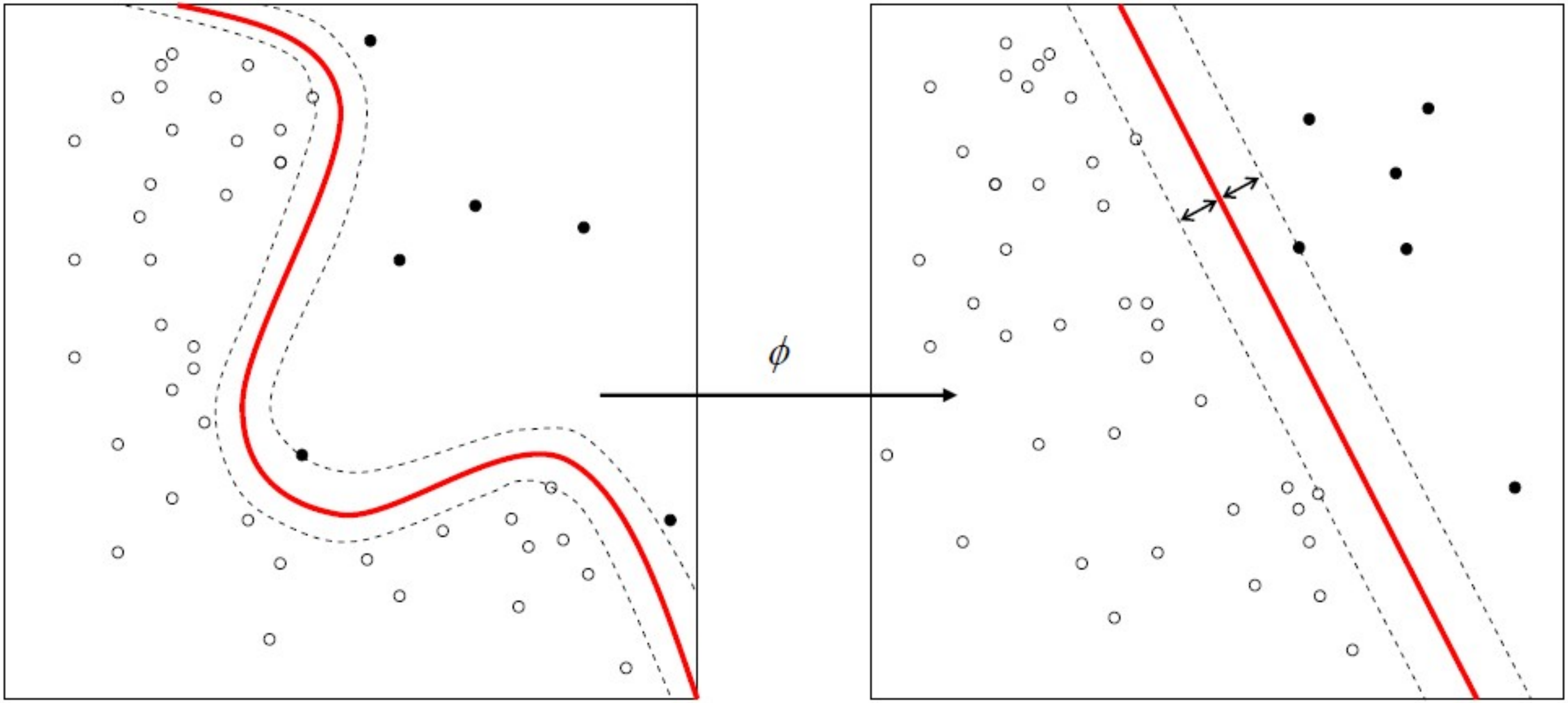
Kernels

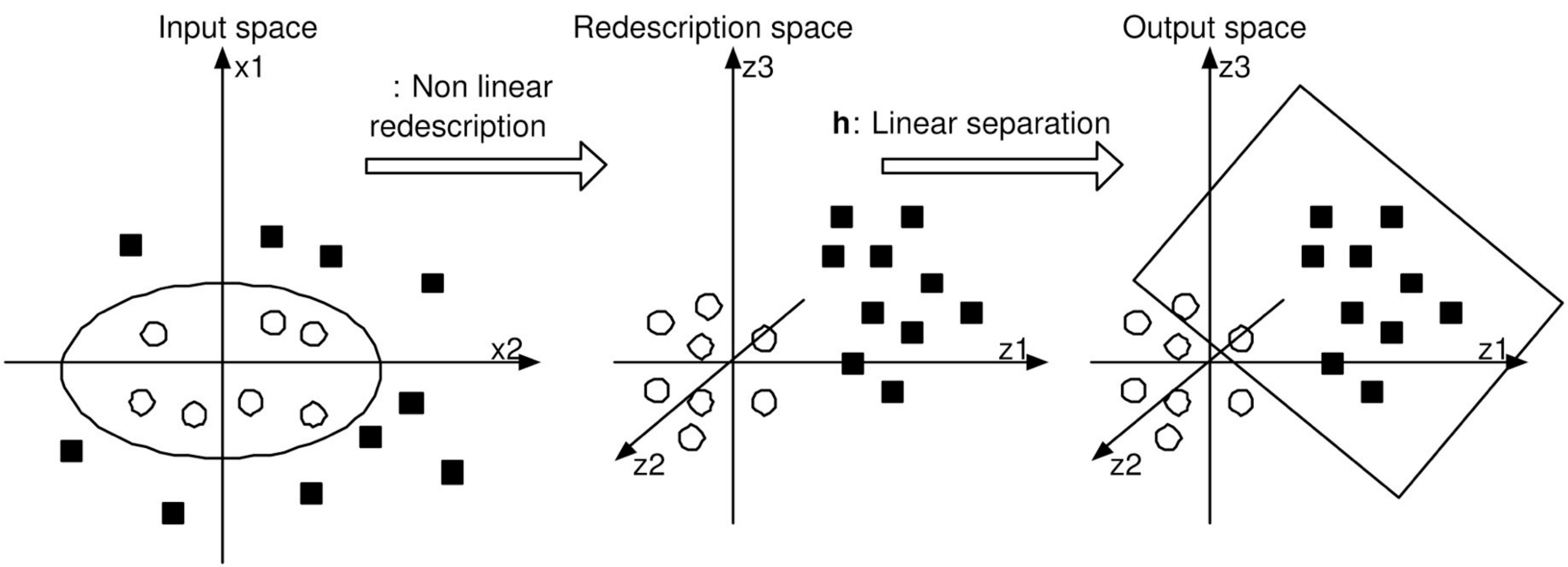
Nonlinear SVMs

Nonlinear data is linear in higher dimensions

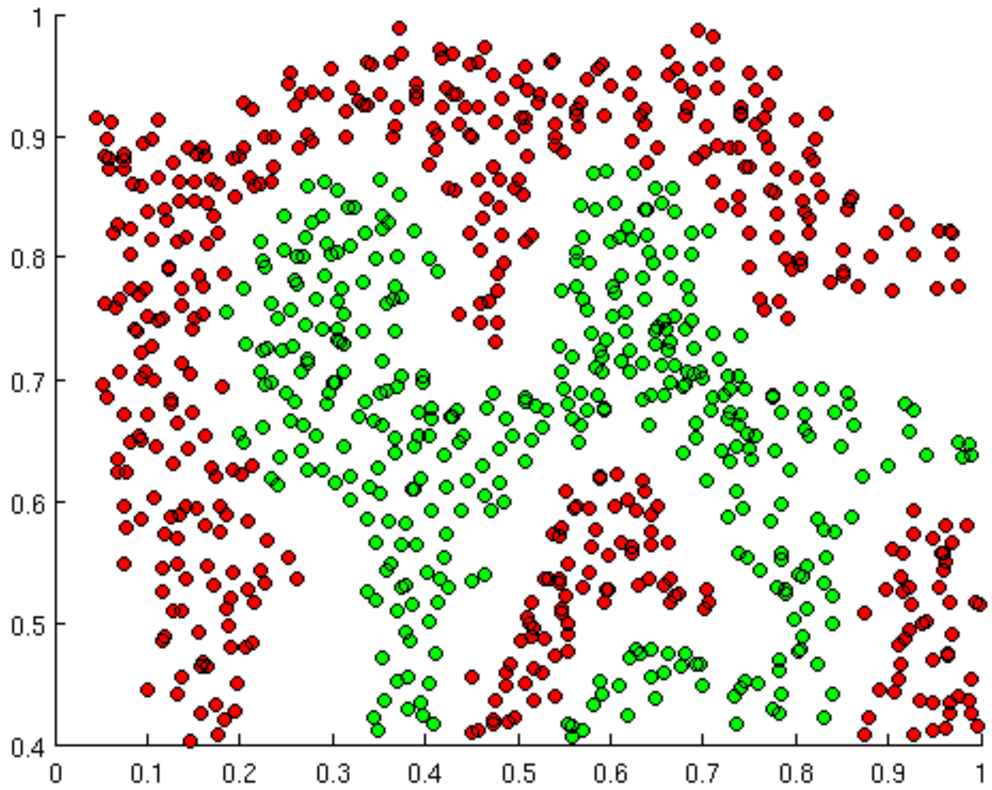


Kernel Trick

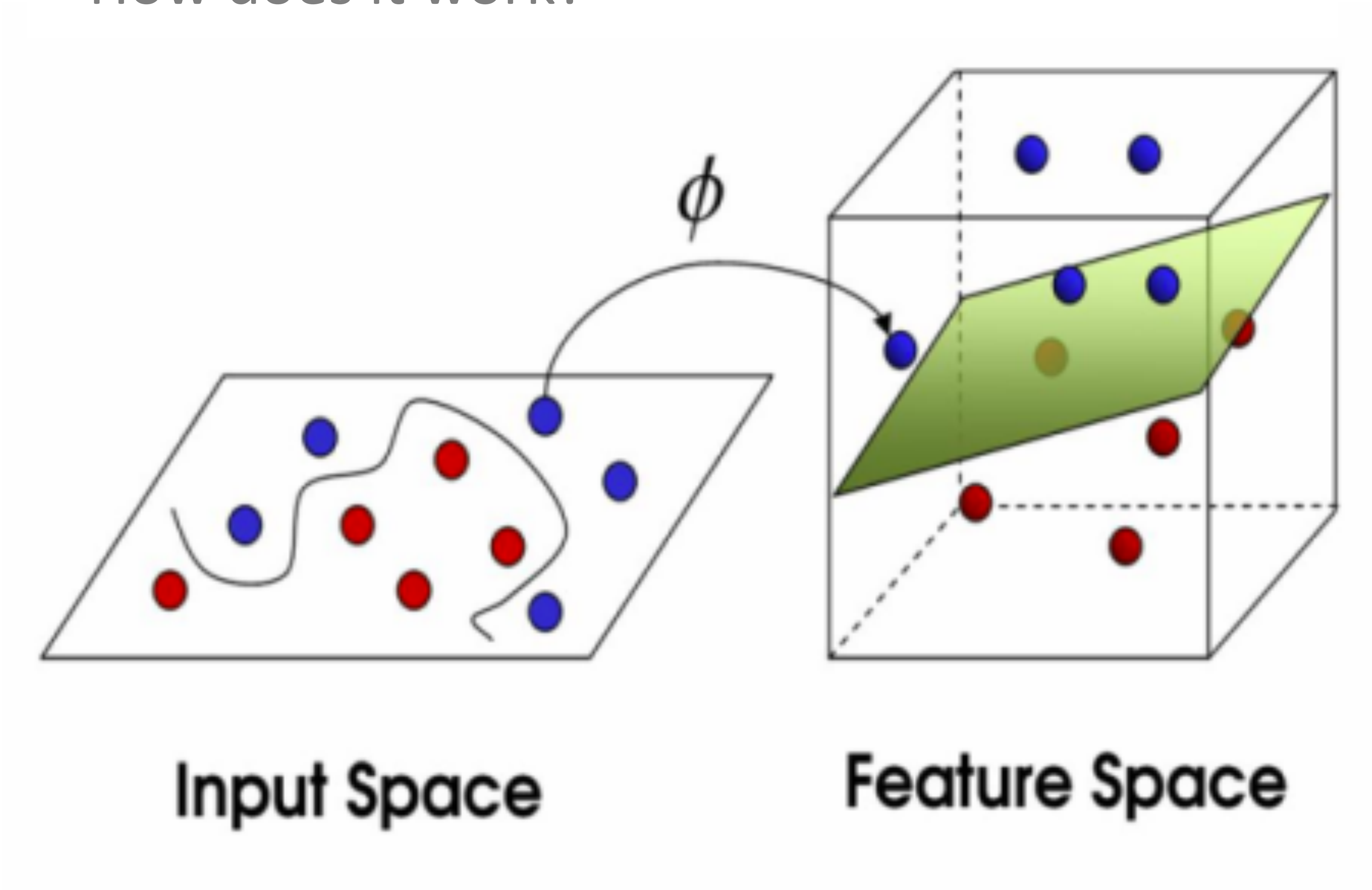
Transform data so it is linearly separable



SVMs can fit intricate boundaries with the Gaussian (RBF) kernel

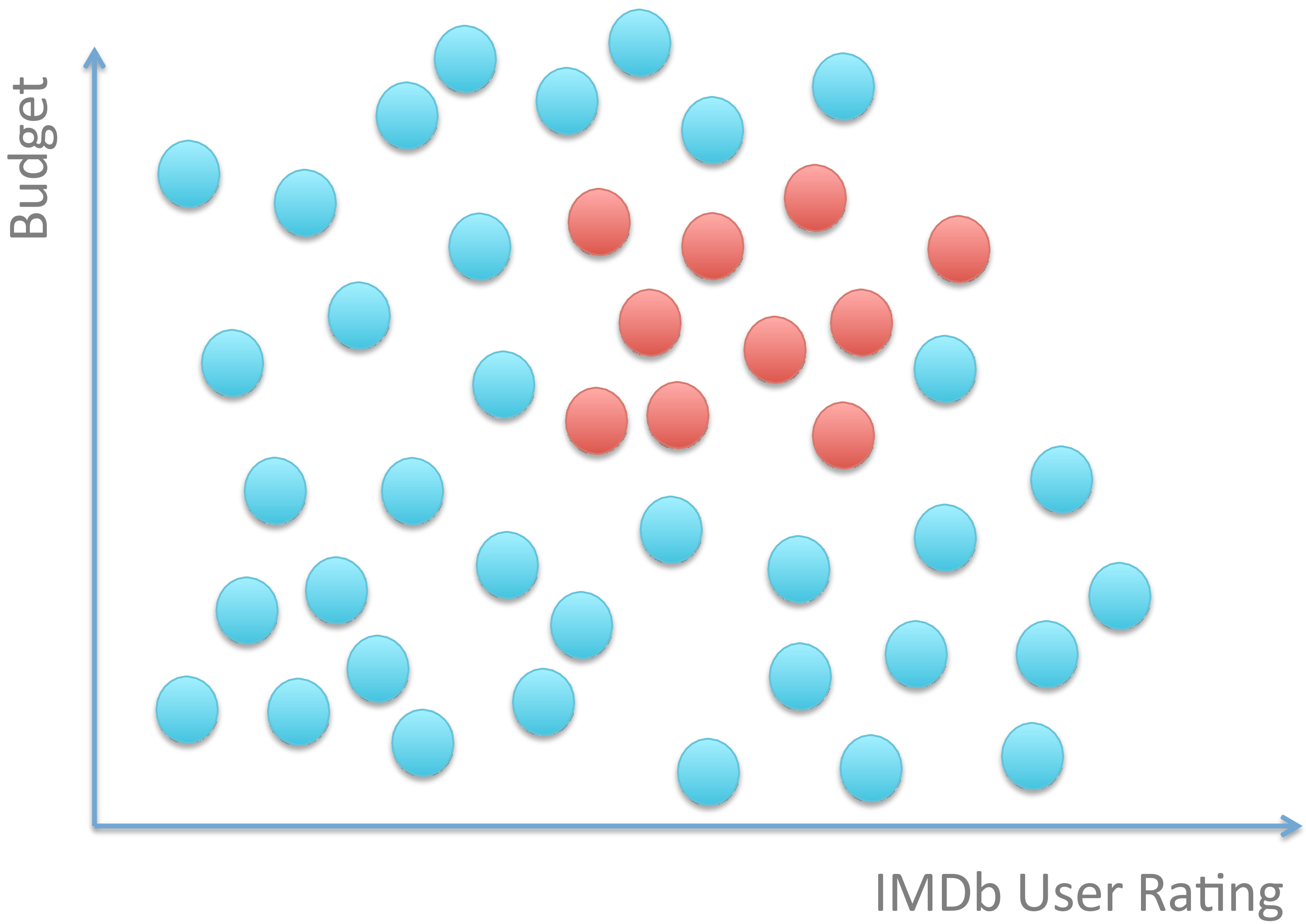


Gaussian Kernel
How does it work?



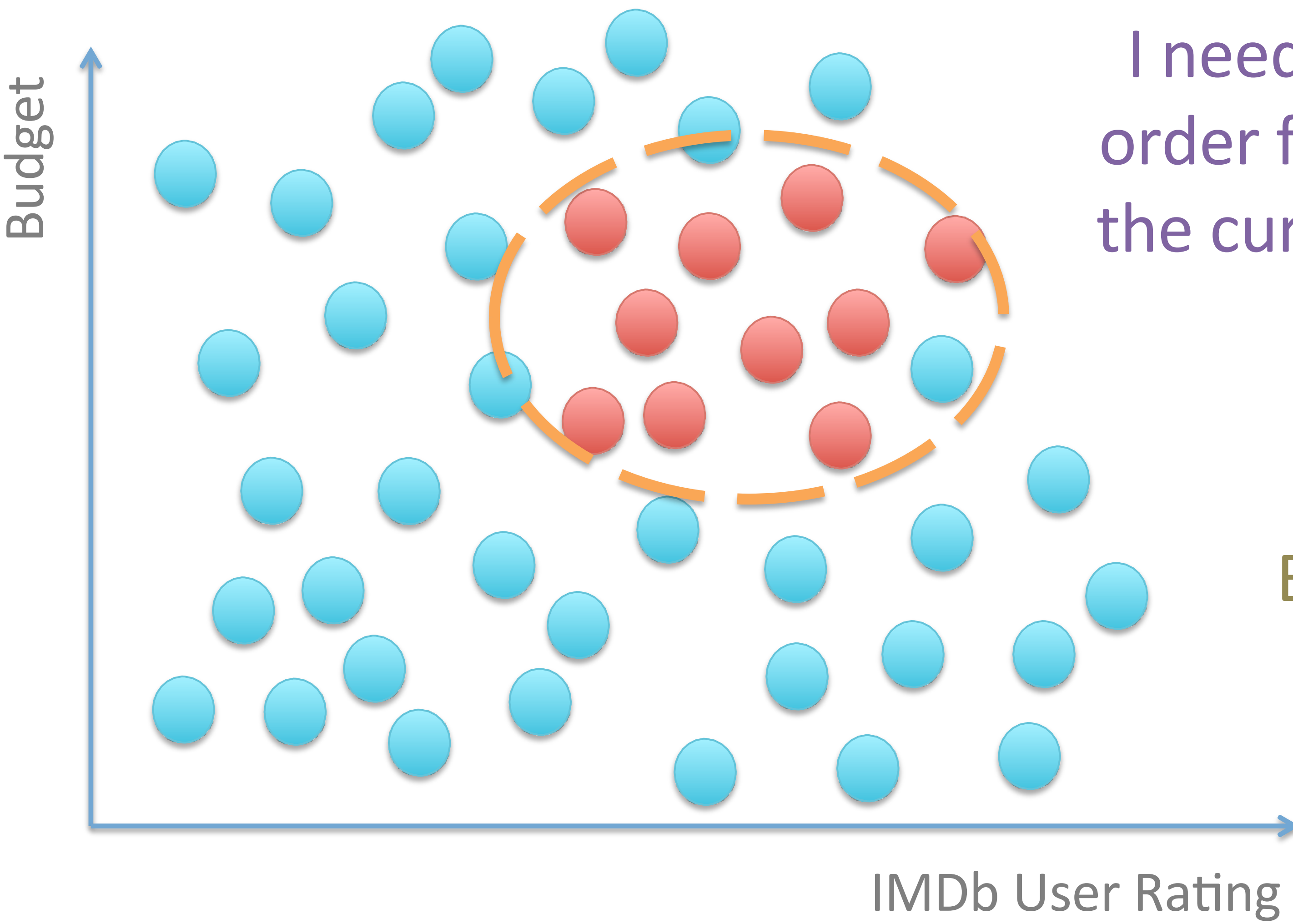
Palme d'Or Winners at Cannes

Pretty intricate boundary



Palme d'Or Winners at Cannes

Pretty intricate boundary



I need to use higher order features to get the curved boundary

$Budget^2$

$Rating^2$

Budget*Rating

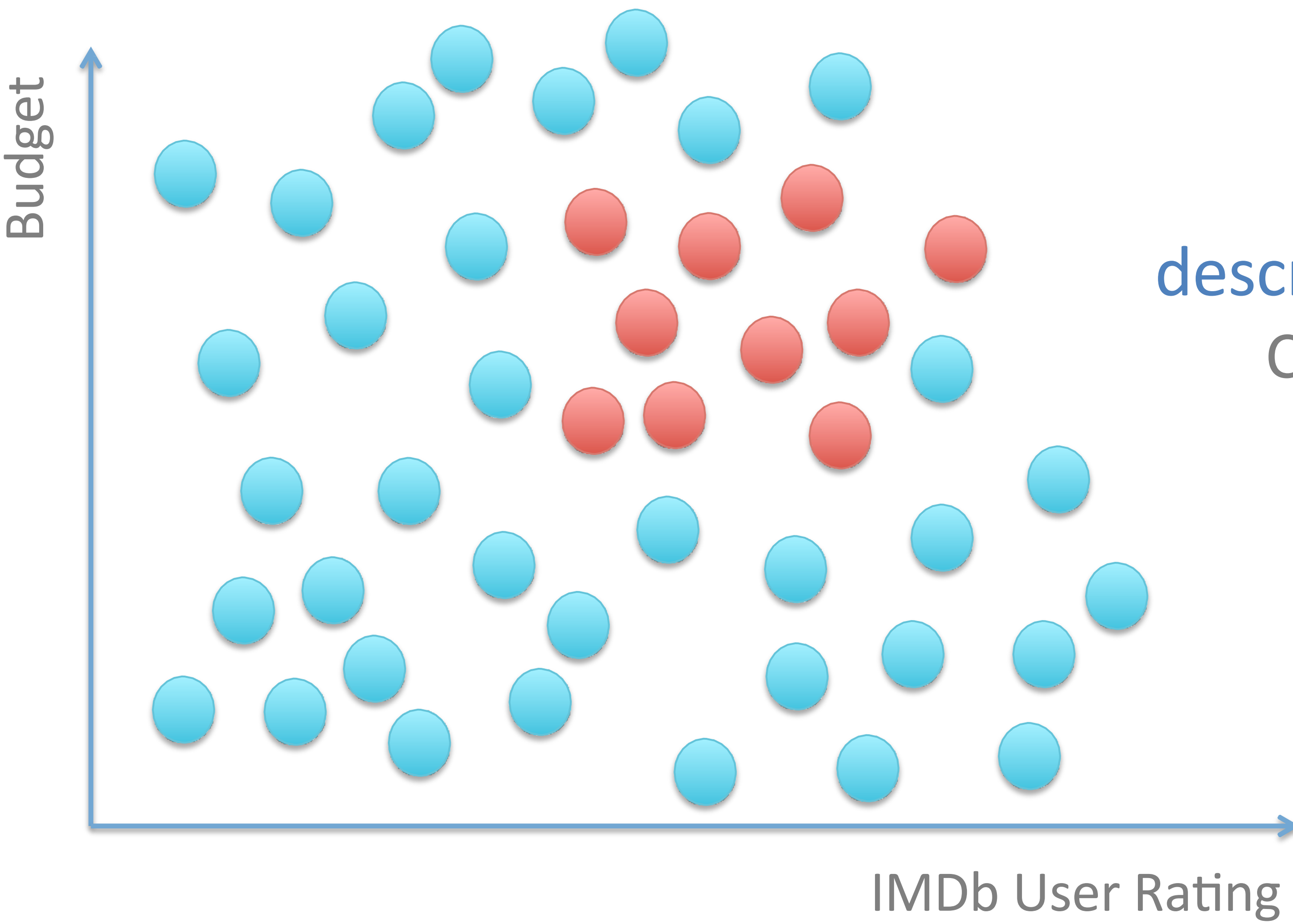
$Budget^3$

$Rating^3$

etc.

Palme d'Or Winners at Cannes

Different Approach: Transform the space



Two features describe this space

Can I describe it differently?

