leaked. Therefore, strengthening data privacy protection and fully exploiting the value of data become a focus of research in recent years, while multi-party joint training resolves a problem of data silos.

[0004] If training members participating in the joint training are in different industries and have a same user group, for example, for a payment platform and a social platform, the payment platform holds more payment data and the social platform holds more entertainment and social data, feature space of data collected by the platforms is different, but sample space is the same. This may be referred to as a vertical data sharding scenario.

[0005] However, current joint training manners applied in the vertical data sharding scenario are limited, and to meet a higher requirement in an actual application is difficult. Therefore, there

is a need for a joint training solution that can better meet the requirement, for example, improving performance of a trained model while protecting data privacy, in the actual application.

SUMMARY

[0006] Embodiments of this specification describe a data privacy protection multi-party joint training method and apparatus, to improve performance of a trained model while protecting data privacy, so as to implement more accurate prediction.

[0007] According to a first aspect, a data privacy protection multi-party joint training method is provided, where a plurality of parties respectively hold different feature parts of a shared sample set, and a label party in the plurality of parties further holds a sample label. The method is applied to the label party, and includes:

processing a target sample in the shared sample set by using a local first embedding model, to obtain a first embedding vector; processing, by using a local first prediction model, a fusion vector between the first embedding vector and embedding vectors provided by the other parties for the target sample, to obtain a first prediction result; and determining a first training loss based on the first prediction result and a corresponding target sample label; processing the target sample by using a local second embedding model, to obtain a second embedding vector; processing the second embedding vector by using a local second prediction model, to obtain a second prediction result; and determining a second training loss based on the second prediction result and the target sample label; and with a target of reducing a gap between the fusion vector and the second embedding vector, determining a third training loss, for the target sample, of an embedding model corresponding to a larger value in the first training loss and the second training loss, wherein the first training loss, the second training loss, and the third training loss are used to correspondingly train models respectively locally deployed by the plurality of parties.

[0008] In an embodiment, the method includes: separately using all shared samples in the shared sample set as the target sample; and after determining the third training loss, the method further includes: training the first embedding model by using the first training loss and a third training loss corresponding to the first embedding model; and training the second embedding model by using the second training loss and a third training loss corresponding to the second embedding model.

[0009] In an embodiment, the label party further holds a private sample set; and the method further includes: processing the private sample set by using the second embedding model and the

second prediction model, to determine a fourth training loss, where the fourth training loss is used to train the second embedding model and the second prediction model.

[0010] In an embodiment, the method further includes: determining respective fifth training losses of the first embedding model and the second embedding model through contrastive learning based on a total original sample set and a corresponding augmented sample set, where the total original sample set includes a private sample set and the shared sample set, and the fifth training loss is used to train a corresponding embedding model.

[0011] In an embodiment, the embedding vectors provided for the label party by the other parties for the target sample are obtained based on the local first embedding model; and the method further includes: training the first prediction model by using the first training loss, and calculating, based on the first training loss, gradients backhauled to first embedding models of the other parties, so that the other parties train the local first embedding models based on the corresponding gradients; and training the second prediction model by using the second training loss.

[0012] In an embodiment, the method further includes: determining a sixth training loss, which is positively correlated to a distance between a model parameter of the first embedding model and a model parameter of the second embedding model, where the sixth training loss is used to train the first embedding model and the second embedding model.

[0013] In an embodiment, the method further includes: determining a seventh training loss, which is positively correlated to a distance between a model parameter of the first prediction model and a model parameter of the second prediction model, where the seventh training loss is used to train the first prediction model and the second prediction model.

[0014] In an embodiment, the first prediction model and the second prediction model are a same prediction model, and a training loss of the same prediction model for the target sample is a smaller value in the first training loss and the second training loss.

[0015] According to a second aspect, a data privacy protection multi-party joint training apparatus is provided, where a plurality of parties respectively hold different feature parts of a shared sample set, and a label party in the plurality of parties further holds a sample label. The apparatus is integrated into the label party, and includes:

a first embedding module, configured to process a target sample in the shared sample set by using a local first embedding model, to obtain a first embedding vector; a first prediction module, configured to process, by using a local first prediction model, a fusion vector between the

first embedding vector and embedding vectors provided by the other parties for the target sample, to obtain a first prediction result; a first loss determining module, configured to determine a first training loss based on the first prediction result and a corresponding target sample label; a second embedding module, configured to process the target sample by using a local second embedding model, to obtain a second embedding vector; a second prediction module, configured to process the second embedding vector by using a local second prediction model, to obtain a second prediction result; a second loss determining module, configured to determine a second training loss based on the second prediction result and the target sample label; and a third loss determining module, configured to determine, by using an embedding vector, for the target sample, of an embedding model corresponding to a smaller value in the first training loss and the second training loss as a label, a third training loss of the other embedding model for the target sample, where the first training loss, the second training loss, and the third training loss are used to correspondingly train models respectively locally deployed by the plurality of parties.

[0016] According to a third aspect, a computer-readable storage medium is provided. The computer-readable storage medium stores a computer program. When the computer program is executed in a computer, the computer is enabled to perform the method according to the first aspect.

[0017] According to a fourth aspect, a computing device is provided, and includes a memory and a processor. The memory stores executable code. When the processor executes the executable code, the method according to the first aspect is implemented.

[0018] In the above-mentioned method and apparatus provided in the embodiments of this specification, the two embedding models and the two prediction models are deployed in the label party, and knowledge distillation is introduced. In a process of knowledge distillation, a distillation direction is determined based on sample prediction accuracy and a sample granularity, to improve effectiveness of distillation learning. Further, the following several improvements are further provided to further enhance an overall effect of joint training: (1) Supervised learning is performed on the second embedding model and the second prediction model in the label party by introducing the private sample set; (2) contrastive learning is separately performed on the two embedding models in the label party by using the shared sample set and the private sample set; and (3) a regularization loss is designed for the two embedding models and/or the two prediction models in the label party.

BRIEF DESCRIPTION OF DRAWINGS

[0019] To describe the technical solutions in the embodiments of the present invention more clearly, the following briefly describes the accompanying drawings required for describing the embodiments. Apparently, the accompanying drawings in the following description show merely
5 some embodiments of the present invention, and a person of ordinary skill in the art can still derive other drawings from these accompanying drawings without creative efforts.

[0020] FIG. 1 is a schematic diagram of distribution in a vertical data sharding scenario according to an embodiment of this specification;

[0021] FIG. 2 is a schematic diagram 1 of an implementation architecture of a data privacy
10 protection multi-party joint training solution according to an embodiment of this specification;

[0022] FIG. 3 is a schematic diagram of a procedure of steps when a multi-party joint training method is applied to a label party according to an embodiment of this specification;

[0023] FIG. 4 is a schematic diagram 2 of an implementation architecture of a data privacy protection multi-party joint training solution according to an embodiment of this specification;

15 [0024] FIG. 5 is a schematic diagram of a procedure of steps when a multi-party joint training method is applied to a non-label party according to an embodiment of this specification;

[0025] FIG. 6 is a schematic diagram of a structure of a multi-party joint training apparatus integrated into a label party according to an embodiment of this specification; and

[0026] FIG. 7 is a schematic diagram of a structure of a multi-party joint training apparatus
20 integrated into a non-label party according to an embodiment of this specification.

DESCRIPTION OF EMBODIMENTS

[0027] The solutions provided in this specification are described below with reference to the accompanying drawings.

[0028] Embodiments of this specification disclose a data privacy protection multi-party joint
25 training solution. In this solution, two embedding models are provided in a label party that holds a label in a plurality of data parties, and prediction accuracy is determined in combination with specific label data of the label party, to determine a learning direction of an embedding vector between the two embedding models, and improve representation accuracy of the embedding vectors, so that an overall effect of multi-party joint training is further improved. It should be

understood that, in the text, the data party is also referred to as a participant or a calculator, or is simply referred to as a party.

[0029] For convenience of understanding, next, a data distribution status in vertical sharding learning is first described. In most vertical sharding scenarios, samples of participants are usually not strictly aligned, to be specific, after a plurality of parties perform data alignment, there are still parts of data that are unique or private to each training member. For example, data alignment may be implemented by private set intersection (Private Set Intersection, PSI for short).

[0030] In the text, a same sample obtained by the plurality of parties through data alignment is referred to as a shared sample, and samples, other than the shared sample, respectively held by the plurality of parties are referred to as private (or exclusive) samples. It may be understood that, the plurality of parties respectively hold different feature parts of the shared sample, and a plurality of shared samples form a shared sample set. In addition, a party that further holds a sample label in the plurality of parties is referred to as a label party, and the other parties are referred to as non-label parties.

