Committee Machine, Ensemble Learning Random Sampling LDA for Face Recognition

Tae-Kyun (T-K) Kim
KAIST, Imperial College London
https://sites.google.com/view/tkkim/

Unsupervised learning ↔ Supervised learning

Generative model ↔ Discriminative model

PCA

FLD/LDA

Maximum margin classifier

Algorithm-independent learning

Ensemble learning

Randomisation

Committee machine

Online learning

Probabilistic model

Regression

→ Feature extraction → Classification →

PCA

NN classification

FLD/LDA

Discriminant functions

Maximum margin classifier

Subspace learning

- PCA
- FLD/LDA
- Projection/pooling
- Orthonormal basis vectors
- Dimension reduction
- Data visualisation
- Reconstruction

Optimisation ↔ Randomisation

- Gradient-based
- Lagrange multipliers
- Eigenvalue problem
- Generalized eigenvalue problem
- Gaussian distributions

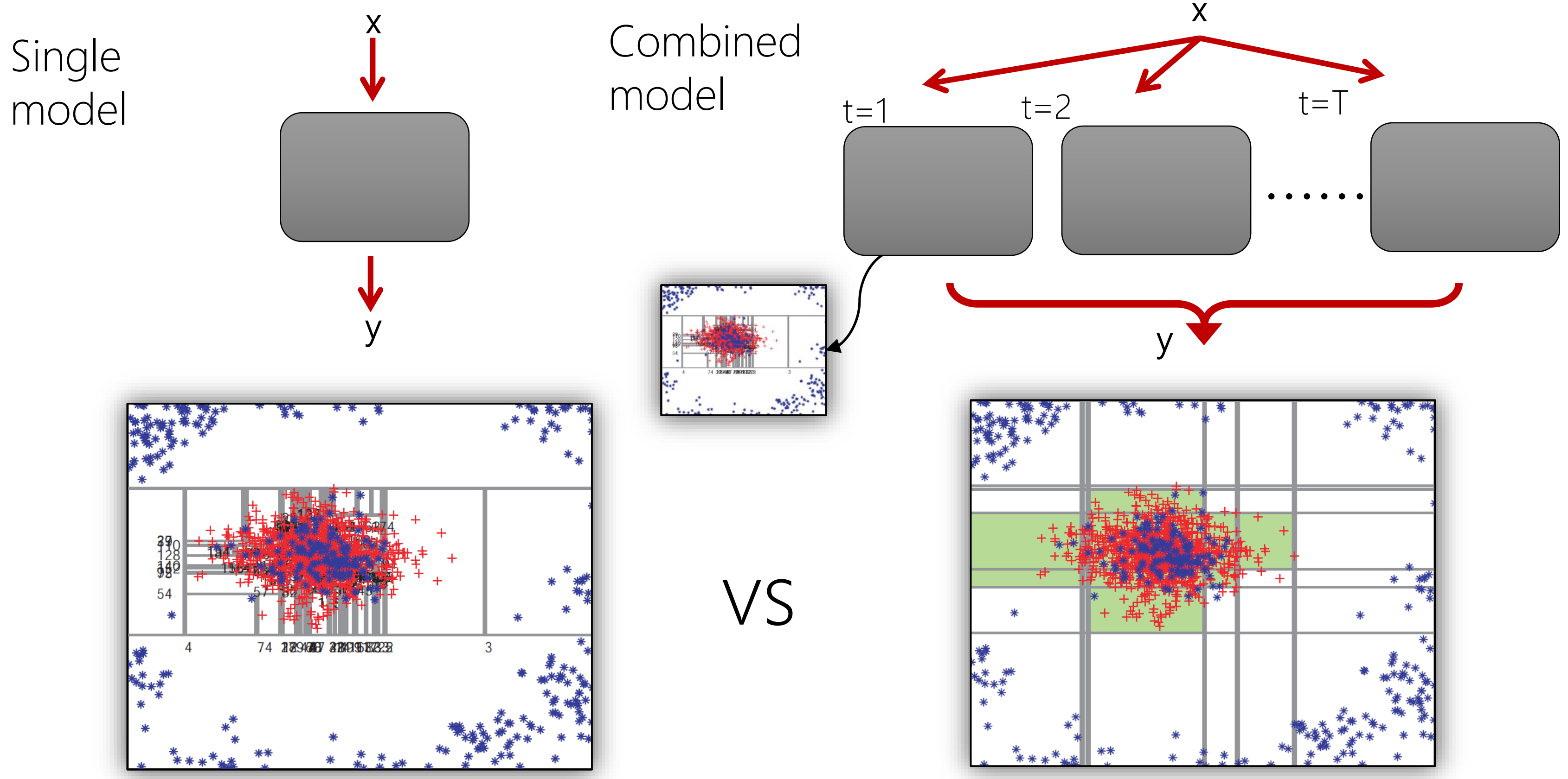
Underfitting → Generalization → Overfitting