Palme d'Or Winners at Cannes

Different Approach: Transform the space

Two features describe this space

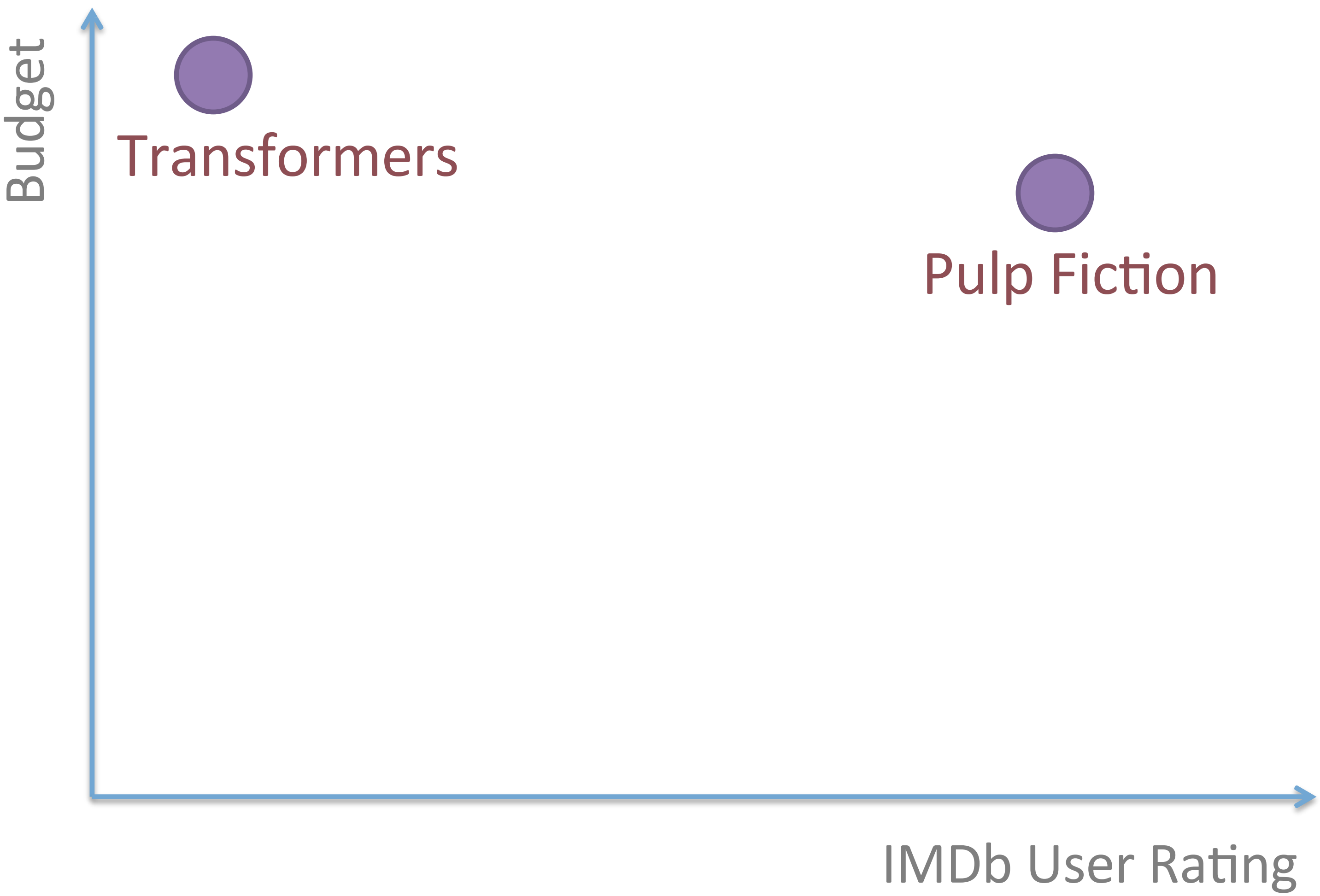
Can I describe it differently?

Palme d'Or Winners at Cannes

Different Approach: Transform the space

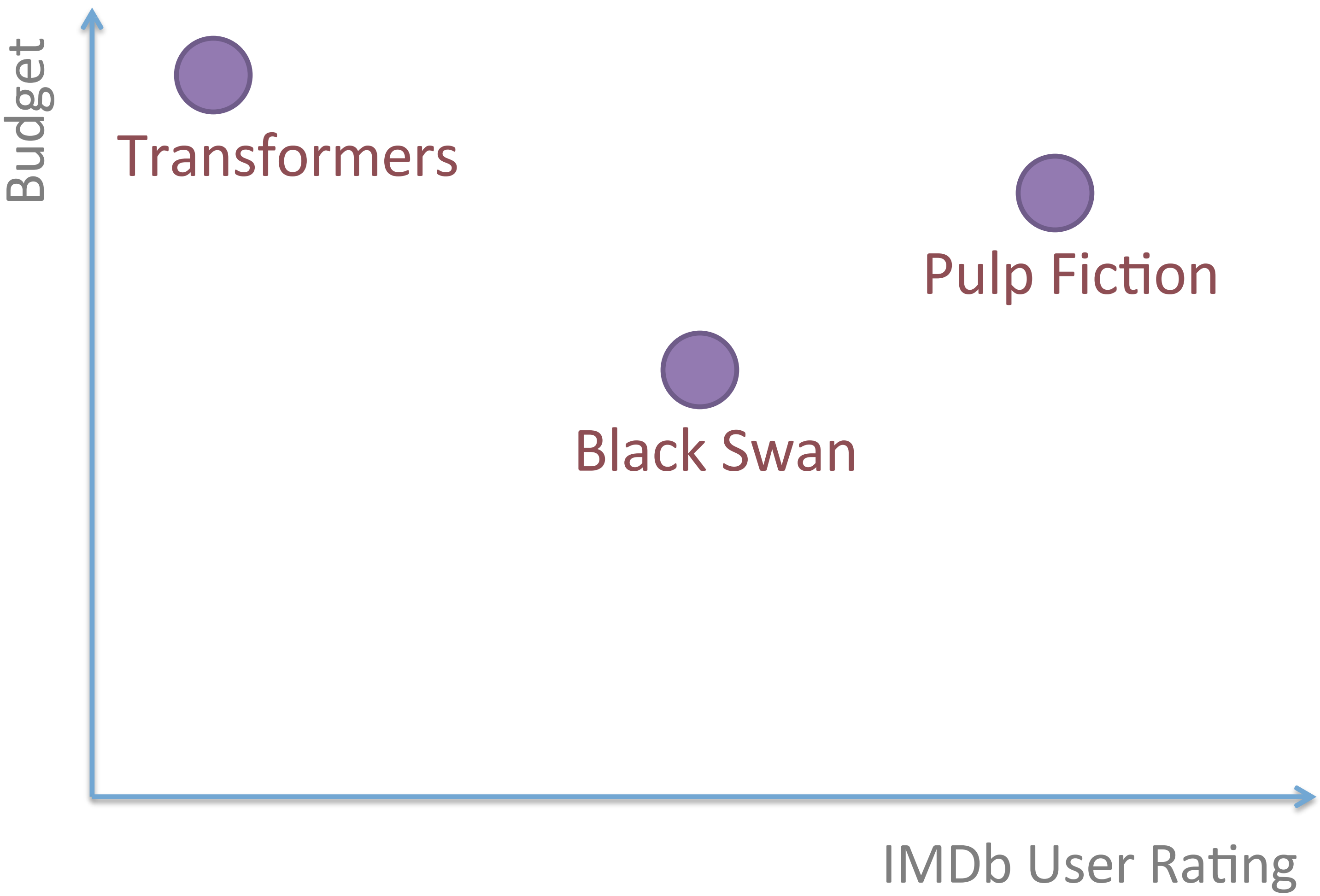
Palme d'Or Winners at Cannes

Different Approach: Transform the space



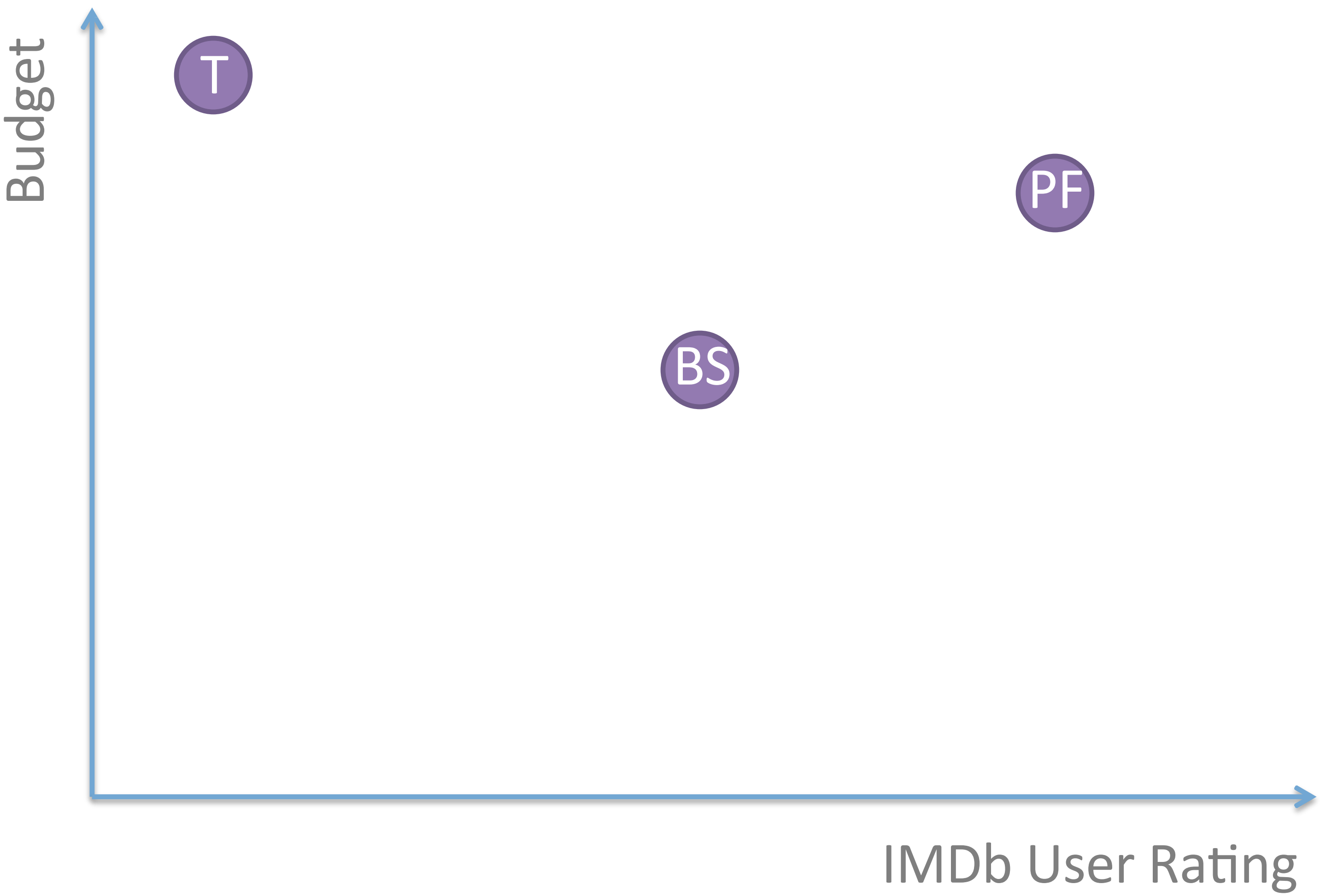
Palme d'Or Winners at Cannes

Different Approach: Transform the space



Define feature a_1: "Pulp Fiction"ness

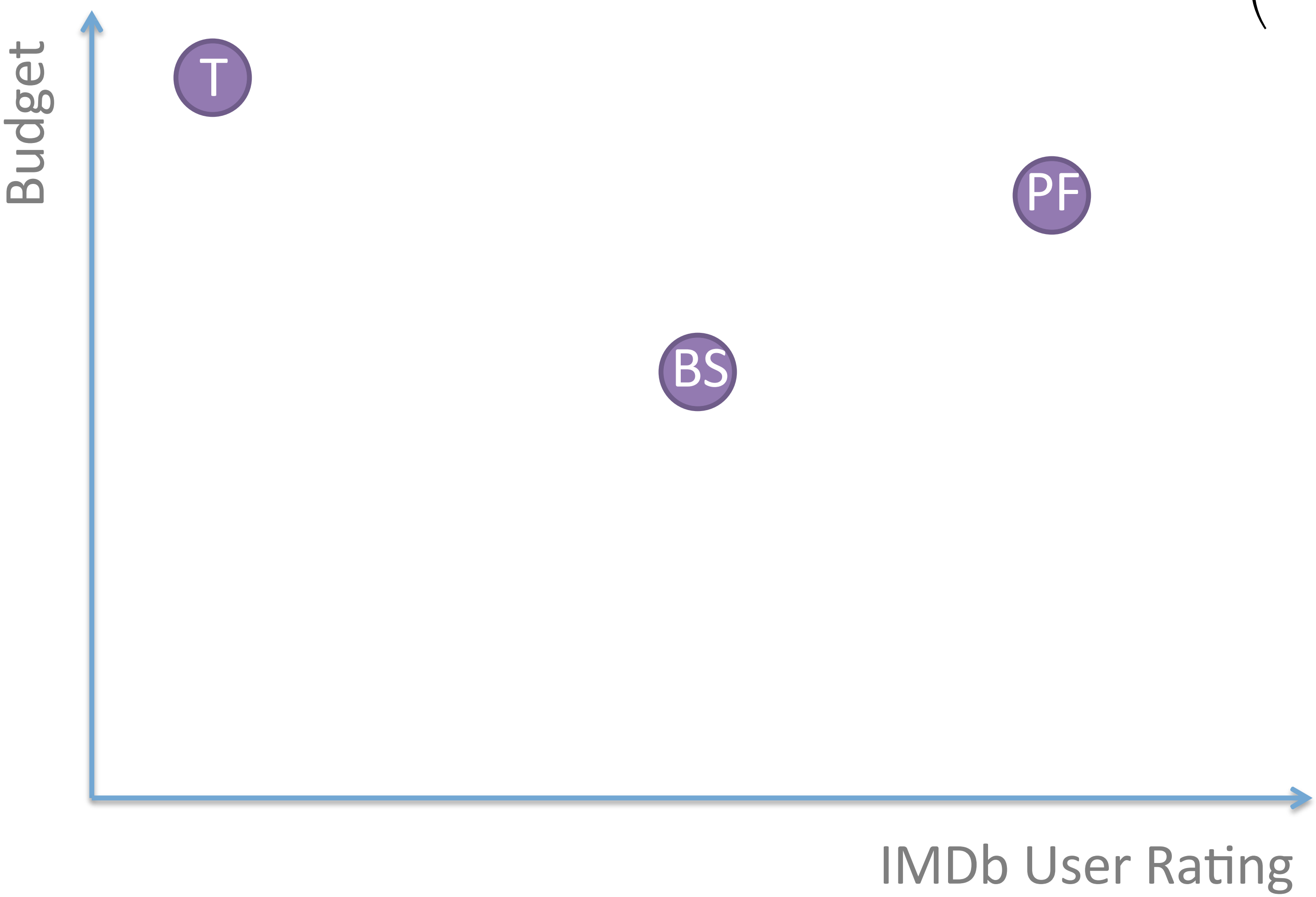
Are you close to Pulp Fiction?



Define feature a_1: "Pulp Fiction"ness

Are you close to Pulp Fiction?

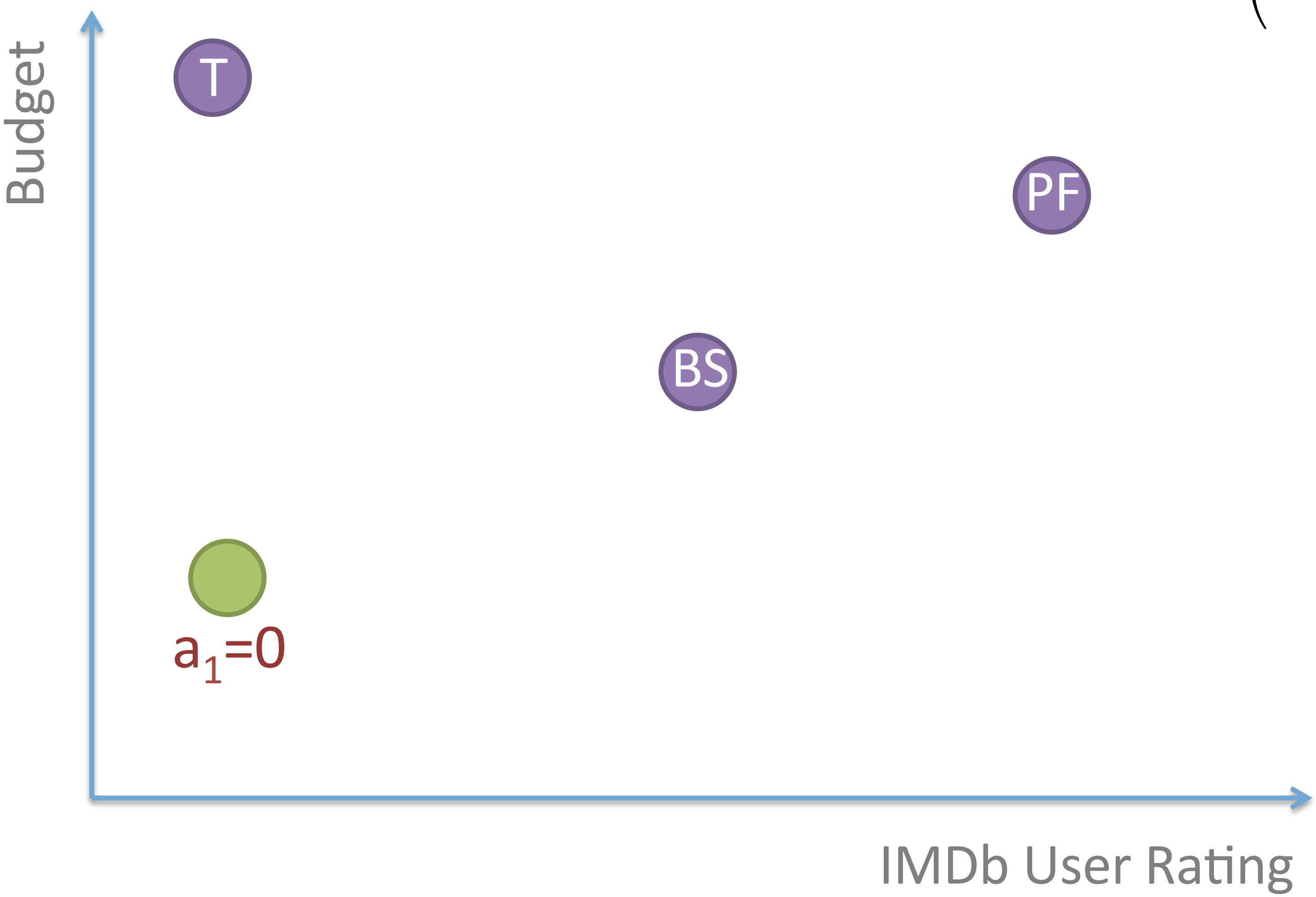
$$a_1(\overrightarrow{x^{obs}}) = \exp\left(\frac{-\sum\left(x_i^{obs} - x_i^{PulpFiction}\right)^2}{2\sigma^2}\right)$$



Define feature a_1: "Pulp Fiction"ness

Are you close to Pulp Fiction?

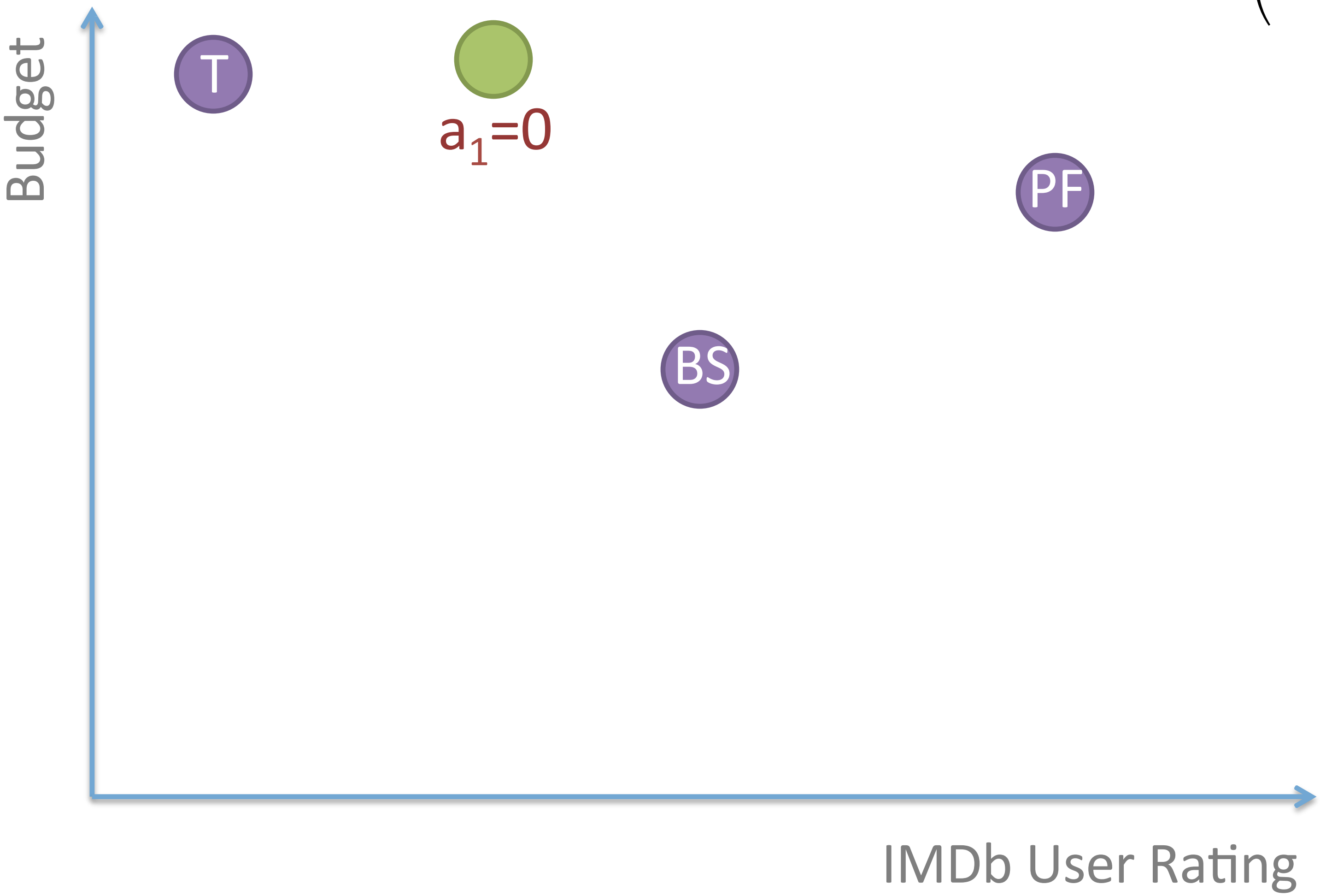
$$a_1(\overrightarrow{x^{obs}}) = \exp\left(\frac{-\sum\left(x_i^{obs} - x_i^{PulpFiction}\right)^2}{2\sigma^2}\right)$$



Define feature a_1: "Pulp Fiction"ness

Are you close to Pulp Fiction?

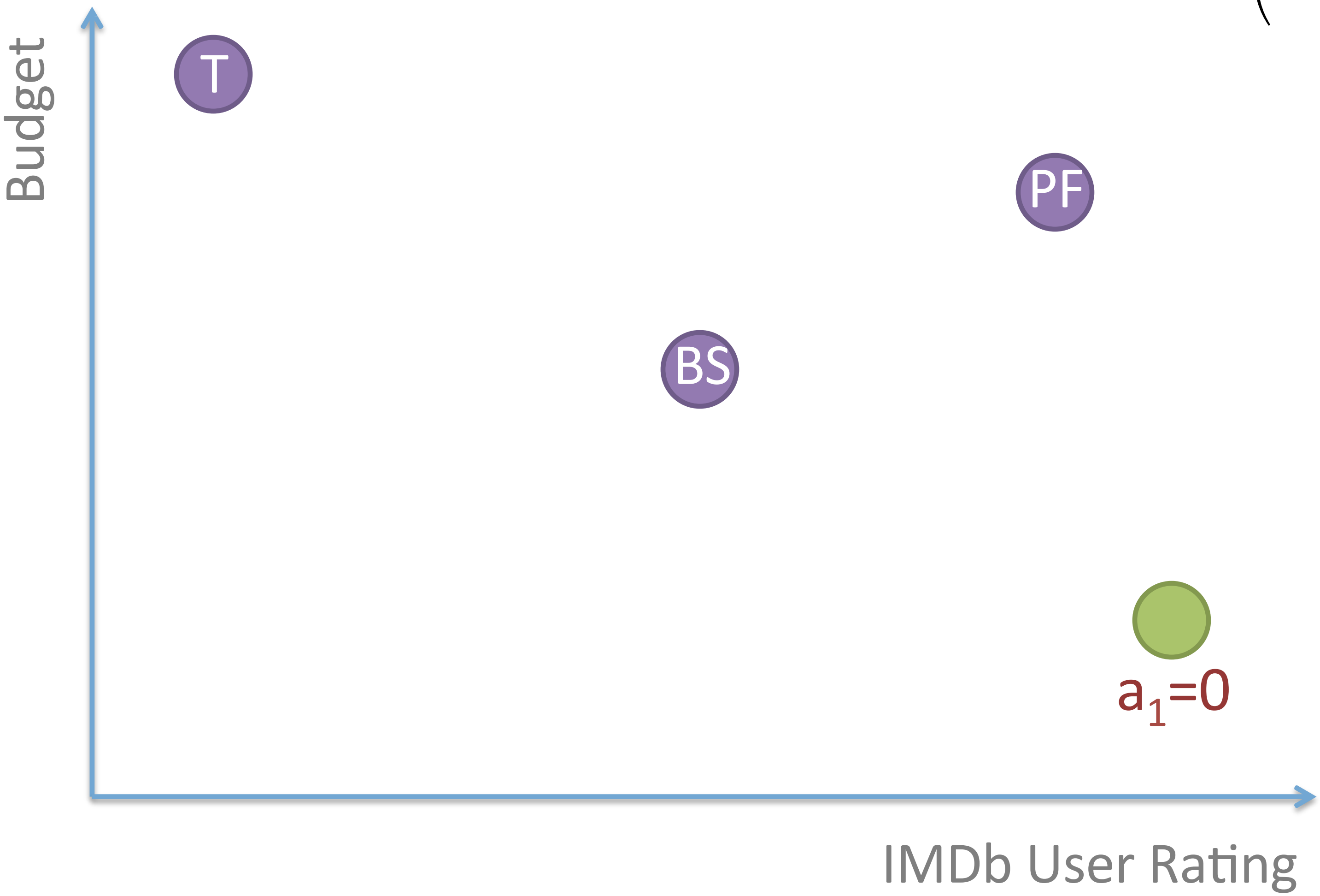
$$a_1(\overrightarrow{x^{obs}}) = \exp\left(\frac{-\sum\left(x_i^{obs} - x_i^{PulpFiction}\right)^2}{2\sigma^2}\right)$$



Define feature a_1: "Pulp Fiction"ness

Are you close to Pulp Fiction?