[0031] FIG. 1 is a schematic diagram of distribution in a vertical data sharding scenario according to an embodiment of this specification, and shows a data party A and a data party B in a plurality of parties, where the data party A is a label party, and the data party B is a non-label party. Specifically, the data party A holds a feature X1 and a label Y1 of a shared sample of the two parties, and the data party B holds a feature Z1 of the shared sample. In addition, the data party A further holds a feature X2 and a label Y2 of a private sample thereof, and the data party further holds a feature Z2 of a private sample thereof.

[0032] The following describes, based on the two data parties shown in FIG. 1, different feature parts held by various parties and a sample label held by a label party in different service scenarios.

[0033] (1) Recommendation scenario

[0034] In the recommendation scenario, the data party A holds object feature data and a user behavior label of a recommended sample, and the data party B holds user feature data of the recommended sample. It needs to be noted that user-related data such as the user feature data mentioned in the text is collected and used after a user confirms authorization.

[0035] In an embodiment, the data party A may be an e-commerce platform, a payment platform, a content platform, or the like, and a service object targeted by the object feature data

owned by the data party A may be goods, a service, a public number, a subscription number, an application, content information, an advertisement, or the like.

[0036] In an example, the object feature data is a product feature, for example, a category, an origin, costs, a selling price, a sales volume, and a target audience. In another example, the object feature data is a public number feature, for example, a classification, a quantity of followers, a frequency of published content, and a quantity of likes and retweets. In still another example, the object feature data is an advertisement feature, for example, an advertising carrier (including an image, text, a video, or an audio), advertising content, and spokesperson information.

[0037] In an embodiment, the user feature data includes a static attribute feature of the user, for example, an occupation, a hobby, a place of residence, and asset information. For example, these features may be extracted from user registration information. In another embodiment, the user data further includes a dynamic behavioral feature of the user. In an example, the data party B is a social platform. In this case, the dynamic behavioral feature may include social duration, a social frequency, a quantity of friends, a quantity of personal dynamic operations, a quantity of likes, and the like. In another example, the data party B is a payment platform. In this case, the dynamic behavioral feature may include a payment frequency, a payment amount, a payment period, and the like.

[0038] The above-mentioned user behavior label indicates whether the user performs a predetermined behavior on a service object. It may be understood that, the predetermined behavior is appropriate to the service object. In an embodiment, the service object is goods or a service. In this case, the predetermined behavior may be purchase or use. In another embodiment, the service object is a public number or a subscription number. In this case, the predetermined behavior may be following or subscription. In still another embodiment, the service object is an application. In this case, the predetermined behavior may be downloading or registration. In yet another embodiment, the service object is an advertisement or content information. In this case, the predetermined behavior may be a click.

[0039] The foregoing describes multi-party data distribution in the recommendation scenario.

[0040] (2) Risk control scenario

[0041] In the risk control scenario, a plurality of parties may respectively hold different behavioral data of a same user. In an embodiment, the data party A is a bank and the data party B is a payment platform. The bank holds bank flows of the user, for example, records of deposits and

withdrawals; and the payment platform holds payment data of the user, for example, a payment amount, and a quantity of payments. In another embodiment, the data party A is a credit service platform and the data party B is a credit bureau. The credit service platform holds service data, for example, records of rentals of books or bicycles without deposit; and the credit bureau holds credit data of the user, for example, a quantity of credit cards that the user has and repayment default records, generated when the user uses a credit service.

[0042] The label party in the plurality of parties holds a risk label. Specifically, the risk label may be a risk classification label, for example, a low risk, a medium risk, and a high risk, or may be a risk score label, for example, 30 points, 90 points, and the like in a hundred-mark system.

[0043] In the foregoing, vertical multi-party data distribution in different service scenarios is described by using two parties in the plurality of parties as an example in combination with the recommendation scenario and the risk control scenario. It may be understood that the plurality of parties are not limited to the two parties, but may alternatively be any other quantity of participants that is greater than 2, the service scenarios are not limited to the recommendation scenario and the risk control scenario, but may alternatively be an image classification scenario, a text classification scenario, an audio classification scenario, or the like, and exhaustion is not provided herein.

[0044] Next, an implementation architecture of a multi-party joint training solution disclosed in embodiments of this specification is still described based on the two data parties shown in FIG. 1. FIG. 2 is a schematic diagram 1 of an implementation architecture of a data privacy protection multi-party joint training solution according to an embodiment of this specification, and shows a first embedding model, a second embedding model, a first prediction model, and a second prediction model that are locally deployed by the data party A, and an embedding model locally deployed by the data party B. It should be noted that the data party B may alternatively deploy another quantity (for example, 2) of embedding models. In addition, "first" in the above-mentioned "first embedding model", and similar terms such as "first" and "second" elsewhere in this specification are used to distinguish between similar things, and do not have another limitation function such as sorting.

[0045] As shown in FIG. 2, the data party A uses knowledge distillation (Knowledge Distillation, KD for short) and supervised learning in a machine learning technology, corresponding to two types of training losses: a distillation loss and a classification loss that are generated. The distillation learning is different from a conventional manner in which a teacher

model and a student model are pre-specified, but instead, a relatively inaccurate embedding model performs representation learning on a more accurate embedding model based on accuracy of prediction results obtained by the two prediction models based on different representations of a same sample, where the different representations of the same sample include an embedding vector
5 Z_A2 output by the second embedding model in the data party A and a fusion vector Z_AB between an embedding vector Z_A1 output by the first embedding model in the data party A and an embedding vector Z_B output by the embedding model in the data party B. The classification loss is determined based on respective prediction results of the first prediction model and the second prediction model and corresponding sample labels. Further, a regularization loss that
10 constrains a model parameter is further designed for the first prediction model and the second prediction model, to prevent the two prediction models from overfitting in a training process.

[0046] Models respectively locally deployed by the plurality of parties may be correspondingly trained based on the distillation loss, the classification loss, and the regularization loss that are determined above. When putting a trained model into actual multi-party joint use, the
15 data party A may make an optional use of each of the two embedding models and the two prediction models, and usually uses a trained first embedding model and first prediction model.

[0047] Specific implementation steps of the above-mentioned joint training solution are described below with reference to FIG. 3 and more embodiments.

[0048] FIG. 3 shows steps of a method performed by a label party that is in a plurality of
20 parties and that participates in joint training, where the plurality of parties respectively hold different feature parts of a shared sample set, and the label party further holds a sample label. For example, the label party may be the data party A shown in FIG. 1 or FIG. 2, and any other party other than the label party in the plurality of parties may be the data party B shown in FIG. 1 or FIG. 2. It should be understood that, all parties in the plurality of parties may be implemented as
25 any apparatus, platform, server, or device cluster, or the like that has a computing and processing capability. FIG. 3 shows the following steps.

[0049] Step S302: Process a target sample in a shared sample set by using a local first embedding model, to obtain a first embedding vector. It should be noted that, the first embedding model may be implemented as different neural network algorithms, for example, a DNN network
30 or a convolutional neural network (Convolutional Neural Network, CNN for short).

[0050] For the above-mentioned target sample, in an embodiment, all shared samples in the

shared sample set may be separately used as the target sample. In another embodiment, it may be understood that, training of a model usually includes a plurality of iterative training rounds, and correspondingly, all shared samples in a batch of shared samples corresponding to a current iterative training round may be separately used as the target sample, where the batch of shared samples is obtained through sampling based on the shared sample set.

[0051] In this step, a first embedding vector $Z1$ may be obtained by inputting a local feature part held by the label party for the target sample into the first embedding model.

[0052] Step S304: Process, by using a local first prediction model, a fusion vector between the first embedding vector and embedding vectors provided by the other parties for the target sample, to obtain a first prediction result. It should be noted that, the first prediction model may be implemented as different neural network algorithms, for example, a DNN network or a CNN network.

[0053] The above-mentioned other parties refer to other participants or other non-label parties other than the label party in the plurality of parties. It may be understood that, the other parties may respectively process local feature parts for the target sample based on local embedding models, to obtain corresponding embedding vectors, and provide the embedding vectors for the label party.

[0054] In this step, fusion processing is first performed on the first embedding vector $Z1$ and embedding vectors Z' provided by the other parties, to obtain a fusion vector Zf , and then the fusion vector Zf is input into the first prediction model, to obtain a first prediction result $Y1$. It should be noted that, there are a plurality of fusion processing manners, for example, splicing, summation, averaging, or introducing an attentional mechanism to calculate attentional weights of the embedding vectors before weighted summation, and the like.

