Supervised Classifiers

- Support vector machine [Cortes1995]
  - Linear (too simple)
  - Non-linear (over fitting)
  - Require a parametric discrimination function
- Regression (sparseness)
  - Logistic regression
  - Neural network
- Nearest neighbor (large memory required)
  - KD-trees
- Bayesian network (probabilistic modeling)
- Ensemble classifiers (long training time)
  - Random Forest [Breiman2001]

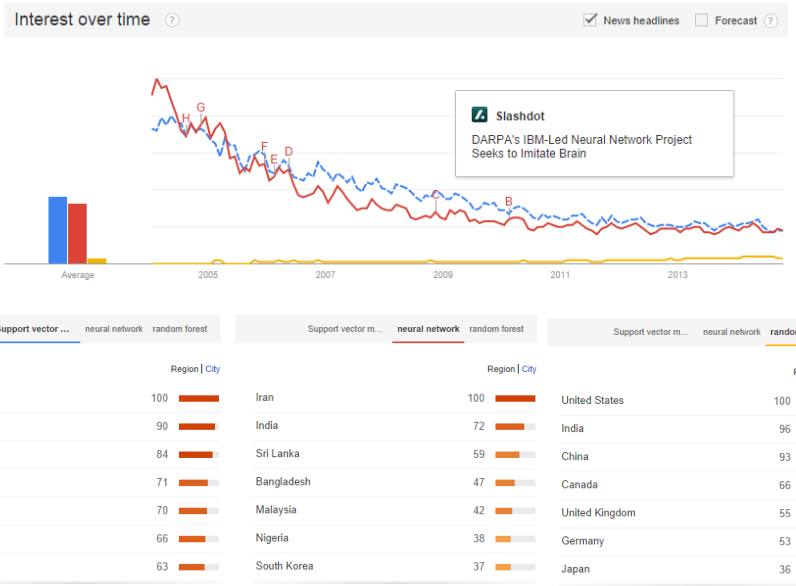
SVM vs RandomForest

Support-vector networks
C Cortes, V Vapnik - Machine learning, 1995 - Springer
Abstract The support-vector network is a new learning machine for two-group classification problems. The machine conceptually implements the following idea: input vectors are non-linearly mapped to a very high-dimension feature space. In this feature space a linear ...
Cited by 15079 Related articles All 39 versions Web of Science: 5510 Cite Save

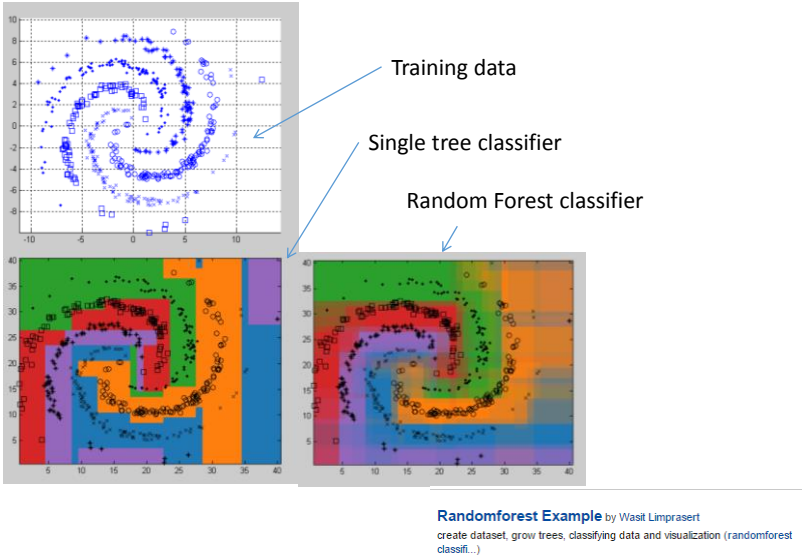
Random forests
L Breiman - Machine learning, 2001 - Springer
Abstract **Random forests** are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error for forests converges as to a limit as the ...
Cited by 13630 Related articles All 83 versions Web of Science: 5942 Cite Save

Real-time human pose recognition in parts from single depth images
J Shotton, T Sharp, A Kipman, A Fitzgibbon... - Communications of the ..., 2013 - dl.acm.org
Abstract We propose a new method to quickly and accurately predict human pose---the 3D positions of body joints---from a single depth image, without depending on information from preceding frames. Our approach is strongly rooted in current object recognition strategies. ...
Cited by 1542 Related articles All 55 versions Web of Science: 26 Cite Save