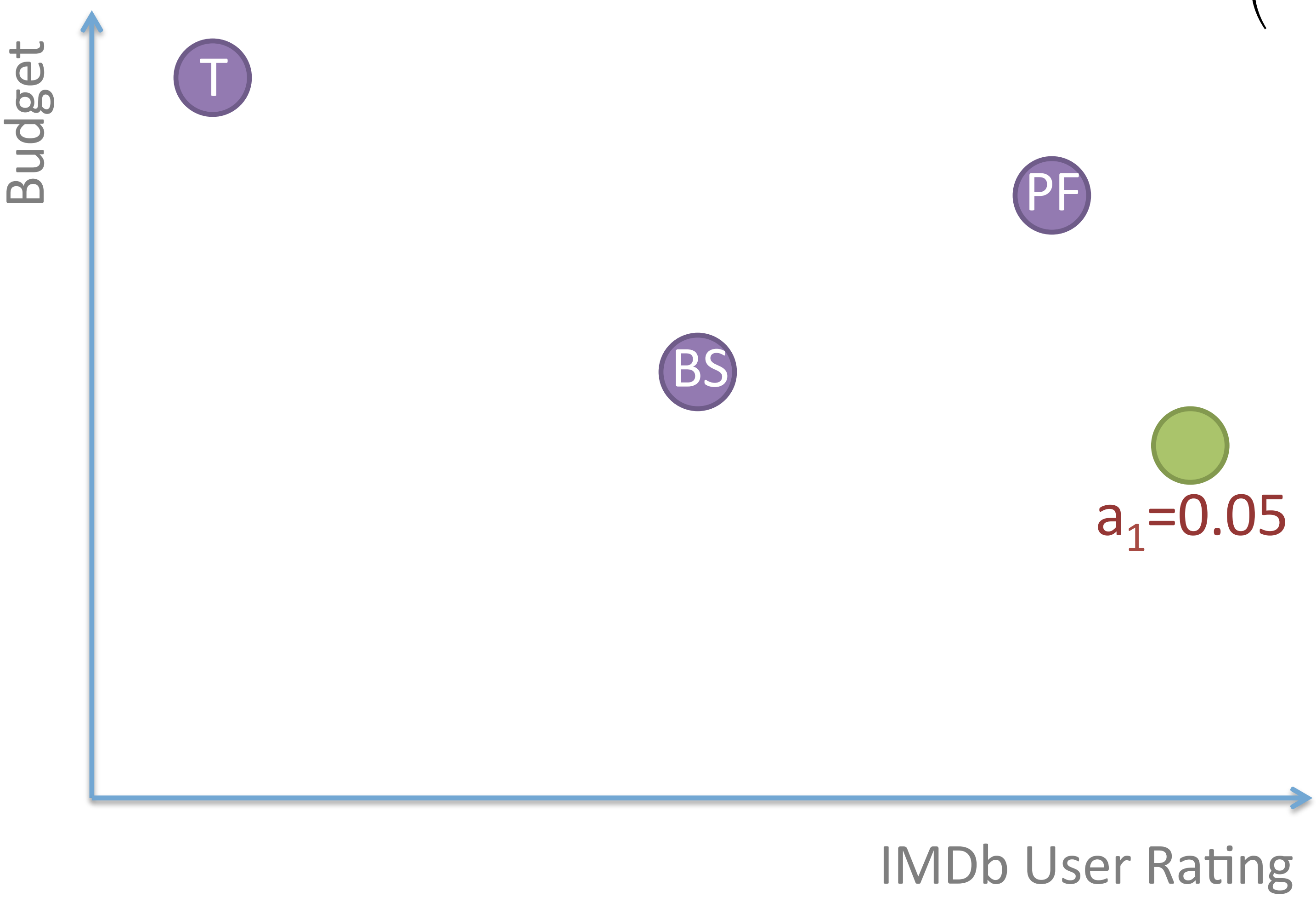
$$a_1(\overrightarrow{x^{obs}}) = \exp\left(\frac{-\sum\left(x_i^{obs} - x_i^{PulpFiction}\right)^2}{2\sigma^2}\right)$$



Define feature a_1: "Pulp Fiction"ness

Are you close to Pulp Fiction?

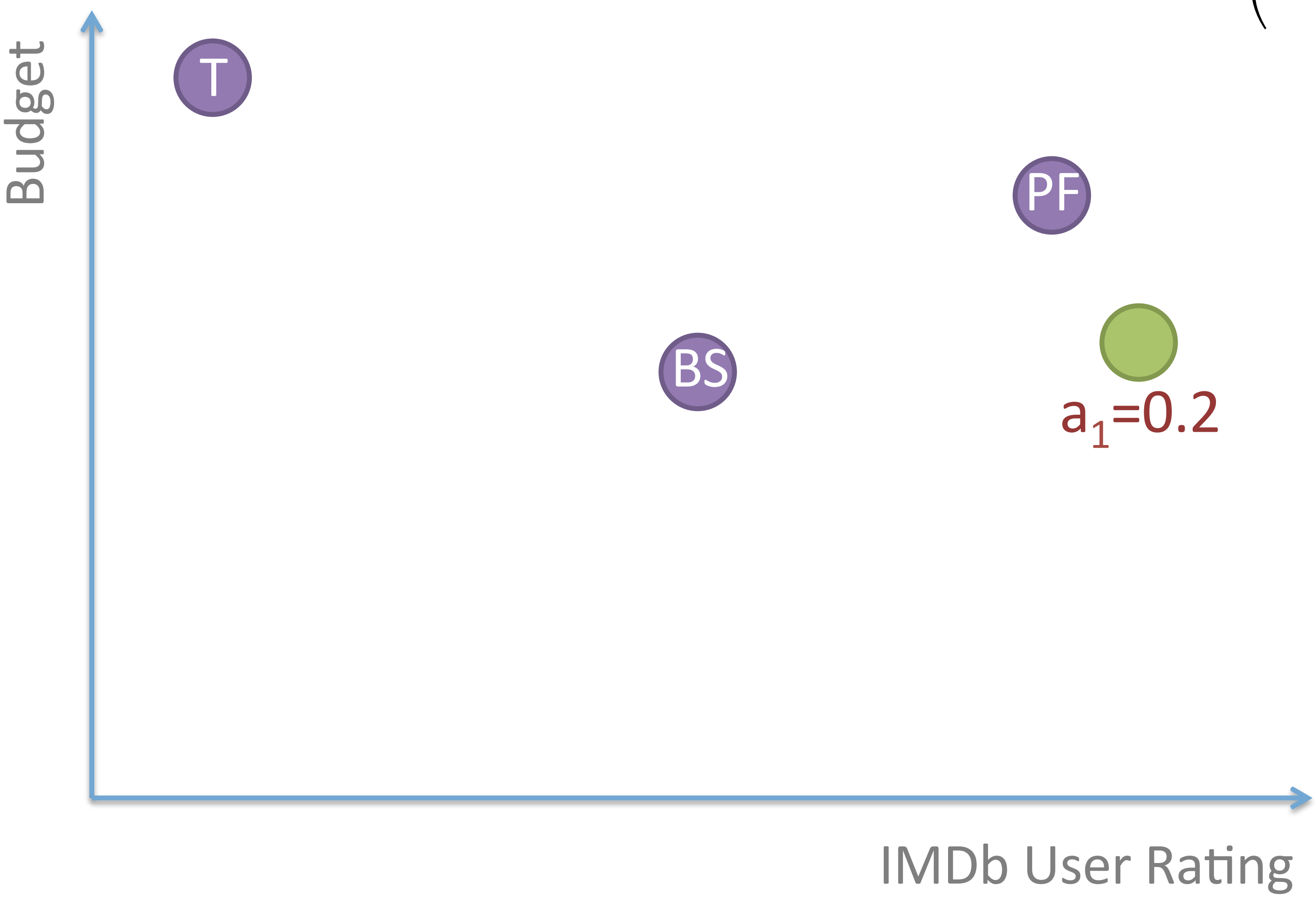
$$a_1(\overrightarrow{x^{obs}}) = \exp\left(\frac{-\sum\left(x_i^{obs} - x_i^{PulpFiction}\right)^2}{2\sigma^2}\right)$$



Define feature a_1: "Pulp Fiction"ness

Are you close to Pulp Fiction?

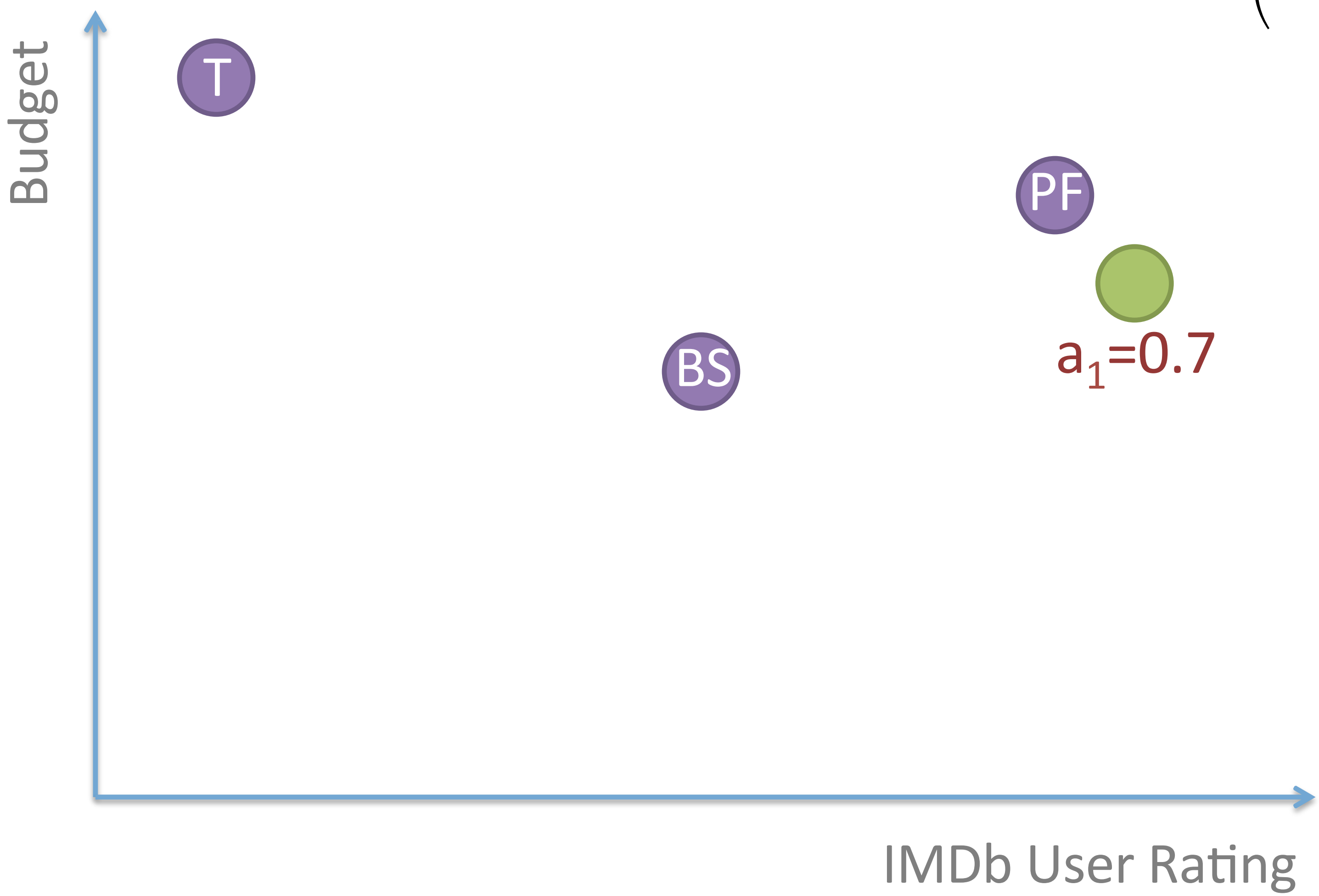
$$a_1(\overrightarrow{x^{obs}}) = \exp\left(\frac{-\sum\left(x_i^{obs} - x_i^{PulpFiction}\right)^2}{2\sigma^2}\right)$$



Define feature a_1: "Pulp Fiction"ness

Are you close to Pulp Fiction?

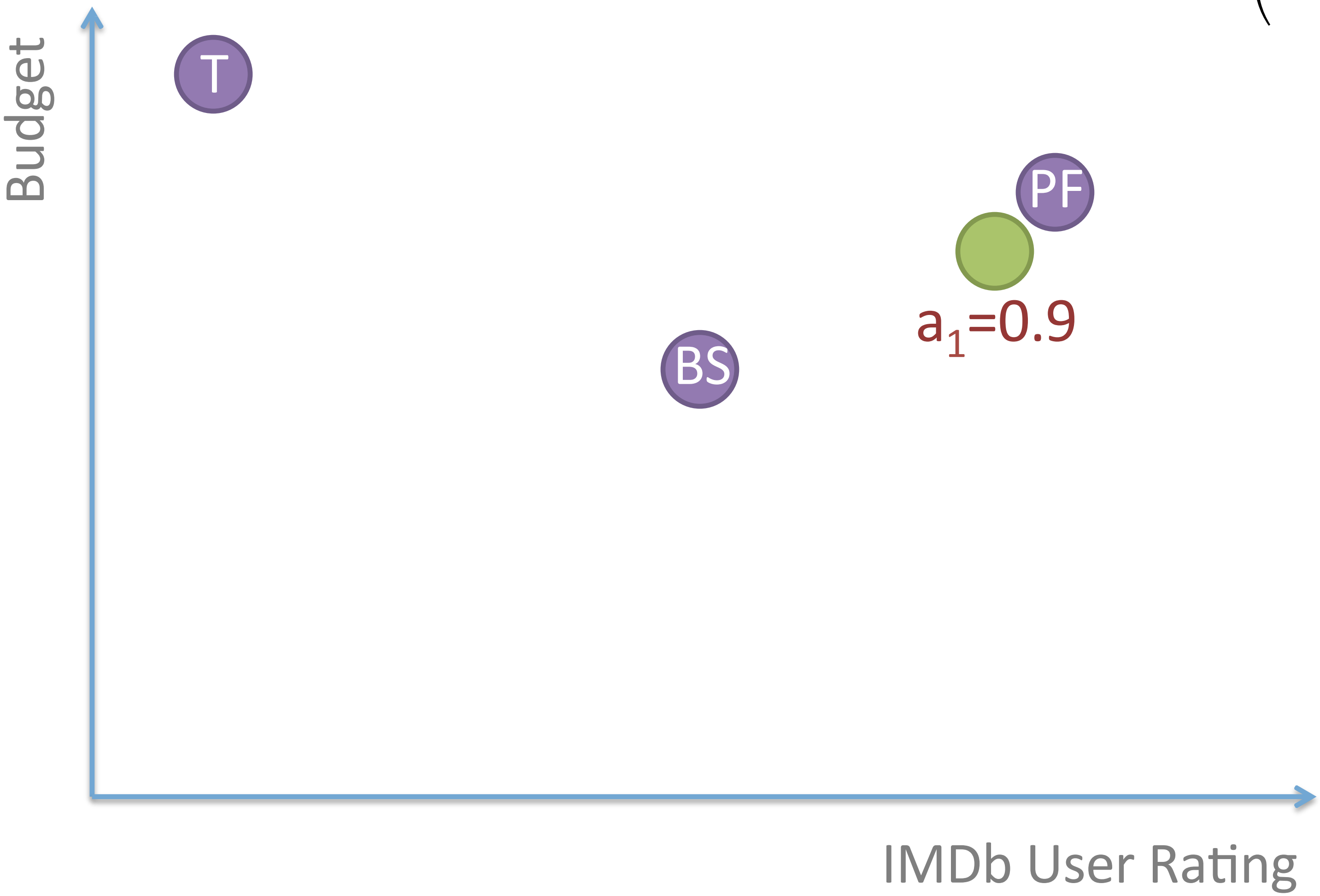
$$a_1(\overrightarrow{x^{obs}}) = \exp\left(\frac{-\sum\left(x_i^{obs} - x_i^{PulpFiction}\right)^2}{2\sigma^2}\right)$$



Define feature a_1: "Pulp Fiction"ness

Are you close to Pulp Fiction?

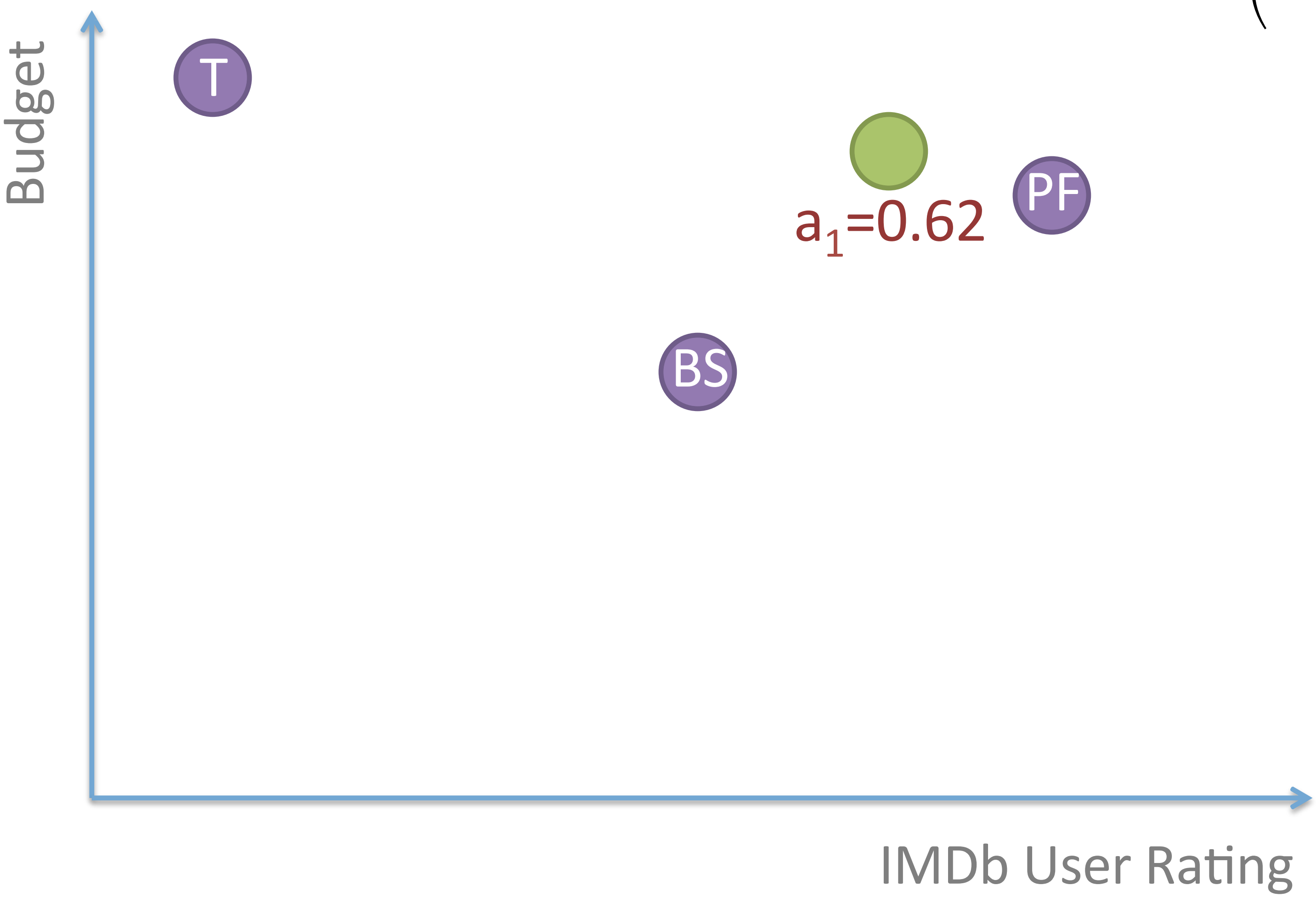
$$a_1(\overrightarrow{x^{obs}}) = \exp\left(\frac{-\sum\left(x_i^{obs} - x_i^{PulpFiction}\right)^2}{2\sigma^2}\right)$$



Define feature a_1: "Pulp Fiction"ness

Are you close to Pulp Fiction?

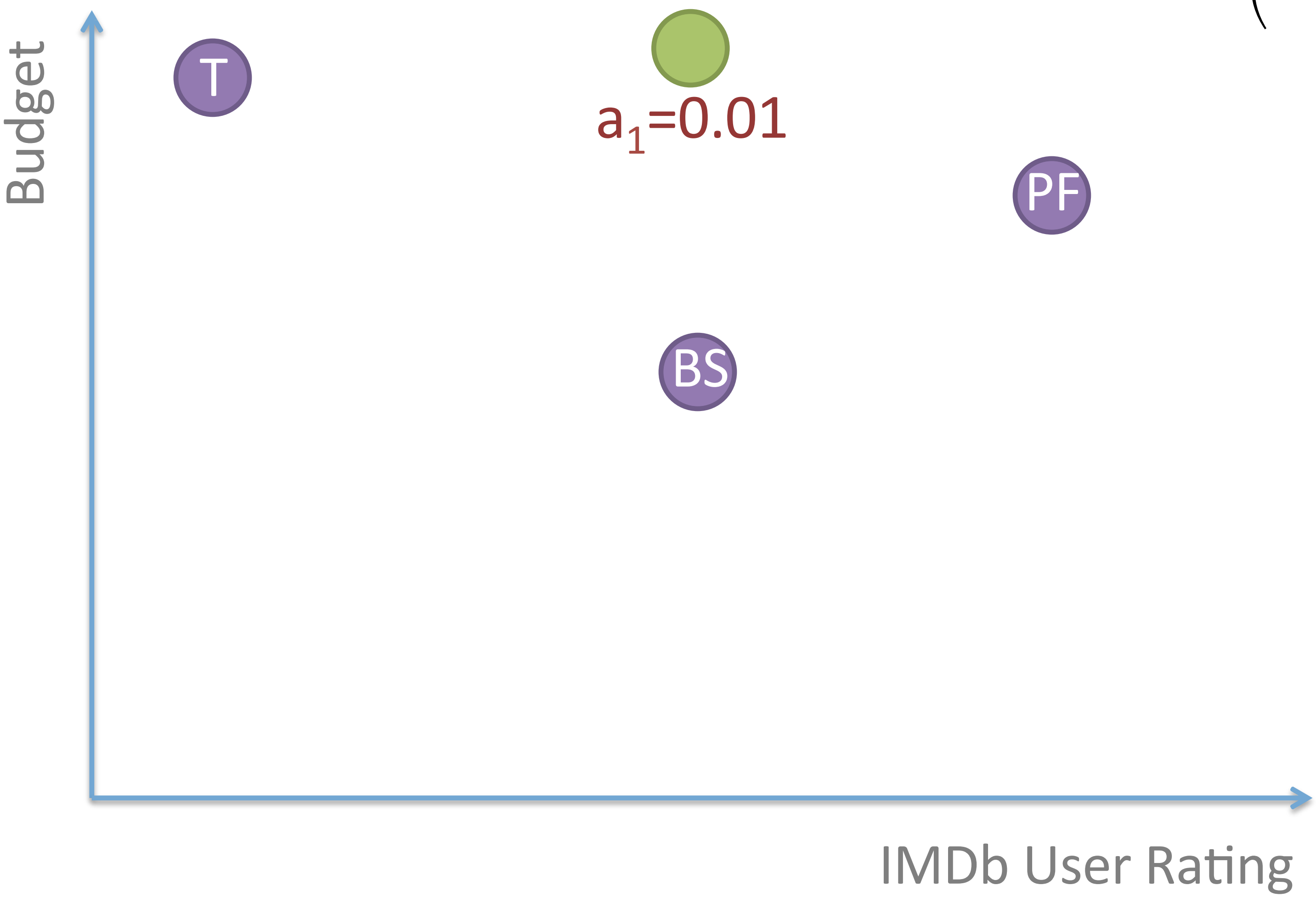
$$a_1(\overrightarrow{x^{obs}}) = \exp\left(\frac{-\sum\left(x_i^{obs} - x_i^{PulpFiction}\right)^2}{2\sigma^2}\right)$$