[0055] From the above, the first prediction result $Y1$ output by the first prediction model based on the fusion vector Zf for the target sample may be obtained.

[0056] Step S306: Determine a first training loss based on the first prediction result and a corresponding target sample label.

[0057] It should be understood that a loss function may be selected as needed to calculate the first training loss. For example, the loss function may be a cross-entropy loss function, a hinge loss function, or the like. In this way, the first prediction result and the sample label of the target sample (or briefly referred to as the target sample label) are assigned to independent variables in the loss function, so that a corresponding function value can be obtained as a first training loss $L_{cla}^{[1]}$, which

belongs to a classification loss.

[0058] The above describes a case in which the label party performs steps S302, S304, and S306, to determine the first prediction result for the target sample by using the local first embedding model and first prediction model based on the local feature part for the target sample and the embedding vectors provided by the other parties for the target sample, and further determine the first training loss L1 in combination with the target sample label.

[0059] The following describes a case in which the label party performs steps S308, S310, and S312, to determine a second prediction result for the target sample by using the first embedding model and the first prediction model based on local sample data only, and further determine a second training loss in combination with the target sample label. It should be noted that, an execution order of different steps in the text is not uniquely limited in the embodiments of this specification, provided that a flow of data is logical, for example, steps S302 and S308 may be performed in sequence, or steps S302 and S308 may be performed in parallel.

[0060] Step S308: Process the target sample by using a local second embedding model, to obtain a second embedding vector. It should be understood that the second embedding model is different from the first embedding model, and even if a same model structure as that of the first embedding model is selected for the second embedding model and model parameters are initialized to be the same, model parameters after training update are different due to a difference between training losses used in subsequent training. In addition, the second embedding model may be implemented as a DNN, a CNN, or another neural network algorithm.

[0061] In this step, the second embedding vector Z2 is obtained by inputting the local feature part held by the label party for the target sample into the second embedding model.

[0062] Step S310: Process the second embedding vector by using a local second prediction model, to obtain a second prediction result. To be specific, an output second prediction result Y2 may be obtained by inputting the second embedding vector Z2 into the second prediction model.

[0063] Step S312: Determine a second training loss based on the second prediction result and the target sample label.

[0064] It should be understood that a loss function may be selected as needed to calculate the second training loss. Generally, it is considered that magnitude of the second training loss needs to be subsequently compared with that of the first training loss. To improve usability of a comparison result, the loss function selected in this step may be designed as a same loss function

as the loss function used in calculating the first training loss in step S306, for example, both are the cross-entropy loss function.

[0065] In this way, a second training loss $L_{cla}^{[2]}$ may be obtained, which also belongs to the classification loss.

5 **[0066]** Afterwards, in step S314, with a target of reducing a gap between the fusion vector Zf and the second embedding vector Z2, a third training loss, for the target sample, of an embedding model corresponding to a larger value in the first training loss L1 and the second training loss L2 is determined.

[0067] It should be understood that, in this step, unlike conventional knowledge distillation (or distillation learning) in which a teacher model and a student model are pre-specified, and then the student model continuously learns from the teacher model, which of the fusion vector Zf and the second embedding vector Z2 has a better representation effect for the target sample is inversely deduced based on prediction accuracy embodied in magnitude of a training loss, so that an embedding model with a poorer representation learns a better representation. It may be understood
10 that a smaller training loss indicates better prediction accuracy and a better representation effect of a vector input into a corresponding prediction model.

[0068] Based on this, in this step, only a distillation loss of an embedding model corresponding to a larger training loss is determined by using the target sample to train the embedding model. Specifically, the third training loss (or referred to as the distillation loss) is determined based on
20 the fusion vector Zf and the second embedding vector Z2. It should be understood that the fusion vector Zf and the second embedding vector Z2 have a same vector dimension. For example, a square of a two-norm of a difference vector between two vectors may be calculated as the third training loss $L_{kd}^{(i)}$. This may be denoted as the following formula:

$$L_{kd}^{[i]} = \|Zf - Z2\|^2 \quad (1)$$

25 **[0069]** $i = \begin{cases} 1, & L_{cla}^{[1]} > L_{cla}^{[2]} \\ 2, & L_{cla}^{[2]} > L_{cla}^{[1]} \end{cases}$, $L_{kd}^{[1]}$ represents a third training loss of the first embedding model,

$L_{kd}^{[2]}$ represents a third training loss of the second embedding model, an operator $\|\cdot\|$ represents the two-norm, and a superscript 2 represents square calculation.

[0070] It should be understood that, calculation of the third training loss is not limited to Formula (1), for example, a one-norm, a two-norm, or the like of a difference vector between two

vectors may alternatively be calculated.

[0071] In addition, determining the distillation loss of one of the two embedding models by using the target sample is described in this step. Actually, for a batch of shared samples (for example, tens of thousands of samples) in different training rounds, after prediction accuracy is determined, a part of the shared samples is used to determine the distillation loss of the first embedding model, and the other part of the shared samples is used to determine the distillation loss of the second embedding model. This means that the first embedding model and the second embedding model have respective corresponding third training losses.

[0072] In the above, the first training loss, the second training loss, and the third training loss are determined. These training losses are used to correspondingly train models respectively locally deployed by the plurality of parties. It should be understood that, for model training, if a forward propagation process is determined, a manner in which back propagation is performed by using a training loss, including which training losses are used to train which models and how to perform parameter update, is accordingly determined, with a target of reducing the training loss in a back propagation process.

[0073] Details are provided as follows: (1) For a model in the label party

[0074] ① A first training loss $L_{cla}^{[1]}$ and a corresponding third training loss $L_{kd}^{[1]}$ thereof are used for training of the first embedding model. For example, a comprehensive training loss of the first embedding model may be denoted as:

$$L^{[1]} = L_{cla}^{[1]} + \lambda L_{kd}^{[1]} \quad (2)$$

[0075] λ is a weighting coefficient of $L_{kd}^{[1]}$, is a hyperparameter, and may be manually set to a value in a value range of (0, 1), for example, set to $\lambda = 0.01$.

[0076] ② The first training loss $L_{cla}^{(1)}$ is used for training of the first prediction model.

[0077] ③ A second training loss $L_{cla}^{[2]}$ and a corresponding third training loss $L_{kd}^{[2]}$ thereof are used for training of the second embedding model. For example, a comprehensive training loss of the second embedding model may be denoted as:

$$L^{[2]} = L_{cla}^{[2]} + \beta L_{kd}^{[2]} \quad (3)$$

[0078] β is a weighting coefficient of $L_{kd}^{[2]}$, is a hyperparameter, and may be manually set to a value in a value range of (0, 1), for example, set to $\beta = 0.02$.

[0079] ④ The second training loss $L_{cla}^{[2]}$ is used for training of the second prediction model.

[0080] (2) For a model in the non-label party

[0081] It may be understood that, at least one embedding model is deployed in the non-label party, where the embedding model is used to process the local feature part of the target sample, to

5 provide an obtained embedding vector to the label party. The first training loss $L_{cla}^{(1)}$ is used for training of the embedding model. Specifically, each non-label party receives a gradient backhauled to a local embedding model from the label party, to update a model parameter of the local embedding model by using this gradient, where the gradient is obtained by the label party through calculation based on the first training loss $L_{cla}^{(1)}$. A specific calculation manner may be
10 implemented by using a current relevant technology, and details are not described.

[0082] The above mainly describes the multi-party joint training solution in which the two embedding models and the two prediction models are deployed in the label party, and knowledge distillation is introduced. In a process of knowledge distillation, a distillation direction is determined based on sample prediction accuracy and a sample granularity, to improve
15 effectiveness of distillation learning.

[0083] Further, the following several improvements are further provided to further improve prediction effects of models trained by the plurality of parties. It may be understood that improvement means for the following several aspects may be used independently or in any combination.

20 [0084] I. Training of the second embedding model and the second prediction model by introducing a private sample set

[0085] It is considered that the label party not only has the shared sample set, but also holds a private sample set formed by private samples other than the shared samples, and the second embedding model and the second prediction model involve only processing of the samples in terms
25 of local features and label data. Therefore, the private sample set is additionally used to perform supervised learning on the second embedding model and the second prediction model.