- Training loss
- Testing (real) loss

Linear model ↔ Nonlinear model

- Eigenface
- Fisherface
- Kernel trick
- Feature space

Overfitting



Overfit (axis-aligned weaklearners, 2 class problem)

Generalised, smooth decision regions (axis-aligned weaklearners, 2 class problem)

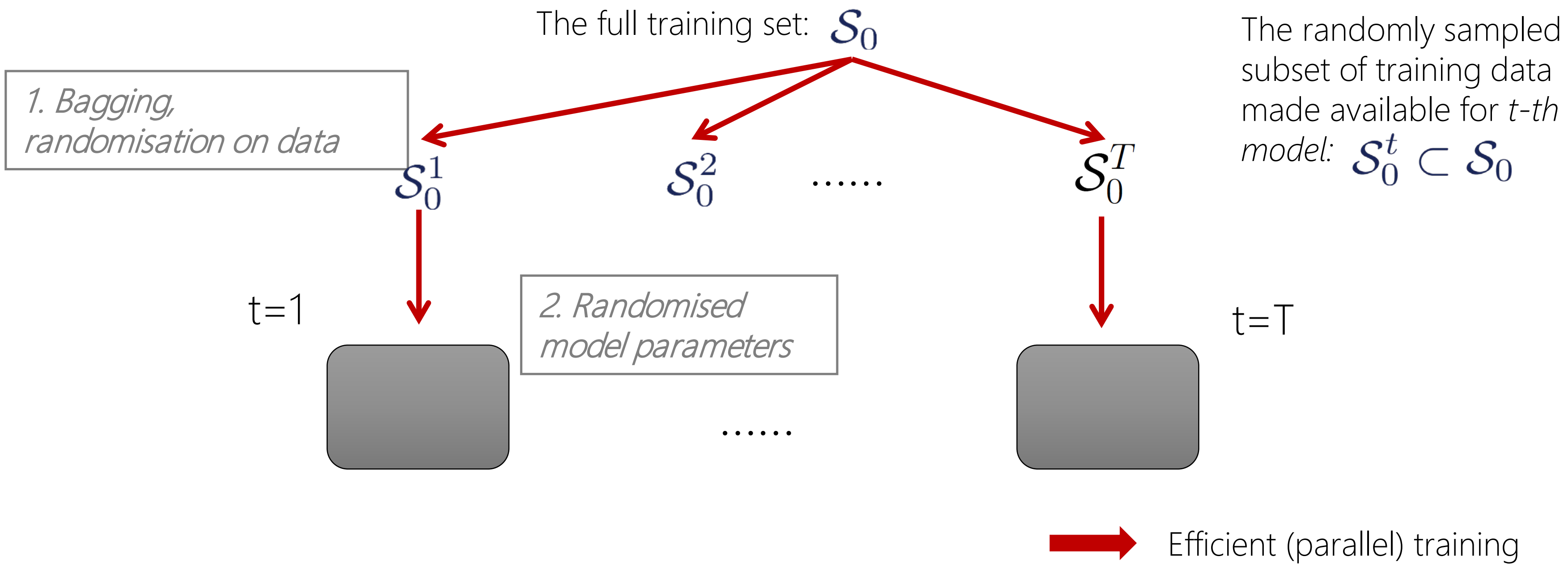
Ensemble of models

- The key aspect of the ensemble model is the fact that its component models are all randomly different from one another.
- This leads to decorrelation between the individual model predictions and, in turn, results in improved generalization and robustness.
- The combined model is characterized by the same components as the individual models.
- The amount of randomness influence the prediction/estimation properties of the models.