Define feature a_1: "Pulp Fiction"ness

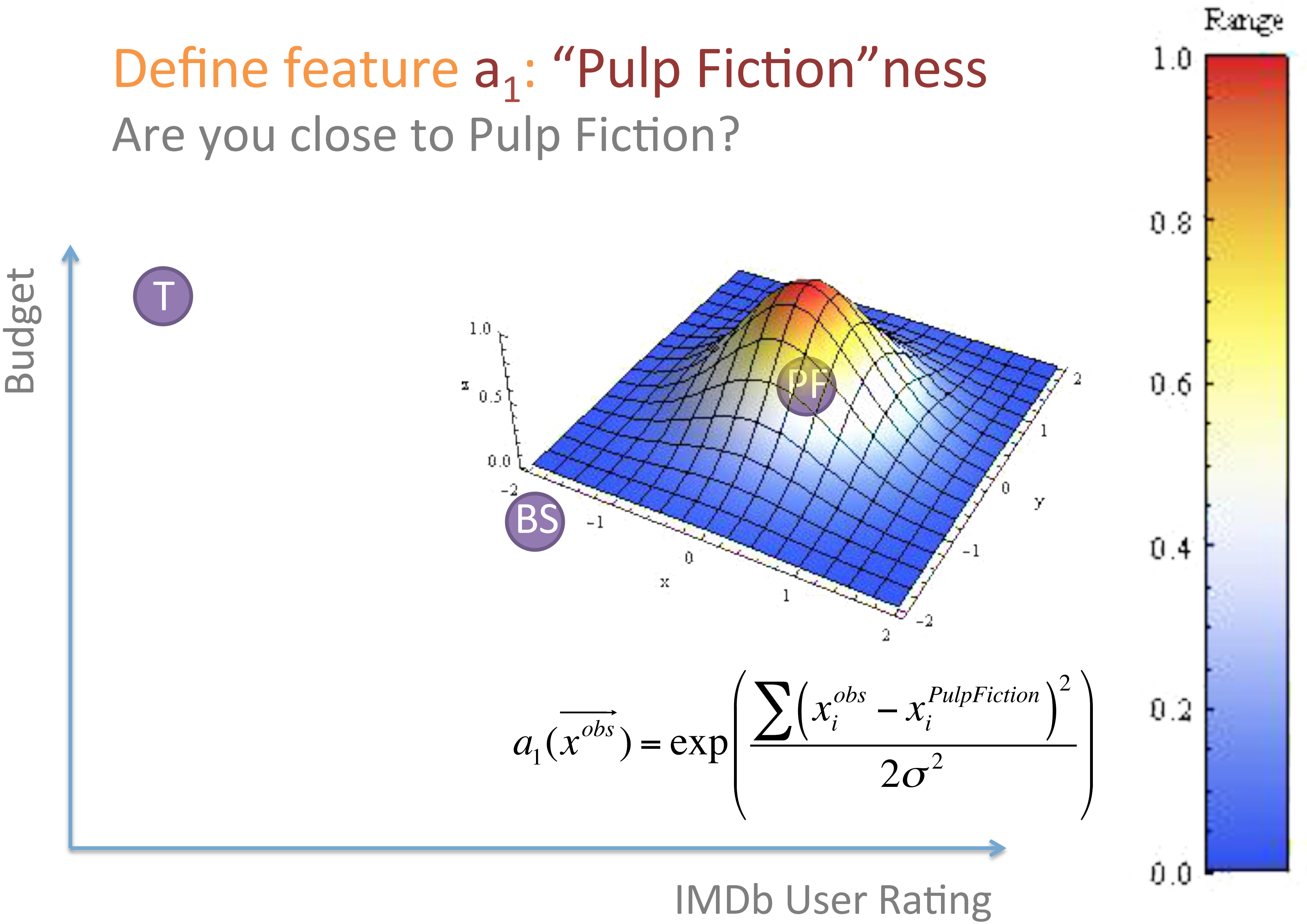
Are you close to Pulp Fiction?

$$a_1(\overrightarrow{x^{obs}}) = \exp\left(\frac{-\sum\left(x_i^{obs} - x_i^{PulpFiction}\right)^2}{2\sigma^2}\right)$$



Define feature a_1: "Pulp Fiction"ness

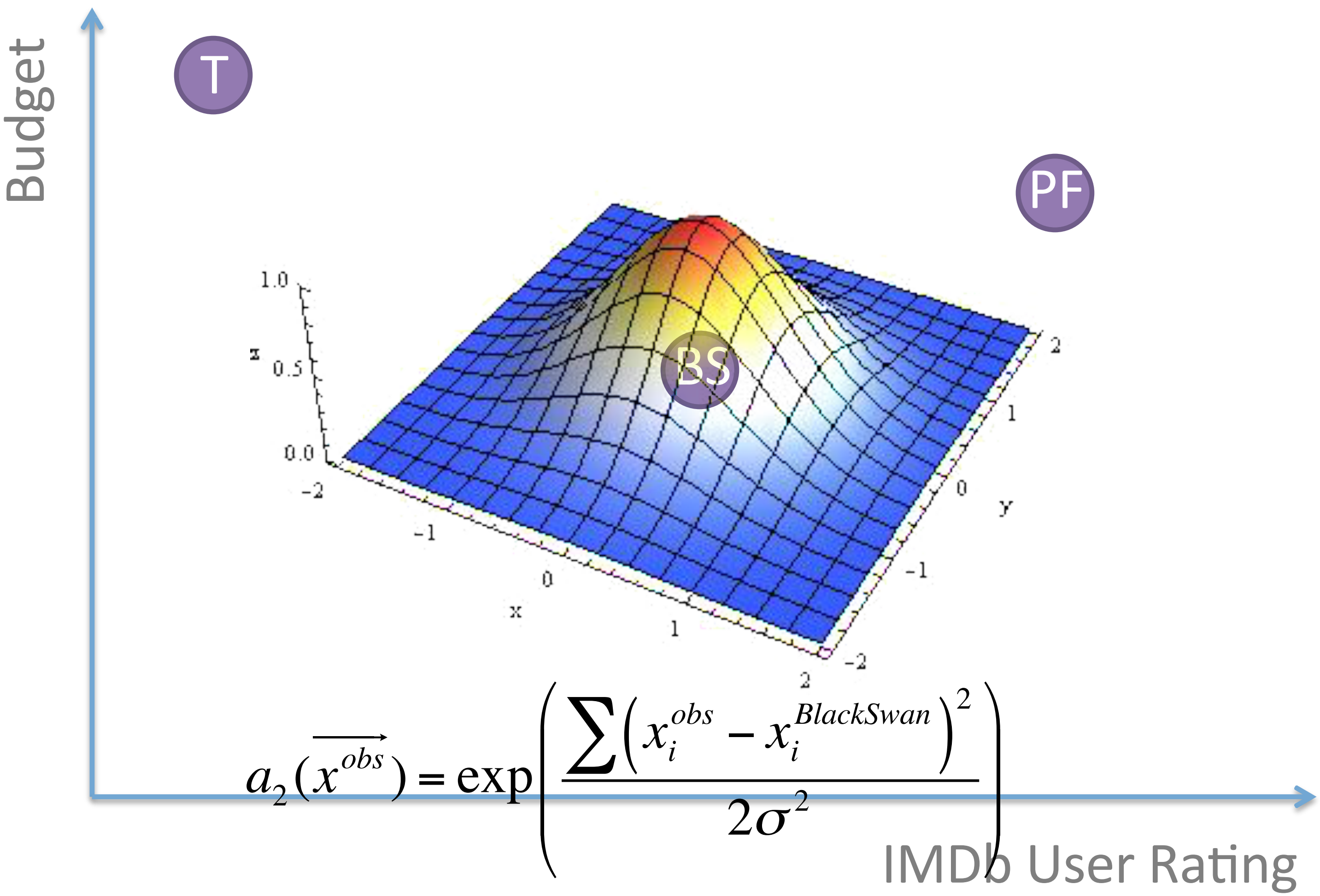
Are you close to Pulp Fiction?



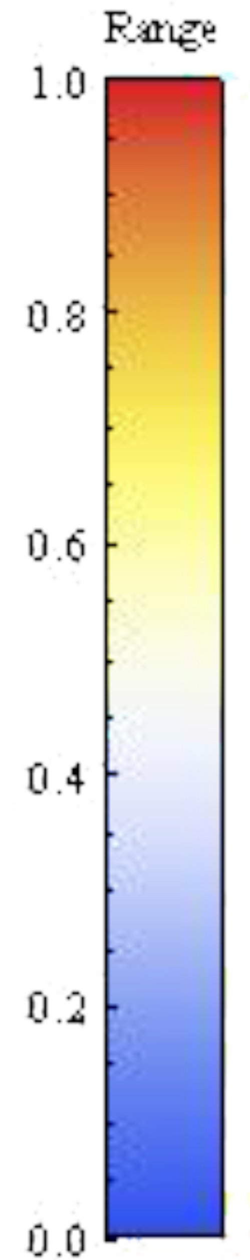
$$a_1(\overrightarrow{x^{obs}}) = \exp\left(\frac{\sum\left(x_i^{obs} - x_i^{PulpFiction}\right)^2}{2\sigma^2}\right)$$

Define feature a_2: "Black Swan"ness

Are you close to Black Swan?

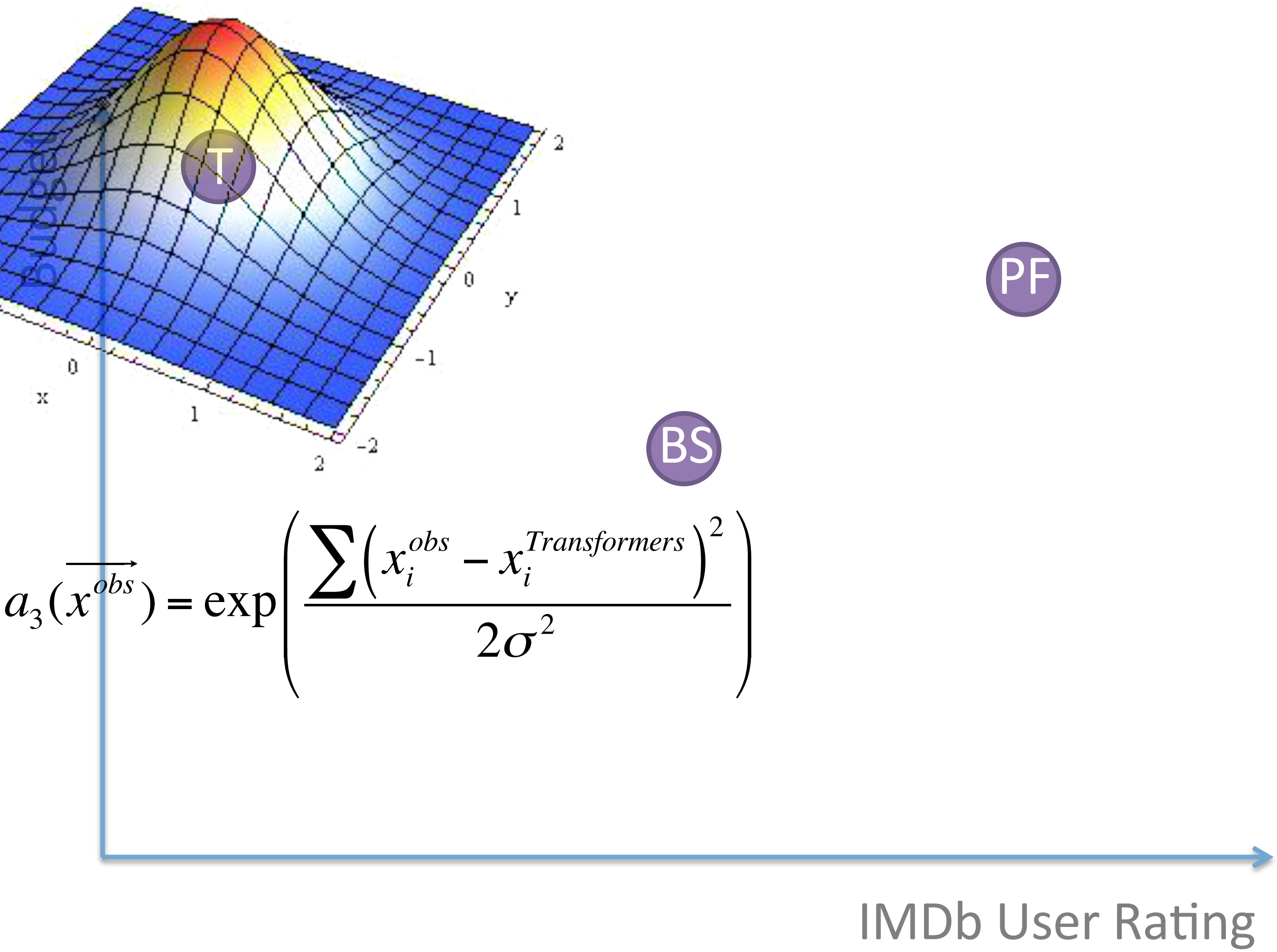


$$a_2(\overrightarrow{x^{obs}}) = \exp\left(\frac{\sum \left(x_i^{obs} - x_i^{BlackSwan} \right)^2}{2\sigma^2} \right)$$

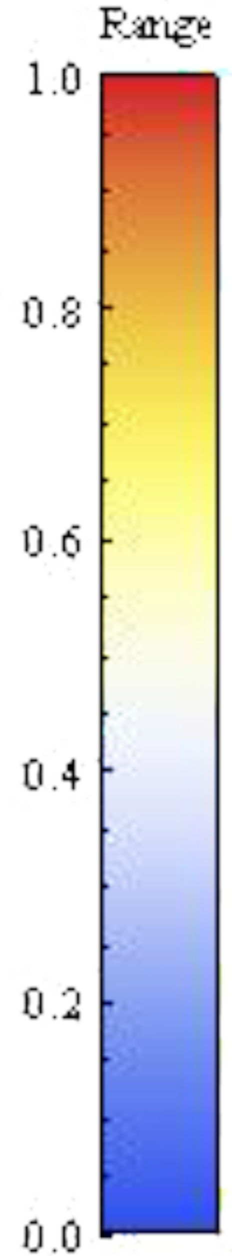


Define feature a_3: "Transformers"ness

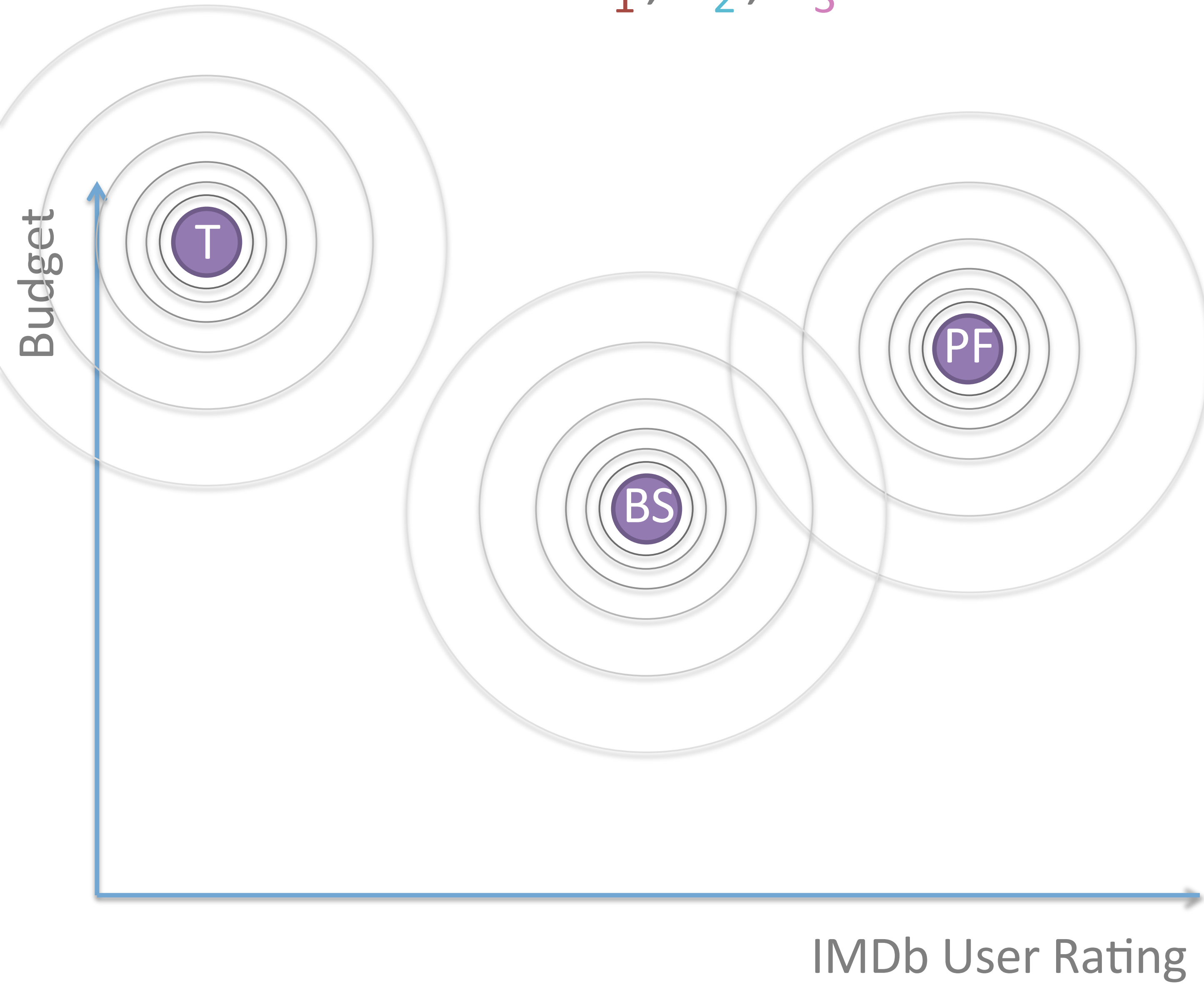
Are you close to Black Swan?



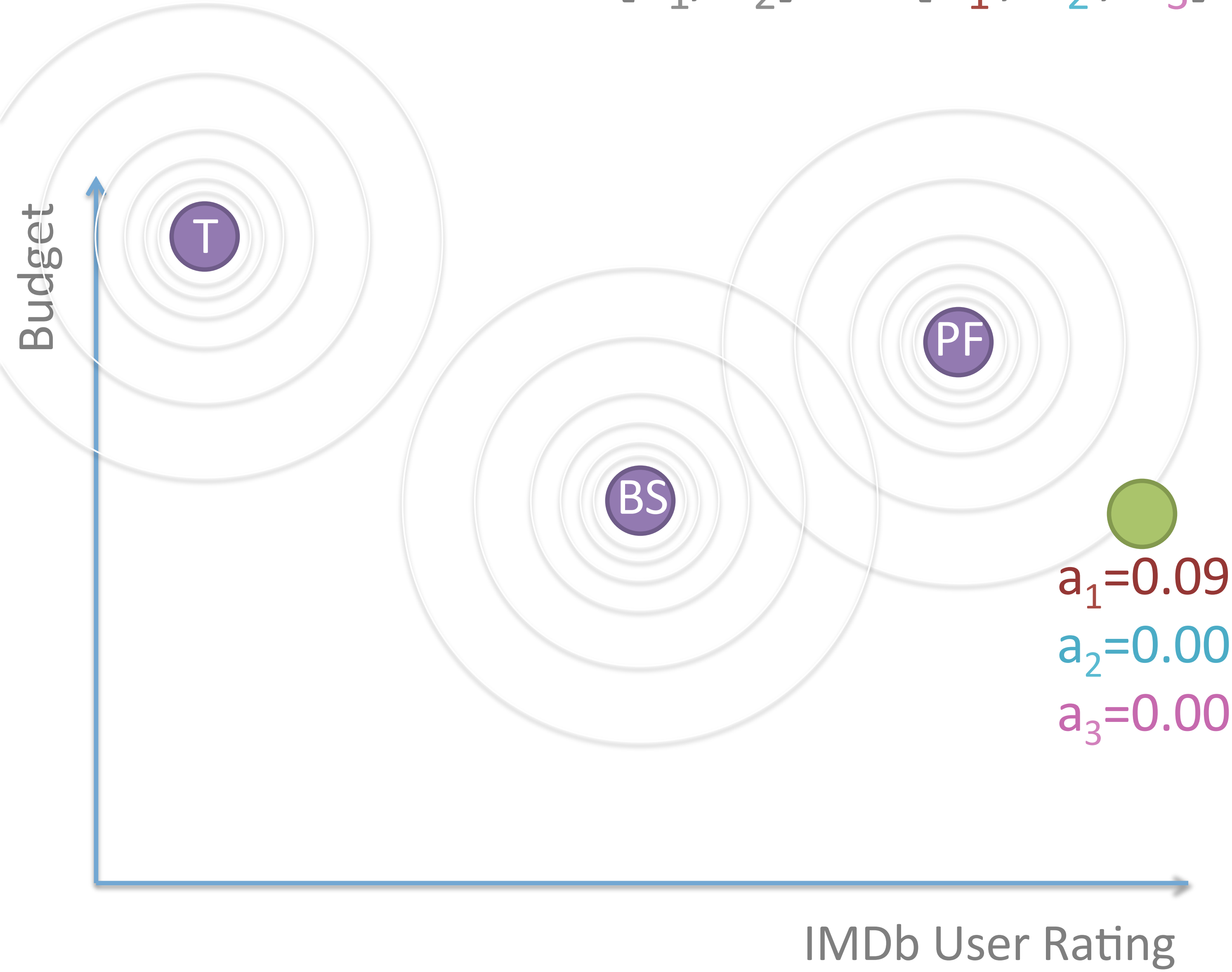
$$a_3(\overrightarrow{x^{obs}}) = \exp\left(\frac{\sum \left(x_i^{obs} - x_i^{Transformers} \right)^2}{2\sigma^2} \right)$$



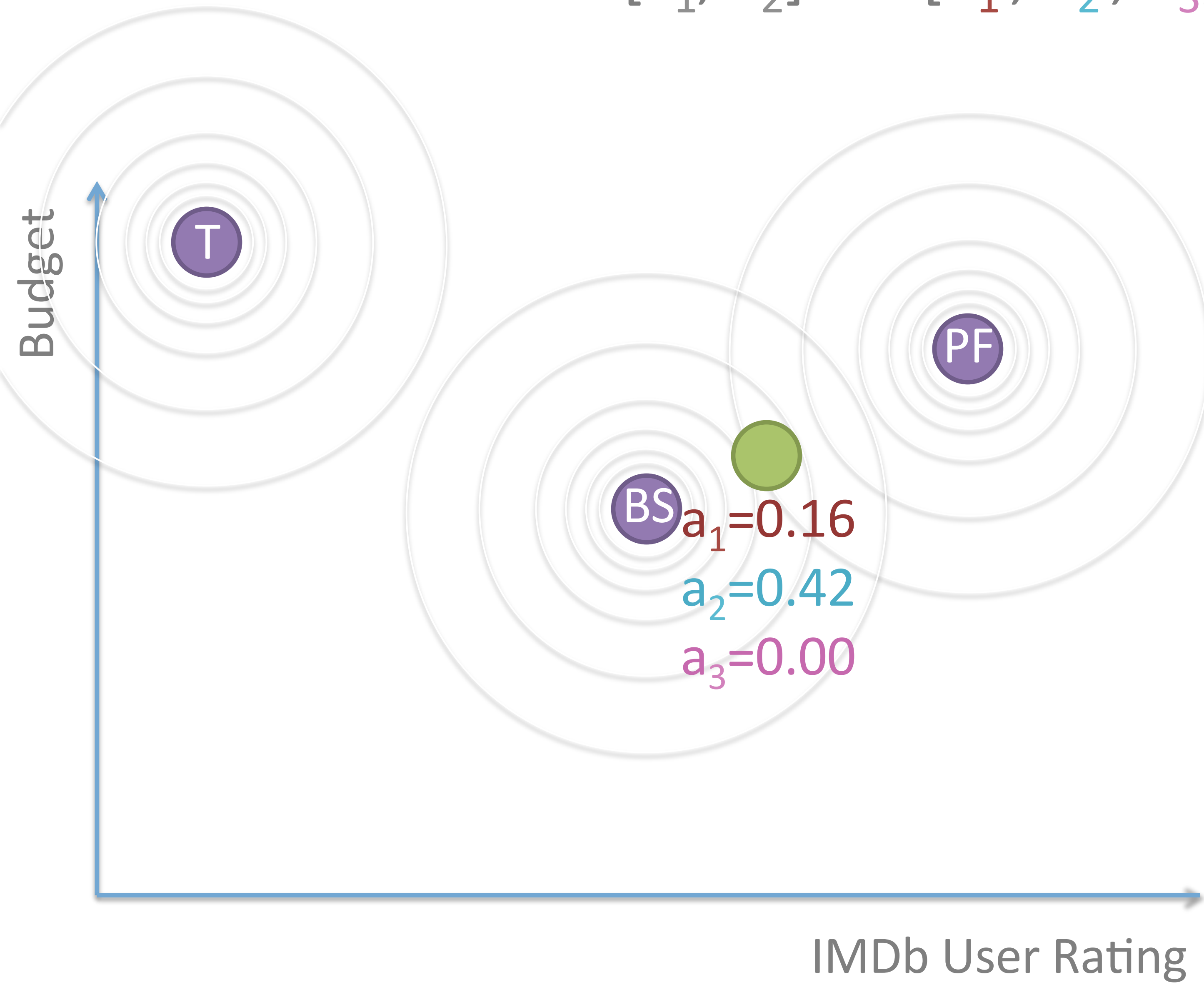
New features: a_1, a_2, a_3



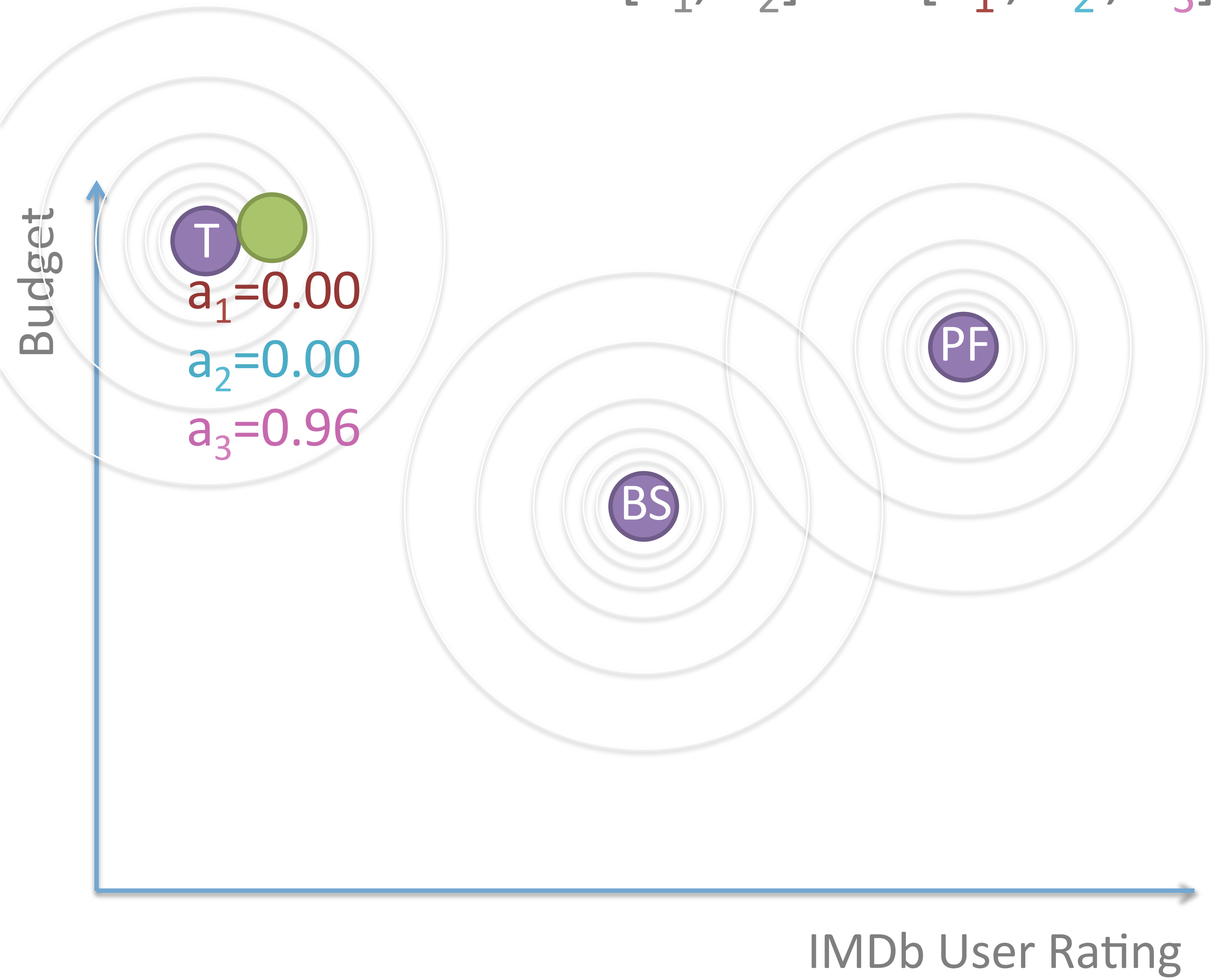
Transformation: $[x_1, x_2] \rightarrow [a_1, a_2, a_3]$