[0086] Specifically, the private sample set is processed by using the second embedding model and the second prediction model, to determine a fourth training loss. It may be understood that, similar to processing the target sample in the above-mentioned shared sample set, the private
30 sample is first processed by using the second embedding model, to obtain a corresponding

embedding vector, and then the embedding vector is input into the second prediction model, to obtain a corresponding prediction result, so that the fourth training loss is determined based on the prediction result and a corresponding sample label.

[0087] Based on this, for example, a comprehensive training loss $L^{[2]}$ of the second embedding model may be denoted as:

$$L^{[2]} = L_{cla}^{[2]}(x \in D_I) + L_{cla}^{[2]}(x \in D_S) + \beta L_{kd}^{[2]}(x \in D_I) \quad (4)$$

[0088] In Formula (4), D_I and D_S respectively represent the shared sample set and the private sample set unique to the label party, $L_{cla}^{[2]}(x \in D_I)$ and $L_{cla}^{[2]}(x \in D_S)$ respectively represent classification losses determined by the second embedding model based on D_I and D_S ,

where the two may be combined and denoted as $L_{cla}^{[2]}(x \in D_I \cup D_S)$, for a meaning of $\beta L_{kd}^{[2]}(x \in D_I)$, reference may be made to a description of a same symbol in Formula (3), and $L_{kd}^{[2]}(x \in D_I)$ is equivalent to $L_{kd}^{[2]}$ in Formula (3).

[0089] In addition, the second prediction model may be trained by using $L_{cla}^{[2]}(x \in D_I \cup D_S)$.

[0090] In this way, the private sample set is introduced in learning of the second embedding model and the second prediction model, to enrich training data and improve training effects of the models, to further improve an overall training effect of all models in the multi-party joint training, and improve prediction accuracy of the trained models.

[0091] II. Training of the first embedding model and the second embedding model by using contrastive learning

[0092] Contrastive learning is used so that the private sample set may also be additionally used for training of the first embedding model. It should be understood that the shared sample set may be further used in addition to the private sample set during contrastive learning, and contrastive learning may also be used for the second embedding model.

[0093] Specifically, based on a total original sample set formed by the shared sample set and the private sample set, data augmentation processing may first be performed on original local features of all samples therein, to obtain a corresponding augmented dataset (or referred to as an augmented sample set in the text). In an embodiment, the data augmentation processing may include adding noise perturbations, masking some of the features, and the like. In another embodiment, for all samples in the total original sample set, several times of data augmentation

processing may be performed on original local features thereof, to obtain several corresponding augmented local features to be grouped into the augmented dataset. It should be understood that "several" in the text refers to one or more.

[0094] According to an example, the total original sample set is a total original user set, and a target party holds original face images of all users therein. In this case, for all the original face images, random cropping, random color distortion, or random Gaussian blurring may be used as a data augmentation means to obtain corresponding augmented face images, to form an augmented face image set.

[0095] According to another example, the target party holds original feature vectors of all the users in the total original user set. In this case, for all the original feature vectors, a manner of randomly masking some of the features may be used, for example, a predetermined quantity of vector elements are randomly set to zero, or null, or default, to obtain corresponding augmented feature vectors, to form the augmented feature vector set.

[0096] Based on the total original sample set and the augmented sample set obtained through data augmentation processing on the total original sample set, the label party may separately perform contrastive learning on the two locally deployed embedding models, to determine respective fifth training losses, where the fifth training loss may be interchangeably referred to as a contrastive loss.

[0097] For ease of understanding, either of the two embedding models is referred to as a target embedding model in the following, and a process of performing contrastive learning on the target embedding model to determine a fifth training loss thereof is described.

[0098] (1) Embedding processing Embedding processing is separately performed on $2N$ samples by using the target embedding model, to correspondingly obtain $2N$ sample embedding vectors. The $2N$ samples include a batch of N original samples sampled from the total original sample set and N augmented samples that correspond to these N original samples and that are in the augmented sample set.

[0099] (2) Calculation of the fifth training loss N positive sample pairs may be constructed by using N original samples and N augmented samples. For each sample in each positive sample pair, each sample in the other $(N - 1)$ positive sample pairs forms a negative sample pair with the sample. With a target of reducing a distance in a positive sample pair as well as increasing a distance in a negative sample pair, the fifth training loss is determined, for example, the following formula is

used:

$$L_{con}(x_j, x_k) = -\log \frac{\exp(\text{sim}(\mathbf{z}_{x_j}, \mathbf{z}_{x_k})/\tau)}{\sum_{l=1}^{2N} 1_{[l \neq j]} \exp(\text{sim}(\mathbf{z}_{x_k}, \mathbf{z}_{x_l})/\tau)} \quad (5)$$

[0100] In Formula (5), samples x_j and x_k constitute a positive sample pair, $\mathbf{z}_{x_j}$ and $\mathbf{z}_{x_k}$ respectively represent embedding vectors output by the target embedding model for the samples x_j and x_k , $\text{sim}(\mathbf{z}_{x_j}, \mathbf{z}_{x_k})$ represents similarity (for example, cosine similarity) between the embedding vectors $\mathbf{z}_{x_j}$ and $\mathbf{z}_{x_k}$, and τ represents a temperature coefficient, for example, $\tau = 0.5$ is manually set.

[0101] The above is an example description of contrast learning on the embedding models. Actually, construction of positive and negative sample pairs and a design of a contrast loss calculation formula are not limited to the above-mentioned manners.

[0102] The respective fifth training losses of the two locally deployed embedding models may be obtained by using the total original sample set and the corresponding augmented dataset. Based on this, for example, comprehensive training losses of the first embedding model and the second embedding model may be respectively denoted as Formula (6) and Formula (7) below:

$$L^{[1]} = L_{cla}^{[1]}(x \in D_I) + \lambda_1 L_{kd}^{[1]}(x \in D_I) + \lambda_2 L_{con}^{[1]}(x \in D_I \cup D_S) \quad (6)$$

$$L^{[2]} = L_{cla}^{[2]}(x \in D_I \cup D_S) + \beta_1 L_{kd}^{[2]}(x \in D_I) + \beta_2 L_{con}^{[2]}(x \in D_I \cup D_S) \quad (7)$$

[0103] In Formula (6) and Formula (7), $L_{con}^{[1]}(x \in D_I \cup D_S)$ and $L_{con}^{[2]}(x \in D_I \cup D_S)$ respectively represent contrastive learning losses, also referred to as the fifth training losses, of the first embedding model and the second embedding model; and λ_1 , λ_2 , β_1 , and β_2 are all hyperparameters, where values thereof may be each manually set to a specific value in a value range of (0, 1), for example, $\lambda_1 = \beta_1 = 0.01$ and $\lambda_2 = \beta_2 = 0.005$ are set. For descriptions of the other symbols, reference may be made to the descriptions in the foregoing embodiment. Details are not described.

[0104] In this way, the first embedding model and the second embedding model are trained by additionally introducing the private sample set through contrastive learning, so that the sample data can be fully used, and training effects of the two embedding models can be effectively improved, thereby improving an overall training effect of all models in the multi-party joint training, and improving prediction accuracy of the trained models.

[0105] III. Training of the first embedding model and the second embedding model by

introducing a regularization loss

[0106] A regularization loss that constrains a model parameter is designed for the first embedding model and the second embedding model, to prevent the two embedding models from overfitting in a training process.

5 **[0107]** Specifically, the first embedding model and the second embedding model are designed to have a same model structure. In this way, with a target of reducing a distance between a model parameter $W^{[e,1]}$ of the first embedding model and a model parameter $W^{[e,2]}$ of the second prediction model, a sixth training loss $L_{reg}^{[e]}$ is determined. In some embodiments, the distance may be measured in the following manners: a Euclidean distance between $W^{[e,1]}$ and $W^{[e,2]}$, a
10 two-norm of a difference vector or a difference matrix between $W^{[e,1]}$ and $W^{[e,2]}$ or a square of the two-norm, and the like. For example, the sixth training loss may be calculated by using the following formula:

$$L_{reg}^{[e]} = \|W^{[e,1]} - W^{[e,2]}\|^2 \quad (8)$$

[0108] An operator $\|\cdot\|$ represents the two-norm and a superscript 2 represents square
15 operation.