* Dropout in deep neural networks ≈ randomisation

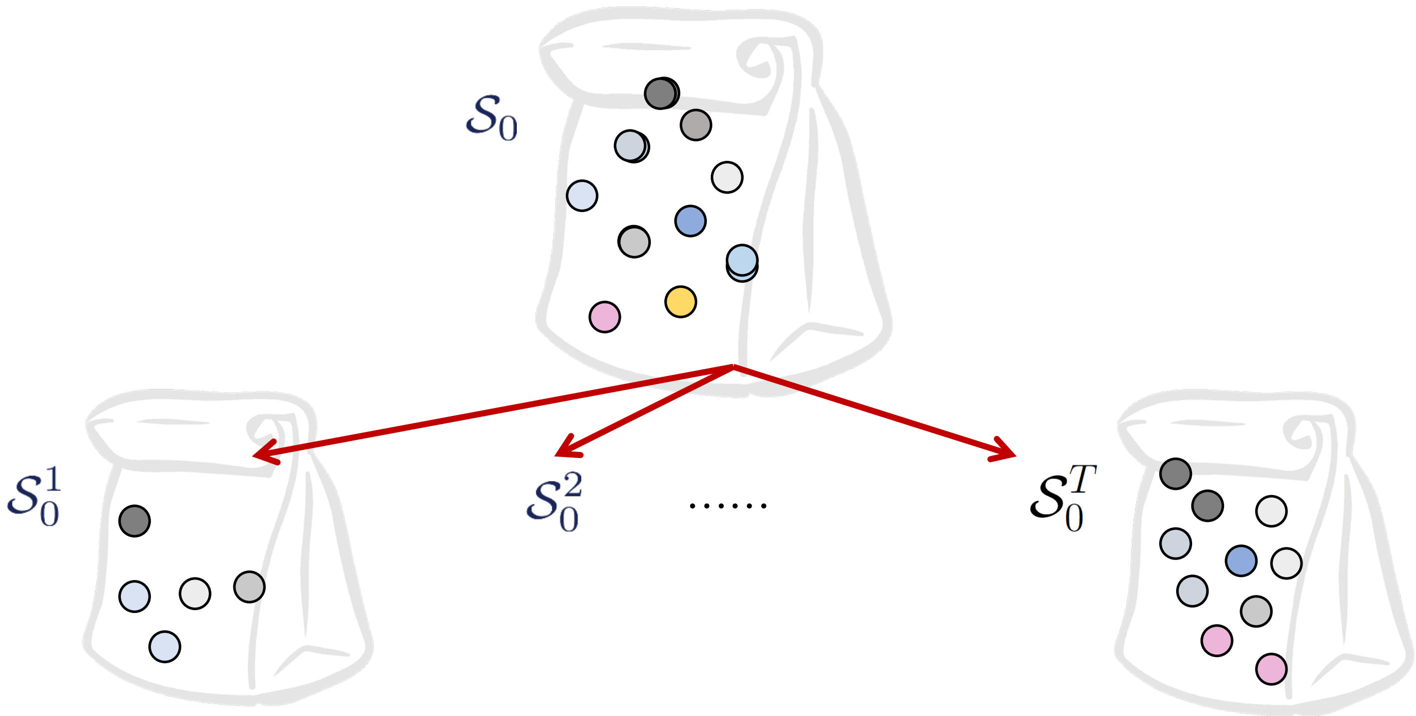
Randomness model

- Randomness is injected into the models during the two phases. Two techniques used together are:
  - random training set sampling (i.e. bagging), and
  - randomized model parameters.



Bagging (Bootstrap AGGregatING) - randomizing the training set

- Given a data set $\mathcal{S}_0$ of size n, it generates T data subsets $\mathcal{S}_0^t$, t=1,…,T.
- Each subset has e.g. n_t=n, by sampling data from $\mathcal{S}_0$ uniformly and with replacement.
- Some data are repeated in $\mathcal{S}_0^t$. If n_t=n and n is large, $\mathcal{S}_0^t$ is likely to have 63.2% of unique data.



Randomizing model parameters

- Given a data subset $\mathcal{S}_0^t$, the t-th model is learnt.
- We may express the model learning as an optimisation problem:

$$\theta^* = \arg\max_{\theta \in T} F$$

where the full set of all possible parameters (or their values) is denoted by T.
- A small random subset $T_t \subset T$ of parameters is considered.
- The randomness parameter $\rho = |T_t|$.
- Thus under the randomness training a model is achieved by optimizing

$$\theta^* = \arg\max_{\theta \in T_t} F$$

Randomizing model parameters

- The randomness parameter $\rho = |T_t|$ controls not only the amount of randomness within each model but also the amount of correlation between different models in the ensemble.
- As illustrated, when $\rho = |T|$ all the models will be identical (if no bagging) and as ρ decreases the models become more decorrelated.

The effect of ρ

$\rho = |T|$

$\rho = 1$

Low randomness, high model correlation

High randomness, low model correlation

Model correlation vs strength

- Randomisation on data and model parameters increases diversity among component models.
- For the fixed data, the randomised model parameters decreases strength of each model.
- This compromising issue is further explained in the perspective of a generic committee machine.

Committee machine

- We consider multiple models or experts, $\mathrm{y}_t(x),\ t = 1, \ldots, T$.
- Output of each model is

$$\mathrm{y}_t(x) = h(x) + \epsilon_t(x)$$

where $h(x), \epsilon_t(x)$ are the true value and error of each model.