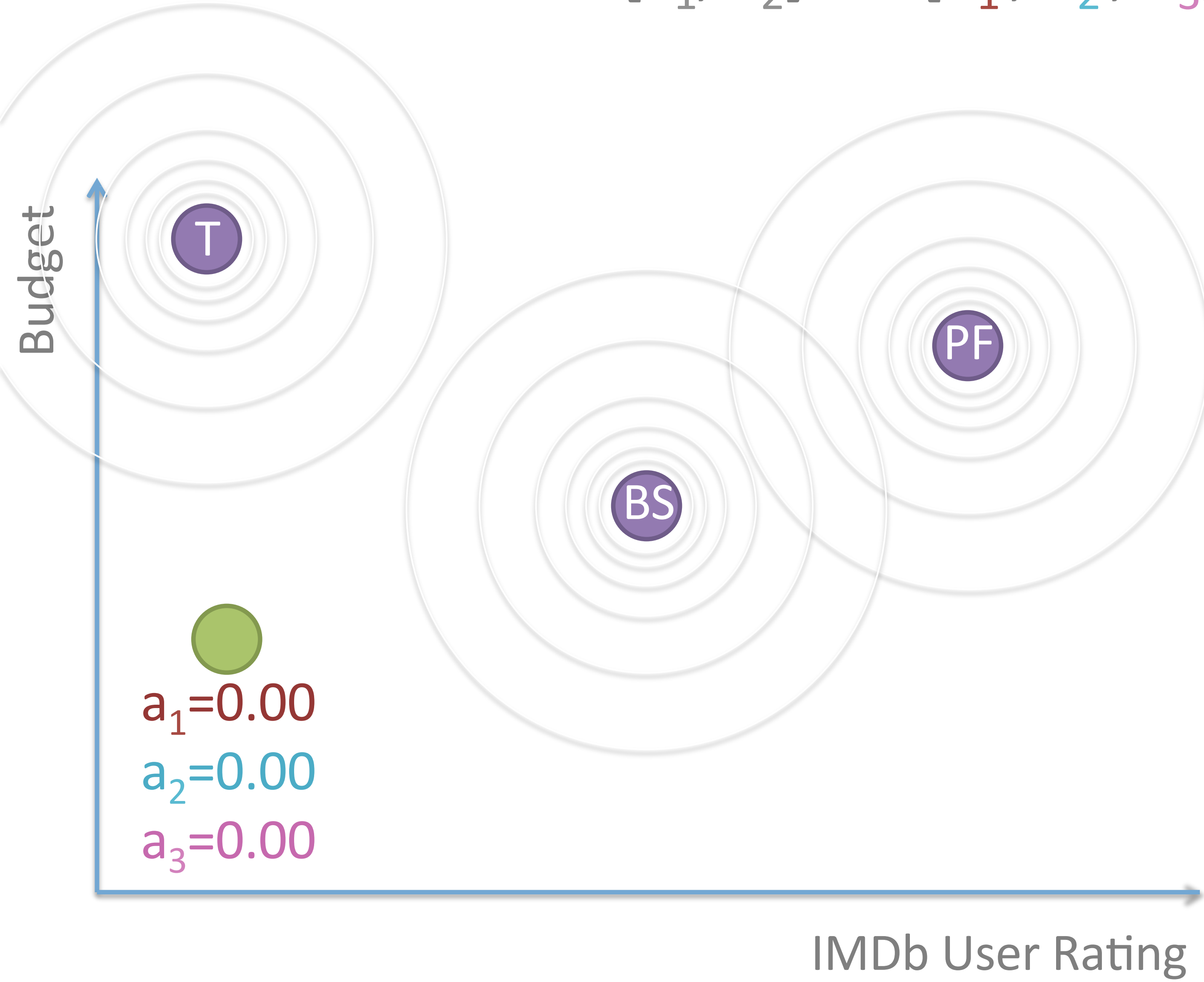
Transformation: $[x_1, x_2] \rightarrow [a_1, a_2, a_3]$



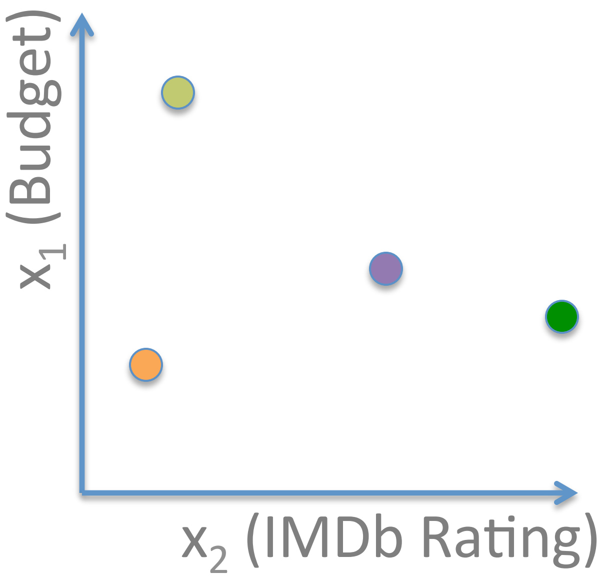
Transformation: $[x_1, x_2] \rightarrow [a_1, a_2, a_3]$



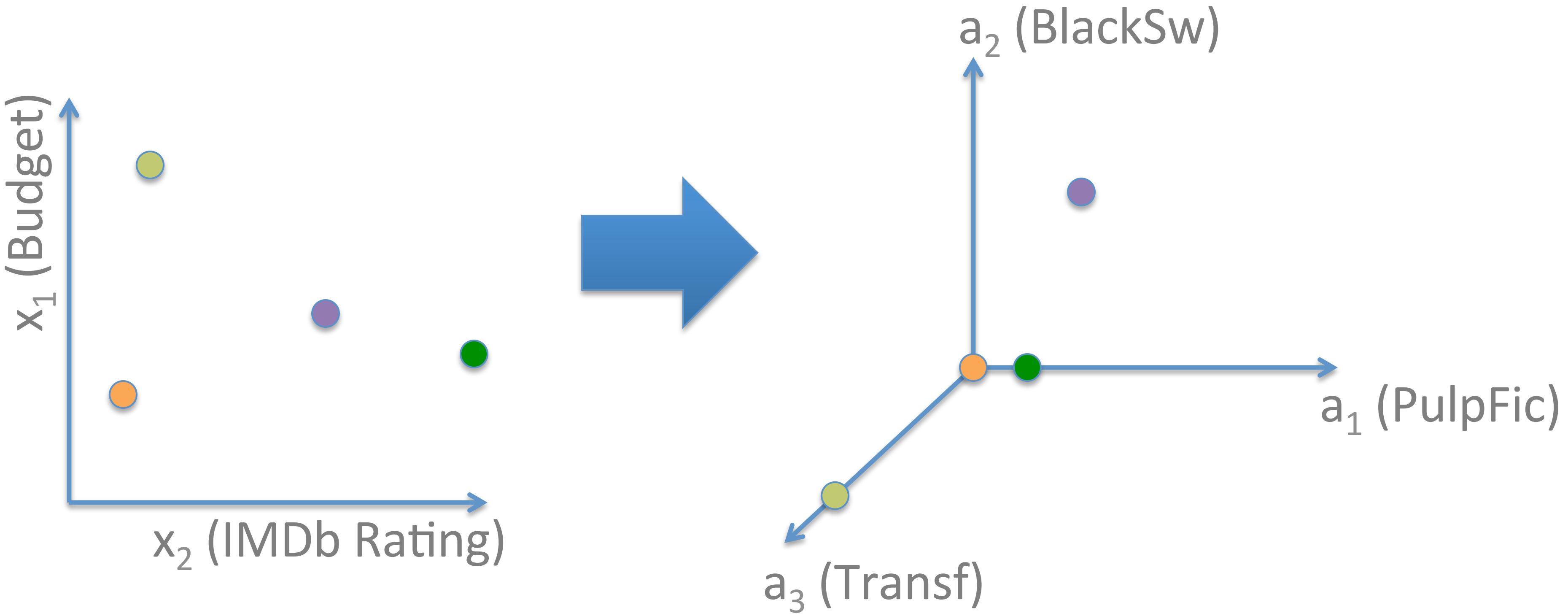
Transformation: $[x_1, x_2] \rightarrow [a_1, a_2, a_3]$



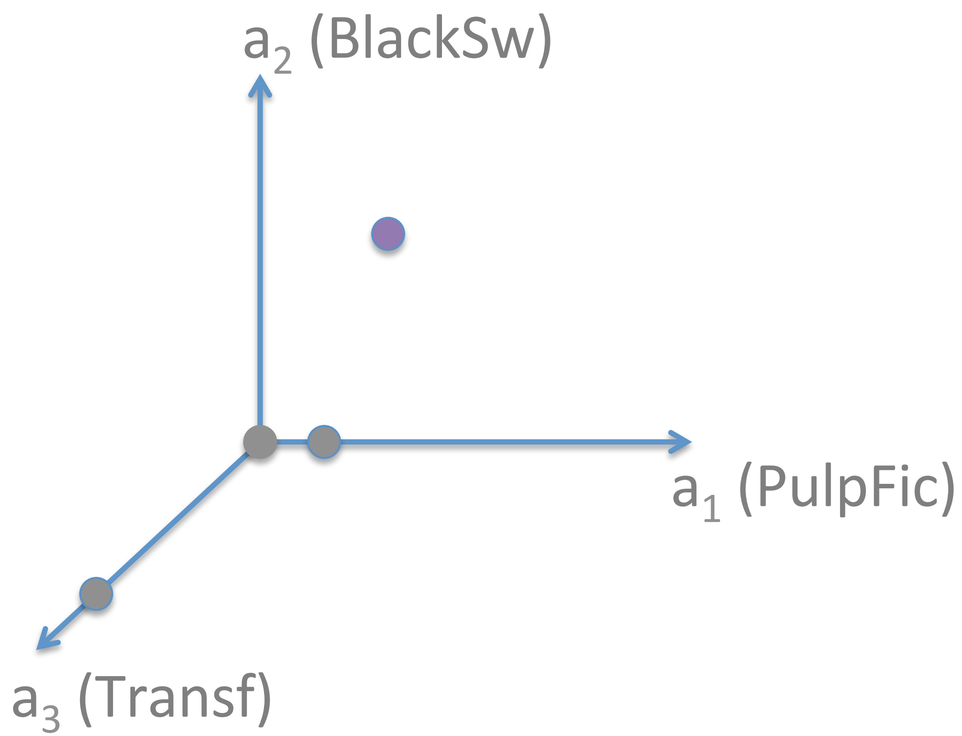
Transformation: $[x_1, x_2] \rightarrow [a_1, a_2, a_3]$



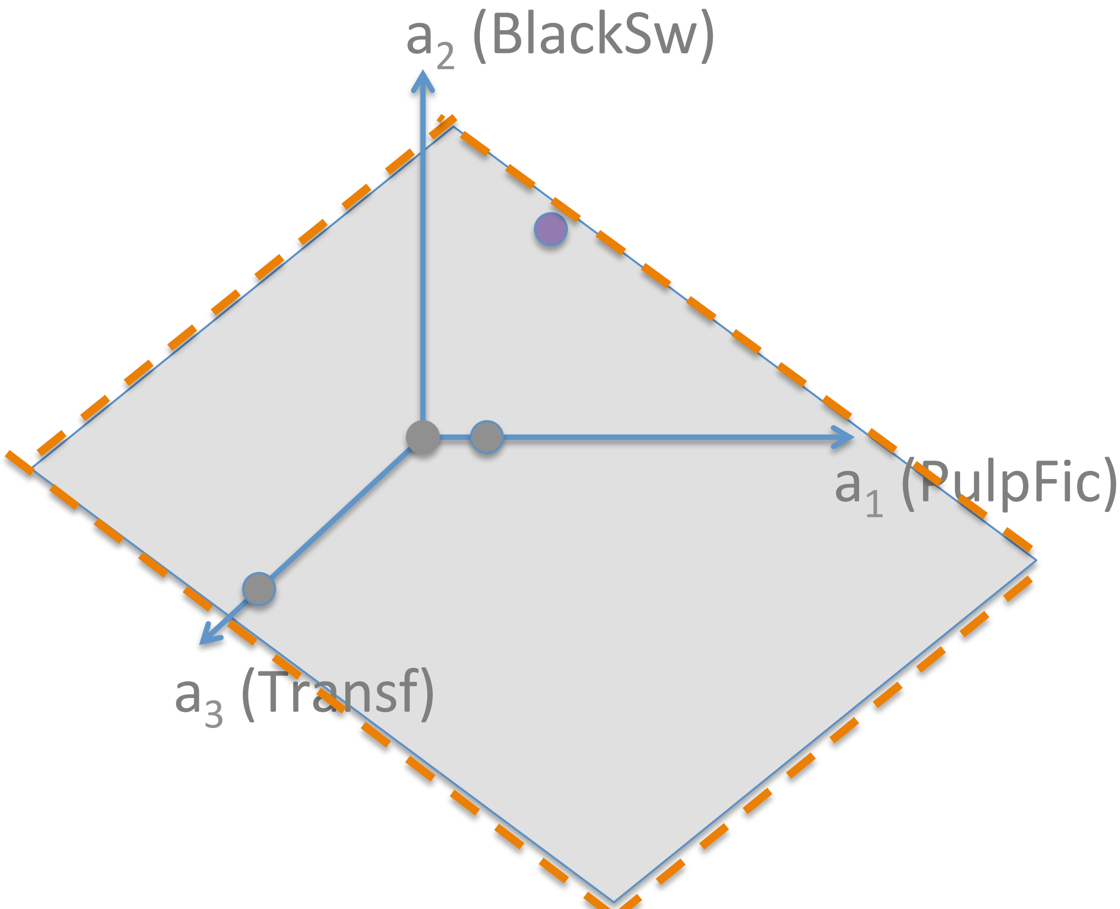
Transformation: $[x_1, x_2] \rightarrow [a_1, a_2, a_3]$