[0109] For example, a regularization loss term may be further added based on Formula (6) in the above, so that comprehensive training losses of the first embedding model and the second embedding model are respectively denoted as Formula (9) and Formula (10) below:

$$L^{[1]} = L_{cla}^{[1]}(x \in D_I) + \lambda_1 L_{kd}^{[1]}(x \in D_I) + \lambda_2 L_{con}^{[1]}(x \in D_I \cup D_S) + \lambda_3 L_{reg}^{[e]} \quad (9)$$

$$20 \quad L^{[2]} = L_{cla}^{[2]}(x \in D_I \cup D_S) + \beta_1 L_{kd}^{[2]}(x \in D_I) + \beta_2 L_{con}^{[2]}(x \in D_I \cup D_S) + \beta_3 L_{reg}^{[e]} \quad (10)$$

[0110] In Formula (9) and Formula (10), λ_3 and β_3 are also hyperparameters, for example, $\lambda_3 = \beta_3 = 0.001$ is manually set.

[0111] IV. Training of the first prediction model and the second prediction model by introducing a regularization loss

25 **[0112]** Further, a regularization loss that constrains a model parameter is further designed for the first prediction model and the second prediction model, to prevent the two prediction models from overfitting in a training process.

[0113] Specifically, the first prediction model and the second prediction model are designed to have a same model structure. In this way, a target of reducing a distance between a model
30 parameter $W^{[p,1]}$ of the first prediction model and a model parameter $W^{[p,2]}$ of the second

prediction model may be designed, to determine a seventh training loss. In some embodiments, the distance may be measured in the following manners: a Euclidean distance between $W^{[p,1]}$ and $W^{[p,2]}$, a two-norm of a difference vector or a difference matrix between $W^{[p,1]}$ and $W^{[p,2]}$ or a square of the two-norm, and the like. For example, a seventh training loss $L_{reg}^{[p]}$ may be calculated

5 by using the following formula:

$$L_{reg}^{[p]} = \|W^{[p,1]} - W^{[p,2]}\|^2 \quad (11)$$

[0114] An operator $\|\cdot\|$ represents the two-norm and a superscript 2 represents square operation.

[0115] For example, comprehensive training losses of the first prediction model and the second prediction model may be respectively denoted as Formula (12) and Formula (13) below:

$$L^{[p,1]} = L_{cla}^{[1]}(x \in D_I \cup D_S) + \alpha L_{reg}^{[p]} \quad (12)$$

$$L^{[p,2]} = L_{cla}^{[2]}(x \in D_I \cup D_S) + \gamma L_{reg}^{[p]} \quad (13)$$

[0116] α and γ are hyperparameters, for example, $\alpha = \gamma = 0.001$ is manually set.

[0117] V. The first prediction model and the second prediction model may alternatively be a same prediction model

[0118] In the foregoing embodiment, the following is mainly described: The first prediction model and the second prediction model are two different models. Specifically, the first prediction model is only used to process the fusion vector Zf but is not used to process the second embedding vector Z2, and the second prediction model is only used to process the second embedding vector Z2 but is not used to process the fusion vector Zf. Correspondingly, there is a difference between training losses of the two.

[0119] Actually, alternatively, the first prediction model and the second prediction model may be implemented as a same prediction model. In other words, a same prediction model is used both for processing the fusion vector Zf to obtain the corresponding first prediction result and for processing the second embedding vector Z2 to obtain the corresponding second prediction result. In this case, a training loss of the prediction model may be determined as a smaller value in a first training loss $L_{cla}^{[1]}$ and a second training loss $L_{cla}^{[2]}$, for example, may be denoted as:

$$L^{[p]} = \min(L_{cla}^{[1]}, L_{cla}^{[2]}) \quad (14)$$

[0120] From the above, several further improvements made to the label party in the multi-party joint training solution are described, to improve a training effect of the model in the label

party, so that the overall effect of the models jointly trained by the plurality of parties is further improved, thereby enhancing accuracy of joint prediction in an actual use stage.

[0121] The above mainly describes improvements made in vertical sharding learning from the label party, and mentions that at least one embedding model is deployed in the non-label party for processing the local feature part of the target sample in the shared sample set, to provide an obtained embedding vector to the label party. Actually, improvements may alternatively be made to an execution manner of any one of non-label parties in the plurality of parties when the non-label party participates in the multi-party joint training, as described below.

[0122] Specifically, contrastive learning and knowledge distillation are used so that the non-label party participating in the joint training may further make effective use of a local private dataset, thereby improving and optimizing prediction performance of the jointly trained models.

[0123] FIG. 4 is a schematic diagram 2 of an implementation architecture of a data privacy protection multi-party joint training solution according to an embodiment of this specification, and shows two embedding models locally deployed by a data party B that belongs to the non-label parties.

[0124] As shown in FIG. 4, training of the models in the data party B involve three types of learning tasks, including contrastive learning, distillation learning, and supervised learning. Specifically, models that perform contrastive learning and distillation learning include two embedding models. The contrastive learning and the distillation learning use full sample data (including a shared sample and a private sample) of all parties, so that stability of the embedding models is improved. In addition, the distillation learning is different from a conventional manner in which a teacher model and a student model are pre-specified, but instead, a learning direction of a sample dimension is determined based on stability of two embedding models in a single data party for a same sample, and specifically, an unstable model learns from a stable model, to improve a learning effect of the model.

[0125] Models that perform supervised learning include a first embedding model whose embedding vector output for a shared sample is provided to a label party, to implement the supervised learning based on a gradient that is provided by the label party for the shared sample and that is backhauled to the first embedding model.

[0126] From the above, the non-label party may obtain two trained embedding models in multi-party joint training, and when putting the trained models into actual joint use, may

selectively use one of the two embedding models, and usually use the first embedding model.

[0127] FIG. 5 is a schematic diagram of a procedure of steps when a multi-party joint training method is applied to a non-label party according to an embodiment of this specification. For brevity of description, any one of non-label parties is hereinafter referred to as a target party, for example, the target party may be the data party B shown in FIG. 1 or FIG. 4. In addition, it should be noted that, for clarity and brevity of description, terms used in steps shown in FIG. 5 may be the same as the terms used in the description of the steps shown in FIG. 3, but there is a difference between actual meanings, which can be accurately distinguished and understood by a person of skill in the art with reference to the context, for example, for a total original sample set, when steps performed by the non-label party are described, what can be used by the non-label party is necessarily a local total original sample set of the non-label party, but is not a total original sample set in a label party, and so on.

[0128] FIG. 5 shows the following steps.

[0129] Step S510: Determine respective first training losses of two locally deployed embedding models through contrastive learning based on a total original sample set and a corresponding augmented sample set, where the total original sample set includes a private sample set and a shared sample set. Step S520: Respectively calculate, by separately using all samples in the total original sample set as a target sample, stability indicators of the two embedding models for the target sample, and determine, by using an embedding vector output by an embedding model with higher stability for the target sample as a label, a second training loss of the other embedding model for the target sample. Step S530: Provide first embedding vectors output by a predetermined first embedding model of the two embedding models for all shared samples to a label party, to enable the label party to determine a third training loss based on the first embedding vectors respectively provided by a plurality of parties, a locally deployed fusion model, and a held sample label, where the first training loss, the second training loss, and the third training loss are used to train models respectively locally deployed by the plurality of parties.

[0130] An expanded description of the above steps is as follows:

[0131] Step S510: Determine the respective first training losses of the two locally deployed embedding models through contrastive learning based on the total original sample set and the corresponding augmented sample set.

[0132] The above-mentioned total original sample set includes the private sample set and the

shared sample set. The shared sample set is a set formed by a plurality of samples shared by the plurality of parties, and the private sample set is a set formed by samples, other than the samples in the shared sample set, further held by the target party.

[0133] It should be noted that, usually, input dimensions of the two embedding models are configured as the same and output dimensions of the two embedding models are configured as the same; but other structural parameters, for example, layer quantities of neural network layers, quantities of neurons, or neural network algorithms on which the two embedding models are based, may be configured as the same or different.

[0134] The respective first training losses of the two locally deployed embedding models may be obtained by using the total original sample set and the corresponding augmented dataset. It should be noted that, for descriptions of the steps, reference may be further made to the related descriptions in the foregoing embodiment. Details are not described.

[0135] In step S520, by separately using all the samples in the total original sample set as the target sample, the stability indicators of the two embedding models for the target sample are respectively calculated, and by using the embedding vector output by the embedding model with higher stability for the target sample as the label, the second training loss (or referred to as the distillation loss) of the other embedding model for the target sample is determined.

[0136] It should be noted that, in this step, distillation learning is refined to a sample granularity, a learning direction between the two embedding models is dynamically determined, and the learning direction may change with different samples. Specifically, an unstable model learns an embedding vector from a stable model based on stability of the two embedding models on a single sample. In this way, training effects of the two embedding models can be effectively improved, thereby improving an overall training effect of all models in the joint training.