- The average sum-of-squares error is

$$\mathrm{E}[\{y_t(x) - h(x)\}^2] = \mathrm{E}[\epsilon_t(x)^2]$$

- The average error by acting individually is $\mathrm{E}_{av} = \frac{1}{T}\sum_{t=1}^{T} \mathrm{E}[\epsilon_t(x)^2]$

Committee machine

- The committee machine is

$$y_{com}(x) = \frac{1}{T}\sum_{t=1}^{T} y_t(x)$$

- The expected error of the committee machine is

$$\mathrm{E}_{com} = E\left[\left\{\frac{1}{T}\sum_{t=1}^{T} y_t(x) - h(x)\right\}^2\right]$$

$$= E\left[\left\{\frac{1}{T}\sum_{t=1}^{T} \epsilon_t(x)\right\}^2\right] = E[\frac{1}{T^2}(\epsilon_1^2 + \epsilon_1\epsilon_2 + \epsilon_2^2 + \cdots)]$$

Committee machine

- If we assume

$$E\left[\epsilon_i(x)\epsilon_j(x)\right] = 0,$$

$$\text{for any } i, j \in \{1, \dots, T\}\ and\ i \neq j$$

then we obtain

$$E_{com} = \frac{1}{T} E_{av}$$

- In practice, the errors are typically highly correlated, but we can still expect that

$$E_{com} \leq E_{av}$$

Prediction models and testing

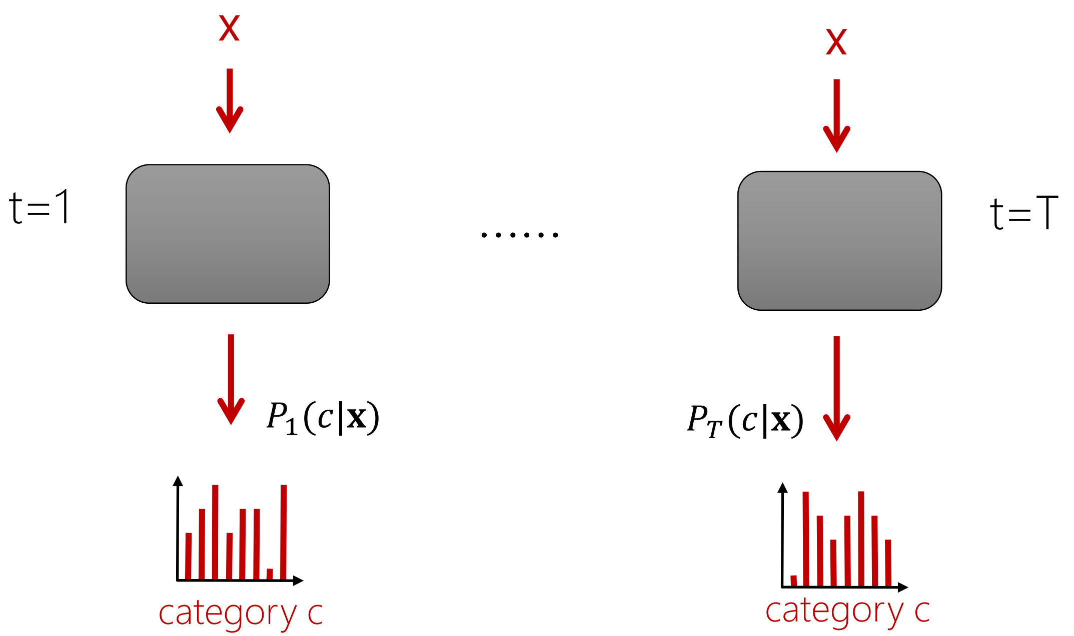
- In an ensemble with T models we use the variable $t \in \{1, . . . , T\}$ to index each component model.
- All models are trained independently (and possibly in parallel).
- During testing, each test point $\mathbf{x}$ is simultaneously pushed through all models.
- Testing can also often be done in parallel, thus achieving high computational efficiency on modern parallel CPU or GPU hardware.
- Combining all model predictions into a single prediction is done by a simple averaging operation. <u>E.g. in classification</u>

$$P(c|\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} P_t(c|\mathbf{x})$$

where $P_t(c|\mathbf{x})$ denotes the class posterior distribution obtained by the t-th model.

Ensemble of models: evaluation

- A data point is passed down all models, and the respective posterior distributions are collected.