Some Statistic of statistical classifiers



Randomforest example on MATLAB



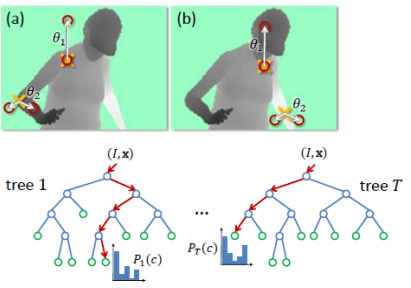
Challenges

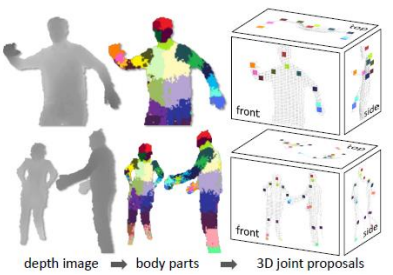
Previous challenges	New challenges
Accuracy Studying on some small dataset and adjusting parameters to get best result	**Big data** Huge amount of data generated.
Small number of samples Collecting data was so expensive	**Tasks distribution** Incoming data larger than computation capacity of single machine to process
Over fitting problem because of spending to much time on adjusting parameters	**Scalability** Able to expand computational power from 10 to 10,000 machines
Limited computational power Computational power was limited. A small number of samples could be processed in a limited time.	**Failure is normal** Standard datacentre has 1% of AFR.

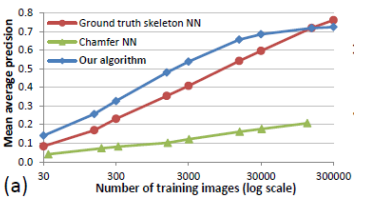
[1] https://blog.codecentric.de/en/2013/11/hardware-will-fail-just-way-expect/
[2]https://gigaom.com/2013/09/19/facebooks-data-centers-are-pioneering-open-source-hardware/

[Shotton2011] Real-Time Human Pose Recognition in Parts from Single Depth Images

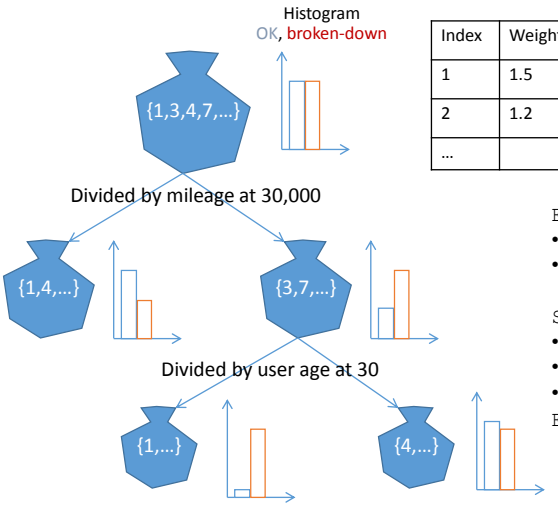
- A pair consists of
  - a depth image and
  - a ground-truth (label) image
- took 24 hr for training with 1,000 cores system
- Classifying at 200fps on Xbox







Splitting Decision tree



Index	Weight	Mileage	User age	province	...
1	1.5	15000	35	1	
2	1.2	40000	25	70	
...					...

Each splitting needs
- field (θ) and
- threshold (τ)

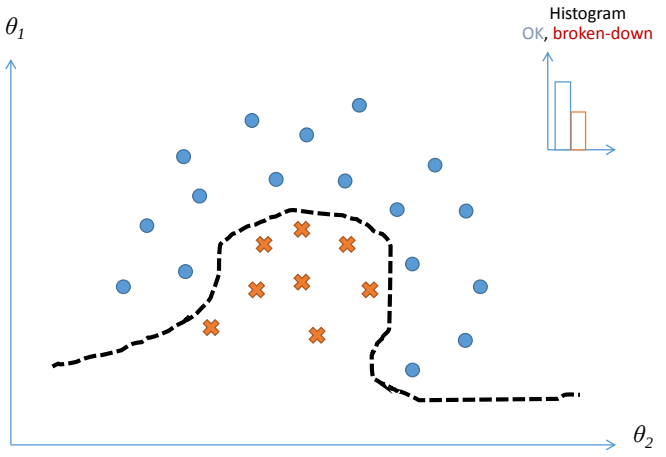
Stop when
- Information gain < 0
- Reaching max depth
- reaching minimum bag size

Entropy (H)