[0137] It should be understood that stability of an embedding model on a sample may be measured by using a predefined stability indicator. In an embodiment, the stability indicator includes an amount of representation information, to be specific, the stability of the model is measured from a perspective of an amount of information contained in an embedding representation obtained by processing the sample by the embedding model. Apparently, a larger amount of representation information indicates a better representation effect and higher stability of the model.

[0138] A calculation manner of the amount of representation information may be obtained

through a reasonable definition and is not limited to a particular calculation manner. In this step, any one of the two embedding models is referred to as a target embedding model. In a specific embodiment, a largest element value in a target embedding vector obtained by processing the target sample through the target embedding model may be determined as an amount of representation information of the target embedding model for the target sample. In another specific embodiment, a sum of absolute values of all elements in the target embedding vector may be determined as the amount of representation information of the target embedding model for the target sample. On the other hand, in a specific embodiment, two amounts of representation information corresponding to two embedding vectors obtained by respectively processing the target sample and a corresponding augmented sample through the target embedding model may alternatively be determined first, and then the amount of representation information of the target embedding model is determined to be a larger value in the two amounts of representation information.

[0139] According to an example, an amount of representation information M_x of the target embedding model for a target sample x may be calculated by using the following formula:

$$M_x = \max(\|z_x\|_\infty, \|z_{\tilde{x}}\|_\infty) \quad (15)$$

[0140] In the above formula, x and $\tilde{x}$ respectively represent the target sample and the corresponding augmented sample, and z_x and $z_{\tilde{x}}$ represent embedding vectors correspondingly obtained by respectively processing samples x and $\tilde{x}$ through the target embedding model; and an operator $\|\cdot\|_\infty$ represents an infinity norm of a vector, and an operation result is a largest element value in the vector.

[0141] From the above, the amount of representation information of the embedding model for the target sample may be calculated.

[0142] In another embodiment, the stability indicator includes a dispersion degree. Specifically, a difference degree between the two embedding vectors obtained by respectively processing the target sample and the corresponding augmented sample through the target embedding model is determined as the dispersion degree of the target embedding model. Apparently, a higher dispersion degree indicates poorer stability of the model.

[0143] A calculation manner of the difference degree may be obtained through a reasonable definition and is not limited to a particular calculation manner. In a specific embodiment, a difference vector between the two embedding vectors is first calculated, and then a sum of squares of vector elements in the difference vector is used as the dispersion degree of the target embedding

model. For example, a dispersion degree ϵ_x may be calculated by using the following formula:

$$\epsilon_x = \|\mathbf{z}_x - \mathbf{z}_{\tilde{x}}\|^2 \quad (16)$$

[0144] In the above formula, x and $\tilde{x}$ respectively represent the target sample and the corresponding augmented sample, and $\mathbf{z}_x$ and $\mathbf{z}_{\tilde{x}}$ represent embedding vectors correspondingly obtained by respectively processing samples x and $\tilde{x}$ through the target embedding model; and an operator $\|\cdot\|$ represents a two-norm of a vector, and a superscript 2 represents obtaining a square.

[0145] In another specific embodiment, a sum of absolute values of vector elements in the difference vector, a one-norm of the difference vector, or a two-norm of the difference vector is used as a dispersion degree of a target model. In still another specific embodiment, a cosine distance or a Euclidean distance between the two vectors is calculated as the dispersion degree of the target model.

[0146] The stability indicator is described above by mainly using the amount of representation information and the dispersion degree as an example. It should be understood that another indicator may be further involved and exhaustion is not provided.

[0147] Based on a calculation result of the stability indicator, the embedding model with higher stability in the two embedding models may be determined. In an embodiment, the stability indicator includes the amount of representation information. In this case, an embedding model with a larger amount of representation information in the two embedding models may be determined as the embedding model with higher stability. In another embodiment, the stability indicator includes the dispersion degree. In this case, an embedding model with a lower dispersion degree in the two embedding models may be determined as the embedding model with higher stability. In still another embodiment, the stability indicator includes the amount of representation information and the dispersion degree. In this case, an embedding model with a larger difference obtained by subtracting the dispersion degree from the amount of representation information may be determined as the embedding model with higher stability.

[0148] After the embedding model with higher stability is determined, an embedding vector output by the embedding model for the target sample may be used as a label, to determine a second training loss of the other embedding model for the target sample, so that an unstable model learns an embedding representation from a stable model.

[0149] It may be understood that a construction purpose of the second training loss is as

follows: reducing a distance between an embedding vector output by the other embedding model for the target sample and the embedding vector label. Calculation of the second training loss is based on the two embedding vectors output by the two embedding models for the target sample and there are a plurality of calculation manners. For example, a distance, for example, a Euclidean distance, a square of the Euclidean distance, or a cosine distance, between the two embedding vectors may be calculated as the second training loss.

$$L_{kd}(x) = Dis(\mathbf{z}_x - \mathbf{z}_{\tilde{x}}) \quad (17)$$

[0150] In Formula (17) above, $L(x)$ represents the second training loss, and $Dis(\mathbf{z}_x - \mathbf{z}_{\tilde{x}})$ represents a distance between two embedding vectors $\mathbf{z}_x$ and $\mathbf{z}_{\tilde{x}}$.

[0151] On the other hand, availability of the target sample is further considered. If stability of the two embedding models on the target sample does not meet a preset stability criterion, the target sample is discarded, to be specific, the target sample does not participate in the distillation learning. If stability of at least one of the two embedding models on the target sample meets the stability criterion, an unstable model learns an embedding representation from a stable model by using the target sample.

[0152] In an embodiment, the stability indicator includes the amount of representation information. In this case, the stability criterion includes the amount of representation information being greater than a preset first threshold. In another embodiment, the stability indicator includes the dispersion degree. In this case, the stability criterion includes the dispersion degree being less than a preset second threshold. In still another embodiment, the stability indicator includes both the amount of representation information and the dispersion degree. In this case, the stability criterion includes the amount of representation information being greater than the first threshold and the dispersion degree being less than the second threshold. For example, the stability criterion may be represented as follows:

$$R_x = \{M_x > \xi\}_1 \& \{\epsilon_x < \eta\}_1 \quad (18)$$

[0153] M_x represents the amount of representation information, ξ represents the first threshold, and $\{M_x > \xi\}_1$ represents that if $M_x > \xi$ is met, 1 is used, otherwise, 0 is used; ϵ_x represents the dispersion degree, η represents the second threshold, and $\{\epsilon_x < \eta\}_1$ represents that if $\epsilon_x < \eta$ is met, 1 used, otherwise, 0 is used; and $\&$ represents a logical operator AND. It may be understood that, if $R_x = 1$ is calculated, it indicates that the stability criterion is met, or if $R_x = 0$ is calculated, it indicates that the stability criterion is not met.

[0154] The above describes content of the stability criterion and a manner of determining whether the criterion is met.

[0155] According to an example, the determining whether the stability meets the criterion may be integrated into a general calculation formula for the second training loss:

$$L_{kd}^{(1)} = \begin{cases} \{\epsilon_x^{(1)} > \epsilon_x^{(2)}\}_1 Dis(\mathbf{z}_x - \mathbf{z}_{\tilde{x}}), & R_x^{(1)} = R_x^{(2)} = 1 \\ R_x^{(2)} Dis(\mathbf{z}_x - \mathbf{z}_{\tilde{x}}), & others \end{cases} \quad (19)$$

$$L_{kd}^{(2)} = \begin{cases} \{\epsilon_x^{(2)} > \epsilon_x^{(1)}\}_1 Dis(\mathbf{z}_x - \mathbf{z}_{\tilde{x}}), & R_x^{(1)} = R_x^{(2)} = 1 \\ R_x^{(1)} Dis(\mathbf{z}_x - \mathbf{z}_{\tilde{x}}), & others \end{cases} \quad (20)$$

[0156] In Formula (19) and Formula (20) above, a superscript (1) corresponds to one of two locally deployed embedding models and a superscript (2) corresponds to the other embedding model. For meanings of other mathematical symbols, reference may be made to the related descriptions in the foregoing embodiment.

[0157] From the above, the learning direction between the two embedding models may be dynamically determined based on the total original sample set and an idea that the unstable model learns the embedding representation from the stable model. Specifically, a second training loss obtained after one embedding model learns an embedding representation from the other embedding model is determined by using a part of the samples, and a second training loss obtained after the other embedding model performs inverse learning is calculated by using a different other part of the samples.