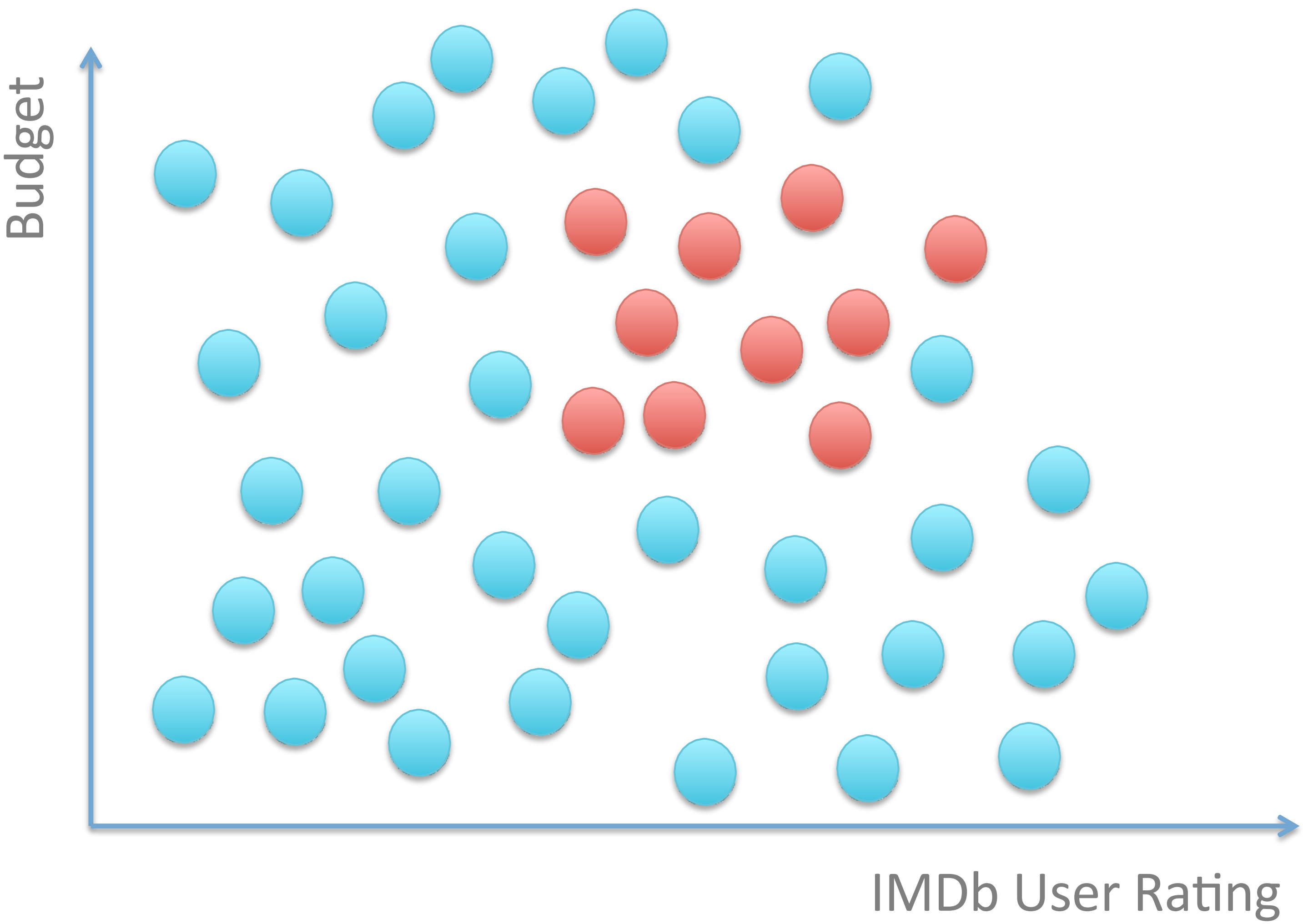
Classification in the new space



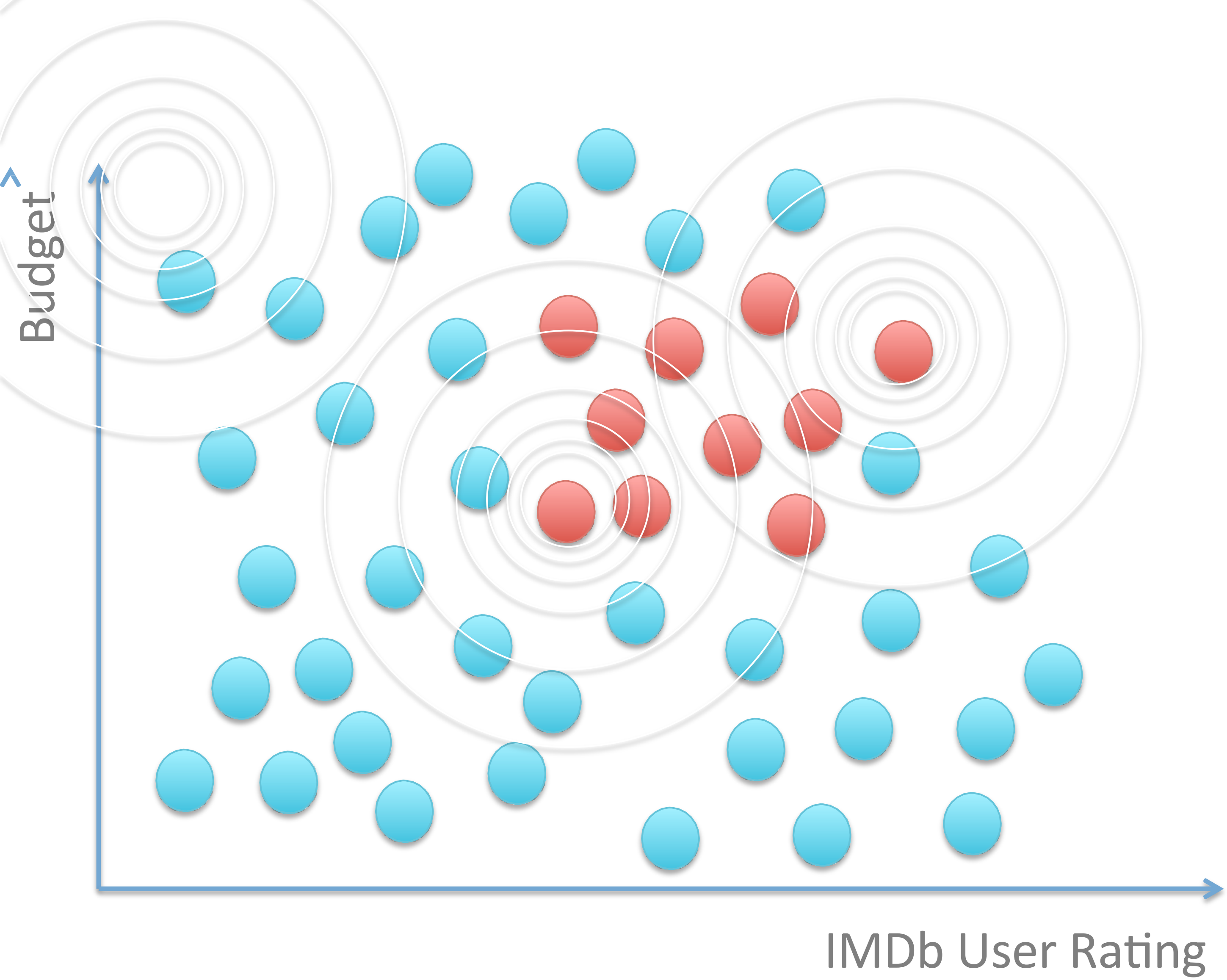
Classification in the new space



Palme d’Or Winners at Cannes

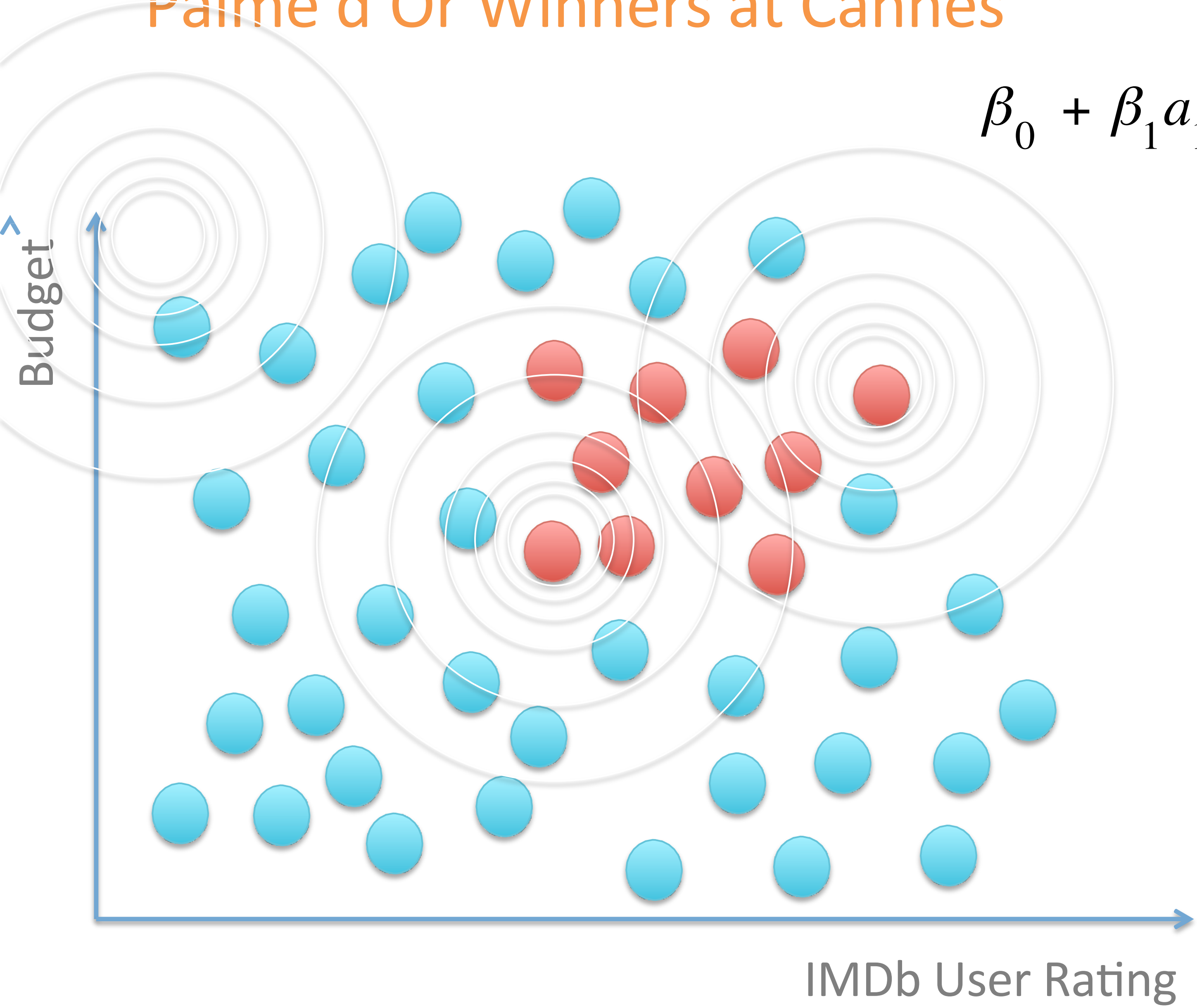


Palme d'Or Winners at Cannes



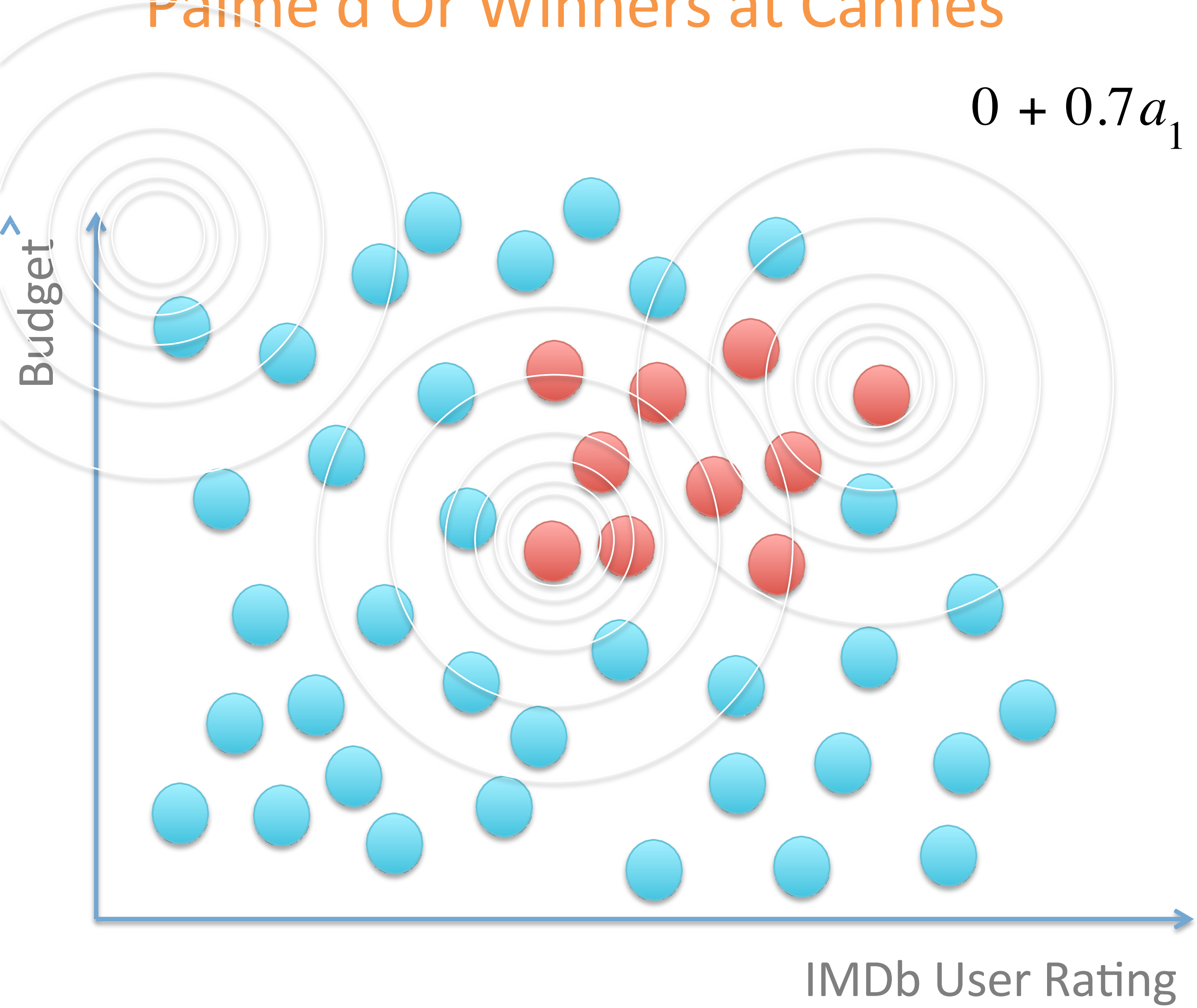
Palme d'Or Winners at Cannes

$$\beta_0 + \beta_1 a_1 + \beta_2 a_2 + \beta_3 a_3$$

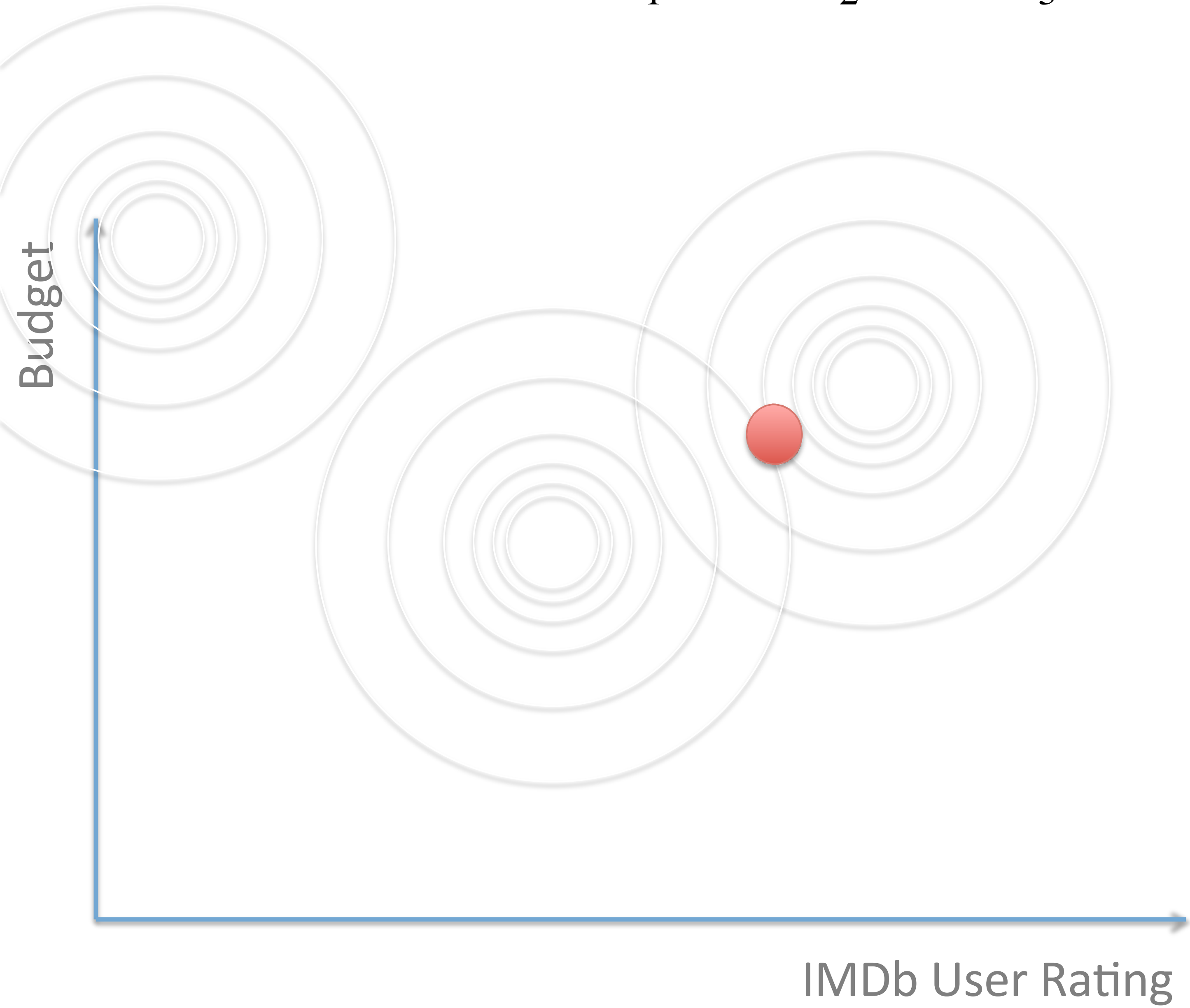


Palme d'Or Winners at Cannes

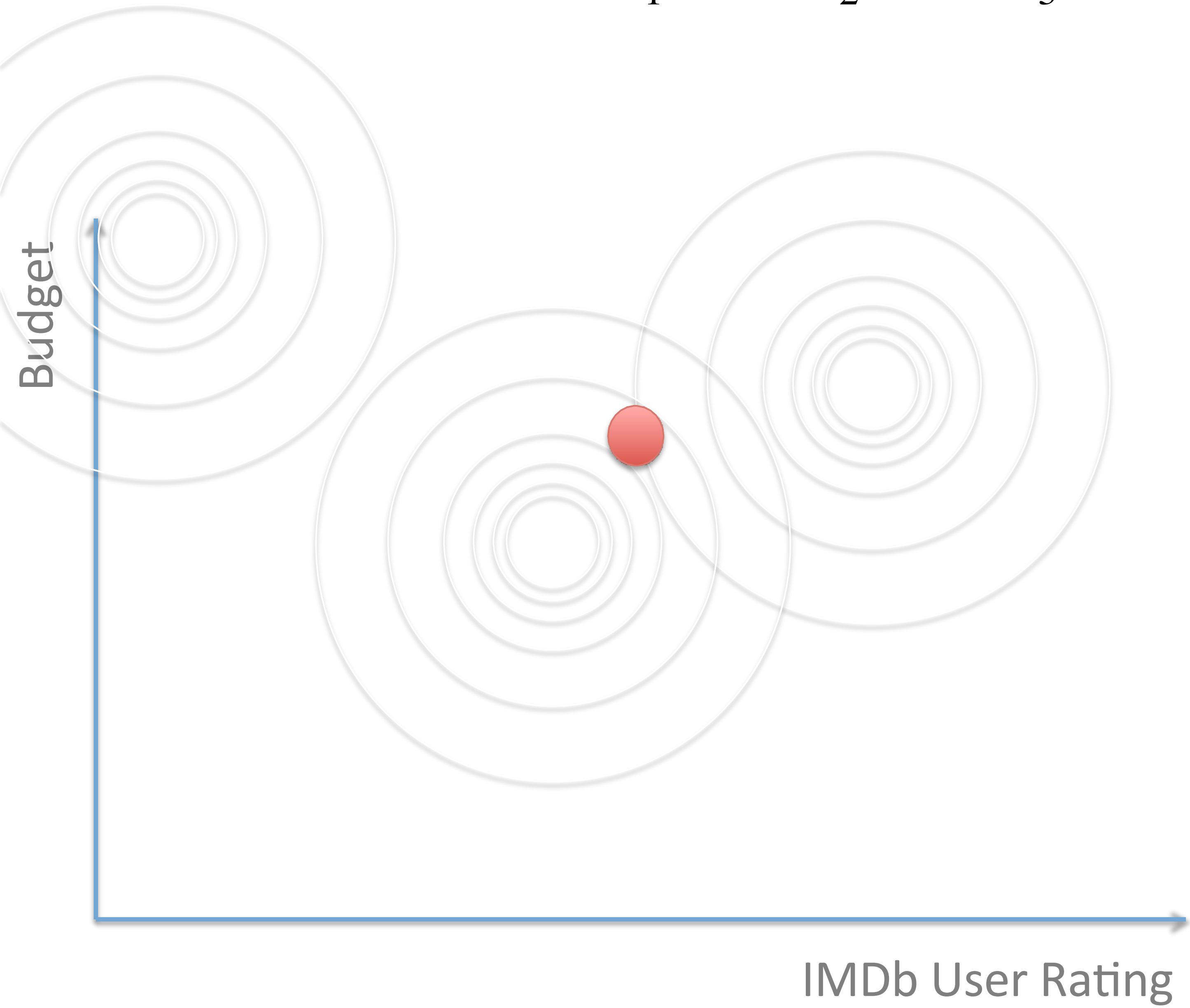
$$0 + 0.7a_1 + 0.9a_2 - 0.6a_3$$



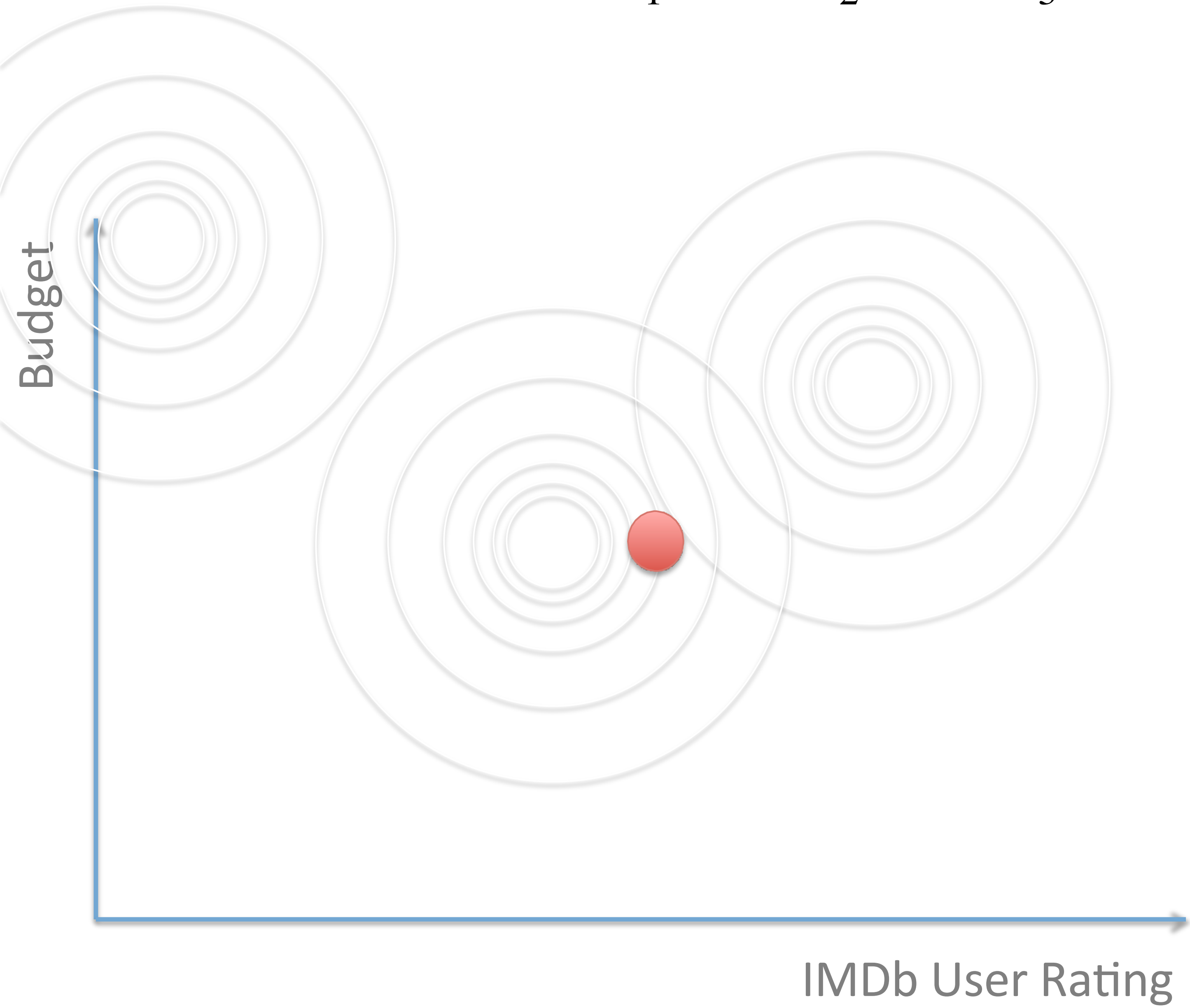
$$0 + 0.7a_1 + 0.9a_2 - 0.6a_3$$



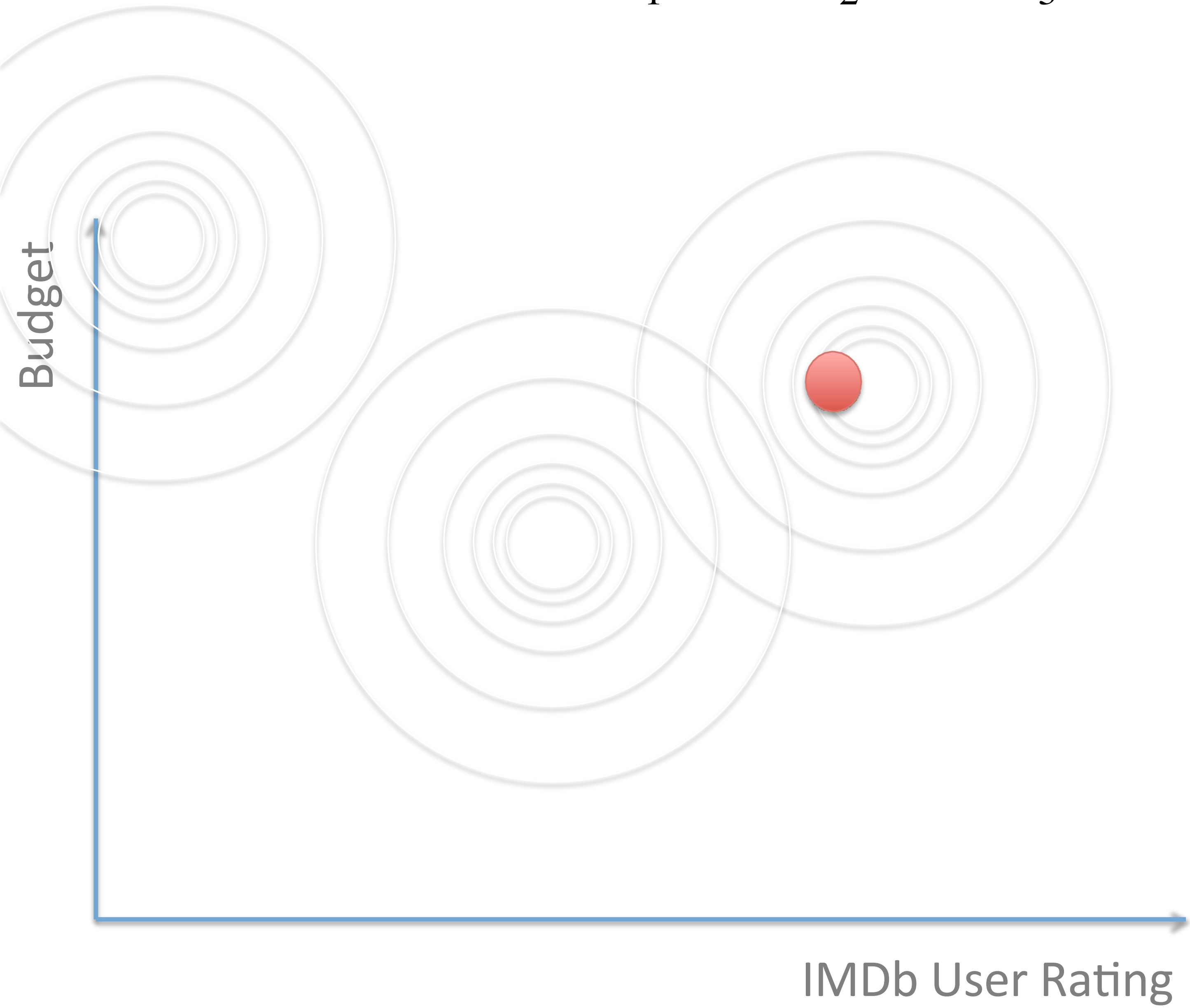
$$0 + 0.7a_1 + 0.9a_2 - 0.6a_3$$



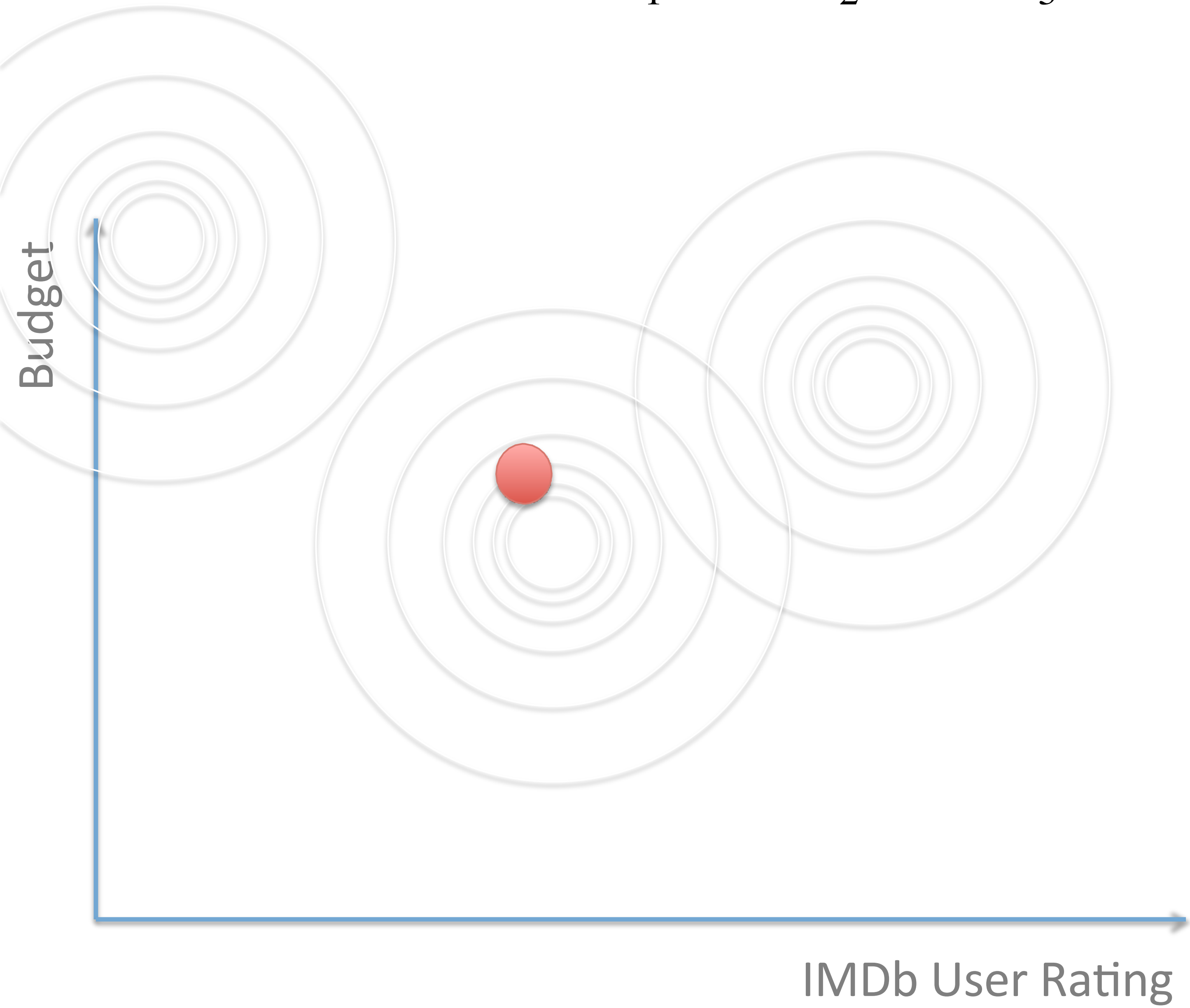
$$0 + 0.7a_1 + 0.9a_2 - 0.6a_3$$



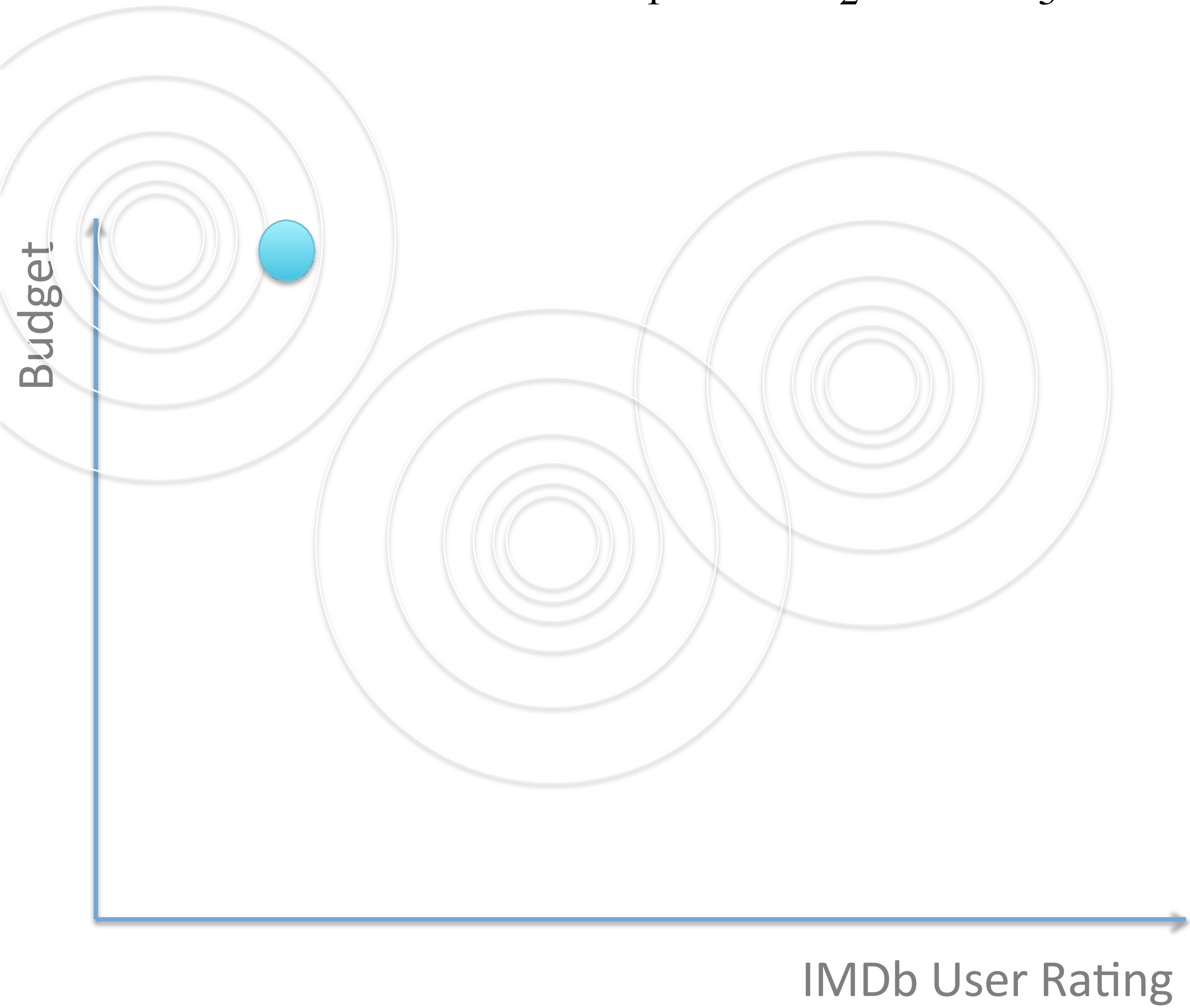
$$0 + 0.7a_1 + 0.9a_2 - 0.6a_3$$



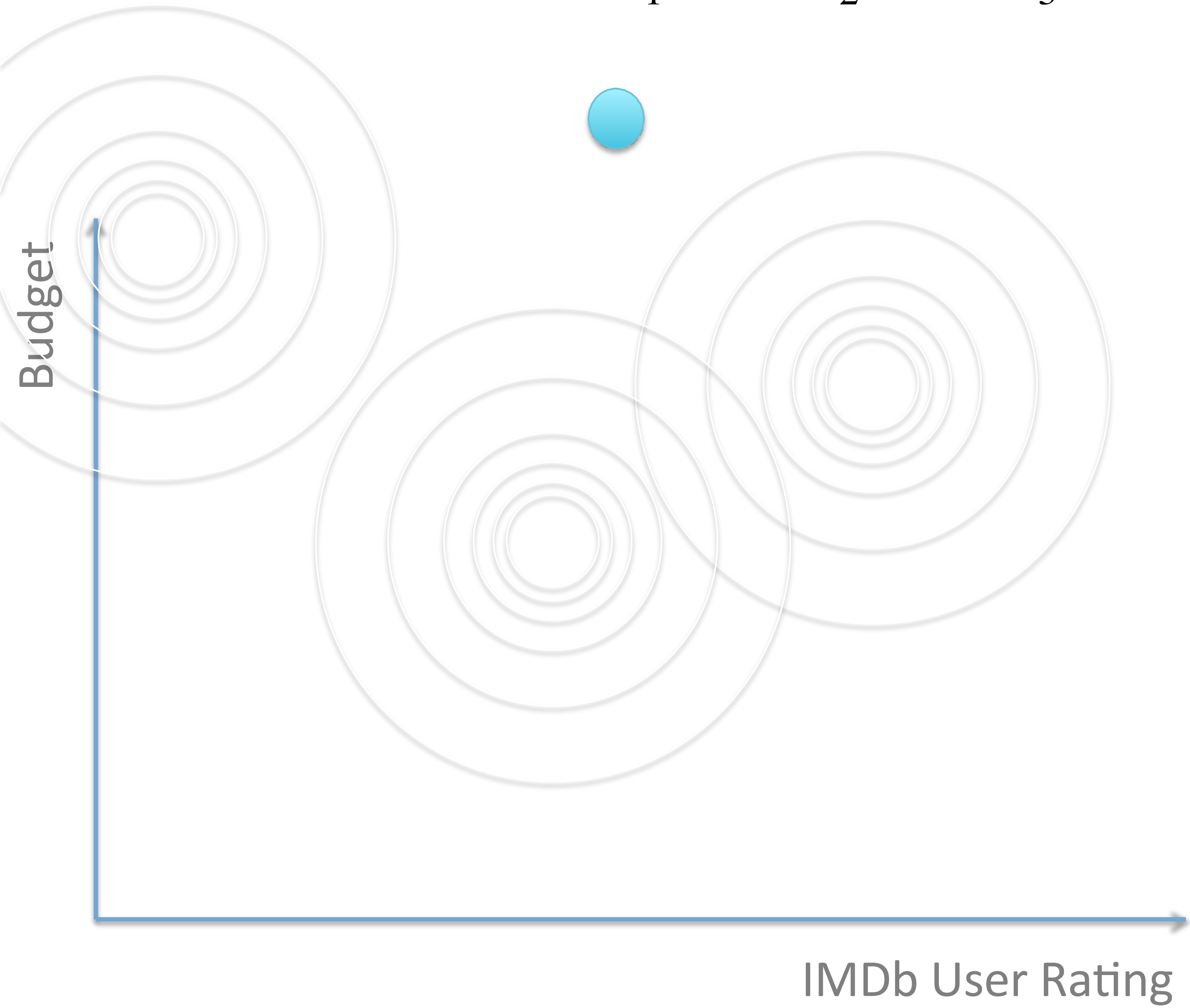