– Classification is done by $$P(c|\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} P_t(c|\mathbf{x})$$

Prediction models and testing

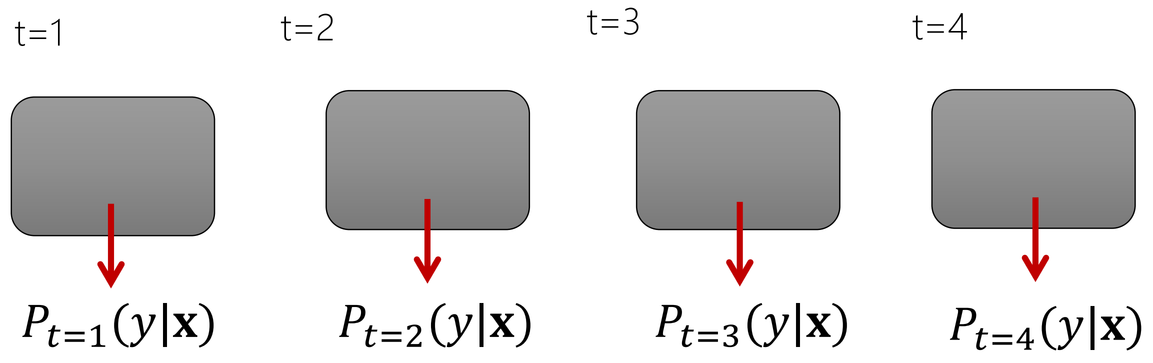
- Alternatively one could also multiply the model outputs together (though the models are not statistically independent)

$$P(c|\mathbf{x}) = \frac{1}{Z}\prod_{t=1}^{T} P_t(c|\mathbf{x})$$

with Z ensuring probabilistic normalization.

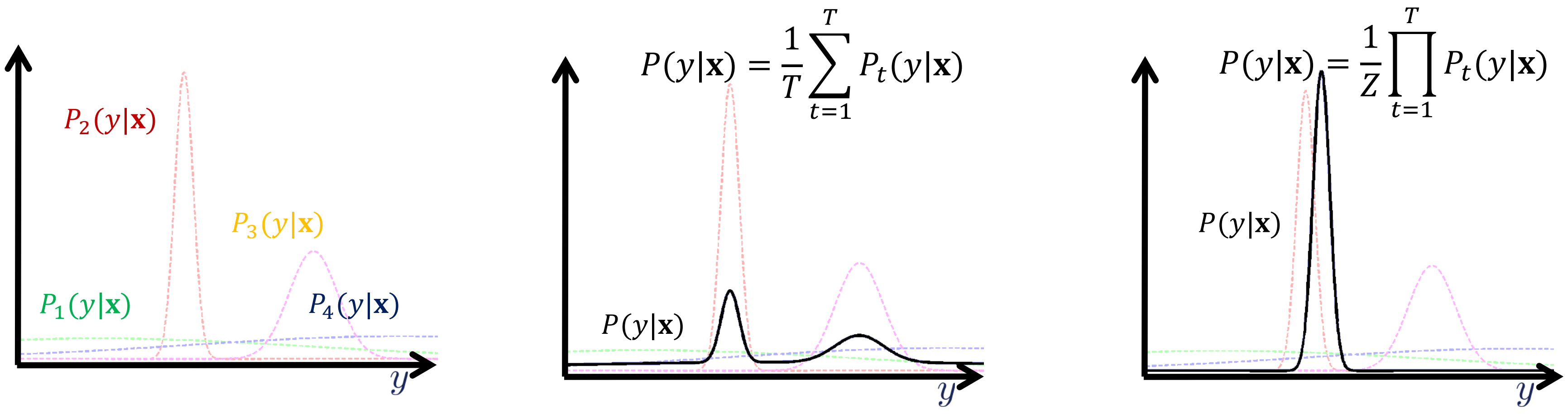
Prediction models and testing

- Model output fusion is illustrated in the next slide, for a simple example where the attribute we want to predict is a continuous variable *y*.
- Imagine that we have trained an ensemble with $T = 4$ models.
- For a test data point **x**, we get the corresponding posteriors $p_t(y|\mathbf{x})$, with $t = \{1, \ldots, 4\}$.



Prediction models and testing

- Some models produce peakier (more confident) predictions than others.
- Both the averaging and the product operations produce combined distributions (shown in black) which are heavily influenced by the most confident i.e. most informative models.
- Therefore, such simple operations have the effect of selecting (softly) the more confident models out of the ensemble.
- Averaging many posteriors also has the advantage of reducing the effect of possibly noisy model contributions.
- In general, the product based ensemble model may be less robust to noise.