[0158] Afterwards, in step S530, the first embedding vectors output by the predetermined first embedding model of the two embedding models for all the shared samples are provided to the label party.

[0159] It should be noted that, any one of the two embedding models may be specified to provide an embedding vector thereof to the label party, and in this way, compared with the other embedding model, the specified first embedding model additionally uses, in a back propagation process, a third training loss generated during supervised learning for tuning a parameter. Usually, if the two embedding models have different parameter magnitude or different complexity, the one with larger parameter magnitude or a more complex structure is usually set as the first embedding model that provides an embedding vector to the label party. In addition, there is no need to repeat the specifying after the first embedding model is specified, in other words, an embedding model

that provides an embedding vector to the label party in a plurality of rounds of iterative training is always the specified model.

[0160] After the target party provides the first embedding vectors output by the first embedding model for all the shared samples to the label party, the label party is enabled to determine a third training loss based on the first embedding vectors respectively provided by a plurality of parties, a locally deployed fusion model, and a held sample label. It should be understood that, when the sample label is a classification label, the third training loss may be interchangeably referred to as a classification loss. The first training loss, the second training loss, and the third training loss described above are used to correspondingly train models respectively locally deployed by the plurality of parties. It should be understood that, some models output calculation not involved in a specific training loss, and in this case, a derivative or a gradient of the specific training loss with respect to these model parameters is zero, in other words, for these models, parameter tuning results are the same with or without use of the specific training loss.

[0161] It may be understood that, after a forward propagation process and calculation of a training loss of the joint training are clarified, a back propagation process is accordingly determined. A brief description of a back propagation process of local models in the non-label party is provided below.

[0162] For the two embedding models of the target party, a loss function of the first embedding model may be denoted as:

$$L^{(1)} = L_{cla} + \lambda_1 L_{con}^{(1)} + \lambda_2 L_{kd}^{(1)} \quad (21)$$

[0163] In the above formula, L_{cla} represents the third training loss, corresponding to a classification loss of the shared sample determined by performing step S302, S304, and S306 by the label party, $L_{con}^{(1)}$ and $L_{kd}^{(1)}$ respectively represent the first training loss and the second training loss of the first embedding model; and λ_1 and λ_2 are both hyperparameters with a value range of (0, 1), for example, $\lambda_1 = 0.01$ and $\lambda_2 = 0.02$ are set.

[0164] A loss function of the second embedding model (a non-first embedding model of the two embedding models) may be denoted as:

$$L^{(2)} = \beta_1 L_{con}^{(2)} + \beta_2 L_{kd}^{(2)} \quad (22)$$

[0165] In the above formula, $L_{con}^{(2)}$ and $L_{kd}^{(2)}$ respectively represent the first training loss and the second training loss of the second embedding model; and β_1 and β_2 are both

hyperparameters with a value range of (0, 1), for example, $\beta_1 = \beta_2 = 0.05$ is set.

[0166] From the above, back propagation may be performed based on the first training loss, the second training loss, and a backhauled gradient provided by the label party.

[0167] (1) The target party trains the local first embedding model by using the gradient that is calculated by the label party based on the third training loss L_{cla} and that is backhauled to the local first embedding model of the target party, and the first training loss $L_{con}^{(1)}$ and the second training loss $L_{kd}^{(1)}$ of the local first embedding model.

[0168] (2) The target party trains the local second embedding model by using the first training loss $L_{con}^{(2)}$ and the second training loss $L_{kd}^{(2)}$ of the local second embedding model, for example, the losses shown in Formula (22).

[0169] From the above, introduction of private sample data other than the shared sample set for model training is achieved by introducing contrastive learning and knowledge distillation in local training of the non-label party, so that performance of the two local embedding models is improved, thereby improving and optimizing overall prediction performance of jointly trained models. In addition, the distillation learning is different from a conventional manner in which a teacher model and a student model are pre-specified, but instead, a learning direction is determined on a sample-by-sample basis based on stability of the two embedding models in the non-label party for a same sample, and an unstable model learns an embedding representation of the sample from a stable model, to improve a learning effect of the model.

[0170] Corresponding to the above-mentioned training method, embodiments of this specification further provide a training apparatus. FIG. 6 is a schematic diagram of a structure of a multi-party joint training apparatus integrated into a label party according to an embodiment of this specification, where a plurality of parties respectively hold different feature parts of a shared sample set, and the label party in the plurality of parties further holds a sample label. The apparatus shown FIG. 6 is integrated into the label party, and includes the following modules:

a first embedding module 601, configured to process a target sample in a shared sample set by using a local first embedding model, to obtain a first embedding vector; a first prediction module 603, configured to process, by using a local first prediction model, a fusion vector between the first embedding vector and embedding vectors provided by the other parties for the target sample, to obtain a first prediction result; a first loss determining module 605, configured to determine a first training loss based on the first prediction result and a corresponding target sample

label; a second embedding module 607, configured to process the target sample by using a local second embedding model, to obtain a second embedding vector; a second prediction module 609, configured to process the second embedding vector by using a local second prediction model, to obtain a second prediction result; a second loss determining module 611, configured to determine
5 a second training loss based on the second prediction result and the target sample label; and a third loss determining module 613, configured to determine, by using an embedding vector, for the target sample, of an embedding model corresponding to a smaller value in the first training loss and the second training loss as a label, a third training loss of the other embedding model for the target sample, where the first training loss, the second training loss, and the third training loss are
10 used to correspondingly train models respectively locally deployed by the plurality of parties.

[0171] In an embodiment, the first embedding module 601 is further configured to separately use all shared samples in the shared sample set as the target sample. The apparatus 600 further includes: a first training module 615, configured to training the first embedding model by using the first training loss and a third training loss corresponding to the first embedding model; and a
15 second training module 617, configured to train the second embedding model by using the second training loss and a third training loss corresponding to the second embedding model.

[0172] In an embodiment, the label party further holds a private sample set; and the apparatus 600 further includes a fourth loss determining module 619, configured to: process the private sample set by using the second embedding model and the second prediction model, to determine
20 a fourth training loss, where the fourth training loss is used to train the second embedding model and the second prediction model.

[0173] In an embodiment, the apparatus 600 further includes a fifth loss determining module 621, configured to: determine respective fifth training losses of the first embedding model and the second embedding model through contrastive learning based on a total original sample set and a
25 corresponding augmented sample set, where the total original sample set includes a private sample set and the shared sample set, and the fifth training loss is used to train a corresponding embedding model.

[0174] In an embodiment, the embedding vectors provided for the label party by the other parties for the target sample are obtained based on the local first embedding model; and the
30 apparatus 600 further includes: a third training module 623, configured to train the first prediction model by using the first training loss; a gradient calculation module 625, configured to calculate, based on the first training loss, gradients backhauled to first embedding models of the other parties,

so that the other parties train the local first embedding models based on the corresponding gradients; and a fourth training module 627, configured to train the second prediction model by using the second training loss.

[0175] In an embodiment, the apparatus 600 further includes: a sixth loss determining module 629, configured to determine a sixth training loss, which is positively correlated to a distance between a model parameter of the first embedding model and a model parameter of the second embedding model, where the sixth training loss is used to train the first embedding model and the second embedding model.

[0176] In an embodiment, the apparatus 600 further includes: a seventh loss determining module 631, configured to determine a seventh training loss, which is positively correlated to a distance between a model parameter of the first prediction model and a model parameter of the second prediction model, where the seventh training loss is used to train the first prediction model and the second prediction model.

[0177] In an embodiment, the first prediction model and the second prediction model are a same prediction model, and a training loss of the same prediction model for the target sample is a smaller value in the first training loss and the second training loss.

[0178] FIG. 7 is a schematic diagram of a structure of a multi-party joint training apparatus integrated into a non-label party according to an embodiment of this specification, where a plurality of parties respectively hold different feature parts of a shared sample set, and a label party in the plurality of parties further holds a sample label. As shown in FIG. 7, the apparatus 700 includes:

a contrastive learning module 710, configured to determine respective first training losses of two locally deployed embedding models through contrastive learning based on a total original sample set and a corresponding augmented sample set, where the total original sample set includes a private sample set and a shared sample set; a distillation learning module 720, configured to: respectively calculate, by separately using all samples in the total original sample set as a target sample, stability indicators of the two embedding models for the target sample, and determine, by using an embedding vector output by an embedding model with higher stability for the target sample as a label, a second training loss of the other embedding model for the target sample; and an embedding vector sending module 730, configured to: provide first embedding vectors output by a predetermined first embedding model of the two embedding models for all shared samples to the label party, to enable the label party to determine a third training loss based

on the first embedding vectors respectively provided by the plurality of parties, a locally deployed fusion model, and a held sample label, where the first training loss, the second training loss, and the third training loss are used to train models respectively locally deployed by the plurality of parties.