$$0 + 0.7a_1 + 0.9a_2 - 0.6a_3$$



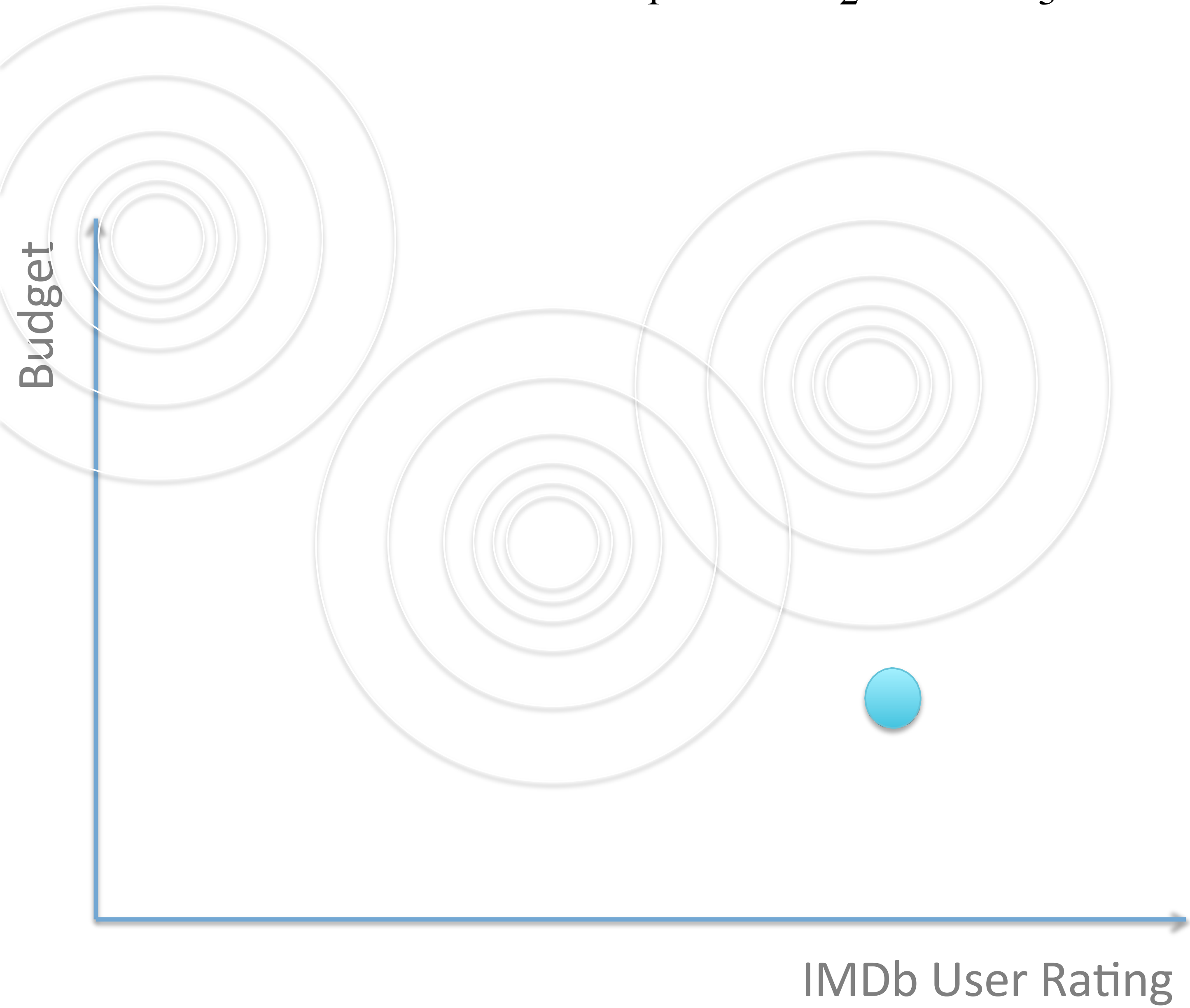
$$0 + 0.7a_1 + 0.9a_2 - 0.6a_3$$



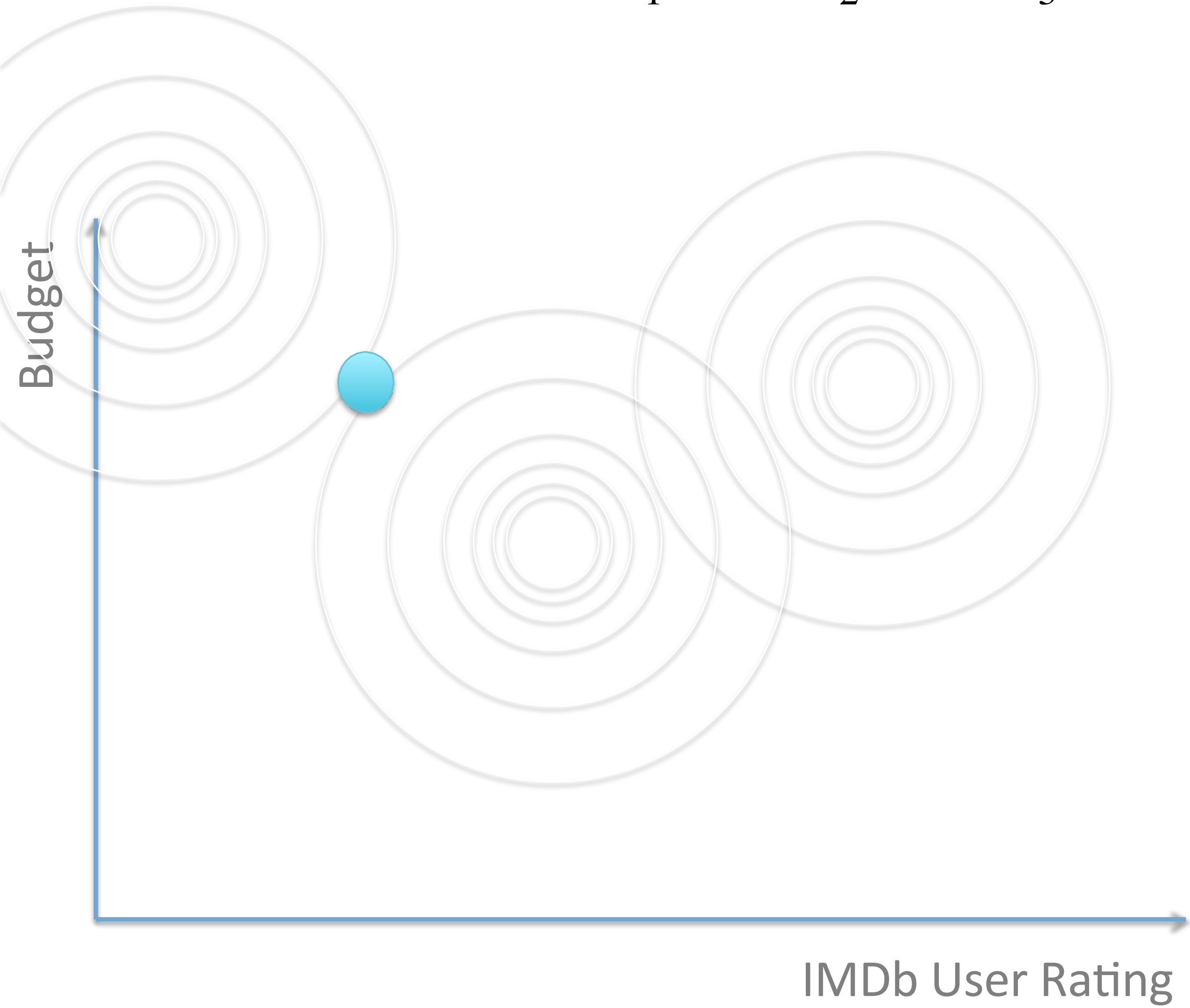
$$0 + 0.7a_1 + 0.9a_2 - 0.6a_3$$



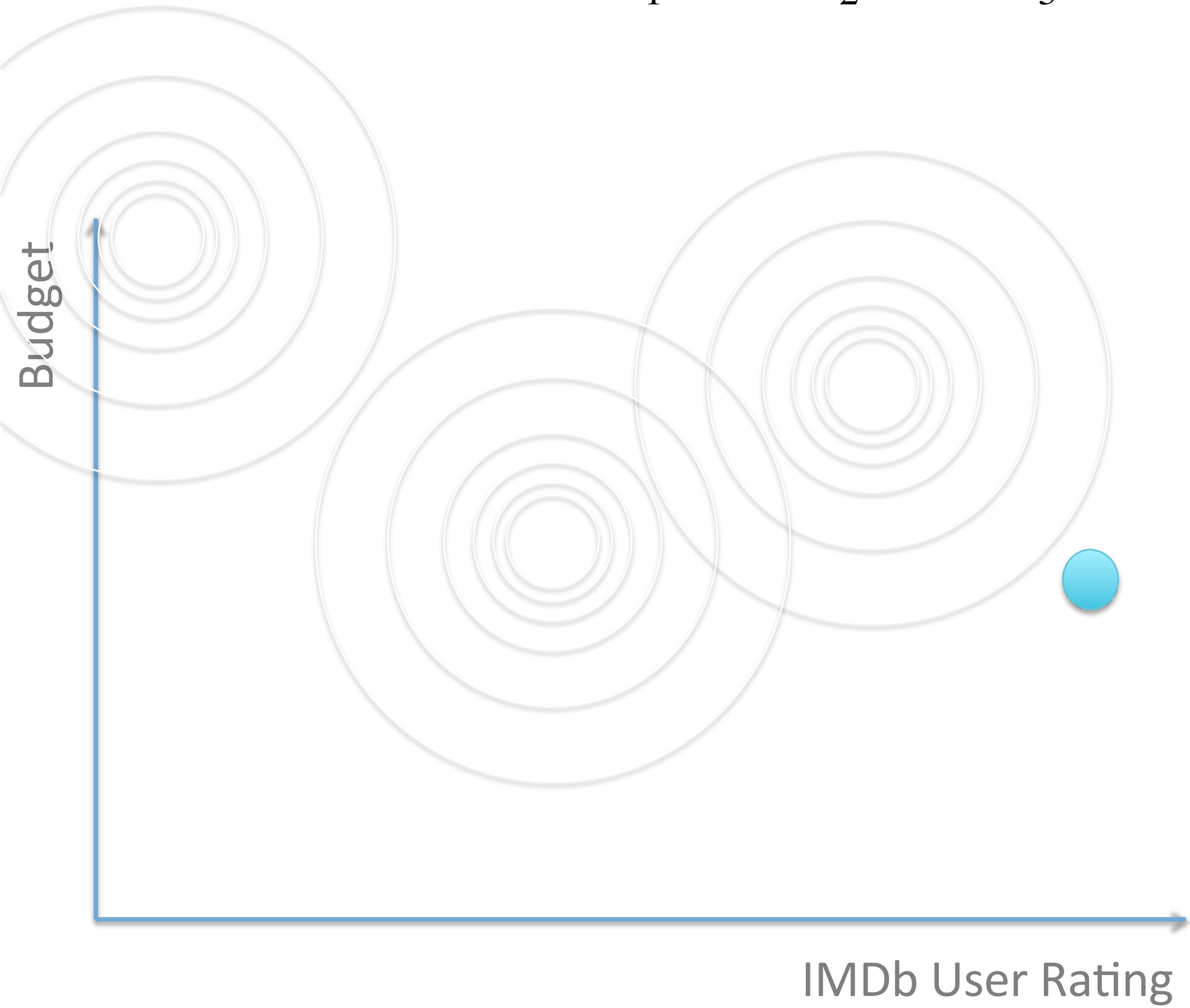
$$0 + 0.7a_1 + 0.9a_2 - 0.6a_3$$



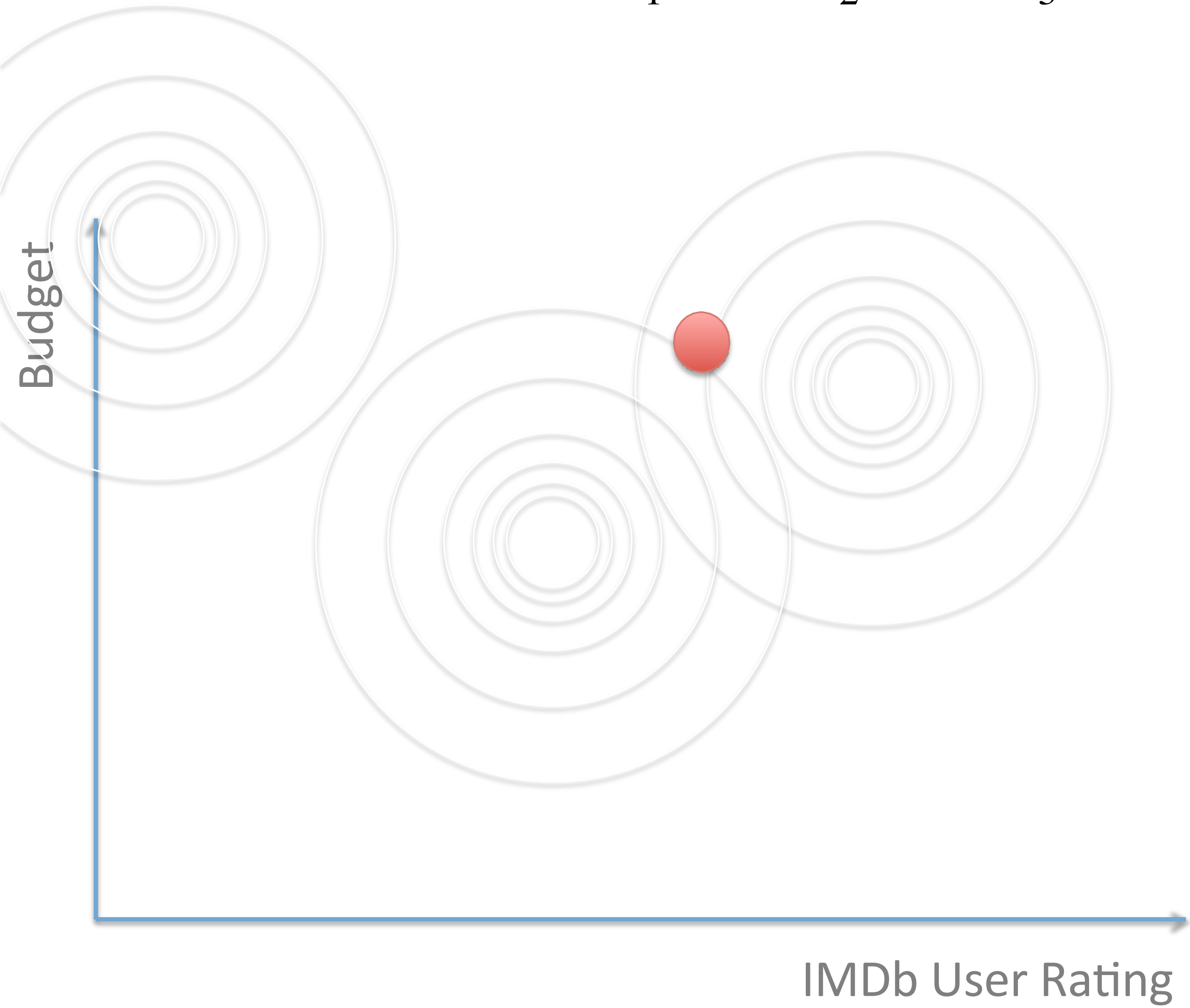
$$0 + 0.7a_1 + 0.9a_2 - 0.6a_3$$



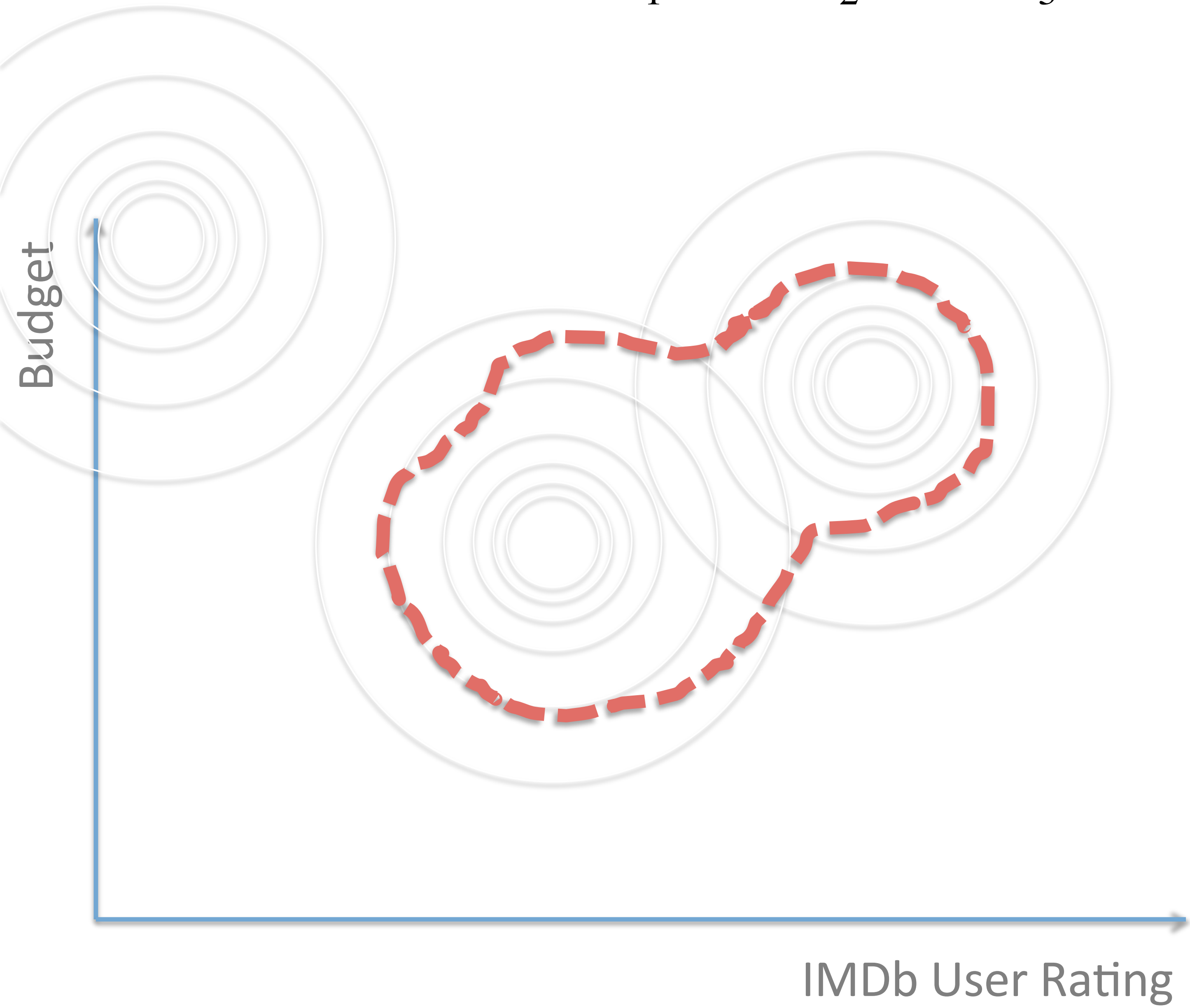
$$0 + 0.7a_1 + 0.9a_2 - 0.6a_3$$



$$0 + 0.7a_1 + 0.9a_2 - 0.6a_3$$

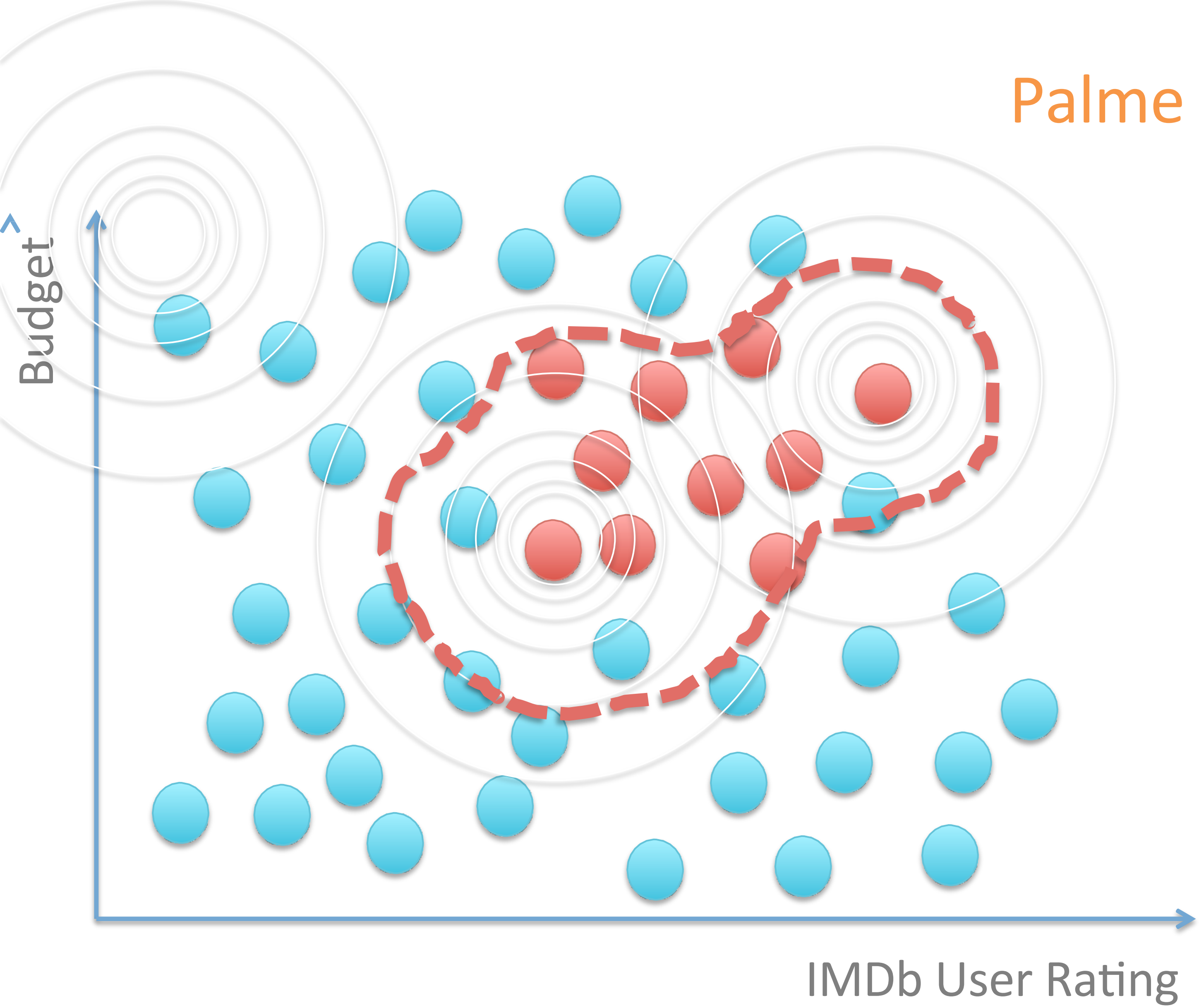


$$0 + 0.7a_1 + 0.9a_2 - 0.6a_3$$



$$0 + 0.7a_1 + 0.9a_2 - 0.6a_3$$

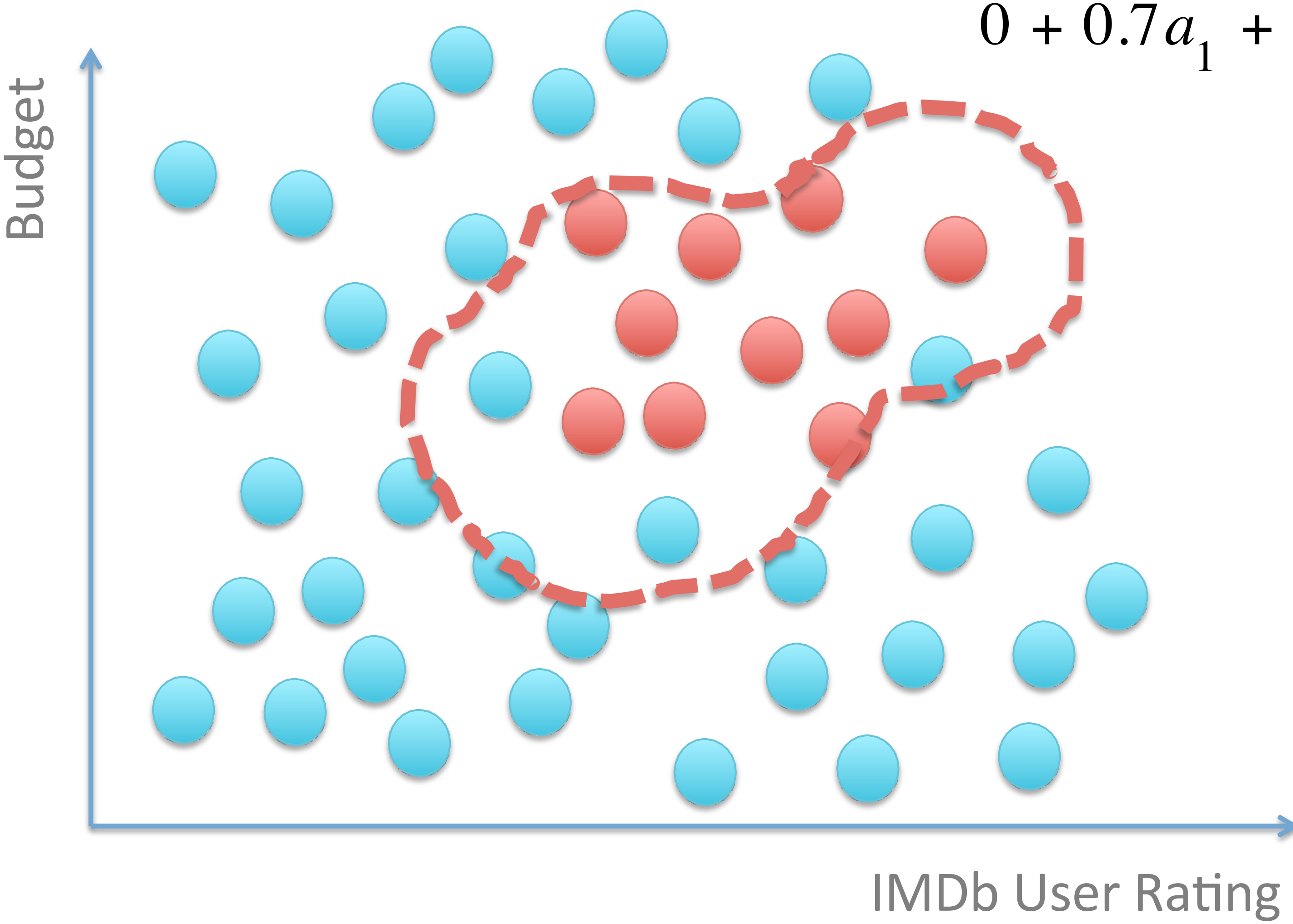
Palme d'Or Winners at Cannes



Palme d'Or Winners at Cannes

Fitting the intricate boundary with the kernel

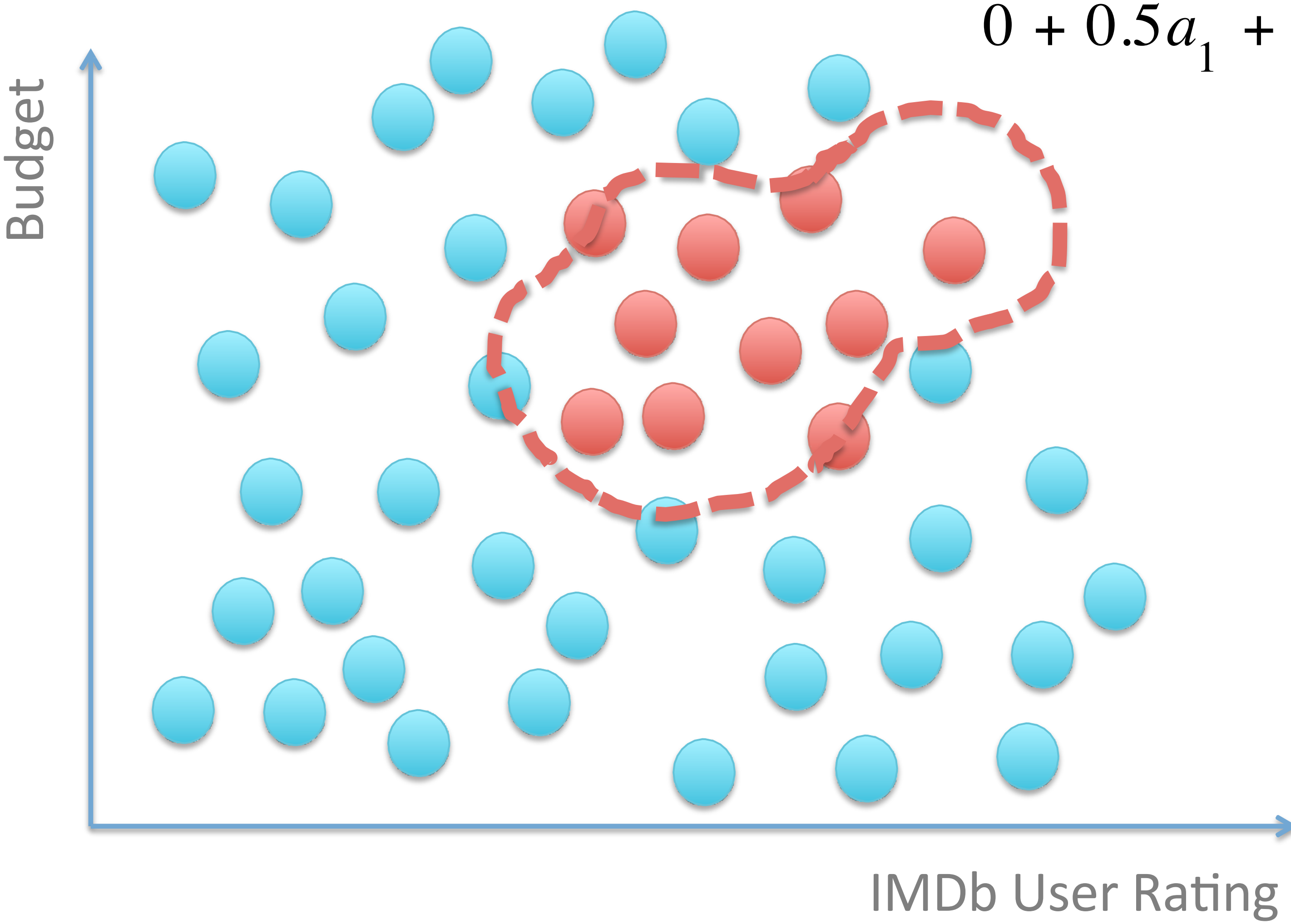
$$0 + 0.7a_1 + 0.9a_2 - 0.6a_3$$



Palme d'Or Winners at Cannes

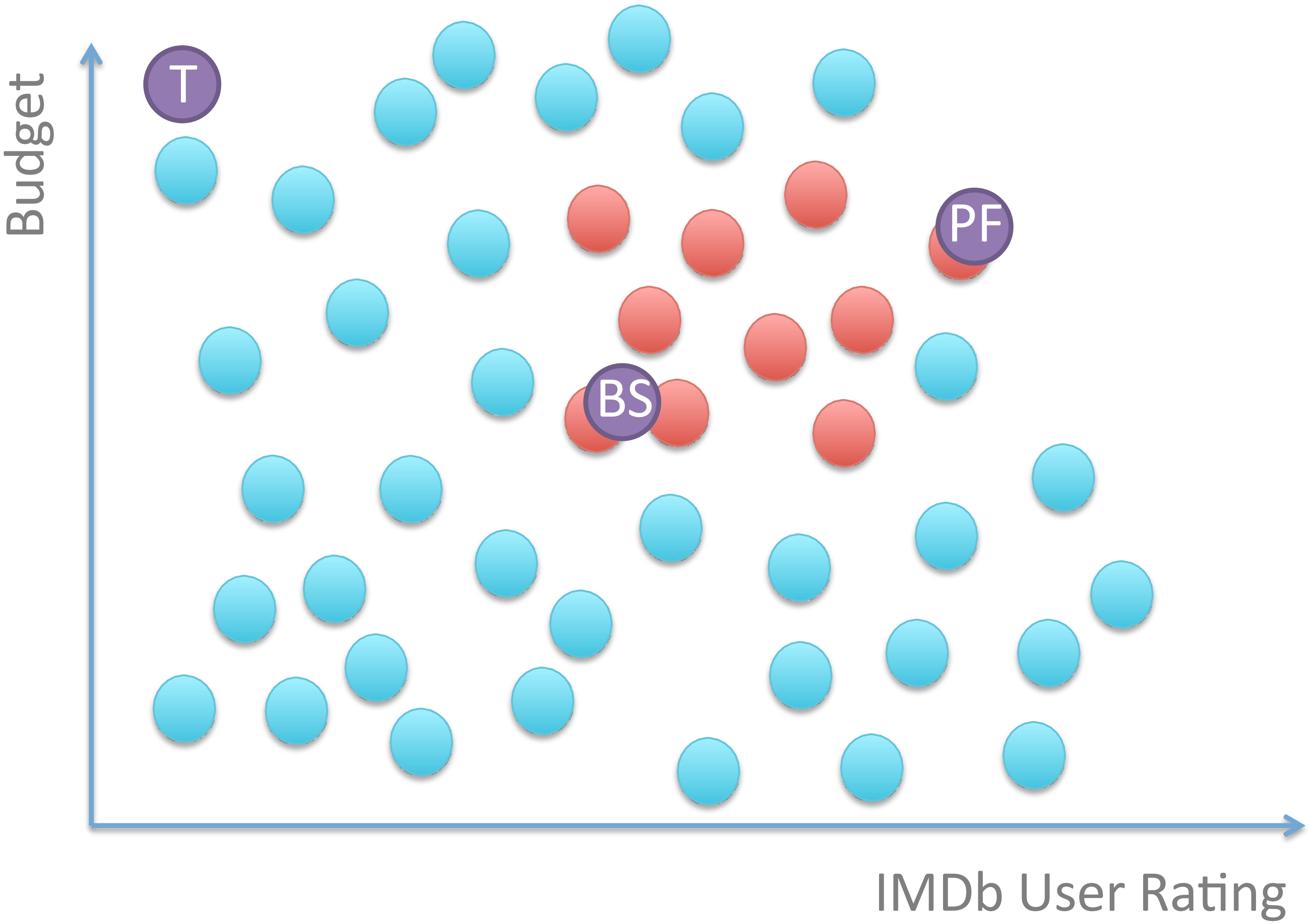