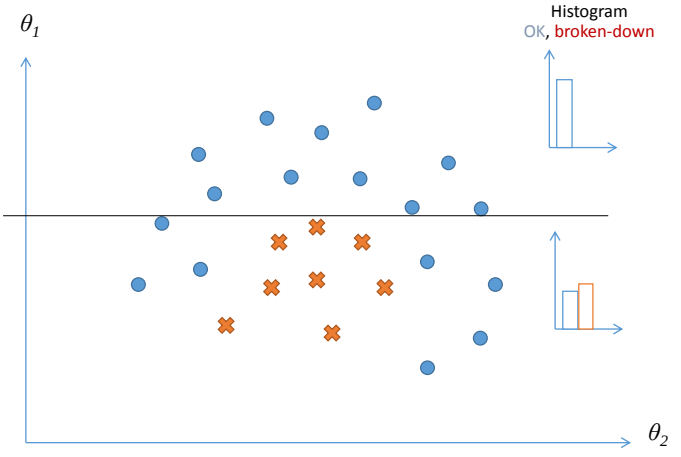
$$H = -\sum_i P_i . log(Pi)$$

Init
  Construct Q={x}
For i
  Random splitting parameter (theta and tau)_i
  calculate G(theta,tau)_i
Find G_max
Use (theta,tau)_max to split Q int QL and QR

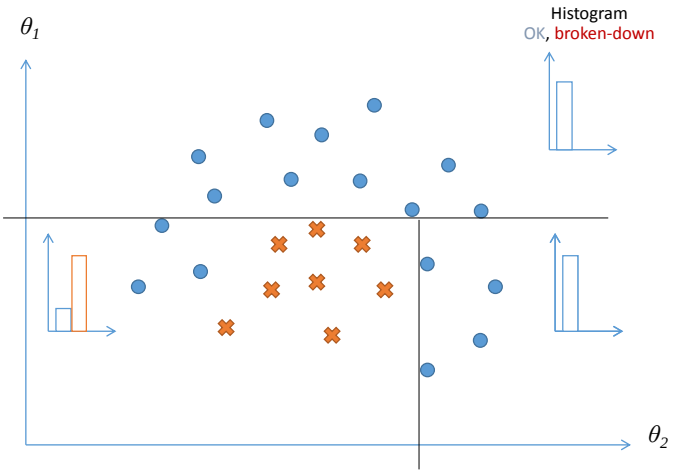
Feature Space



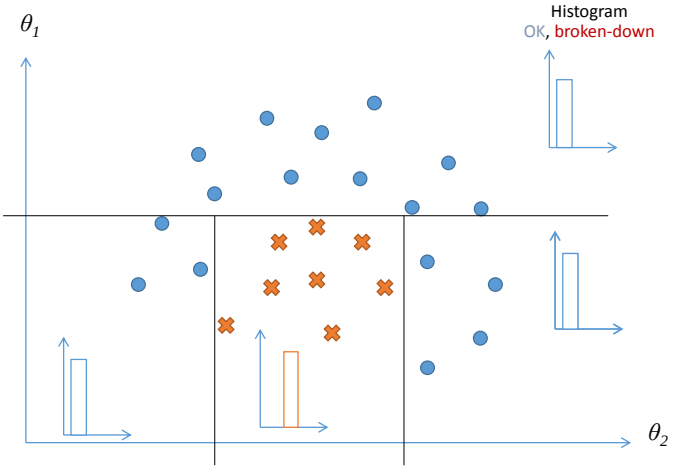
Feature Space



Feature Space



Feature Space



Algorithm

- A bag Q contains a collections of samples
- For each proposal
  - Randomly propose splitting parameters θ and find the threshold τ, $\phi=\{\theta, \tau\}$
  - Use ϕ to split the bag into Q_L and Q_R
  - Compute information gain
  - $G = H - \frac{|Q_L|.HL+|Q_R|.HR}{|Q|}$
    - $H = \sum_i P_i log(Pi)$
- Repeat the splitting recursively until reaching the maximum depth

Output tree

```
<node param=" 3 5 12 4 9 0 0" theshold="0.00">
    <node param=" 1 7 14 15 2 0 0" theshold="6.00">
        <node hist=" 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000"/>
        <node hist=" 0.000 0.000 0.004 0.000 0.008 0.000 0.000 0.000 0.008 0.004 0.977"/>
    </node>
    <node param=" 9 15 8 14 10 0 0" theshold="0.00">
        <node param=" 1 15 7 2 9 0 0" theshold="18.00">
            <node hist=" 0.044 0.032 0.032 0.187 0.162 0.156 0.004 0.011 0.008 0.013 0.352"/>
            <node hist=" 0.000 0.000 0.837 0.000 0.000 0.002 0.000 0.000 0.000 0.000 0.161"/>
        </node>
        <node hist=" 0.002 0.436 0.001 0.011 0.011 0.001 0.008 0.154 0.005 0.008 0.362"/>
    </node>
</node>
```