5 **[0179]** In an embodiment, the apparatus 700 further includes a data augmentation module 740, configured to perform data augmentation processing on the total original sample set, to obtain the augmented sample set.

10 **[0180]** In an embodiment, the stability indicator includes an amount of representation information; and the distillation learning module 720 is configured to calculate the amount of representation information, specifically including: determining, for each of the two embedding models, two amounts of representation information corresponding to two embedding vectors obtained by separately processing the target sample and a corresponding augmented sample through the embedding model, and determining an amount of representation information of the embedding model to be a larger value in the two amounts of representation information.

15 **[0181]** In a specific embodiment, the distillation learning module 720 is configured to determine the two amounts of representation information, specifically including: using, for each of the two embedding vectors, a largest value element in the embedding vector as the amount of representation information of the embedding vector.

20 **[0182]** In an embodiment, the stability indicator includes a dispersion degree; and the distillation learning module 720 is configured to calculate the dispersion degree, specifically including: determining, for each of the two embedding models, a difference degree between two embedding vectors obtained by separately processing the target sample and a corresponding augmented sample through the embedding model, as the dispersion degree of the embedding model.

25 **[0183]** In a specific embodiment, the distillation learning module 720 is configured to determine the difference degree, specifically including: calculating a sum of squares of vector elements in a difference vector between the two embedding vectors, a one-norm of the difference vector, or a two-norm of the difference vector, as the difference degree.

30 **[0184]** In an embodiment, the distillation learning module 720 is configured to determine the second training loss, specifically including: based on a calculation result of the stability indicator, when determining that at least one of the two embedding models meets a stability criterion, using

an embedding vector output by an embedding model with higher stability for the target sample as a label, to determine a second training loss of the other embedding model for the target sample.

[0185] In a specific embodiment, the distillation learning module 720 is further configured to discard the target sample when determining that both the two embedding models do not meet the stability criterion.

[0186] Further, in an example, the stability indicator includes the amount of representation information and the dispersion degree, where the stability criterion includes: The amount of representation information is greater than a first threshold and the dispersion degree is less than a second threshold.

[0187] In an embodiment, the apparatus 700 further includes: a target gradient receiving module 750, configured to receive a target gradient from the label party, where the target gradient is a gradient that is determined based on the third training loss and that is backhauled to the first embedding model deployed by the target party; and a first training module 760, configured to train the first embedding model by using the target gradient and the first training loss and the second training loss of the first embedding model.

[0188] In an embodiment, the two embedding models include the first embedding model and the second embedding model; and the apparatus 700 further includes: a second training module 770, configured to train the second embedding model by using the first training loss and the second training loss of the second embedding model.

[0189] An embodiment of another aspect further provides a computer-readable storage medium. The computer-readable storage medium stores a computer program, and when the computer program is executed on a computer, the computer is enabled to perform the method described with reference to FIG. 3 or FIG. 5.

[0190] An embodiment of still another aspect further provides a computing device, including a memory and a processor. The memory stores executable code, and when the processor executes the executable code, the method described with reference to FIG. 3 or FIG. 5 is implemented. A person skilled in the art should be aware that in the above-mentioned one or more examples, the functions described in the present invention can be implemented by hardware, software, firmware, or any combination thereof. When being implemented by software, these functions can be stored in a computer-readable medium or transmitted as one or more instructions or code on a computer-readable medium.

[0191] In the above-mentioned specific implementations, the objectives, technical solutions, and beneficial effects of the present invention are further described in detail. It should be understood that the above-mentioned descriptions are merely specific implementations of the present invention, but are not intended to limit the protection scope of the present invention. Any
5 modification, equivalent replacement, improvement, or the like made based on the technical solutions of the present invention shall fall within the protection scope of the present invention.

CLAIMS

1. A data privacy protection multi-party joint training method, wherein a plurality of parties respectively hold different feature parts of a shared sample set, and a label party in the plurality of parties further holds a sample label; and the method is applied to the label party and comprises:

5 processing a target sample in the shared sample set by using a local first embedding model, to obtain a first embedding vector; processing, by using a local first prediction model, a fusion vector between the first embedding vector and embedding vectors provided by the other parties for the target sample, to obtain a first prediction result; and determining a first training loss based on the first prediction result and a corresponding target sample label;

10 processing the target sample by using a local second embedding model, to obtain a second embedding vector; processing the second embedding vector by using a local second prediction model, to obtain a second prediction result; and determining a second training loss based on the second prediction result and the target sample label; and

15 with a target of reducing a gap between the fusion vector and the second embedding vector, determining a third training loss, for the target sample, of an embedding model corresponding to a larger value in the first training loss and the second training loss, wherein the first training loss, the second training loss, and the third training loss are used to correspondingly train models respectively locally deployed by the plurality of parties.

20 2. The method according to claim 1, comprising: separately using all shared samples in the shared sample set as the target sample; and after determining the third training loss, the method further comprises:

 training the first embedding model by using the first training loss and a third training loss corresponding to the first embedding model; and

25 training the second embedding model by using the second training loss and a third training loss corresponding to the second embedding model.

 3. The method according to claim 1, wherein the label party further holds a private sample set; and the method further comprises:

30 processing the private sample set by using the second embedding model and the second prediction model, to determine a fourth training loss, wherein the fourth training loss is used to train the second embedding model and the second prediction model.

4. The method according to claim 1, further comprising:

determining respective fifth training losses of the first embedding model and the second embedding model based on a total original sample set and a corresponding augmented sample set through contrastive learning, wherein the total original sample set comprises a private sample set and the shared sample set, and the fifth training loss is used to train a corresponding embedding model.

5. The method according to claim 1, wherein the embedding vectors provided for the label party by the other parties for the target sample are obtained based on the local first embedding model; and the method further comprises:

training the first prediction model by using the first training loss, and calculating, based on the first training loss, gradients backhauled to first embedding models of the other parties, so that the other parties train the local first embedding models based on the corresponding gradients; and training the second prediction model by using the second training loss.

6. The method according to claim 1, further comprising:

determining a sixth training loss, which is positively correlated to a distance between a model parameter of the first embedding model and a model parameter of the second embedding model, wherein the sixth training loss is used to train the first embedding model and the second embedding model.

7. The method according to claim 1, further comprising:

determining a seventh training loss, which is positively correlated to a distance between a model parameter of the first prediction model and a model parameter of the second prediction model, wherein the seventh training loss is used to train the first prediction model and the second prediction model.

8. The method according to claim 1, wherein the first prediction model and the second prediction model are a same prediction model, and a training loss of the same prediction model for the target sample is a smaller value in the first training loss and the second training loss.

9. A data privacy protection multi-party joint training apparatus, wherein a plurality of parties respectively hold different feature parts of a shared sample set, and a label party in the plurality of parties further holds a sample label; and the apparatus is integrated into the label party and comprises:

a first embedding module, configured to process a target sample in the shared sample set by

using a local first embedding model, to obtain a first embedding vector;

a first prediction module, configured to process, by using a local first prediction model, a fusion vector between the first embedding vector and embedding vectors provided by the other parties for the target sample, to obtain a first prediction result;

5 a first loss determining module, configured to determine a first training loss based on the first prediction result and a corresponding target sample label;

a second embedding module, configured to process the target sample by using a local second embedding model, to obtain a second embedding vector;

10 a second prediction module, configured to process the second embedding vector by using a local second prediction model, to obtain a second prediction result;

a second loss determining module, configured to determine a second training loss based on the second prediction result and the target sample label; and

15 a third loss determining module, configured to determine, by using an embedding vector, for the target sample, of an embedding model corresponding to a smaller value in the first training loss and the second training loss as a label, a third training loss of the other embedding model for the target sample, wherein the first training loss, the second training loss, and the third training loss are used to correspondingly train models respectively locally deployed by the plurality of parties.

20 10. A computer-readable storage medium, wherein the computer-readable storage medium stores a computer program, and when the computer program is executed in a computer, the computer is enabled to perform the method according to any one of claims 1 to 8.

11. A computing device, comprising a memory and a processor, wherein the memory stores executable code, and when the processor executes the executable code, the method according to any one of claims 1 to 8 is implemented.

25