Prediction models and testing

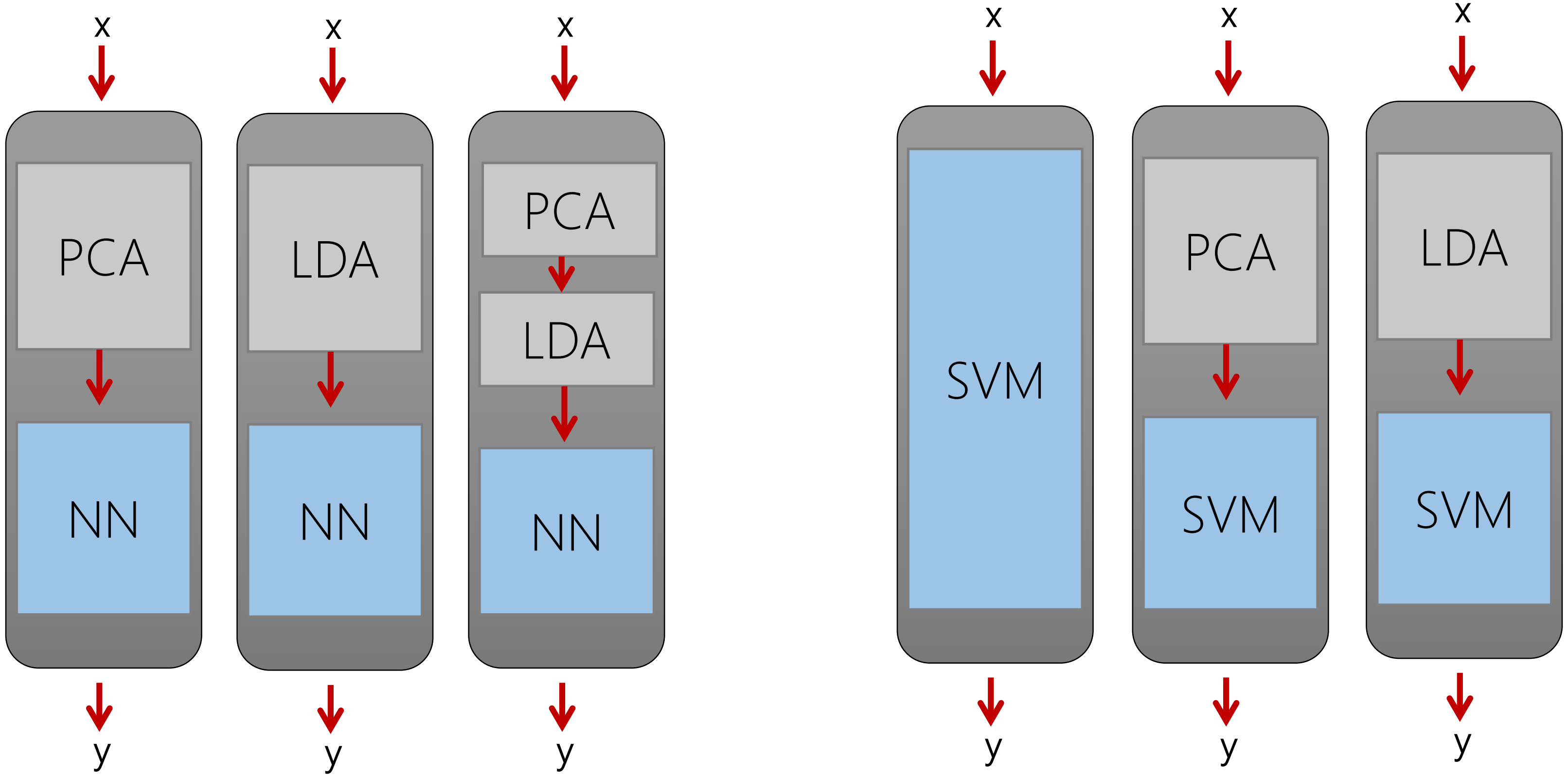
- Alternative ensemble models are possible, where for instance one may choose to select individual models in a hard way, or may do majority voting.

- Min: $P(y|\mathbf{x}) = \min_t P_t(y|\mathbf{x})$

- Max: $P(y|\mathbf{x}) = \max_t P_t(y|\mathbf{x})$

- Majority voting (in classification):
  - each learned model votes for a class to assign to a query image.
  - Classification of the query image is by assigning the class has the highest number of 'votes'.

In our case, each single model can be



More discriminative than PCA

When $\mathbf{S}_W^{-1}$ is not attainable

SVM is a global optimiser?

PCA helps the computational complexity of SVM, but accuracy?