Fitting the intricate boundary with the kernel

$$0 + 0.5a_1 + 0.6a_2 - 0.8a_3$$



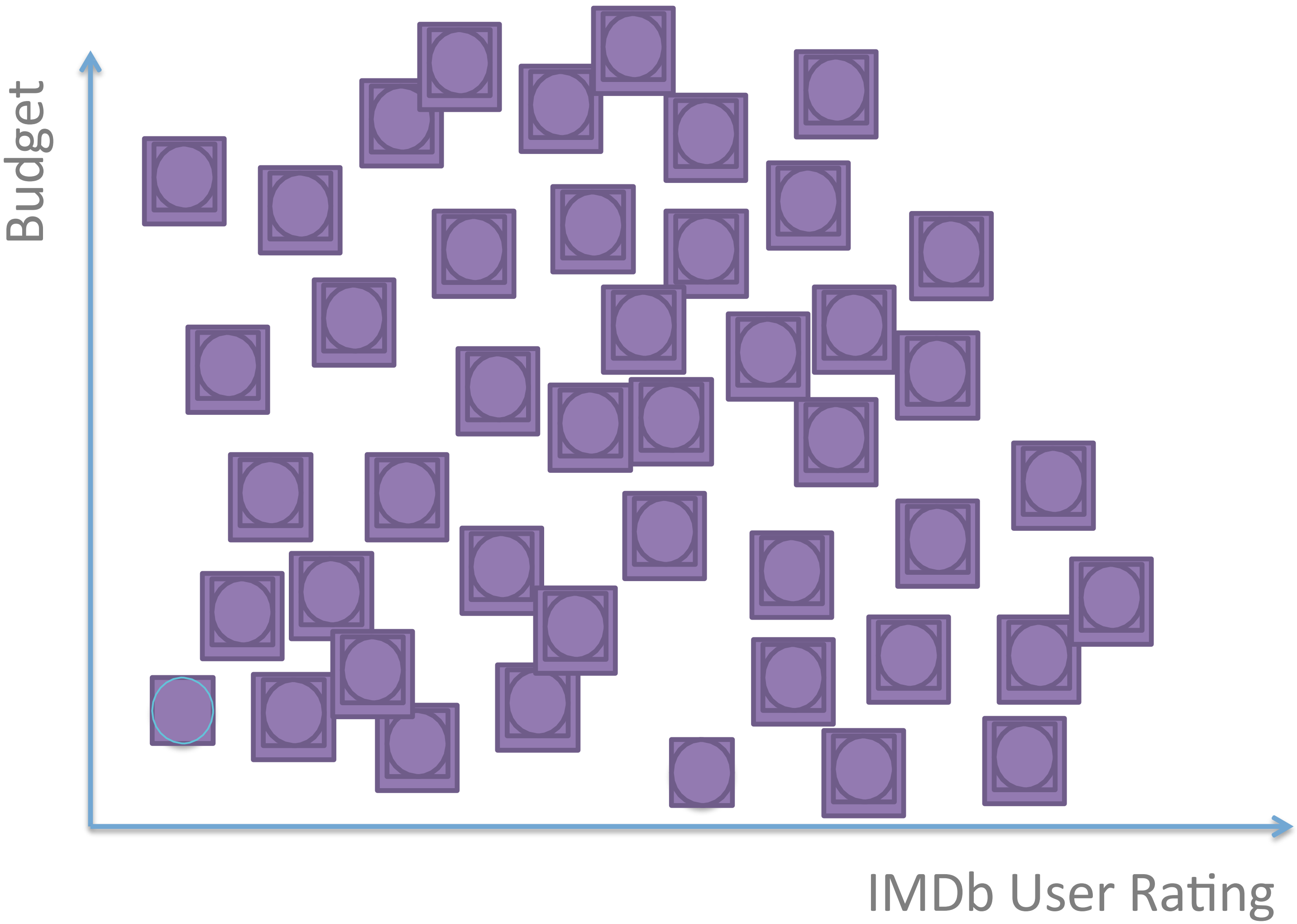
Ok, but how do I choose my feature movies?

What is my Pulp Fiction, Black Swan and Transformers?



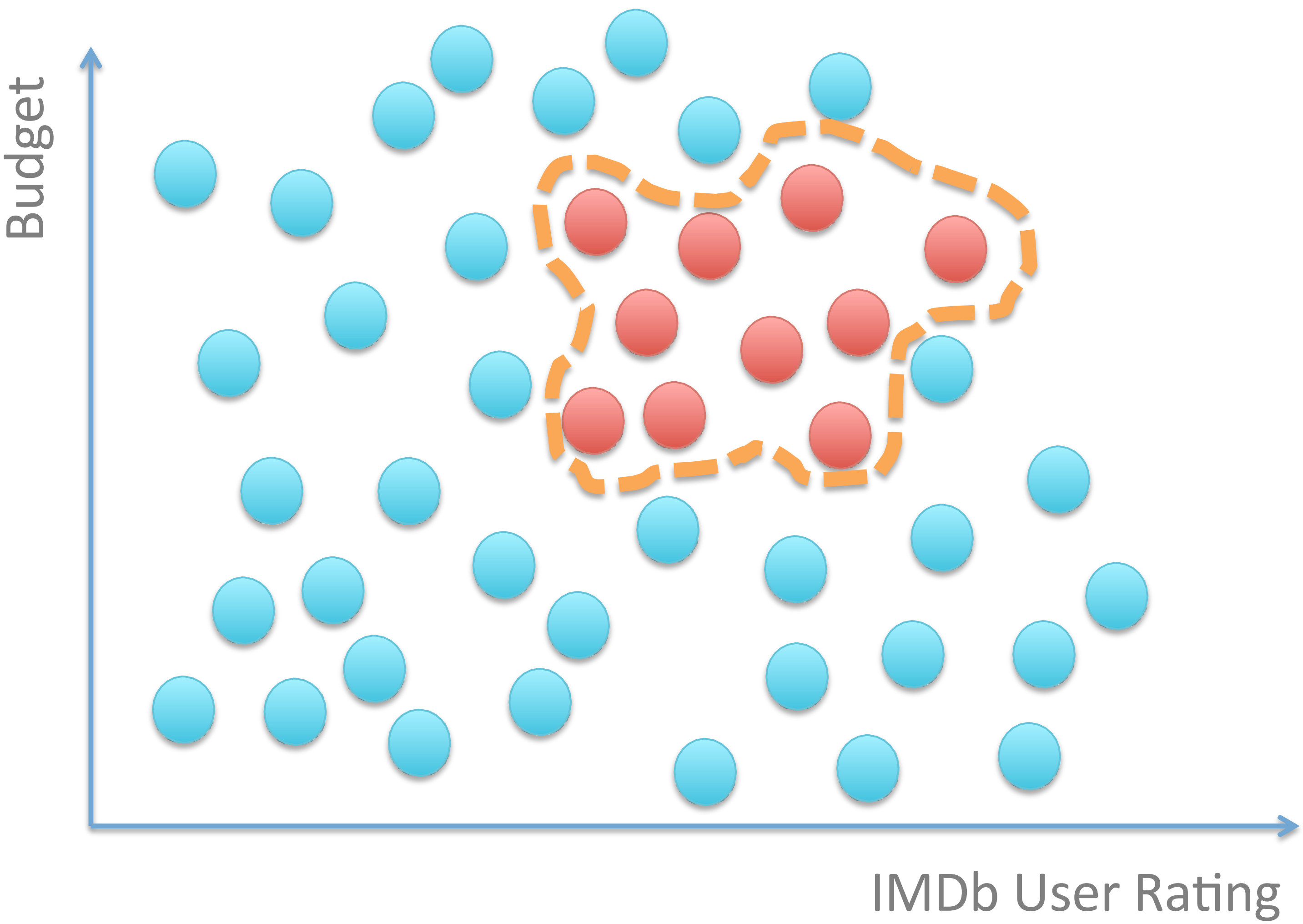
ALL OF THE TRAINING POINTS

Every movie I know is a feature itself



Palme d'Or Winners at Cannes

SVM: Fitting the intricate boundary with RBF kernel



Oh no! Too many points! What do?

Approximate the feature map

from sklearn.kernel_approximation **import** Nystroem

Use a small random sample of points as new features

from sklearn.kernel_approximation **import** Nystroem

Use a small random sample of points as new features

from sklearn.kernel_approximation **import** RBFSampler

Try to mathematically approximate transformation with Monte Carlo sampling (faster, less accurate)