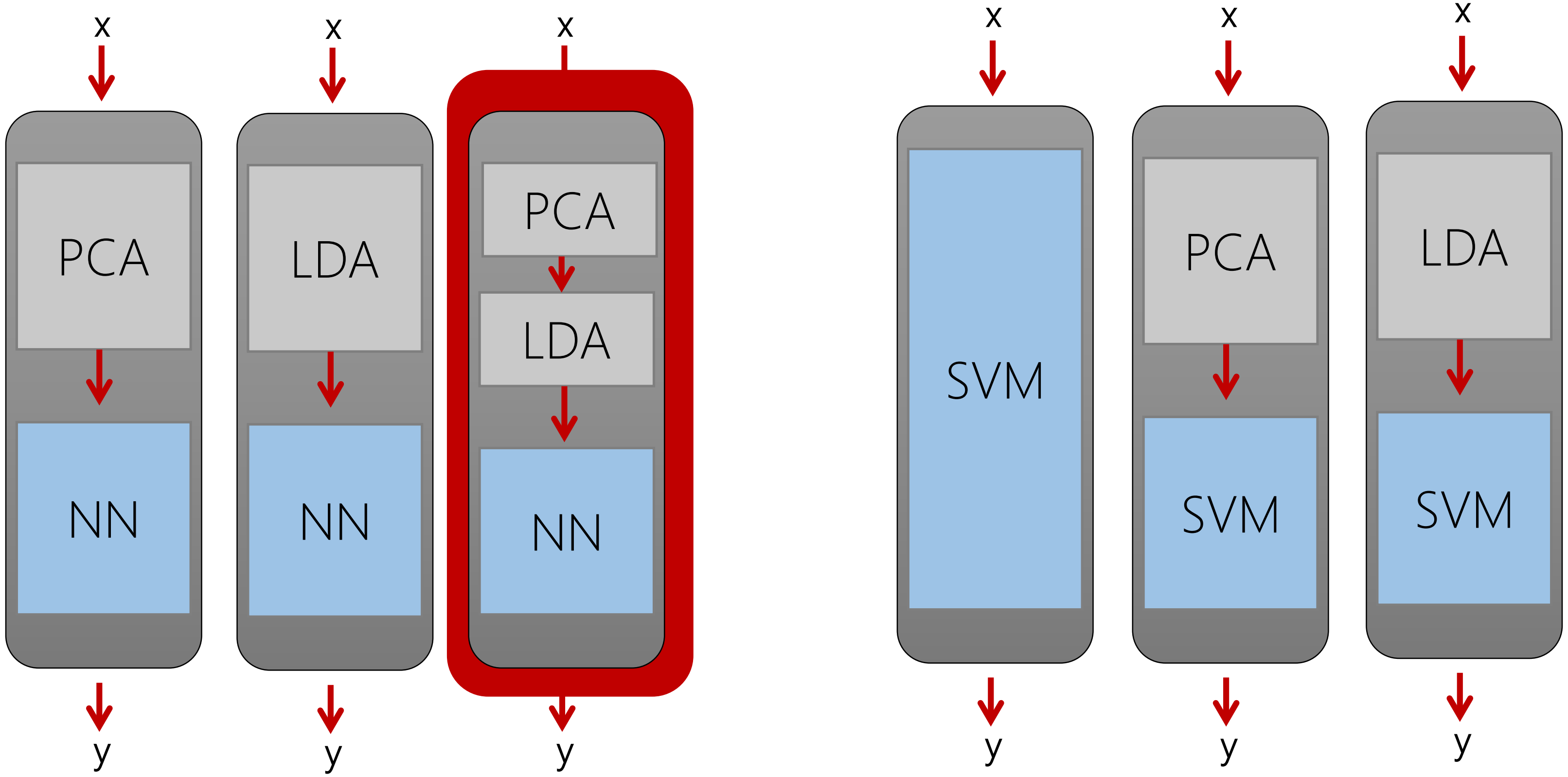
Two discriminative parts in a sequence?

Random Sampling LDA for Face Recognition

Tae-Kyun (T-K) Kim
KAIST, Imperial College London
https://sites.google.com/view/tkkim/

X. Wang and X. Tang, Random Sampling for Subspace Face Recognition, IJCV, 70(1), 91–104, 2006

A base model for ensemble learning is



More discriminative than PCA

When $\mathbf{S}_W^{-1}$ is not attainable

SVM is a global optimiser?

PCA helps the computational complexity of SVM, but accuracy?

Two discriminative parts in a sequence?

Random sampling on training data

- In bagging, random bootstrap replicates are generated by sampling the training set, so each replicate has a smaller number of (unique) training samples.
- We first project the high dimensional image data to the N−1 dimension PCA subspace. For N training samples, there are at most N−1 eigenvectors with nonzero eigenvalues.

(1) Apply PCA to the face training set with N samples for c classes.
Project all the face data to the N−1 eigenfaces $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{\mathrm{N}-1}]$.

(2) Generate *T* bootstrap replicates $\{S_t\}_{t=1}^{T}$.
Each replicate contains the training images of c1 individuals randomly selected from the c classes, or a random subset of images for each of the c classes.

(3) Construct a PCA-LDA classifier from each replicate and combine the multiple classifiers using a fusion rule.
Mpca and Mlda need to be chosen.

Random sampling in feature space

- We first project the high dimensional image data to the N−1 dimension PCA subspace before random sampling.
- In Fisherface, overfitting happens when the training set is relatively small compared to the high dimensionality of the feature vector.
- In order to construct a stable LDA classifier, we sample a small subset of features.
- By the random sampling, we construct multiple stable LDA classifiers.
- We then combine these classifiers to construct a more powerful classifier that covers the entire feature space without losing discriminant information.

Random sampling in feature space

At the training stage:

Consider N images {$\mathbf{x}_n$}, n = 1,…,N and $\mathbf{x}_n \in \mathrm{R}^D$ in an D-dimensional image space, and assume that each image belongs to one of c classes.

(1) Apply PCA to the face training set:

All the eigenfaces with zero eigenvalues are removed, and N−1 eigenfaces $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{\mathrm{N}-1}]$ are retained.

(2) Generate T random subspaces $\{R_t\}_{t=1}^T$:

Each random subspace R_t is spanned by M0 + M1 dimensions.

The first M0 dimensions are fixed as the M0 largest eigenfaces in $\mathbf{W}$.

The remaining M1 dimensions are randomly selected from the other N−1−M0 eigenfaces in $\mathbf{W}$.

(3) T LDA classifiers $\{y_t^R(\mathrm{x})\}_{t=1}^T$ are constructed from the T random subspaces.

Mpca (=M0+M1) and Mlda need to be chosen.

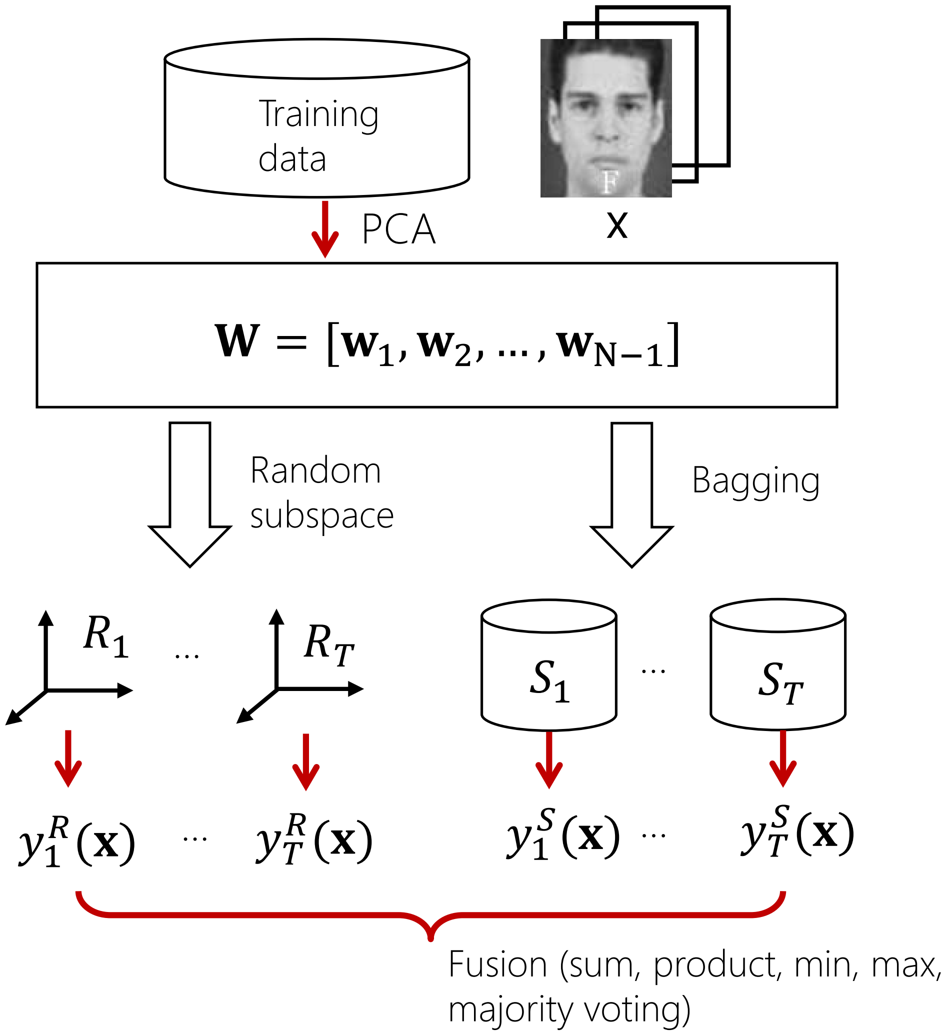
Random sampling in feature space

At the testing stage:

(1) The input face data is projected to *T* random subspaces and fed to *T* PCA-LDA classifiers in parallel.

(2) The outputs of the *T* PCA-LDA classifiers are combined using a fusion scheme (e.g. sum, product, min, max, majority voting) to make the final decision.

Random sampling based PCA-LDA (Fisherface)



X. Wang and X. Tang, Random Sampling for Subspace Face Recognition, IJCV, 70(1), 91–104, 2006